

## PC-BASIC Commandes

### Kit de survie de PC-BASIC

PC-BASIC a une interface de style des années 1980 exploitée en exécutant des commandes typées. Il n'y a pas de menu, et il n'y a aucun des indices visuels que nous attendons des logiciels modernes.

Voici quelques commandes essentielles pour vous aider :

**FILES** affiche le répertoire de travail actuel et son contenu

**LOAD "PROGRAM"** charge le fichier programme nommé PROGRAM.BAS en mémoire

**LIST** affiche le code BASIC du programme en cours

**RUN** démarre le programme en cours

**SAVE "PROGRAM",A** enregistre le programme actuel dans un fichier texte nommé PROGRAM.BAS

**NEW** supprime immédiatement le programme en cours de la mémoire

**SYSTEM** quitte PC-BASIC immédiatement, supprimant tout programme non enregistré

Utilisez l'une des combinaisons de touches Ctrl+Break , Ctrl+Scroll Lock , Ctrl+C ou F12+B pour interrompre un programme en cours.

### Raccourcis clavier

Les touches de fonction et la touche alt peuvent être utilisées comme raccourcis clavier pour certains mots-clés.

Les valeurs par défaut des touches de fonction sont :

F1 LIST	Les mots clés suivants peuvent être saisis avec Alt + première lettre. Les raccourcis Alt ne peuvent pas être redéfinis.
F2 RUN Enter	
F3 LOAD"	
F4 SAVE"	
F5 CONT Enter	
F6 ,"LPT1:" Enter	
F7 TRON Enter	
F8 TROFF Enter	
F9 KEY Space	
F10 SCREEN 0,0,0 Enter	

Dans l'interface de ligne de commande sous Windows, Ctrl + C termine PC-BASIC.

Vous pouvez écrire votre programme avec un éditeur de texte comme Bloc-Notes de Windows et ensuite utiliser la commande LOAD" de PC-BASIC. Exemple LOAD"Blackjac pour charger le programme [BLACKJAC.BAS](#).

## L'environnement de travail

La première chose que vous verrez au démarrage de PC-BASIC est l'environnement de travail. Comme GW-BASIC, mais contrairement à pratiquement tous les compilateurs et interpréteurs modernes, le fonctionnement de PC-BASIC environnement sert à la fois d'environnement de développement et de canevas sur lequel exécuter directement les commandes BASIC.

À quelques exceptions près, pratiquement toutes les commandes pouvant être exécuté dans l'environnement de travail peut être utilisé dans un programme, et vice versa.

L'écran PC-BASIC par défaut comporte 25 lignes et 80 colonnes. La 25ème rangée est utilisée par PC-BASIC pour afficher les raccourcis clavier, ce qui signifie que vous ne pouvez pas l'utiliser pour taper dessus.

Dans certaines vidéos modes, il n'y a que 40 ou 20 colonnes. Les lignes logiques dépassent la largeur de la ligne physique : si vous continuez à taper au-delà de l'écran largeur, le texte passera à la ligne suivante mais PC-BASIC le considèrera toujours comme faisant partie de la même ligne.

Une ligne logique peut comporter au maximum 255 caractères ; si vous tapez plus de 255 caractères, il ignorera le reste. Une ligne peut également être enveloppée par un saut de ligne, entré avec Ctrl + Entrée. Si vous appuyez sur Entrée, PC-BASIC tentera d'exécuter la ligne logique sur laquelle se trouve le curseur placé comme une commande.

Lorsque la commande est exécutée correctement, PC-BASIC affiche le invite Ok. S'il y a une erreur, il affichera un message d'erreur suivi de Ok. Si la ligne commence par un nombre, il sera stocké en tant que ligne de programme. Aucune invite ne s'affiche.

### Clés spéciales

Les touches suivantes ont un effet spécial dans l'environnement de travail :

↑ or Ctrl + 6	Déplace le curseur vers le haut, sauf sur la ligne du haut.
↓ or Ctrl + -	Déplace le curseur vers le bas, sauf à la ligne 24.
← or Ctrl + ]	Déplace le curseur vers la gauche. Le bord gauche de l'écran s'enroule, sauf au rang supérieur.
→ or Ctrl + /	Déplace le curseur vers la droite. Le bord droit de l'écran s'enroule, sauf au rang 24.
Ctrl + ← ou Ctrl + B	Passes à la première lettre du mot précédent. Les mots sont composés de lettres A—Z et chiffres 0—9.
Ctrl + → ou Ctrl + F	Passes à la première lettre du mot suivant.
Tab or Ctrl + I	Déplace le curseur sur le taquet de tabulation suivant. Les taquets de tabulation ont une largeur de 8 colonnes.

Backspace ou Ctrl + H	Supprime le caractère à gauche du curseur, déplacez tous les autres caractères sur la ligne logique une position vers la gauche et modifier les attributs de ces caractères à l'attribut courant. Au bord gauche de l'écran, cela fait la même chose que Del.
Del ou Ctrl + Backspace	Supprime le caractère au niveau du curseur et décaler tous les autres caractères d'une position vers la gauche, changeant les attributs en courant.
Esc or Ctrl + [	Supprimer la ligne logique courante.
Ctrl + End ou Ctrl + E	Supprime tous les caractères depuis le curseur jusqu'à la fin de la ligne logique.
Ctrl + Break ou Ctrl + C ou Ctrl + Scroll Lock	Passe à la première colonne de la ligne suivante, sans exécuter ni stocker la ligne sous le curseur.
Enter ou Ctrl + M	Exécute ou stocke la ligne logique courante. La ligne complète sur l'écran est considéré comme faisant partie de la commande, même si vous n'avez pas tapé ce. Une ligne commençant par un chiffre est stockée en tant que ligne de programme.
End ou Ctrl + N	Déplace le curseur à la première position après la fin de la ligne logique.
Home ou Ctrl + K	Déplace le curseur en haut à gauche de l'écran.
Ctrl + Home ou Ctrl + L	Efface l'écran et déplacez le curseur en haut à gauche de l'écran.
Ctrl + Enter ou Ctrl + J	Passe à la première colonne de la ligne suivante, reliant les deux lignes en une seule ligne logique.
Ctrl + G	Bip le haut-parleur.
Pause ou Ctrl + Num Lock	Pause. Appuyez sur une autre touche pour reprendre. La dernière pression de touche ne sera pas détectée par BASIC.
Ctrl + Prt Sc	Bascule la sortie d'écran en écho vers l'imprimante (ou un autre périphérique attaché à LPT1 :).
Shift + Prt Sc	Imprime l'écran.
Ins or Ctrl + R	Bascule le mode d'insertion. En mode insertion, les caractères sont insérés plutôt qu'écrasé à l'emplacement actuel. Si l'insertion fait que la ligne étendre la largeur physique de l'écran, la ligne logique s'étend sur ligne suivante. Les touches fléchées quittent le mode d'insertion.
Lorsqu'un programme est démarré, les commandes du programme sont suivies jusqu'à ce que le programme quitte et revient en mode direct ou jusqu'à ce qu'une intervention de l'utilisateur soit requise. Lorsqu'un programme est en cours d'exécution, quelques touches ont un effet immédiat :	

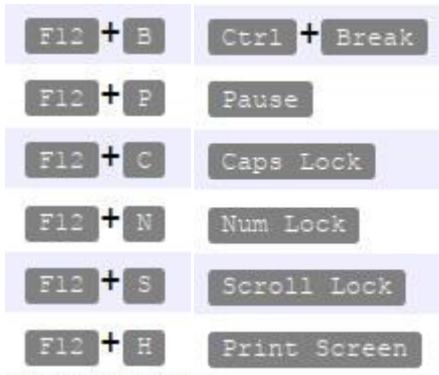
Pause ou Ctrl + Num Lock	Suspendre l'exécution. Appuyez sur une autre touche pour reprendre.
Ctrl + Break ou Ctrl + Scroll Lock	Arrête l'exécution et retourne en mode direct. Un message Break est imprimé.
Ctrl + C	Si ctrl-c-break=True : arrête l'exécution et revient en mode direct. Un message de pause est imprimé.
Si une entrée utilisateur est requise par les instructions INPUT, LINE INPUT ou RANDOMIZE, la plupart des touches ont le même effet qu'en mode direct. Les touches suivantes ont un effet différent :	
Ctrl + Break or Ctrl + C ou Ctrl + Scroll Lock	Arrête l'exécution et retourne en mode direct. Un message de pause est imprimé.
Enter	Termine la saisie et revenez au mode précédent.
<b>Raccourcis clavier</b>	
Les touches de fonction et la touche alt peuvent être utilisées comme raccourcis clavier pour certains mots clés. Les valeurs par défaut des touches de fonction sont :	
F1	LIST
F2	RUN Enter
F3	LOAD"
F4	SAVE"
F5	CONT Enter
F6	,"LPT1:" Enter
F7	TRON Enter
F8	TROFF Enter
F9	KEY Space
F10	SCREEN 0,0,0 Enter

Les raccourcis des touches de fonction peuvent être redéfinis avec l'instruction KEY. Les raccourcis sont affichés en bas de l'écran. Les mots clés suivants peuvent être saisis avec Alt + première lettre. Les raccourcis Alt ne peuvent pas être redéfinis.

AUTO, BSAVE, COLOR, DELETE, ELSE FOR, GOTO, HEX\$, INPUT, KEY, LOCATE MOTOR, NEXT OPEN, PRINT, RUN, SCREEN, THEN, USING, VAL, WIDTH, XOR

## Clés alternatives

Dans PC-BASIC, la touche F12 peut être utilisée pour saisir des touches spéciales qui ne sont pas présentes sur certains claviers.



La touche F12 peut également être utilisée en combinaison avec les touches numériques régulières et les touches fléchées pour saisir des chiffres à partir du pavé numérique. Les combinaisons F12 ne sont pas présentes dans GW- DE BASE.

De plus, comme dans GW-BASIC, la touche Alt peut être utilisée pour saisir des caractères par leur code points (valeurs ASCII). Cela se fait en appuyant sur la touche Alt et en tapant le point de code comme une valeur décimale sur le pavé numérique, puis relâchez la touche Alt.

## Opérations du Presse-papiers

Contrairement à GW-BASIC, vous pouvez copier et coller du texte dans le presse-papiers. Cela peut être fait avec la souris ou avec la touche F11.

L'utilisation du presse-papiers avec la souris fonctionne dans le style de X11 : le bouton gauche est sélectionné et copie; le bouton du milieu est coller.

Les combinaisons de clavier suivantes font également fonctionner le presse-papiers :

F11 + ↑ ↓ ← → Sélectionnez une zone d'écran.

F11 + A Tout sélectionner.

F11 + C Copier dans le presse-papier.

F11 + V Coller depuis le presse-papiers.

**Touches de contrôle de l'émulateur** Dans PC-BASIC, F11 + F bascule en mode plein écran.

## Compatibilité

- Certaines combinaisons de touches peuvent avoir un effet différent de celui décrit ci-dessus, selon le système d'exploitation et le choix de l'interface à utiliser avec PC-BASIC.

- Certaines combinaisons de touches seront interprétées par le système d'exploitation ou la fenêtre gestionnaire et provoquer des actions spéciales.

Par exemple, sur la plupart des systèmes, Alt + F4 terminer PC-BASIC à moins que l'option prevent\_close ne soit définie ; F1 peut ouvrir votre le système d'aide du système d'exploitation. Il peut être possible d'éviter certains de ces effets en utilisant l'interface graphique en mode plein écran.

- Dans l'interface de ligne de commande sous Windows, Ctrl + C termine PC-BASIC.
- Dans l'interface de ligne de commande sous Linux et Mac, Ctrl + D termine PC-BASIC.

## Programmes et fichiers

PC-BASIC peut contenir qu'un programme BASIC à la fois en mémoire.

Pour saisir une ligne de programme, lancez avec un numéro de ligne et entrez les commandes BASIC après cela. La longueur maximale d'une ligne de programme est de 255 caractères, y compris le numéro de ligne et les espaces.

La ligne de programme ne sera pas être exécuté immédiatement, mais stocké dans le programme. Les lignes de programme sont triées par ligne nombre, de sorte que la ligne 10 est exécutée avant la ligne 20.

Toutes les lignes de programme doivent avoir un numéro de ligne. Les numéros de ligne vont de 0 à 65535 inclus. Il n'est pas possible d'entrer une ligne nombre supérieur à 65529, mais ceux-ci peuvent exister dans les programmes chargés.

Au sein d'une ligne de programme, les instructions sont séparées par deux-points :

EX.

```
195 C=1:W(X,1)=C:C=C+1  
200 R=X:S=1:GOTO 260
```

Pour exécuter le programme, tapez la commande RUN. PC-BASIC va maintenant exécuter toutes les lignes de programme dans l'ordre à l'intérieur de l'environnement de travail.

Vous ne pouvez pas déplacer le curseur ou entrer des commandes pendant l'exécution du programme.

Si et quand le programme se termine, le contrôle de l'environnement de travail reviendra pour vous.

Vous pouvez interrompre un programme à tout moment en utilisant l'une des combinaisons de touches Ctrl + Break ou Ctrl + Scroll Lock. Le programme s'arrêtera immédiatement, imprimera un message Break et vous rend le contrôle.

Dans GW-BASIC, vous ne pouvez pas utiliser Ctrl + C pour interrompre un programme en cours d'exécution.

Toutefois plusieurs les claviers modernes n'ont pas de touche Break ou Scroll Lock, ce qui rendrait impossible pour interrompre un programme qui ne se termine pas.

Par conséquent, par défaut, PC-BASIC traite **Ctrl + C** comme s'il s'agissait de Ctrl + Break.

Définissez l'option Ctrl-c-break=False si vous préférez le comportement GW-BASIC stylé.

Lors de l'utilisation de l'interface textuelle ou de ligne de commande, cette option est ignorée.

Un programme peut être stocké sur un lecteur à l'aide de la commande SAVE, de l'une des trois manières suivantes :

1. Texte brut, lisible par n'importe quel éditeur de texte : **SAVE "MYPROG",A**
2. Tokénisé, occupant moins d'espace de stockage : **SAVE "MYPROG"**
3. Protégé, qui est un format crypté : **SAVE "MYPROG",P**

Dans les trois cas, le programme sera écrit dans le répertoire de travail courant avec le nom **MONPROG.BAS**.

PC-BASIC peut lire et écrire des fichiers protégés créés par GW-BASIC. Contrairement à GW-BASIC, cependant, il ne désactive pas l'accès au contenu non chiffré du fichier.

Le cryptage utilisé par GW-BASIC a été cassé il y a plusieurs décennies, donc le mode protégé offrait peu de protection de toute façon ; interdire l'accès est un petit danger pour la sécurité car cela permettrait à quelqu'un de vous envoyer un programme que vous ne pouvez pas inspecter avant de l'exécuter.

Cependant, il est possible de désactiver l'accès aux fichiers protégés en activant l'option `hide-protected`.

Vous pouvez lire un fichier programme en mémoire avec `LOAD "MYPROG"`. Cela effacera le programme actuellement en mémoire et le remplacera par celui lu dans le répertoire de travail courant.

Pour accéder aux fichiers dans un répertoire différent, spécifiez un chemin à partir du répertoire courant.

la spécification du chemin suit les conventions DOS.

Le seul séparateur de chemin valide est la barre oblique inverse `\`.

Par exemple, `LOAD "PROGRAMS\MYPROG"`.

Vous pouvez charger ou exécuter un programme immédiatement au démarrage de PC-BASIC en utilisant la commande `load` ou options d'exécution.

Par exemple, `pcbasic --run=MONPROG.BAS`

Les arguments de ces options peuvent être fournis sous forme de chemins PC-BASIC ou de chemins dans le formulaire standard pour votre système d'exploitation.

PC-BASIC permet de convertir entre les trois formats de programme :

**Soit en chargeant le programme et enregistrement dans le format souhaité,**

ou à partir de la ligne de commande en utilisant l'option de convertisseur.

Pour convertir un fichier tokenisé ou protégé en texte brut, vous pouvez utiliser, par exemple :

`pcbasic --convert=A PROGRAMME.BAS PROGRAMME.BAS`

## Accéder à vos disques

PC-BASIC émule les périphériques de disque DOS, qui sont désignés par des lettres de lecteur telles que Z:, C:

L'une des lettres de lecteur correspond au périphérique actuel.

Sous Windows :

- Par défaut, les périphériques de disque PC-BASIC s'accorderont avec les lettres de lecteur Windows au démarrage de la session PC-BASIC.
- Si PC-BASIC est démarré à partir du raccourci du menu Démarrer, le périphérique actuel sera votre Dossier Documents (ou Mes Documents sur certaines versions de Windows). Vous pouvez modifier cet emplacement en définissant le dossier Démarrer du raccourci.
- Si PC-BASIC est démarré à partir de l'invite de commande, le périphérique actuel sera défini sur le répertoire de travail actuel de l'invite de commande.
- Si le périphérique actuel de PC-BASIC ou le dossier Démarrer dans est remplacé par un dossier système tel que comme C:\Program Files\PC-BASIC , Windows déplacera les fichiers qui y sont écrits vers %LocalAppData%\VirtualStore à la place.
- Notez que les périphériques de disque DOS de PC-BASIC ne sont pas le même que le lecteur Windows. Le périphérique C: sur PC-BASIC n'est pas toujours votre lecteur Windows C:.

Par défaut, les lettres de lecteur Windows sont mappées sur les périphériques PC-BASIC au démarrage de la Session PC-BASIC. Cependant, si vous utilisez l'option mount ; ou si les lettres du lecteur Windows changent pendant que PC-BASIC est en cours d'exécution (via, par exemple, l'utilisation du réseau ou la carte du réseau Drive Operations), ils ne seront plus d'accord.

## Sur d'autres systèmes :

- Par défaut, Z : pointera vers le répertoire de travail actuel à partir duquel PC-BASIC a été lancé. Ce sera l'appareil actuel.
- S'il est démarré à partir d'un menu ou d'un package d'application, il s'agira généralement de votre répertoire personnel ~.

Ce périphérique actuel est l'endroit où les fichiers seront enregistrés et chargés dans BASIC. Si vous ne le faites pas, spécifier un autre endroit.

Vous pouvez changer le périphérique actuel à l'aide de l'option de périphérique actuel dans le fichier de configuration ou sur l'invite de commande.

ou installer PC-BASIC dans votre répertoire de programme...

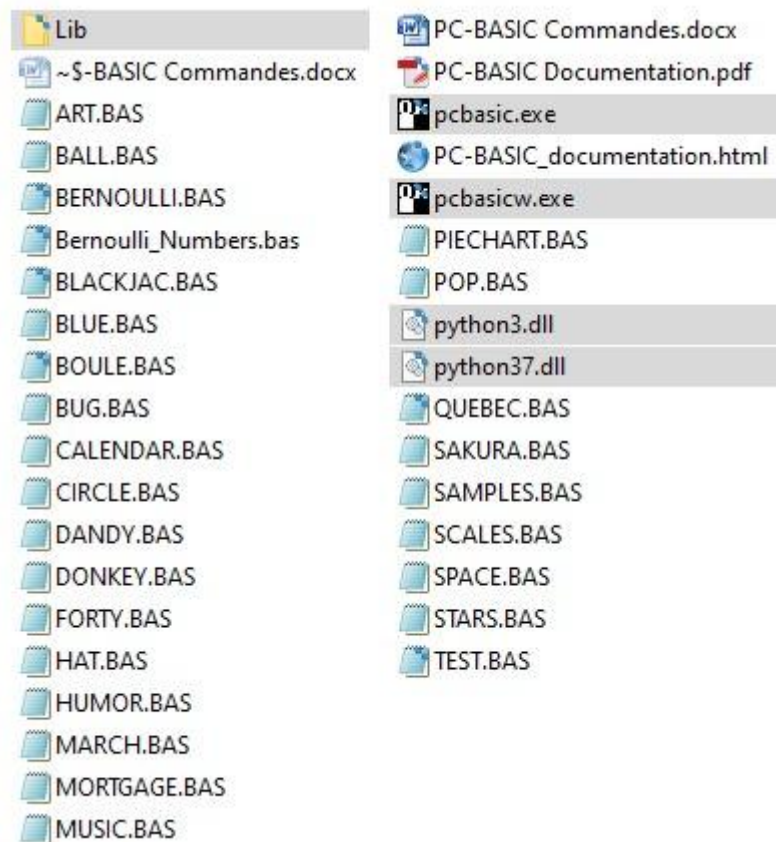
ou si vous utilisez des sous répertoire pour vos programmes, vous pouvez créer un répertoire racine du disque.

Ex. Dans un répertoire nommé C:\Basic



Fichiers à copier dans votre répertoire de travail ou dans tous vos répertoires si vous voulez en utiliser plusieurs

**Soit: Le dossier Lib, les fichiers pcbasic.exe, pcbasicw.exe, python3.dll, python37.dll**



Vous pouvez mapper des lecteurs et d'autres emplacements de système de fichiers en tant que périphériques PC-BASIC en utilisant l'option de montage.

Par exemple, sous Windows, l'option `mount=A:C:\Users\Me\BasicFloppy` rendra le dossier `C:\Users\Me\BasicFloppy` disponible en tant que périphérique A: de PC-BASIC.

Sur d'autres plates-formes, un exemple d'option de montage pourrait ressembler à `mount=A:/home/me/BasicFloppy` ce qui rendrait le répertoire `/home/me/BasicFloppy` disponible en tant que le périphérique A: de PC-BASIC.

PC-BASIC utilise les conventions DOS pour les noms de fichiers et les chemins. Ceux-ci ont des différents de Conventions de noms de fichiers courts Windows et pas si différentes des conventions Unix.

Cette peut conduire à des effets surprenants en présence de plusieurs fichiers correspondant au même Nom DOS. Pour éviter de telles surprises, il est préférable d'exécuter PC-BASIC dans un répertoire de travail qui lui est propre et utilisez le format tout en majuscules pour tous les fichiers.

## Fichiers BASIC compatibles

De nombreux dialectes BASIC utilisent la même extension.BAS, mais leurs fichiers ne sont pas compatibles. PC- BASIC exécute uniquement les fichiers de programme **GW-BASIC**.

Quelques conseils pour reconnaître les programmes GW-BASIC :

- Les fichiers GW-BASIC stockés sous forme de texte sont des fichiers de texte brut avec des numéros de ligne.
- Les fichiers tokenisés sont des fichiers binaires commençant par l'octet magique &hFF.
- Les fichiers protégés sont des fichiers binaires commençant par l'octet magique &hFE.

En particulier, les fichiers QBASIC (qui n'ont pas de numéro de ligne) et les fichiers QuickBASIC (magic byte &hFC) ne fonctionnera pas.

PC-BASIC acceptera les conventions de retour à la ligne DOS et Unix pour les programmes stockés en clair texte. Ce comportement est différent de GW-BASIC, qui n'accepte que les fichiers texte avec CR LF fins de ligne.

Par conséquent, dans des cas exceptionnels où une ligne de programme est continuée via LF correct, les fichiers texte GW-BASIC peuvent ne pas être chargés correctement.

Si vous rencontrez de tels un cas, utilisez l'option soft-linefeed pour activer le comportement GW-BASIC.

Si le saut de ligne progressif est activé, les fichiers texte au format Unix standard (fins de ligne LF, pas de caractère de fin de fichier) échoueront à se charger, comme ils le font dans GW-BASIC.

Sous Linux ou Mac, utilisez un utilitaire tel qu'unix2dos pour convertir programmes enregistrés sous forme de fichiers texte avant de les charger.

Lors de l'enregistrement en tant que texte, PC-BASIC utilise les fins de ligne CR LF et &h1A à la fin du fichier.

Paquets PC-BASIC peut exécuter des programmes packagés.

Un package est simplement un répertoire ou une archive zip.

Le contenu du répertoire ou du fichier zip sera chargé en tant que répertoire de travail actuel.

Si une configuration fichier nommé PCBASIC.INI est présent dans ce répertoire, ses paramètres sont chargés ; habituellement, un de ces paramètres sera un argument d'exécution lié à un programme BASIC inclus dans l'archive ou le répertoire.

PC-BASIC reconnaîtra les fichiers zip quelle que soit leur extension.

## **Une suggestion**

L'extension pour les packages PC-BASIC est.BAZ.

Les packages sont un choix pratique si un programme doit modifier de nombreuses options PC-BASIC pour fonctionner comme vous le souhaitez, ou s'il a besoin d'une configuration du répertoire de travail.

Les packages de fichiers zip sont décompressés dans un répertoire temporaire chaque fois qu'ils sont chargés. Le répertoire temporaire est supprimé à la fermeture de PC-BASIC.

Avec les packages zipfile, il est donc impossible d'enregistrer les fichiers et de les rouvrir lors de la prochaine exécution du package.

## **Cassettes audio**

Le périphérique CAS1 s'interface avec l'émulateur de cassette. Les cassettes n'ont jamais été très populaires sur IBM PC, et en effet uniquement disponible avec le PC d'origine et le PCjr. Il n'y a pas de nombreuses cassettes IBM PC dans la nature.

Cependant, si vous en croisez un, il vous suffit pour le lire avec PC-BASIC est de l'enregistrer dans un fichier.WAV (RIFF WAVE) et de le joindre au CAS1 : périphérique avec l'option cas1=WAV:filename.

Vous pouvez également générer votre propre bande images et stockez-y vos programmes. Les fichiers WAV générés par PC-BASIC sont volumineux mais très facilement compressé dans une archive ZIP; cela fonctionne mieux et conduit à des fichiers plus petits qu'un transcodage vers un format audio avec perte comme MP3.

Comme alternative à.WAV, vous pouvez stocker des bandes au format CAS. Il s'agit simplement d'un petit vidage de la bande et est interchangeable avec les images de bande pour l'émulateur PC IBM PCE.

Les versions précédentes de PC-BASIC incluait la prise en charge des cassettes BASICODE; cela a été abandonné au profit d'un décodeur BASICODE séparé. Utilisez ce décodeur pour convertir le programme BASICODE au format PC-BASIC avant de le charger dans PC-BASIC.

Sécurité PC-BASIC fait des choix par défaut avec la sécurité de base, mais ne sécurise pas ses programmes de manière significative.

Les programmes BASIC ont un accès plus ou moins complet à votre ordinateur. Vous devez les traiter avec la même prudence que vous le feriez avec des scripts shell ou binaires.

Par conséquent, n'exécutez pas un programme à partir d'Internet que vous n'avez pas inspecté au préalable. En utilisant LIST ou pbasic --convert=Un nom de fichier sur la ligne de commande. Vous ne vous contenteriez pas de télécharger un exécutable sur Internet et de l'exécuter soit, non?

## Connexion aux périphériques

### Impression

Vous pouvez imprimer à partir de programmes PC-BASIC en accédant au périphérique LPT1:.

PC-BASIC envoyer la sortie à l'imprimante par défaut de votre système d'exploitation, sauf si vous modifiez le `lpt1= option`.

Pour imprimer via une imprimante nommée MyPrinter , définissez `lpt1=PRINTER:MyPrinter`.

Vous pouvez connectez également des imprimantes aux périphériques LPT2: et LPT3:.

La sortie sera envoyée à l'imprimante dans l'un des cas suivants : un fichier ouvert dans LPT1 : est fermé, un programme se termine ou PC-BASIC est fermé.

Si vous préférez, vous pouvez à la place, envoyez chaque page séparément à l'imprimante en définissant `lpt1=PRINTER:MyPrinter:page`.

Vous pouvez même envoyer chaque ligne séparément, mais cela n'a de sens que sur une imprimante alimentée par un tracteur (comme c'était courant à l'apogée de GW-BASIC).

Il est facile d'imprimer dans un fichier au lieu d'une imprimante : définissez `lpt1=FILE:output.txt` pour envoyer tous les sorties LPT1 de l'imprimante dans le fichier texte `output.txt`.

Les instructions d'impression LPRINT et LLIST envoient toujours leur sortie au LPT1 de PC-BASIC.

La présentation des documents imprimés est laissée à votre système d'exploitation : ce sera celui par défaut de présentation de fichiers texte.

Si vous souhaitez modifier le mode d'impression des documents, veuillez consulter les paramètres de votre système d'exploitation.

- Sous Windows, les fichiers texte sont imprimés de la même manière que lorsque vous faites glisser un fichier texte et vous le déposez sur l'icône de l'imprimante.

Cela signifie que le formatage est géré par l'application associée aux fichiers.txt ; il s'agit généralement de `notepad.exe`.

Changer la manière dont les documents PC-BASIC sont imprimés, modifiez la mise en page de l'application. Vous devrez définir une police d'imprimante comprenant les caractères que vous avez besoin d'imprimer.

- Sur les systèmes Unix, PC-BASIC utilisera l'utilitaire `paps` s'il est disponible ; sélectionne automatiquement les polices qui prennent en charge les caractères dont vous avez besoin.

## Ports série et parallèle

PC-BASIC fournit les périphériques série COM1 : et COM2 :

Pour les utiliser, vous avez besoin pour les connecter à un port de communication de votre ordinateur avec l'option com1= ou com2=.

Pour vous connecter au premier port série physique, définissez com1=PORT:0 (ou, alternativement, com1=PORT:COM1 sous Windows ou com1=PORT:/dev/ttyS0 sous Linux).

Si vous n'avez pas de port série, vous pouvez en émuler un en envoyant les communications sur une prise réseau : set com1=SOCKET:localhost:7000 et tout le trafic COM1: sera envoyé via le socket 7000.

Pour accéder à un port parallèle, connectez-le à l'un des ports LPT1 : , LPT2 : ou LPT3 :

Par exemple, définissez lpt2=PARPORT:0 pour connecter le premier port parallèle de votre ordinateur à LPT2 :.

## Modification de l'interface

### Cibles d'émulation

Par défaut, PC-BASIC émule GW-BASIC sur un système avec des capacités vidéo VGA. Cependant, il peut émuler plusieurs autres configurations, qui diffèrent les unes des autres en termes de vidéo et la capacité audio, les polices, la taille de la mémoire, ainsi que la syntaxe BASIC disponible.

Le moyen le plus simple pour définir la cible d'émulation, utilisez une option prédéfinie. Par exemple, exécutez pcbasic -- preset=pcjr.

Les autres préréglages cibles d'émulation disponibles sont :

Préréglé	Cible d'émulation
pcjr	IBM PCjr avec cartouche BASIC, y compris la vidéo PCjr et le son à 3 voix capacités et syntaxe BASIC étendue.
tandy	Tandy 1000 avec GW-BASIC, y compris la vidéo Tandy et le son à 3 voix capacités et syntaxe BASIC étendue.
olivetti	Olivetti M24 ou AT&T PC 6300.
cga	IBM ou compatible avec Color/Graphics Adapter et un moniteur composite. Cela permet l'émulation composite colorburst.
ega	IBM ou compatible avec Extended Graphics Adapter.
vga	IBM ou compatible avec Video Graphics Array.
mda	IBM ou compatible avec Monochrome Display Adapter et teinté vert moniteur monochrome.
hercules	IBM compatible avec Hercules Graphics Adapter et monochrome teinté vert moniteur.
strict	Choisissez une compatibilité stricte avec GW-BASIC plutôt que la commodité, la sécurité et la rime ou la raison.

Les préréglages sont des groupes d'options définies dans le fichier de configuration par défaut. Vous pouvez créer vos propres préréglages en créant un entête dans votre fichier de configuration privé avec le nom du nouveau préréglage, suivi des options que vous souhaitez appliquer.

Par exemple, si vous définissez : `[mon_preset] vidéo=vga syntaxe=pcjr` vous pouvez maintenant exécuter `pcbasic --preset=my_preset` pour démarrer une émulation d'un hypothétique machine avec une carte vidéo VGA exécutant PCjr Cartridge BASIC.

Les Fonctionnalités de compatibilité GW-BASIC PC-BASIC vise un très haut niveau de compatibilité avec GW-BASIC. Cependant, certaines fonctionnalités de compatibilité sont désactivées par défaut pour des raisons de commodité ou de sécurité.

Ces fonctionnalités peuvent être activées à l'aide d'options de ligne de commande individuelles.

Le plus haut niveau de la compatibilité avec GW-BASIC peut être obtenue en définissant `preset=strict`, qui éteint toutes les fonctionnalités de commodité et de sécurité qui causent des différences avec GW-BASIC.

## **Pages de code**

PC-BASIC prend en charge un grand nombre de pages de codes héritées qui étaient courantes à l'époque où GW-BASIC était populaire, y compris les pages de code de jeu de caractères à deux octets utilisées pour le chinois, le japonais et le coréen.

Vous pouvez sélectionner votre page de code en utilisant l'option `codepage=`.

Par exemple, `codepage=936` sélectionne la page de code GBK couramment utilisée par le chinois continentale.

PC-BASIC chargera et enregistrera tous les fichiers de programme comme s'ils étaient encodés dans la page de code que vous sélectionner.

Il est également possible de charger et de sauvegarder des programmes dans un codage standard en activant la possibilité d'encodage de texte.

Par exemple, si `--text-encoding=utf-8` est défini, la source du programme en texte brut sera enregistré et chargé dans l'encodage standard UTF-8.

Veillez noter que vous devrez toujours sélectionner une page de code qui fournit tous les caractères Unicode dont votre programme a besoin.

Notez que PC-BASIC n'implémente pas les fonctionnalités suivantes pour certains de ces pages de codes :

## **Texte bidirectionnel**

Tout le texte est imprimé de gauche à droite indépendamment de la page de code sélectionnée. Pour écrire des chaînes dans une langue qui s'écrit de droite à gauche, la suite logique des caractères doit être inversée afin que la commande apparaisse visuellement correcte.

Bien que cela ne soit pas pratique, cela va dans le sens du comportement de GW-BASIC. Cela affecte les pages de code marquées d'un B dans le tableau.

La combinaison de caractères PC-BASIC reconnaît les points de code à un seul octet (où chaque glyphe s'affiche sur une seule cellule à l'écran) et des points de code à deux octets (où un seul glyphe occupe deux cellules à l'écran).

Combinaison de caractères (tels que les signes diacritiques combinés de les pages de codes 874 et 1258 ) ne s'affichent donc pas correctement : au lieu d'être combiné avec leur caractère de base précédent en un seul glyphe combiné, tel que les combinaisons seront affichées sous forme de glyphes séparés.

Si disponible, l'alternative des pages de code avec des caractères précomposés donneront de meilleurs résultats. Cela affecte le code pages marquées d'un C dans le tableau.

Les pages de code suivantes sont disponibles.

PC-BASIC utilise la page de codes Microsoft OEM numéro où il n'y a pas d'ambiguïté.

Les pages de code sont censées être en accord avec Microsoft sources pour les gammes &h80 – &hFF.

Les plages &h00 – &h1F et &h7F sont implémentées comme les caractères graphiques spéciaux IBM où certaines sources de page de codes les répertorie comme caractères de contrôle correspondants.

Pour les pages de code non officielles et celles avec des conflits de numérotation, les noms de pages de code sont utilisés à la place des chiffres.

<i>codepage_id</i>	<b>Codepage</b>	<b>Languages</b>	<b>Notes</b>
437	DOS Latin USA	English	
720	Transparent ASMO	Arabic	<u>B</u>
737	DOS Greek	Greek	
775	DOS Baltic Rim	Estonian, Latvian and Lithuanian	
850	DOS Latin 1	Western European languages	
851	DOS Greek 1	Greek	
852	DOS Latin 2	Central European languages	
853	DOS Latin 3	Southern European languages	
855	DOS Cyrillic 1	Serbian, Macedonian and Bulgarian	
856	DOS Hebrew	Hebrew	<u>B</u>
857	DOS Latin 5	Turkish	
858	DOS Latin 1 with Euro	Western European languages	
860	DOS Portuguese	Portuguese	
861	DOS Icelandic	Icelandic	
862	DOS Hebrew	Hebrew	<u>B</u>
863	DOS Canadian French	French	
864	DOS Arabic	Arabic	<u>B</u>
865	DOS Nordic	Danish and Norwegian	
866	DOS Cyrillic 2	Russian	
868	DOS Urdu	Urdu	<u>B</u>
869	DOS Greek 2	Greek	
874	TIS-620	Thai	<u>C</u>
932	Shift-JIS (variant)	Japanese	
934	DOS/V Korea	Korean	
936	GBK; GB2312/EUC-CN superset	Simplified Chinese	
938	DOS/V Taiwan	Traditional Chinese	
949	IBM-PC Korea KS; EUC-KR superset	Korean	
950	Big-5 (variant)	Traditional Chinese	
1258	Vietnamese	Vietnamese	<u>C</u>



alternativnyj	GOST Alternativnyj Variant	Russian	
armscii8a	ArmSCII-8a; FreeDOS cp899	Armenian	
big5-2003	Big-5 (Taiwan 2003)	Traditional Chinese	
big5-hkscs	Big-5 (Hong Kong 2008)	Traditional Chinese	
georgian-academy	Academy Standard	Georgian	
georgian-ps	Parliament Standard	Georgian	
iransystem	Iran System	Persian	<i>B</i>
kamenicky	Kamenický; cp895	Czech	
koi8-r	KOI8-R	Russian	
koi8-ru	KOI8-RU	Ukrainian, Belarusian, Russian	
koi8-u	KOI8-U	Ukrainian, Russian	
mazovia	Mazovia; cp667, 991, 790	Polish	
mik	MIK, FreeDOS cp3021	Bulgarian	
osnovnoj	GOST Osnovnoj Variant	Russian	
ruscii	RUSCII	Ukrainian, Russian	
russup3	Cornell Russian Support for DOS v3	Russian	
russup4ac	Exceller Software Russian Support for DOS v4 Academic	Russian	
russup4na	Exceller Software Russian Support for DOS v4 Non-Academic	Russian	
viscii	VISCII, FreeDOS cp30006	Vietnamese	

Vous pouvez ajouter des pages de code personnalisées à PC-BASIC, en ajoutant un fichier avec son mappage à Unicode au répertoire `codepage/`.

## Polices

PC-BASIC émule les polices distinctives des machines compatibles IBM.

Les polices ROM des adaptateurs IBM et Tandy d'origine (qui sont dans le domaine public dans un certain nombre de pays) ont été inclus dans PC-BASIC. Ceux-ci fournissent l'émulation la plus précise. Cependant, les ROM de polices ne comprenaient qu'une seule page de code - DOS Latin USA 437.

PC-BASIC utilise par défaut une police dont le style est très similaire à celui de la police IBM VGA mais qui a pris en charge de nombreuses autres pages de codes, en particulier les alphabets occidentaux et moyen-orientaux.

Le chinois, le japonais et le coréen sont pris en charge par des glyphes "pleine largeur" qui prennent l'espace de deux caractères réguliers. Il est possible de changer le choix de la police à l'aide de l'option `font=`.

Vous pouvez fournir une liste de polices, où la dernière police spécifiée est la plus préférée.

PC-BASIC lit les polices dans une variante du format.hex introduit par UniFont. C'est facile de définir des polices personnalisées dans ce format : il peut être modifié dans un éditeur de texte standard.

Voir UniFont Projet d'outil auteur.

Vous pouvez ajouter des polices personnalisées à PC-BASIC en les installant dans le sous-répertoire font/ du répertoire d'installation de PC-BASIC.

Par défaut, les polices suivantes sont disponibles :

<i>font_name</i>	<b>Name</b>	<b>Sizes</b>	<b>Codepages</b>
default	PC-BASIC default font	8, 14, 16	all bundled codepages
cga	IBM Colour/Graphics Adapter font	8	437 only
mda	IBM Monochrome Display Adapter font	14	437 only
vga	IBM Video Graphics Array font	8, 14, 16	437 only
olivetti	Olivetti/AT&T font	16	437 only
tandy1	Tandy-1000 font old version	8	437 only
tandy2	Tandy-1000 font new version	8	437 only

Si tous les glyphes ne sont pas trouvés dans la ou les polices spécifiées, la police par défaut est utilisée comme alternative. Les noms de police freedos , univga et unifont sont traités comme des synonymes de default sauf si une police portant l'un de ces noms est disponible. Ce comportement est obsolète.

### **Redirection des E/S**

PC-BASIC prend en charge la redirection d'entrée et de sortie de la manière que GW-BASIC : sortie redirigée avec l'option **output=** sera envoyée à l'écran ainsi que le fichier spécifié, tandis que l'entrée redirigé avec **input=** est extrait uniquement du fichier spécifié.

Notez que la sortie d'écran via le SCRNL, l'appareil et la saisie au clavier via le KYBD : l'appareil ne sont pas redirigés.

Des dossiers sont lus et écrits dans la page de codes définie avec PC-BASIC.

Notez qu'il est également possible d'utiliser la fonction de votre système d'exploitation pour rediriger la sortie de la console à l'aide des opérateurs < et >.

Il est préférable de définir interface=none pour que les E/S soient redirigées via la console. Cela produira des fichiers dans l'encodage standard de votre console, qui est souvent UTF-8 sous Unix et Windows-1252 sous Windows.

## Interface de ligne de commande

Vous pouvez exécuter PC-BASIC en tant qu'interface de ligne de commande en définissant `interface=cli` (ou `-b`) option.

Aucune fenêtre ne s'ouvrira : vous pouvez taper des commandes BASIC directement dans votre Invite de commande/Terminal.

Utilisez les touches fléchées horizontales pour vous déplacer sur la ligne actuelle où vous vous trouvez. Utilisez les touches fléchées verticales pour afficher les lignes d'écran au-dessus et au-dessous.

Copier et coller ne sont disponibles que si le shell appelant les fournit.

Sous Windows, `Ctrl + Break` terminera PC-BASIC immédiatement. Vous pouvez utiliser `Ctrl + C` pour interrompre le programme. La clé de fin de fichier combinaison (`Ctrl + D` sous Unix, `Ctrl + Z` sous Windows) quittera PC-BASIC.

Vous pouvez utiliser l'interface de ligne de commande pour exécuter une ou quelques commandes BASIC directement, comme alors: `moi@maboîte$ pcbasic -c '?1+1' 2 moi@mabox$`

À des fins de script, il est également possible d'exécuter PC-BASIC sans aucune interface en définissant `interface=aucun` ou `-n`. Si cela est défini, PC-BASIC prendra l'entrée et enverra la sortie à la console en UTF-8 sans autre modification. Ceci est utile en combinaison avec la redirection.

## Interface textuelle

Une interface texte plein écran est également disponible : activez-la en définissant `interface=text` (ou `-t`). L'interface textuelle est très similaire à l'interface graphique par défaut, mais fonctionne en invite de commande ou avec votre fenêtre de terminal.

Les modes d'écran graphiques peuvent être utilisés dans l'interface de texte et de ligne de commande, mais seul le texte sur l'écran sera visible. De nombreuses combinaisons de touches, `pre`, `Ctrl` et `Alt` ne sont pas disponibles.

Les interfaces de texte et de ligne de commande tenteront d'utiliser le haut-parleur du PC pour le son. Seulement le son d'une seule voix peut être produit de cette façon.

Sur les systèmes Linux sous X11, vous devrez peut-être pour installer l'utilitaire de bip et activer le pilote ou l'émulation du haut-parleur PC ; l'accès direct aux haut-parleurs est souvent limité aux connexions `root` ou `tty`, et sur les systèmes Ubuntu, il est désactivé par défaut.

## Guide de configuration

Cette documentation explique comment modifier les paramètres et les options de PC-BASIC.

### Modification des paramètres

PC-BASIC possède un certain nombre de paramètres qui modifient son fonctionnement. Les paramètres peuvent être modifiés en définissant des options sur la ligne de commande ou en modifiant le fichier de configuration.

Quelle que soit la méthode, les options ont le même nom et la même syntaxe. Dans ce qui suit, nous allons souvent se référer à un paramètre d'option particulier ; rappelez-vous que vous pouvez le définir à partir de la ligne de commande ainsi qu'à partir du fichier de configuration.

### Options de ligne de commande

Vous pouvez entrer des options de ligne de commande si vous démarrez PC-BASIC à partir de l'invite de commande, de la console ou du terminal (l'invite C:\> sous Windows), en fournissant le option avec deux tirets devant, comme ceci :

```
pcbasic --preset=tandy --ctrl-c-break=Vrai
```

Sur la ligne de commande, vous pouvez omettre l'expression =True qui est courante dans la commutation d'options.

Certaines options ont un autre nom court composé d'une seule lettre précédée par un seul tiret, que vous pouvez utiliser sur la ligne de commande. Vous pouvez combiner plusieurs courtes options avec un seul tiret.

### Fichiers de configuration

Vous pouvez modifier les options en ajoutant ou en supprimant des lignes dans votre fichier de configuration local, ce qui peut être trouvé à l'emplacement suivant:

Windows

```
%AppData%\pcbasic-2.0\PCBASIC.INI
```

OS X

```
~/Bibliothèque/Application Support/pcbasic-2.0/PCBASIC.INI
```

Linux

```
~/config/pcbasic-2.0/PCBASIC.INI
```

Modifier une option dans le fichier de configuration en ajoutant une ligne dans la section nommée [pcbasic] , ainsi:

```
[pcbasique] pré-réglé=tandy ctrl-c-break=Vrai
```

Vous ne pouvez pas utiliser d'arguments positionnels ou le nom court des options dans le fichier de configuration.

Vous ne pouvez pas non plus omettre l'expression =True. Le fichier de configuration doit être un fichier texte encodé en ASCII ou UTF-8.

## Synopsis

```
pcbasic [program|package [output]] [--allow-code-poke[=True|=False]] [--aspect=x,y] [-b]
[--border=width] [-c=statement[:statement...]] [--caption=title] [--cas1=type:value]
[--codepage=codepage_id[:nobox]] [--config=config_file] [--com1=type:value]
[--com2=type:value] [--convert={A|B|P}] [--mouse-clipboard[=True|=False]]
[--ctrl-c-break[=True|=False]] [--current-device={CAS1|@|A|B... |Z}]
[--debug[=True|=False]] [--dimensions=x,y] [-d] [--double[=True|=False]]
[-e=statement[:statement...]] [--exec=statement[:statement...]]
[--extension=module_name[,module_name... ]] [--font=font_name[,font_name... ]]
[--fullscreen[=True|=False]] [-h] [--help] [--hide-listing=line_number]
[--hide-protected[=True|=False]] [-i={input_file|{STDIO|STDIN}[:RAW]}]
[--input={input_file|{STDIO|STDIN}[:RAW]}] [--interface=[none|cli|text|graphical]]
[-k=keystring] [--keys=keystring] [-l=program] [--load=program] [--logfile=log_file]
[--lpt1=type:value] [--lpt2=type:value] [--lpt3=type:value] [-f=number_of_files]
[--max-files=number_of_files] [--max-memory=max_memory[,basic_memory_blocks]]
[-s=record_length] [--max-reclen=record_length]
[--monitor={rgb|composite|green|amber|grey|mono}]
[--mount=[drive:path[,drive:path... ]]] [-n] [-o=output_file[:append]]
[--output=output_file[:append]] [--peek=[seg:addr:val[,seg:addr:val... ]]]
[--preset=option_block] [--prevent-close[=True|=False]] [-q] [--quit[=True|=False]]
[--reserved-memory=number_of_bytes] [--resume[=True|=False]] [-r=program]
[--run=program] [--scaling={smooth|crisp|native}] [--serial-buffer-size=size]
[--shell=[shell-executable]] [--soft-linefeed[=True|=False]] [--sound[=True|=False]]
[--state=state_file] [--syntax={advanced|pcjr|tandy}] [-t] [--term=terminal_program]
[--text-width={40|80}] [--text-encoding=[encoding]] [--utf8[=True|=False]] [-v]
[--version] [--video=adapter] [--video-memory=size] [-w] [--wait[=True|=False]]
[--options=gwbasic_options]
```

## Arguments positionnels

Les arguments positionnels doivent précéder toutes les options, ne doivent pas commencer par un tiret -. N'importe quel les arguments positionnels qui suivent les options seront ignorés.

## Programme

Si un programme.BAS est spécifié comme premier argument positionnel, il sera exécuté. Les options -- run , -- load et --convert remplacent ce comportement.



## Package

Si un package de fichier zip ou un répertoire est spécifié comme premier argument de position, tout le fichier de configuration PCBASIC.INI contenu sera chargé ; généralement, il exécutera un programme fichier dans le package.

Toutes les autres options de ligne de commande remplaceront le package fichier de configuration, notez en particulier le potentiel des --run , --load et -- convertir les options pour modifier le comportement du package.

## Production

Si un deuxième argument de position est spécifié, il définit le fichier de sortie pour le format de fichier conversion. Cet argument est ignoré sauf si l'option --convert est donnée.

Choix --allow-code-poke[=Vrai|=Faux]

Autoriser les programmes à POKE dans la mémoire de code.

--aspect=x,y

Réglez le format d'affichage sur x:y. N'a d'effet que s'il est combiné avec -- interface=graphique.

-b

Utilisez l'interface de ligne de commande. Ceci est identique à --interface=cli.

--border=largeur

Définissez la largeur de la bordure de l'écran sous forme de pourcentage de 0 à 100. Le pourcentage fait référence à la largeur totale des bordures des deux côtés en tant que fraction de l'écran utilisable largeur. N'a d'effet que s'il est combiné avec --interface=graphical.

-c=instruction[:instruction...]

Exécutez des commandes en tant que shell. Ceci est un raccourci pratique et identique à -- interface=aucun -- quit=True --exec=instruction[:instruction...].

--caption=titre

Définissez la légende de la barre de titre de la fenêtre PC-BASIC. Le titre par défaut est PC-BASIC.

--cas1=type:valeur

Attachez une ressource au CAS1 : périphérique de cassette. type:la valeur peut être

WAV :fichier\_wav Connectez-vous au fichier RIFF Wave wav\_file avec des données modulées dans IBM PC format cassette.

CAS : fichier\_cas Connectez-vous à l'image de bande PCE/PC-BASIC CAS cas\_file.

--codepage=codepage\_id[:nobox]

Charger la page de codes spécifiée. La page de code détermine quels caractères sont associé à un octet de caractère donné ou, dans le cas de pages de code à deux octets, deux octets de caractères.

Les pages de code disponibles sont stockées dans le répertoire codepage/ ; par par défaut, ce sont :

437	720	737	775	806	850
851	852	853	855	856	857
858	860	861	862	863	864
865	866	868	869	874	932
934	936	938	949	950	1258
				georgian-	georgian-
alternativ	armscii8a	big5-2003	big5-hkscsacademy		ps
iransystem	iscii-as	iscii-be	iscii-de	iscii-gu	iscii-ka
iscii-ma	iscii-or	iscii-pa	iscii-ta	iscii-te	kamenicky
koi8-r	koi8-ru	koi8-u	mazovia	mik	osnovnoj
pascii	ruscii	russup3	russup4ac	russup4na	viscii

Voir la liste des pages de code dans le Guide de l'utilisateur pour plus de détails.

Le spécificateur nobox désactive la reconnaissance de dessin de boîte pour le jeu de caractères à deux octets des codepage. Par défaut, les séquences de caractères de dessin de boîte sont reconnues par un algorithme qui n'est pas aussi intelligent qu'il le pense, et affiche sous forme de dessin de boîte plutôt que en tant que caractères DBCS.

Si aucune boîte n'est définie, ils seront affichés en tant que DBCS. --config=fichier\_config

### Lire un fichier de configuration

La configuration par défaut du système est toujours lue en premier, mais tout groupe prédéfini d'options dans un fichier de configuration remplace l'ensemble équivalent par défaut groupe prédéfini.

--com1=type:valeur

Attachez une ressource au COM1 : périphérique série. type:vaue peut être l'un des Suivant.

PORT : nom\_appareil

Connecte à un périphérique série. device\_name peut être un nom de périphérique tel que COM1 ou /dev/ttyS0 ou un nombre, où le premier port série est le numéro 0.

SOCKET:hôte:socket

Connecte à un socket TCP sur un hôte distant ou local.



RFC2217:hôte:socket

Connecte en utilisant le protocole RFC2217 à un socket TCP sur un réseau distant ou local héberger.

STDIO :[CRLF]

Connecte aux E/S standard du shell appelant. Si CRLF est spécifié, PC-BASIC remplace les caractères CR par LF sur sa sortie et LF par CR sur son entrée.

Ceci est plus intuitif sur les shells Unix. Lorsque vous utilisez une console Unix, vous devez utiliser stty -icanon pour permettre à PC-BASIC de lire correctement l'entrée.

Si cette option n'est pas spécifiée, le périphérique COM1 : n'est pas disponible.

--com2=type:valeur

Attachez une ressource au COM2 : périphérique série. Voir --com1.

--convert={A|B|P}

Convertissez le programme dans l'un des formats suivants :

<b>A</b>	<b>Plain text</b>	
<b>B</b>		<b>Tokenised</b>
<b>P</b>		<b>Protected</b>

Si la sortie n'est pas spécifiée, vous écrivez sur la sortie standard.

Si le programme n'est pas spécifié, utilisez l'argument de --run ou --load.

Si aucun de ceux-ci n'est donné, lisez à partir de la norme saisir.

Remplace --resume , --run et --load.

--mouse-clipboard[=Vrai|=Faux]

Activer les opérations du presse-papiers avec la souris.

Si True (par défaut), sélectionnez le texte avec la gauche bouton de la souris pour copier et coller avec le bouton central de la souris.

--ctrl-c-break[=Vrai|=Faux]

Si False , le comportement GW-BASIC où Ctrl + C interrompt AUTO et INPUT mais pas l'exécution du programme ou LIST.

Si True , traitez Ctrl + C exactement comme Ctrl + Break et Ctrl + Scroll Lock lorsque

--interface=graphique

Avec --interface={text|cli} , Ctrl + C est toujours traité comme Ctrl + Break. La valeur par défaut est True.

--current-device={CAS1|@|A|B... |Z}

Réglez le périphérique actuel sur la lettre de lecteur PC-BASIC indiquée ou CAS1 pour la cassette dispositif. L'appareil choisi doit être monté à un emplacement réel en utilisant --mount (ou --cas1 si le périphérique cassette est choisi).

--debug[=Vrai|=Faux]

Option développeur - à n'utiliser que si vous savez ce que vous faites. Activer l'extension de débogage.

--dimensions=x,y

Définissez les dimensions de la fenêtre sur x par y pixels. Cela remplace --scaling=native et -- aspect. N'a d'effet que s'il est combiné avec --interface=graphical.

-d --double[=Vrai|=Faux]

Activer les fonctions mathématiques transcendantales à double précision. Ceci est équivalent au /d option dans GW-BASIC.

-e=instruction[:instruction...] ou --exec=instruction[:instruction...]

### **Exécutez les instructions BASIC.**

Les instructions sont exécutées après le chargement de n'importe quel programme mais avant d'entrer en mode direct ou de l'exécuter. Plusieurs déclarations peuvent être saisies en les séparant par deux-points :

Ceux-ci seront exécutés comme s'ils avaient été saisis comme instructions séparées, et non comme une seule instruction composée : même si des instructions telles que GOTO ou LIST sont inclus, les instructions suivantes seront toujours exécutées.

Le caractère : sera interprété comme faisant partie d'une chaîne s'il est entre guillemets simples ".

Si votre shell appelant interprète ces guillemets, vous devez leur échapper correctement.

--extension=nom\_module[,nom\_module... ]

Option développeur - à n'utiliser que si vous savez ce que vous faites. Charger le(s) module(s) d'extension.

--font=nom\_police[,nom\_police... ]

Utilisez les polices spécifiées pour l'interface. Les dernières polices spécifiées sont prioritaires, les précédents sont de secours. Les polices disponibles sont stockées dans font/.

Par défaut, les polices suivantes sont disponibles :

défaut cga mda vga olivetti tandy1 tandy2.

Les noms de police freedos , univga et unifont sont traités comme des synonymes de par défaut sauf si une police portant l'un de ces noms est disponible. Ce comportement est obsolète et ces synonymes seront supprimés dans une future version. Voir la liste des polices dans le Guide de l'utilisateur pour plus de détails.

--fullscreen[=Vrai|=Faux]

Mode plein écran. N'a d'effet que s'il est combiné avec --interface=graphical.

-h --aide

Afficher un message d'utilisation et d'aide.

--hide-listing=numéro\_ligne

Désactiver la liste et l'enregistrement en texte brut des lignes au-delà de line\_number , comme dans GW-BASIC au-delà de 65530. À utiliser avec précaution car cela permet l'exécution de lignes de code cachées. La valeur par défaut est de lister toutes les lignes.

--hide-protected[=Vrai|=Faux]

Désactivez la liste et l'enregistrement en texte brut des fichiers protégés, comme dans GW-BASIC. Utiliser avec attention car cela permet l'exécution de lignes de code cachées.

-i={ fichier\_entrée|{STDIO|STDIN}[:RAW]} ou --input={ fichier\_entrée|{STDIO|STDIN}[:RAW]}

Récupère l'entrée clavier de input\_file , sauf si KYBD: est lu explicitement. Saisi depuis KYBD : les fichiers sont toujours lus depuis le clavier, selon le comportement GW-BASIC.

Si input\_file est STDIO: ou STDIN: , l'entrée au clavier sera lue à partir de l'entrée standard.

Si RAW est spécifié, l'entrée sera traitée comme des octets de page de codes. Sinon, il sera traité comme l'encodage des paramètres régionaux (probablement UTF-8).

--interface=[aucun|cli|texte|graphique]

Choisissez le type d'interface. Toutes les interfaces ne seront pas disponibles sur tous les systèmes. Les types d'interface suivants peuvent être disponibles :

none Filtre à utiliser avec des tuyaux. Aussi -n.

cli Interface de ligne de commande. Aussi -b.

texte Interface texte ANSI. Aussi -t.

graphical graphique SDL2.

Les valeurs suivantes pour cette option sont obsolètes :

ansi Interface texte ANSI. Synonyme de texte.

sdl2 Interface graphique SDL2. Synonyme de graphique.

pygame Interface graphique PyGame. Veuillez utiliser graphique à la place.

curses interface texte NCurses. Veuillez utiliser du texte à la place. La valeur par défaut est graphical.

-k=keystring --keys=keystring

Insérez la chaîne de touches dans la mémoire tampon du clavier. keystring peut contenir des codes d'échappement comme `\r` pour retour, `\n` pour saut de ligne et `\xXX` pour entrer `CHR$(&HXX)`, keystring peut contenir des codes e-ASCII pour indiquer les pressions de touches qui n'ont pas d'encodage de caractère. Par exemple, `\0\x0F` indique Maj+Tabulation.

-l=programme ou --load=programme

Démarrez en mode direct avec le programme BASIC chargé.

--logfile=fichier\_journal

Écrivez les messages d'erreur et d'avertissement dans log\_file au lieu de stderr.

--lpt1=type:valeur

Déterminez où va la sortie lors de l'écriture sur le LPT1 : périphérique parallèle. type:la valeur peut être

PRINTER:[printer\_name][:trigger]

La sortie est écrite sur une imprimante. Si printer\_name n'est pas spécifié, la valeur par défaut est l'imprimante utilisée. Les imprimantes Windows et CUPS sont prises en charge. L'imprimante sera activée lorsqu'un fichier est envoyé sur LPT1 : le déclencheur définit un gâchette pour activer l'imprimante :

line Après chaque saut de ligne.

page Après chaque saut de page.

close Aucun déclencheur supplémentaire

La valeur par défaut est close.

FILE:nom\_fichier

La sortie est écrite dans un fichier ou un périphérique de caractères tel que `/dev/stdout` sous Unix ou LPT1 sous Windows.

STDIO :[CRLF]

La sortie est écrite sur la sortie standard du shell appelant. Si CRLF est spécifié, PC-BASIC remplace les caractères CR par LF sur sa sortie. C'est plus intuitif sur les shells Unix.

PARPORT : numéro\_port

La sortie est écrite sur un port parallèle Centronics, où port\_number est 0 pour le premier port parallèle, etc. Cette option ne fonctionne qu'avec les ports parallèles physiques. Pour écrire sur une imprimante Windows ou un autre périphérique mappé avec NET USE LPT1 : , utilisez FILE:LPT1 à la place.

La valeur par défaut est IMPRIMANTE : , de sorte que la sortie aille vers l'imprimante par défaut spécifiée par le système opérateur.

--lpt2=type:valeur

Attachez une ressource au LPT2 : périphérique parallèle. Voir --lpt1. A noter que, contrairement à LPT1 : , les imprimantes connectées à LPT2 : ne sont pas activées lorsqu'un programme se termine. Si cette option n'est pas spécifiée, LPT2 : n'est pas disponible.

--lpt3=type:valeur

Attachez une ressource au LPT3 : périphérique parallèle. Voir --lpt1. A noter que, contrairement à LPT1 : , les imprimantes connectées à LPT3 : ne sont pas activées lorsqu'un programme se termine. Si cette option n'est pas spécifiée, LPT3 : n'est pas disponible.

-f=nombre\_de\_fichiers --max-files=nombre\_de\_fichiers

Définissez le nombre maximum de fichiers ouverts sur number\_of\_files. Ceci est équivalent à l'option /f dans GW-BASIC. La valeur par défaut est 3.

--max-memory=max\_memory[,basic\_memory\_blocks]

Définissez la taille maximale du segment de mémoire de données sur max\_memory et la taille maximale de la mémoire de données disponible pour BASIC à basic\_memory\_blocks\*16.

Dans PC-BASIC, le minimum de ces valeurs est simplement la taille de la mémoire de données ; les deux valeurs sont autorisés pour la compatibilité avec l'option /m dans GW-BASIC.

-s=longueur\_enregistrement --max-reclen=longueur\_enregistrement

Définissez la longueur d'enregistrement maximale pour les fichiers RANDOM sur record\_length. La valeur par défaut est 128, le maximum est 32767. Cela équivaut à l'option /s dans GW-BASIC.

--monitor={rgb|composite|vert|ambre|gris|mono}

Définit le type de moniteur à émuler. Les types disponibles sont :

rgb Moniteur couleur RVB (par défaut).

Composite Moniteur couleur composite.

green Moniteur monochrome teinté de vert.

amber Moniteur monochrome teinté d'ambre.

grey monochrome gris en niveaux de gris.

mono Moniteur monochrome teinté en vert (identique au vert ).

Sur SCREEN 2 avec --video={pcjr|tandy|cga} , --monitor=composite active artefacts de couleur (bruts).

--mount=[lecteur:chemin[,lecteur:chemin... ]]

Attribuez le chemin d'accès à la lettre de lecteur drive :. Le chemin peut être absolu ou relatif.

Si cette option n'est pas spécifiée : sous Windows, toutes les lettres de lecteur Windows seront attribuées à Lettres de lecteur PC-BASIC ; sur d'autres systèmes, le répertoire de travail courant est attribué à Z :

Si cette option est spécifiée mais vide, ne montez aucun lecteur (à l'exception du dispositif @: ).

-n

Exécutez PC-BASIC comme filtre de ligne de commande. Identique à --interface=none.

-o=fichier\_sortie[:append] ou --output=fichier\_sortie[:append]

Envoie la sortie d'écran à output\_file , sauf si SCRN: est écrit explicitement. Sortie vers SCRN : les fichiers seront toujours affichés à l'écran, comme dans GW-BASIC.

Si le spécificateur append est donné, le fichier de sortie est ajouté plutôt qu'écrasé.

Si output\_file est STDIO: ou STDOUT: , la sortie d'écran sera envoyée à la sortie standard.

--peek=[seg:addr:val[,seg:addr:val... ]]

Définissez les valeurs prédéfinies PEEK. Si défini, DEF SEG seg : ? PEEK(addr) renverra val.

--preset=bloc\_options

Charger les options de préréglage de la machine. Une option prédéfinie correspond à une section définie dans un fichier de configuration par un nom entre crochets, comme [cette]

--preset=this

Chargera tous les paramètres définis dans cette section. Les préréglages disponibles dépendent de votre fichier de configuration. Consultez la liste des préréglages par défaut dans le Guide de l'utilisateur.

--prevent-close[=Vrai|=Faux]

Supprime l'évènement de fermeture de fenêtre. Cela permet à BASIC de capturer des combinaisons de touches qui fermer normalement la fenêtre. Interface graphique uniquement. Par défaut, le fonctionnement de la combinaison de touches du système pour fermer une fenêtre (généralement Alt + F4 ) termine PC- DE BASE.

Définissez --prevent-close pour permettre à BASIC de capturer cette combinaison de touches à la place. Ceci est utile si votre programme utilise cette combinaison de touches.

-q --quit[=Vrai|=Faux]

Quitte l'interpréteur lorsque l'exécution s'arrête. S'il est combiné avec --run , PC-BASIC se ferme lorsque le programme se termine. S'il est défini en mode direct, PC-BASIC quitte après la première commande réalisé.

--reserved-memory=nombre\_d'octets

Réservez number\_of\_bytes de mémoire au bas du segment de données. Par compatibilité avec GW-BASIC. La valeur par défaut est 3429 octets. La diminution de cette valeur rend plus d'espace de chaîne et de variable disponible pour une utilisation par les programmes.

--resume[=Vrai|=Faux]

Reprendre à partir de l'état enregistré. Remplace --run et --load.

-r=programme ou --run=programme

Exécutez le programme spécifié. Remplace --load.

--scaling={lisse|net|natif}

Choisissez la méthode de mise à l'échelle.

smooth L'affichage est mis à l'échelle en douceur à la plus grande taille qui permet le bon ratio d'aspect.

crisp L'affichage est mis à l'échelle à la même taille qu'avec smooth , mais sans lissage.

native La mise à l'échelle et le format d'image sont optimisés pour la taille de pixel native de l'écran, sans lissage.

--scaling=native remplace --aspect.

La valeur par défaut est lisse. N'a d'effet que s'il est combiné avec --interface=graphical.

--serial-buffer-size=taille

Définissez la taille du tampon d'entrée série. La valeur par défaut est 256. S'il est défini sur 0, les communications série sont désactivées.

--shell=[shell-exécutable]

Activer l'instruction SHELL pour exécuter l'interpréteur de commandes du système d'exploitation shell-exécutable. L'exécutable shell-exécutable doit prendre en charge les Conventions MS-DOS d'appel de COMMAND.COM, en particulier son commutateur /C.

Exemple de commande les interpréteurs sont CMD.EXE sous Windows et "wine cmd.exe" sous Unix.

Si shell-exécutable est vide (comme c'est le cas par défaut), l'instruction SHELL est désactivée.

--soft-linefeed[=Vrai|=Faux]

Ne traite pas LF dans les fichiers texte et programme comme un saut de ligne. Cela permet le plus haut niveau de compatibilité avec les fichiers GW-BASIC.

Si cette option est définie, tout texte Linux ou Mac les fichiers doivent être convertis en texte DOS avant de les utiliser avec PC-BASIC.

--sound[=True|=False]

False Supprime la sortie audio.

True Son de sortie, si un pilote audio est disponible (par défaut).

Si le son est activé, PC-BASIC essaiera d'abord d'utiliser la bibliothèque SDL2 ; s'il n'est pas disponible, il essaiera

PortAudio.

Si aucun n'est disponible, le son sera désactivé.

Les valeurs suivantes pour cette option est obsolète :

none Supprime la sortie audio. Utilisez False à la place.

interface Utilise le moteur de son natif de l'interface. Utilisez True à la place.

SDL2 Utilisez le générateur de sons SDL2.

portaudio Utilisez le générateur de son PortAudio.

bip Utilisez le haut-parleur intégré.



--state=fichier\_état

Définissez le fichier d'état de sauvegarde sur state\_file. La valeur par défaut est pcbasic.session dans l'application Répertoire de données.

--syntax={ avancé|pcjr|tandy }

Choisissez le dialecte BASIC.

Les dialectes disponibles sont :

advanced Microsoft GW-BASIC et IBM BASICA

pcjr Cartouche IBM PCjr BASIC

tandy Tandy 1000 GW-BASIQUE.

La valeur par défaut est advanced.

-t

Utiliser l'interface textuelle. Identique à --interface=text.

--term=programme\_terminal


Définissez le programme de terminal exécuté par la commande PCjr TERM sur terminal\_program. N'a d'effet qu'avec --syntax={pcjr|tandy}.

--text-width={40|80}

Définissez le nombre de colonnes en mode texte au démarrage. La valeur par défaut est 80.

--text-encoding=[encodage]

Définissez l'encodage du texte. Fichiers texte (c'est-à-dire programmes en texte brut et fichiers ouverts pour INPUT et OUTPUT ) stockés sur un périphérique de disque sera supposé être encodé dans encoding.

Exemples de valides les encodages sont utf-8 , utf-16 , latin-1. Veuillez vous assurer que tous les caractères de la page de codes actuelle sont inclus dans l'encodage choisis; si ce n'est pas le cas, ces caractères seront remplacés par  ou ?.

Si le codage n'est pas défini, les fichiers texte seront traités comme des octets bruts dans le PC-BASIC actuel de codes de page.

--utf8[=Vrai|=Faux]

Définissez l'encodage du texte sur UTF-8. Cette option est obsolète et ignorée si --text-encoding est défini. Utilisez --text-encoding=utf-8 à la place.

-v ou --version

Imprimez la chaîne de version PC-BASIC et quittez.

--video=adapter

Définissez la carte vidéo à émuler. Adaptateurs disponibles :

vga	Video Graphics Array
ega	Enhanced Graphics Adapter
cga	Color/Graphics Adapter
mda	Monochrome Display Adapter
hercules	Hercules Graphics Adapter
pcjr	IBM PCjr graphics
tandy	Tandy 1000 graphics
olivetti	Olivetti M24 graphics

La valeur par défaut est vga.

--video-memory=size

Définissez la quantité de mémoire vidéo émulée disponible. Cela affecte le nombre de pages vidéo utilisables. Sur PCjr et Tandy, cela peut être modifié au moment de l'exécution via l'instruction CLEAR ; au moins 32768 doivent être disponibles pour accéder à l'ÉCRAN 5 et ÉCRAN 6.

La valeur par défaut est 16384 ou PCjr et Tandy et 262144 sur une autre machine préconfigurée.

-w --wait[=True|=False]

Si True , PC-BASIC attend une frappe avant de fermer la fenêtre en quittant. Fonctionne seulement pour --interface=graphical ou --interface=text. La valeur par défaut est False. -

-options=gwbasic\_options

Définissez les commutateurs de ligne de commande de style GW-BASIC. Il s'agit d'une option pratique pour faciliter la migration depuis GW-BASIC. gwbasic\_options est une chaîne qui peut contenir les options suivantes :

/d Activer les fonctions mathématiques à virgule flottante double précision. Voir aussi --double.

/f:n Définissez le nombre maximum de fichiers ouverts. Voir aussi --max-files.

/s:n Définissez la longueur d'enregistrement maximale pour les fichiers RANDOM. Voir aussi --max-reclen.

/c:n Définissez la taille du tampon de réception pour les périphériques COM. Voir aussi --serial-buffer- Taille.

/i Allouez statiquement des blocs de contrôle de fichiers et un tampon de données. Notez que c'est déjà l'approche par défaut dans GW-BASIC et PC-BASIC de sorte que cette option n'a pas effet.

/m:n,m Définissez l'emplacement de mémoire le plus élevé sur n et la taille de mémoire BASIC maximale sur m\*16 octets. Voir aussi --max-memory.

>filename Écrire la sortie d'écran dans filename. Voir aussi --output.

>>filename Ajoute la sortie d'écran au nom de fichier. Voir aussi --output.

<filename Lire la saisie au clavier à partir du nom de fichier. Voir aussi --input.

Les commutateurs de style GW-BASIC ne sont pas sensibles à la casse. Notez que les symboles utilisés dans ces les commutateurs peuvent avoir une signification différente dans le shell à partir duquel PC-BASIC est appelé ; vous devez citer et échapper les options si nécessaire.

## Exemples

pcbasique

Démarrez PC-BASIC en mode direct, en émulant GW-BASIC/BASICA avec des graphiques VGA.

```
pcbasic --codepage=950
```

Démarrez PC-BASIC en utilisant la page de codes Big-5.

```
pcbasic Foobar.baz
```

Démarrez PC-BASIC avec le package Foobar. Chargez les paramètres du package ; d'habitude cela exécutera un programme principal contenu dans le package.

```
pcbasic Foobar.baz --convert=A --text-encoding=utf-8
```

Répertoriez le programme principal du package Foobar sur la sortie standard au format UTF-8.

```
pcbasic MONPROG.BAS --mount=A:./files,B:./morefiles
```

Montez les fichiers du sous-répertoire du répertoire actuel en tant que lecteur A: et sous-répertoire morefiles en tant que lecteur B: , puis exécutez MYPROG.BAS.

```
pcbasic --mount=A:C:\fakeflop
```

Exécutez PC-BASIC avec le répertoire Windows C:\fakeflop monté en tant que lecteur A:.

```
pcbasic Z:\INFO.BAS --preset=mda --monitor=ambre
```

Exécutez INFO.BAS dans le répertoire courant sur un MDA émulé avec une teinte ambrée moniteur.

```
pcbasic /home/me/retro/COMP.BAS --preset=cga --monitor=composite
```

Exécutez COMP.BAS stocké dans /home/me/retro sur une machine CGA émulée avec un moniteur composite.

```
pcbasic PCJRGAME.BAS --preset=pcjr -k='start\r'
```

Exécutez PCJRGAME.BAS sur un PCjr émulé et insérez les frappes démarrer Entrez.

```
pcbasic BANNER.BAS --lpt2=IMPRIMANTE :
```

Exécutez BANNER.BAS en mode par défaut avec l'imprimante par défaut attachée à LPT2 :

```
pcbasic --resume
```

Reprenez la dernière session PC-BASIC fermée.

```
pcbasic -c ?1+1
```

Exécutez la commande BASIC PRINT 1+1 dans l'interface de ligne de commande et revenez à le shell appelant.

## PC-BASIC, Guide de langage

Cette documentation décrit le langage PC-BASIC, qui vise à émuler fidèlement GW- BASIC 3.23, IBM Advanced BASIC, IBM Cartridge BASIC et Tandy 1000 GW-BASIC.

Le Guide de langage BASIC couvre la langage sujet par sujet, regroupant thématiquement les éléments de langage utilisés dans un but connexe. Veuillez vous référer au Guide de langage BASIC pour une description formelle des éléments du langage et de leur syntaxe.

## Travailler avec les programmes

Vous pouvez écrire votre programme avec un éditeur de texte comme Bloc-Notes de Windows et ensuite utiliser la commande LOAD" de PC-BASIC. Exemple LOAD"Blackjac pour charger le programme [BLACKJAC.BAS](#).

Commande	Description
AUTO	Entrer en mode de numérotation automatique des lignes
CHAIN	Chargez un nouveau programme et exécutez-le en préservant les variables communes
COMMON	Défini les variables communes
DELETE	Supprimer des lignes du programme
EDIT	Imprimer une ligne de programme à l'écran pour l'édition
LIST	Imprimer des lignes de programme à l'écran
LLIST	Impression des lignes de programme sur l'imprimante
LOAD	Lire un nouveau programme à partir du fichier
MERGE	Superposition d'un fichier de programme sur le programme en cours

<b>NEW</b>	Efface le programme en cours de la mémoire
<b>RENUM</b>	Renumérote les numéros de ligne du programme
<b>RUN</b>	Démarrer le programme en cours
<b>SAVE</b>	Stocker le programme actuel dans un fichier
<b>TRON</b>	Activer le suivi des numéros de ligne
<b>TROFF</b>	Désactiver le suivi des numéros de ligne
<b>SYSTEM</b>	Quitte l'interpréteur BASIC

Pour modifier un programme, j'utilise le Bloc-Notes de Windows et ensuite j'utiliser la commande LOAD" (F3) de PC-BASIC. Par la suite, J'exécute le programme avec la commande RUN (F2).

### **Flux de contrôle**

Un programme est normalement exécuté en commençant par son numéro de ligne le plus bas (ou le numéro de ligne appelé par RUN).

Les instructions sur une ligne sont exécutées de gauche à droite.

Lorsque toutes les instructions sur une ligne sont terminées, l'exécution passe au numéro de ligne le plus bas suivant, et ainsi de suite jusqu'à ce qu'il n'y ait plus de ligne.

Les instructions de flux de contrôle peuvent être utilisées pour modifier ce flux normal d'exécution. Les instructions END et STOP servent dans un programme à arrêter son exécution. Lorsque STOP est utilisé, un message Break est imprimé.

En mode direct, CONT peut être exécuté pour reprendre le programme là où il a été arrêté. Alors que END semble destiné à terminer le programme, cela n'empêche pas l'utilisateur de le reprendre avec CONT.

Des sauts inconditionnels peuvent être effectués avec GOTO. Le déroulement du programme se poursuivra à la ligne numéro indiqué dans l'instruction GOTO. En raison du manque de constructions sophistiquées de bouclage du langage PC-BASIC, de branchement et de rupture, les sauts inconditionnels sont essentiels et utilisés fréquemment.

L'instruction GOSUB saute à un sous-programme. Semblable à GOTO, il s'agit d'un saut inconditionnel ; cependant, l'emplacement de l'appel est stocké et le programme et continuera son flux à partir de là, le sous-programme se termine par une instruction RETURN.

Les sous-programmes sont un peu comme des procédures en ce sens qu'elles permettent de séparer des morceaux de code qui exécutent une tâche donnée du corps principal du programme, mais ils n'ont pas de portée distincte puisque toutes les variables de PC-BASIC sont globales. Ils n'ont pas de valeur de retour.

Il est même possible de sauter d'un sous-programme n'importe où dans le programme en fournissant l'instruction RETURN avec une ligne de Numéro.

L'instruction ON fournit une construction de branchement alternative. Une valeur entière est utilisée pour sélectionner l'une d'une liste de numéros de ligne, et l'exécution se poursuit à partir de là. Elle peut être utilisée avec un saut GOTO ainsi qu'avec un appel de sous-programme GOSUB.

ON, GOTO et GOSUB peuvent également être utilisés en mode direct pour démarrer un programme ou un sous-programme sans réinitialiser les variables.

La construction IF–THEN–ELSE teste une condition et exécute différentes branches de code en fonction de sa valeur de vérité. Ce n'est pas une construction de bloc ; tout le code dans THEN et ELSE les branches doivent tenir sur une seule ligne. Pour cette raison, la ramification est souvent utilisée en combinaison avec GOTO.

Par exemple:

```
10 INPUT "How old are you"; AGE%
20 IF AGE%>30 THEN 100
30 IF AGE%<30 THEN 200 ELSE PRINT "You are 30 years old."
40 END
100 PRINT "You are over 30."
110 END
200 PRINT "You are not yet 30."
210 END
```

La construction en boucle WHILE–WEND répète le bloc de code entre WHILE et WEND comme tant qu'une condition donnée reste vraie.

La construction FOR–NEXT répète un bloc de code pendant qu'un compteur reste dans une plage donnée. Le compteur est défini sur une valeur de départ à la première passe de l'instruction FOR et incrémenté par la valeur STEP à chaque passage de NEXT.

Par exemple:

```
10 FOR I=1 TO 10
20 PRINT STRING$(I, "*"); USING " [##]"; I
30 NEXT I
```

Les constructions en boucle peuvent être imbriquées. Le flux de contrôle est également affecté par l'interception d'événements et d'erreurs.

<b>Commande</b>	<b>Description</b>
CONT	Continue le programme interrompu
ELSE	Ignore le reste de la ligne (ELSE autonome)
END	Arrête l'exécution du programme
FOR	Démarre une boucle FOR
GOSUB	Appelle un sous-programme
GOTO	Passe à un autre emplacement dans le programme

IF	Branche de programme sous condition
NEXT	Itération d'une boucle FOR
ON	Saut calculé ou appel de sous-programme
RETURN	Retour du sous-programme GOSUB
STOP	Interrompre l'exécution du programme
WEND	Itération d'une boucle WHILE
WHILE	Entre dans une boucle WHILE

### Tableaux et variables

Commande	Description
DEFDBL	Spécifier la plage de noms pour les variables flottantes à double précision
DEFINT	Spécifiez la plage de noms pour les variables à nombres entiers
DEFSNG	Spécifier la plage de noms pour les variables flottantes simple précision
DEFSTR	Spécifier la plage de noms de variables pour les chaînes
DIM	Allouer un tableau
ERASE	Libérer un tableau
LET	Affecter une valeur à une variable
OPTION BASE	Définir l'index de départ des tableaux
SWAP	Échange de deux variables

### Conversion de types

Fonction	Description
ASC	Caractère à valeur ordinale
CHR\$	Valeur ordinale au caractère
HEX\$	Représentation d'un entier à une chaîne hexadécimale
OCT\$	Représentation d'un entier à une chaîne octale
STR\$	Valeur numérique en représentation sous forme de chaîne décimale
VAL	Représentation sous forme de chaîne en valeur numérique
CDBL	Valeur numérique en flottant double précision
CINT	Valeur numérique en nombre entier

CSNG	Valeur numérique en flottant simple précision
CVD	Représentation d'octet en flottant double précision
CVI	Représentation d'octet en nombre entier
CVS	Représentation d'octet en flottant simple précision
MKD\$	Représentation flottante à double précision en octets
MKI\$	Représentation d'entier à octet
MKS\$	Représentation flottante en octet simple précision

### Opérations sur les chaînes

Commande	Description
LSET	Copie une valeur justifiée à gauche dans un tampon de chaîne
MID\$	Copie une valeur dans une partie d'un tampon de chaîne
RSET	Copie une valeur justifiée à droite dans un tampon de chaîne
<b>Fonction</b>	
INSTR	Trouver
LEFT\$	Sous-chaîne gauche
LEN	Longueur de chaîne
MID\$	Sous-chaîne
RIGHT\$	Sous-chaîne droite
SPACE\$	Espaces de répétition
STRING\$	Répéter les caractères

### Le texte et l'écran

Commande	Description
CLS	Efface l'écran
COLOR	Définit les valeurs de couleur et de palette
LOCATE	Définit la position et la forme du curseur de l'écran de texte
PALETTE	Attribue une couleur à un attribut
PALETTE USING	Affecte un tableau de couleurs aux attributs
PCOPY	Copie une page écran



PRINT	Imprime des expressions à l'écran
VIEW PRINT	Définit la zone de défilement du texte
WIDTH	Définit le nombre de colonnes de texte à l'écran
<b>Fonction</b>	
CSRLIN	Ligne actuelle du curseur
POS	Colonne actuelle du curseur
SCREEN	Caractère ou attribut à un emplacement donné

### Imprimer

Commande	Description
LPRINT	Imprime des expressions sur l'imprimante
<b>Fonction</b>	
LPOS	Position de la colonne de la tête d'impression

### La saisie au clavier

Commande	Description
INPUT	Récupérer l'entrée utilisateur sur la console
LINE INPUT	Récupérer une ligne d'entrée utilisateur sur la console
<b>Fonction</b>	
INKEY\$	Lecture non bloquante depuis le clavier
INPUT\$	Blocage de la lecture depuis le clavier

### Macros de touches de fonction

Commande	Description
Key	Gérer la visibilité de la liste des macros de touches de fonction
Key	Définir une macro de touche de fonction

## Calculs et mathématiques

Fonction	
ABS	Valeur absolue
ATN	Arc tangente
COS	Cosinus
EXP	Exponentiel
FIX	Troncature
INT	Sol
LOG	Un algorithme naturel
SIN	Sinus
SGN	Signe
SQR	Racine carrée
TAN	Tangente

## Nombres aléatoires

Commande	Description
RANDOMIZE	Amorcer le générateur de nombres aléatoires
Fonction	
RND	Nombre pseudo-aléatoire

## Périphériques et fichiers

Commande	Description
CLOSE	Fermer un fichier
FIELD	Attribuer une chaîne à un tampon d'enregistrement à accès aléatoire
GET	Lire un enregistrement à partir d'un fichier à accès aléatoire
INPUT	Lire une variable d'un fichier
LINE INPUT	Lire une ligne d'un fichier
LOCK	Verrouille un fichier ou une plage d'enregistrements contre toute autre utilisation
OPEN	Ouvrir un fichier de données
PUT	Écrire le tampon d'enregistrement à accès aléatoire sur le disque
RESET	Fermer tous les fichiers

UNLOCK	Déverrouille un fichier ou une plage d'enregistrements contre une autre utilisation
WIDTH	Définir le nombre de colonnes de texte dans un fichier
WRITE	Écrire des expressions dans un fichier
<b>Fonction</b>	
EOF	Fin de fichier
LOC	Emplacement dans le dossier
LOF	Longueur du fichier
INPUT\$	Lire une chaîne à partir d'un fichier

## Graphique

Commande	Description
CIRCLE	Dessiner une section d'ellipse ou d'arc
DRAW	Dessiner une forme définie par une chaîne Graphiques de Macro Langage
GET	Stocker une zone d'écran en tant que sprite
LINE	Dessiner un segment de ligne
PAINT	Remplir une région connectée
PSET	Mettre un pixel
PRESET	Changer un pixel en attribut d'arrière-plan
PUT	Dessiner un sprite à l'écran
SCREEN	Changer le mode vidéo
VIEW	Défini la fenêtre graphique
WINDOW	Défini des coordonnées logiques
<b>Fonction</b>	
POINT	Coordonnées du pointeur graphique
POINT	Attribut de pixel
PMAP	Convertir entre les coordonnées physiques et logiques

## Son

Commande	Description
BEEP	Bip le haut-parleur
BEEP	Commutateur de haut-parleur
NOISE	Générer du bruit
PLAY	Jouer un morceau encodé en Music Macro Language
SOUND	Générer une tonalité
SOUND	Commutateur de son
<b>Fonction</b>	
PLAY	Longueur de la file d'attente de la musique de fond

## Disque et DOS

Commande	Description
CHDIR	Changer le répertoire courant
FILES	Lister les fichiers du répertoire courant
KILL	Supprimer un fichier sur un périphérique de disque
MKDIR	Créer un nouveau répertoire
NAME	Renommer un fichier sur disque
RMDIR	Supprimer un répertoire
ENVIRON	Définir une chaîne d'environnement shell
<b>Fonction</b>	
ENVIRON\$	Chaîne de la table d'environnement shell

## Fonctions définies par l'utilisateur

Commande	Description
DEF FN	Définir une nouvelle fonction
<b>Fonction</b>	
FN	Fonction définie par l'utilisateur

## Date et Heure

Commande	Description
DATE\$	Définir la date du système
TIME\$	Régler l'heure du système
<b>Fonction</b>	
DATE\$	Date système sous forme de chaîne
TIME\$	Heure système sous forme de chaîne
TIMER	Heure système en secondes depuis minuit

## Inclure des données dans un programme

Commande	Description
DATA	Définir les données à utiliser par le programme
READ	Récupérer une entrée de données
RESTORE	Réinitialiser le pointeur de données

## Référence du langage

### Méta syntaxe

Dans les descriptions de la syntaxe BASIC, les conventions suivantes s'appliquent. Rendu exact de la balise peut varier selon le moyen utilisé pour afficher cette documentation.

### Caractère gras

Tapez exactement comme indiqué.

### *Italique*

Remplacez par la méta variable appropriée.

[a

Les entités entre crochets sont facultatives.

{ a | b }

Alternatives disjointes dont une doit être choisie.

[ a | b ]

Alternatives facultatives disjointes.

a...

L'entité précédente peut être répétée.

## Définitions

Une ligne de programme se compose d'un numéro de ligne suivi d'une instruction composée.

Les Lignes de programme se terminent par un CR ou ou par la fin du fichier (éventuellement par un caractère EOF).

Tout ce qui se trouve sur une ligne de programme après un caractère NUL est ignoré.

Un numéro de ligne est un nombre entier dans la plage [0—65535]. Notez que les numéros de ligne 65530—65535 ne peut pas être entré à partir de la console ou d'un fichier de programme texte, mais peut faire partie d'un fichier programme tokenisé.

Une instruction composée se compose d'instructions séparées par des deux-points :

instruction [: instruction ]...

Une expression prend l'une des formes suivantes :

opérateur\_unaire { littéral | variables | élément\_tableau | fonction }

expression expression de l'opérateur\_binaire

(expression)

dont les éléments sont décrits dans les sections Littéraux, Variables, Opérateurs et Fonctions.

Un élément de tableau prend la forme

tableau { [( { expression\_numérique [, expression\_numérique ]... } ) ] }

## Littéraux

Littéraux de chaîne

Les littéraux de chaîne sont de la forme suivante :

"[caractères]{NUL|CR|EOF}"

où caractères est une chaîne de caractères.

Tout caractère de la page de code actuelle peut être utilisé, avec les exceptions suivantes, qui terminent toutes la chaîne littérale (à part d'autres effets qu'ils peuvent avoir):

NUL ( CHR\$( &h00 ) )

CR ( CHR\$( &h0D ) )

EOF ( CHR\$( &h1A ) )

" ( CHR\$( &h22 ) )

Les chaînes sont également légalement terminées à la fin du fichier en l'absence d'un caractère EOF.

En dehors de ceux-ci, les littéraux de chaîne ne doivent contenir aucun des caractères de la plage ASCII &h0D — &h1F, qui conduisent à des résultats imprévisibles.

Il n'y a pas de mécanisme d'échappement. Pour inclure l'un des caractères ci-dessus dans une chaîne, utiliser la concaténation de chaînes et le CHR\$ fonction.

## Littéraux numériques

Les littéraux numériques ont l'une des formes suivantes :

[+|-] [0|1|2|3|4|5|6|7|8|9]... [.] [0|1|2|3|4|5|6|7|8|9]...

[{E|e|D|d}[+|-][0|1|2|3|4|5|6|7|8|9]...] [#|!|%]

&{H|h}[0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F|a|b|c|d|e|f]...

&[O|o][0|1|2|3|4|5|6|7]...

Les littéraux hexadécimaux ne doivent pas contenir d'espaces, contrairement aux littéraux décimaux et octaux.

Le o caractère dans les littéraux octaux est facultatif : ils peuvent être spécifiés de la même manière que &o777 ou &777.

Les littéraux hexadécimaux et octaux désignent des entiers et n'incluent pas de signe.

Ils doivent s'étendre entre [ &h0 — &hFFFF ], dont la plage [ &h8000 — &hFFFF ] est interprétée comme un deux complément entier négatif ; par exemple, &hFFFF = -1.

Des signes peuvent apparaître à gauche du & mais ceux-ci forment une expression et ne font pas partie du littéral lui-même.

Les littéraux à virgule flottante doivent être spécifiés en notation décimale. Le séparateur décimal est le point.

Un exposant en base 10 peut être spécifié après E dans les flottants simple précision, ou après D dans floteurs à double précision.

Le % de fin est ignoré et n'indique pas un entier littéral.

Le ! ou # marquent le littéral comme simple ou double précision, respectivement. Des exemples de littéraux numériques valides sont:

-1 42 42! 42# 1.3523523.235435 -3 3...e.D 1.1e+7 1.1d+7 1e2 1e-2 &7 &hffff &O20 &h & 65537% 1.1%

Notez que les expressions telles que &o-77 sont légales ; ce ne sont cependant pas des octaux négatifs mais plutôt l'expression &o (octal vide; zéro) moins 77 (décimal 77).

## Variables

Les noms de variables doivent commencer par une lettre ; tous les caractères du nom de la variable (sauf le sigil) doit être des lettres A—Z , des chiffres 0—9 , ou un point.

Seuls les 40 premiers caractères du nom sont importants. Un nom de variable ne doit pas être identique à un mot réservé ou un mot réservé plus sceau. Ainsi, par exemple, vous ne pouvez pas nommer une variable TO ! mais vous pouvez la nommer AS !.

Les noms de variables peuvent contenir n'importe quel mot réservé. Les noms de variables peuvent également commencer par un mot réservé, à l'exception de USR et FN.

Ainsi, FNORD% et USRNME\$ ne sont pas noms de variable légaux alors que FRECKLE% et LUSR\$ le sont.

Pour chaque nom, quatre variables différentes peuvent exister correspondant aux quatre types. C'est-à-dire, vous pouvez avoir A\$, A%, A ! et A# comme différentes variables. Lequel d'entre eux est aussi connu sous le nom de A dépend des paramètres dans DEFINT / DEFDBL / DEFSNG / DEFSTR.

Par défaut, A est égal à la simple précision A!. De plus, les tableaux A\$( ) , A%( ) , A!( ) , A#( ) sont séparés des variables scalaires du même nom.

## Types et Sigil

PC-BASIC reconnaît quatre types de variables, distingués par leur sigil ou caractère de type, le dernier caractère du nom complet de la variable :

sigil	type	Taille	Intervalle	Précision
\$	Chaîne de caractères	3 octets plus caractères espace	0—255 caractères	
%	Entier	2 octets	-32768—32767	exacte
!	Flottant simple précision	4 octets	$\pm 2.938726 \cdot 10^{-39}$ à $\pm 1.701412 \cdot 10^{38}$	~6 chiffres significatifs
#	Flotteur double précision	8 octets	$\pm 2.938735877055719 \cdot 10^{-39}$ à $\pm 1.701411834604692 \cdot 10^{38}$	~16 chiffres significatifs

Notez que les flottants à double précision peuvent contenir plus de décimales que les flottants à simple précision, mais pas de nombres plus grands ou plus petits.

Bien que tous les entiers soient signés, certaines instructions interpréteront les entiers négatifs comme leurs deux compléments.



## Tableaux

Les tableaux sont indexés avec des crochets ou des crochets ; même le mélange des parenthèses est autorisé.

Voici tous les éléments de tableau autorisés :  $A[0]$  ,  $A(0)$  ,  $A(0)$  ,  $A(0)$ .

Les tableaux multidimensionnels sont spécifiés en séparant les indices par des virgules :  $A(0, 0)$  ,  $A[0, 0, 0]$  , etc.

Par défaut, les tableaux sont indexés à partir de 0. Cela peut être changé à 1 en utilisant `OPTION BASE 1`.

Les tableaux peuvent être alloués en spécifiant le plus grand index autorisé à l'aide de `DIM`. Si tous les indices du tableau sont 10 ou moins, ils n'ont pas besoin d'être explicitement alloués.

Le premier accès du tableau (lecture ou écriture) l'allouera automatiquement avec un index maximum de 10 et le même nombre d'indices comme lors du premier accès.

Pour réallouer un tableau, l'ancien tableau doit d'abord être supprimé avec `CLEAR` ou `ERASE`.

Les tableaux multidimensionnels sont stockés dans l'ordre des colonnes majeures, de sorte que  $A \% (2,0)$  immédiatement suit  $A\%(1,0)$ .

## Conversions

PC-BASIC effectuera implicitement la conversion entre les trois types de données numériques. Lorsqu'une valeur de un type est affectée à une variable, un élément de tableau ou un paramètre d'un autre type, il est converti selon les règles suivantes :

- Les flottants simple et double précision sont convertis en nombre entier en arrondissant au nombre entier le plus proche. Les moitiés sont arrondies à partir de zéro. Si l'ensemble résultant est nombre en dehors de la plage autorisée pour les entiers, le dépassement de capacité est déclenché.
- Les flottants à double précision sont convertis en flottants à simple précision par arrondissement Gaussien de la mantisse, où le nouveau bit le moins significatif de la mantisse est arrondi si la fraction binaire tronquée est supérieure à la moitié ; les moitiés sont arrondi à pair.
- Les nombres entiers sont convertis dans leur représentation exacte en simple ou double précision flote.
- Les flottants à simple précision sont convertis en leur représentation exacte en flottants double précision.
- Il n'y a pas de conversion implicite entre les chaînes et les types numériques.

Une tentative d'assignation d'une valeur de chaîne à une variable numérique, un élément de tableau ou un paramètre (ou vice versa) augmentera `Type mismatch`.

## Les opérateurs

### Ordre de préséance

L'ordre de priorité des opérateurs est le suivant, de étroitement lié (priorité élevée) à lâche (faible priorité) :

1. ^
2. \* /
3. \
4. MOD
5. + - (unaire et binaire)
6. = <> >< < > <= =< >= =>
7. NOT (unaire)
8. AND
9. OR
10. XOR
11. EQV
12. IMP

Les expressions entre parenthèses () sont évaluées en premier. Tous les opérateurs binaires sont associatif à gauche : les opérateurs de même priorité sont évalués de gauche à droite.

### Exemples

- L'exponentiation est plus étroitement liée que la négation :  $-1^2 = -(1^2) = -1$  mais  $(-1)^2 = 1$ .
- L'exponentiation est associative à gauche :  $2^3^4 = (2^3)^4 = 4096$ . les erreurs
- Si un opérateur autre que + , - ou NOT est utilisé sans opérande gauche, Syntaxe erreur est levée.
- À la fin d'une instruction, si un opérateur est utilisé sans opérande droit, l'opérande est élevé Missing. Si cela se produit ailleurs dans une instruction, comme dans des crochets, une erreur de syntaxe est générée.

### Opérateurs mathématiques

Les opérateurs mathématiques fonctionnent uniquement sur des expressions numériques. Notez cependant que + peut prendre le rôle de l'opérateur de concaténation de chaînes si les deux opérandes sont des chaînes.

Code	Opération	Résultat
$x \wedge y$	Exponentiation	x élevé à la puissance y
$x * y$	Multiplication	Produit de x et y
$x / y$	Division	Quotient de x et y
$x \setminus y$	Division tronquée à l'entier	Quotient entier de x et y
$x \text{ MOD } y$	Modulo	Reste entier de x par y (avec le signe de x)
$x + y$	Addition	Somme de x et y

$x - y$	Soustraction	Différence de x et y
$+ y$	Unaire Plus	Valeur de y
$- y$	Négation	Valeur négative de y

## Remarques

- Si nécessaire, le résultat de l'opération sera mis à niveau vers un type de données capable de retenir le résultat. Par exemple, diviser des nombres entiers 3 par 2 donnera une simple précision 1.5.
- Cependant, l'opérateur d'exponentiation  $^$  donnera au plus un résultat simple précision sauf si l'option double est utilisée.
- L'expression  $0^0$  renverra 1 et ne générera pas d'erreur, même si, mathématiquement, élever zéro à la puissance zéro n'est pas défini.

## Les erreurs

- Si l'un ou l'autre des opérandes est une chaîne, l'incompatibilité de type sera déclenchée. L'exception est  $+$  qui ne soulèvera une incompatibilité de type que si l'un ou l'autre des opérandes, mais pas les deux, sont des chaînes.
- Si  $y=0$ ,  $x / y$ ,  $x \text{ MOD } y$  et  $x \setminus y$  augmenteront la division par zéro.
- Si  $x=0$  et  $y<0$ ,  $x^y$  augmentera Division de zéro.
- Si le résultat d'une opération est trop volumineux pour tenir dans un type de données à virgule flottante, Le Overflow est relevé.
- Si les opérandes ou le résultat de  $\setminus$  ou MOD ne sont pas dans  $[-32768-32767]$ , Overflow est soulevé.
- Si  $x<0$  et y est un nombre fractionnaire,  $x ^ y$  déclenchera Illegal function call.

## Opérateurs relationnels

Les opérateurs relationnels peuvent fonctionner sur des opérandes numériques ainsi que sur des opérandes de chaîne ; cependant, si l'un l'opérande est une chaîne et l'autre numérique, l'incompatibilité de type est levée. Les opérateurs relationnels renvoient soit 0 (pour faux) soit -1 pour vrai.

Code	Opération	Résultat
=	Égal	Vrai si a est égal à b, faux sinon
<> ><	Inégal	Faux si a est égal à b, vrai sinon
<	Moins que	Vrai si a est inférieur à b, faux sinon
>	Plus grand que	Vrai si a est supérieur à b, faux sinon
<= =<	Inférieur ou égal	Vrai si a est inférieur ou égal à b, faux sinon
>= =>	Plus grand ou égal	Vrai si a est supérieur ou égal à b, faux sinon

Lorsque vous travaillez sur des opérandes numériques, les deux opérandes sont comparés en nombres flottants selon l'ordre habituel des nombres.

L'opérateur égal teste l'égalité à la précision près de la machine pour la plus grande précision des deux types d'opérateurs.

Lors de la comparaison de chaînes, l'ordre est le suivant.

- Deux chaînes sont égales uniquement si elles ont la même longueur et chaque code de caractère de la première chaîne correspond au code de caractère correspondant de la seconde. Ceci inclut tout espace ou caractère non imprimable.
- Chaque position de caractère des chaînes est comparée en commençant par le plus à gauche. Lorsqu'une paire de caractères différents est rencontrée, la chaîne avec le caractère de point de code inférieur est inférieure à la chaîne avec le caractère de plus grand point de code.
- Si les chaînes sont de longueur différente, mais égales jusqu'à la longueur de la chaînes la plus courte, alors la chaîne la plus courte est inférieure à la chaîne la plus longue.

### Opérateurs de chaîne

L'opérateur de concaténation de chaînes est +. Il a une forme binaire ainsi qu'une forme unaire. L'unaire moins peut également être utilisé sur des chaînes, mais n'a aucun effet.

Code	Opération	Résultat
x + y	Enchaînement	La chaîne formée par x suivi de y
+ y	Unaire Plus	Valeur de y
- y	Moins unaire	Valeur de y

### Les erreurs

- Si l'un des opérandes (mais pas les deux) d'une concaténation est numérique, Type mismatch sera relevé.
- Si  $LEN(x) + LEN(y) > 255$ , x + y augmentera La chaîne trop longtemps.

### Les fonctions

Les fonctions ne peuvent être utilisées que dans le cadre d'une expression au sein d'une instruction ; ils peuvent prendre l'entrée des valeurs entre parenthèses et produire une valeur de retour.

Par exemple, dans PRINT ABS(-1) la fonction ABS est utilisée dans une expression au sein d'une instruction PRINT ; dans Y = CARRÉ(X) + 2 la fonction SQR est utilisée dans une expression au sein d'une instruction LET.

Certains ouvrages de référence utilisent également des termes tels que variable système pour les fonctions qui ne prennent pas une entrée, probablement puisque dans la syntaxe GW-BASIC, ces fonctions n'ont pas de parenthèses, contrairement aux langages de la famille C (et même à certains BASIC modernes).

Cependant, cela est simplement la syntaxe GW-BASIC pour les fonctions sans entrées.

Par exemple, on peut faire DEF FNA=1 : PRINT FNA dans lequel aucune parenthèse n'est autorisée.

<b>ABS</b>	$y = \text{ABS}(x)$	x est une expression.
Renvoie la valeur absolue de x si x est un nombre et la valeur de x si x est une chaîne.		
<b>ASC</b>	$\text{val} = \text{ASC}(\text{char})$	char est une expression avec une valeur de chaîne.
Renvoie le point de code (valeur ASCII) pour le premier caractère de char.		
<b>Les erreurs</b> <ul style="list-style-type: none"><li>• char a une valeur numérique : Type mismatch.</li><li>• char est égal à "" : Illegal function call.</li></ul>		
<b>ATN</b>	$y = \text{ATN}(x)$	x est une expression numérique qui donne l'angle en radians.
<b>Remarques</b> <ul style="list-style-type: none"><li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur de précision.</li><li>• ATN(x) diffère par le chiffre le moins significatif de GW-BASIC.</li></ul>		
<b>Les erreurs</b> <ul style="list-style-type: none"><li>• x a une valeur de chaîne : Type mismatch.</li></ul>		
<b>CDBL</b>	$y = \text{CDBL}(x)$	Convertit l'expression numérique x en valeur double précision.
<b>Les erreurs</b> <ul style="list-style-type: none"><li>• x a une valeur de chaîne : Type mismatch.</li></ul>		

<b>CHR\$</b>	char = CHR\$(x)	x est une expression numérique dans la plage [0—255]
<p>Renvoie le caractère avec le point de code x</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> <li>• x n'est pas dans [-32768—32767] : Overflow.</li> <li>• x n'est pas dans [0—255] : Illegal function call.</li> </ul>		
<b>CINT</b>	y = CINT(x)	Convertit l'expression numérique x en entier signé. Les moitiés sont arrondies à partir de zéro, de sorte que par ex. CINT(2.5) = 3 et CINT(-2.5) = -3.
<p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> <li>• x n'est pas dans [-32768—32767] : Overflow.</li> </ul>		
<b>COS</b>	cosine = COS(angle)	angle est une expression numérique qui donne l'angle en radians.
<p>Renvoie le cosinus de l'angle. A moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie une valeur simple précision.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• La valeur de retour diffère généralement de la valeur renvoyée par GW-BASIC au dernier chiffre significatif.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• angle a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>CSNG</b>	y = CSNG(x)	Convertit l'expression numérique x en une valeur simple précision par arrondi gaussien.
<p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> </ul>		

<b>CSRLIN</b>	$y = \text{CSRLIN}$	Cette fonction ne prend aucun argument.
Renvoie la ligne d'écran du curseur sur la page active. La valeur de retour est dans la plage [1—25].		
<b>CVI</b>	$y = \text{CVI}(s)$	Convertit une chaîne de deux octets en entier signé.
<p><math>s</math> est une expression de chaîne qui représente un entier en utilisant deux little-endian de codage complémentaire. Seuls les deux premiers octets sont utilisés.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• <math>s</math> a une valeur numérique : Incompatibilité de type.</li> </ul>		
<b>CVS</b>	$y = \text{CVS}(s)$	Convertit une chaîne de quatre octets en un nombre à virgule flottante simple précision.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• <math>s</math> est une expression de chaîne qui représente un nombre simple précision dans le Format binaire Microsoft. Seuls les quatre premiers octets sont utilisés.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• <math>s</math> a une valeur numérique : Incompatibilité de type.</li> </ul>		
<b>CVD</b>	$y = \text{CVD}(s)$	Convertit une chaîne de huit octets en nombre à virgule flottante double précision.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• <math>s</math> est une expression de chaîne qui représente un nombre à double précision dans le Format binaire Microsoft. Seuls les huit premiers octets sont utilisés.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• <math>s</math> a une valeur numérique : Incompatibilité de type.</li> </ul>		

<b>DATE\$</b> (fonction)	s = DATE\$	Cette fonction ne prend aucun argument.
Renvoie la date système sous forme de chaîne au format "mm-dd-yyyy".		
<b>ENVIRON\$</b>	value = ENVIRON[ ]\$(x)	Renvoie une variable d'environnement.
<p>Paramètres</p> <p>x est une expression.</p> <ul style="list-style-type: none"> <li>• Si x a une valeur de chaîne, renvoie la valeur de la variable d'environnement x ou chaîne vide si aucune variable portant le nom x n'est définie dans la table d'environnement. La variable d'environnement doit être une chaîne ASCII et sera convertie en majuscule sur les systèmes sensibles à la casse.</li> <li>• Si x a une valeur numérique, elle doit être dans [1—255]. Renvoie la x ième entrée dans le tableau d'environnement.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x est la chaîne vide : Illegal function call.</li> <li>• x contient des caractères non ASCII : Illegal function call.</li> <li>• x est un nombre qui n'est pas dans [-32768—32767] : Overflow.</li> <li>• x est un nombre qui n'est pas dans [1—255] : Illegal function call.</li> </ul>		
<b>EOF</b>	file = EOF(file_num)	Renvoie -1 si le fichier portant le numéro file_num a atteint la fin du fichier ; 0 sinon. Le dossier doit être ouvert en mode INPUT ou RANDOM. EOF(0) renvoie 0.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• Si num_fichier est ouvert sur KYBD : effectue une lecture bloquante et renvoie -1 si CTRL + Z est entré, 0 sinon. Le caractère saisi est alors renvoyé en écho à la console.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• file_num est un nombre qui n'est pas dans [-32768—32767] : Overflow.</li> <li>• file_num est un nombre qui n'est pas dans [0—255] : Illegal function call.</li> <li>• file_num n'est pas 0 ou le numéro d'un fichier ouvert : Bad file number.</li> <li>• Le fichier portant le numéro file_num est en mode OUTPUT ou APPEND : fichier incorrect.</li> </ul>		



<b>ERDEV</b>	zero = ERDEV	Renvoi 0
Remarques <ul style="list-style-type: none"> <li>• Dans GW-BASIC, renvoie la valeur d'une erreur de périphérique.</li> <li>• Cette fonction n'est pas implémentée dans PC-BASIC.</li> <li>• Cette fonction ne prend aucun argument.</li> </ul>		
<b>ERDEV\$</b>	empty = ERDEV[ ]\$	Renvoie une chaîne vide.
Remarques <ul style="list-style-type: none"> <li>• Dans GW-BASIC, renvoie le nom de périphérique d'une erreur de périphérique.</li> <li>• Cette fonction n'est pas implémentée dans PC-BASIC.</li> <li>• Cette fonction ne prend aucun argument.</li> </ul>		
<b>ERL</b>	error_line = ERL	Renvoie le numéro de ligne où la dernière erreur a été déclenchée.
Remarques <ul style="list-style-type: none"> <li>• Si l'erreur a été déclenchée par une instruction directe, renvoie 65535.</li> <li>• Si aucune erreur n'a été générée, renvoie 0.</li> <li>• Cette fonction ne prend aucun argument.</li> </ul>		
<b>ERR</b>	error_code = ERR	Renvoie le numéro de la dernière erreur.
Remarques <ul style="list-style-type: none"> <li>• Si aucune erreur n'a été générée, renvoie 0.</li> <li>• Si la dernière erreur était une erreur de syntaxe déclenchée par une instruction directe, renvoie 0.</li> <li>• Cette fonction ne prend aucun argument.</li> </ul>		

<b>EXP</b>	$y = \text{EXP}(x)$	Renvoie l'exponentielle de $x$ , c'est-à-dire $e$ à la puissance $x$ .
<p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <math>x</math> est une expression à valeur numérique.</li> </ul> <p><b>Remarques</b></p> <ul style="list-style-type: none"> <li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur de précision.</li> <li>• La valeur de retour diffère parfois du chiffre le moins significatif de GW-BASIC. Pour les grandes valeurs de <math>x</math>, la différence peut être de 3 chiffres.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• <math>x</math> a une valeur de chaîne : Type mismatch.</li> <li>• <math>x</math> est supérieur au logarithme népérien de la valeur simple précision maximale : Overflow.</li> </ul>		
<b>EXTERR</b>	$\text{zero} = \text{EXTERR}(x)$	Renvoie 0.
<p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <math>x</math> est une expression numérique dans <math>[0—3]</math>.</li> </ul> <p><b>Remarques</b></p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, cette fonction renvoie des informations d'erreur étendues à partir de MS-DOS.</li> <li>• Cette fonction n'est pas implémentée dans PC-BASIC.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• <math>x</math> a une valeur de chaîne : Type mismatch.</li> <li>• <math>x</math> n'est pas dans <math>[-32768—32767]</math> : Overflow.</li> <li>• <math>x</math> n'est pas dans <math>[0—3]</math> : Illegal function call.</li> </ul>		

<b>FIX</b>	whole = FIX(nombre)	Renvoie le nombre tronqué vers zéro.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• nombre est une expression numérique.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• FIX tronque vers zéro : il supprime la partie fractionnaire. En revanche, INT tronque vers moins l'infini.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• nombre est une expression de chaîne : Type mismatch.</li> </ul>		
<b>FN</b>	result = FN[ ]name [(arg_0 [, arg_1]...)]	
<p>Évalue la fonction définie par l'utilisateur précédemment définie avec DEF FN name. Espaces entre Le FN et le nom sont facultatifs.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• name est le nom d'une fonction précédemment définie.</li> <li>• arg_0, arg_1,... sont des expressions, données en paramètres à la fonction.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Aucune fonction nommée nom n'est définie : Fonction utilisateur non définie.</li> <li>• Le nombre de paramètres diffère de la définition de la fonction : Erreur de syntaxe.</li> <li>• Le type d'un ou plusieurs paramètres diffère de la définition de la fonction : Type décalage.</li> <li>• Le type de retour est incompatible avec le sigil du nom de la fonction : Type mismatch.</li> <li>• La fonction appelée est récursive ou mutuellement récursive : Mémoire insuffisante.</li> </ul>		
<b>FRE</b>	free_mem = FRE(x)	Renvoie la mémoire BASIC disponible.
<p>Paramètres</p> <p>x est une expression.</p> <ul style="list-style-type: none"> <li>• Si x a une valeur numérique, elle est ignorée.</li> <li>• Si x a une valeur de chaîne, la récupération de place est effectuée avant de renvoyer la mémoire disponible.</li> </ul>		

<b>HEX\$</b>	hex_repr = HEX\$(x)	Renvoie une chaîne avec la représentation hexadécimale de x
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• x est une expression numérique dans [-32768—65535]. Les valeurs pour x négatif sont affichées en complément à deux.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x n'est pas dans [-32768—65535] : Overflow.</li> <li>• x a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>INKEY\$</b>	key = INKEY\$	Renvoie une pression sur une touche à partir du tampon du clavier. Si le tampon du clavier est vide, renvoie la chaîne vide. Sinon, la valeur de retour est une chaîne à un ou deux caractères contenant le code ASCII de la touche enfoncée.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• Cette fonction ne prend aucun argument.</li> <li>• Lorsqu'une touche de fonction F1 – F10 est enfoncée, INKEY\$ renverra les lettres de la macro associée - sauf si cette macro a été définie sur vide avec la clé instruction, auquel cas elle renvoie le code e-ASCII de la touche de fonction.</li> </ul>		
<b>INP</b>	code = INP(port)	Renvoie la valeur d'un port de machine émulé
<p>Paramètres</p> <p>port est une expression numérique dans [0—65535].</p> <p>port Effet</p> <p>&amp;h60 Renvoie le scancode du clavier pour la touche actuellement enfoncée ou la dernière touche publié.</p> <p>Les scancodes retournés par INP(&amp;h60) sont ceux listés dans le tableau clavier des scancodes (Voir à la fin du document).</p> <p>Si une clé est actuellement en panne, la valeur de retour est son scancode. Sinon la clé est enfoncée, la valeur de retour est le scancode de la dernière clé relâchée, incrémenté de 128.</p>		

&h201 Renvoie la valeur du port de jeu (port du joystick). Cette valeur est construite comme suit :

#### Signification des bits

- 0 joystick 2 axe x
- 1 joystick 1 axe y
- 2 joystick 1 axe x
- 3 joystick 2 axe y
- 4 joystick 2 bouton 1
- 5 joystick 1 bouton 2
- 6 joystick 1 bouton 1
- 7 joystick 2 bouton 2

Les bits du bouton sont 0 lorsque le bouton est déclenché, 1 sinon. Les valeurs des axes sont normalement 0 mais sont mis à 1 par OUT &h201, x puis retombent à 0 après un délai. Plus le délai est long, plus la valeur de l'axe est élevée.

Autre valeurs, Renvoie zéro.

#### Remarques

- Seul un nombre limité de ports machine sont émulés dans PC-BASIC.

#### Les erreurs

- le port n'est pas dans [-32768—65535] : Overflow.
- port a une valeur de chaîne : Type mismatch.

#### **INPUT\$**

chars = INPUT[ ]\$ (num\_chars [, [#] file\_num])

Renvoie une chaîne de num\_chars caractères à partir du clavier ou, si file\_num est fourni, à partir d'un fichier texte.

#### Paramètres

- num\_chars est une expression numérique dans [1—255].
- file\_num est une expression numérique qui renvoie le numéro d'un fichier texte ouvert dans Mode INPUT. Le # est facultatif et n'a aucun effet.

#### Remarques

- Il s'agit d'une lecture bloquante. Il attendra les caractères s'il n'y en a pas dans le tampon.
- Tous les caractères de contrôle sauf Ctrl + Pause , Ctrl + Arrêt défil et Pause sont passé à la chaîne par INPUT\$. Ctrl + pause et Ctrl + arrêt du défilement l'exécution tandis que Pause s'arrête jusqu'à ce qu'une autre

touche soit enfoncée (et non lue).

- Lors de la lecture à partir du clavier directement ou via KYBD : , touches fléchées, Suppr , Home , End , Pg Up , Pg Dn sont passés en tant que caractères NUL. Les touches de fonction sont ignorés s'ils sont interceptés par des évènements, sinon le remplacement de la macro de la touche de fonction est actif normalement.

### Les erreurs

- num\_chars n'est pas dans [-32768—32767] : Overflow.
- num\_chars n'est pas dans [1—255] : Illegal function call.
- num\_fichier n'est pas un fichier ouvert : numéro de fichier incorrect.
- file\_num est inférieur à zéro : Illegal function call.
- num\_fichier est supérieur à 32767 : Overflow.
- num\_fichier n'est pas ouvert pour INPUT : mode de fichier incorrect.
- num\_chars ou file\_num sont des chaînes : Type mismatch.
  
- file\_num est ouvert sur un port COM et il s'agit de la première INPUT , LINE INPUT ou Appel INPUT\$ sur ce port car le tampon est complètement rempli (c'est-à-dire LOF(num\_fichier) est devenu nul) : Overflow du tampon de communication.

### INSTR

position = INSTR([start,] parent, child)

Renvoie l'emplacement de la première occurrence de la sous-chaîne enfant dans parent.

#### Paramètres

- parent et child sont des expressions de chaîne.
- start est une expression numérique dans [1—255] , spécifiant la position de départ à partir de laquelle chercher ; s'il n'est pas spécifié, la recherche commence au caractère 1.

#### Notes

- Si child n'est pas une sous-chaîne de parent apparaissant au début ou avant, INSTR renvoie 0.

### Les erreurs

- start a une valeur de chaîne ou parent ou child a des valeurs numériques : Type mismatch.
- start n'est pas dans [-32768—32767] : Overflow.
- start n'est pas dans [1—255] : Illegal function call.

<b>INT</b>	whole = INT(number)	Renvoie le nombre tronqué vers moins l'infini.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• nombre est une expression.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• <b>FIX</b> tronque vers zéro : il supprime la partie fractionnaire. En revanche, INT tronque vers l'infini négatif.</li> <li>• Si nombre est une expression de chaîne, INT renvoie sa valeur inchangée.</li> </ul>		
<b>IOCTL\$</b>	result = IOCTL[ ]\$ ([#] file_num)	Raises Illegal function call.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, IOCTL\$ lit la réponse à IOCTL à partir d'un périphérique.</li> <li>• Cette fonction n'est pas implémentée dans PC-BASIC.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• num_fichier n'est pas dans [-32768—32767] : Overflow.</li> <li>• num_fichier n'est pas un fichier ouvert : Bad file number.</li> <li>• Sinon : Illegal function call</li> </ul>		
<b>LEFT\$</b>	child = LEFT\$(parent,num_chars)	Renvoie les num_chars caractères les plus à gauche du parent.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• parent est une expression de chaîne.</li> <li>• num_chars est une expression numérique dans [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si num_chars est égal à zéro ou si parent est vide, LEFT\$ renvoie une chaîne vide.</li> <li>• Si num_chars est supérieur à la longueur de parent , renvoie parent.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• parent a une valeur numérique ou num_chars a une valeur de chaîne : Type mismatch.</li> <li>• num_chars n'est pas dans [-32768—32767] : Overflow.</li> <li>• num_chars n'est pas dans [0—255] : Illegal function call.</li> </ul>		

<b>LEN</b>	length = LEN(string)	Renvoie le nombre de caractères dans string.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• string est une expression de chaîne.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• string a une valeur numérique : Type mismatch.</li> </ul>		
<b>LOC</b>	location = LOC(file_num)	
<p>Renvoie l'emplacement actuel dans le fichier ouvert sous le numéro file_num.</p> <ul style="list-style-type: none"> <li>• Si le fichier est ouvert pour INPUT , OUTPUT ou APPEND , LOC renvoie le nombre de Blocs de 128 octets lus ou écrits depuis l'ouverture du fichier.</li> <li>• Si le fichier est ouvert pour RANDOM , LOC renvoie le dernier numéro d'enregistrement lu ou écrit.</li> <li>• Si le fichier est ouvert sur un périphérique COM, LOC renvoie le nombre de caractères dans le tampon d'entrée, avec un maximum de 255.</li> <li>• Si le fichier est ouvert vers KYBD : , LOC renvoie 0.</li> </ul> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• file_num est une expression numérique dans la plage [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• num_fichier ne doit pas être précédé d'un #.</li> <li>• En mode OUTPUT ou APPEND, avant toute écriture, LOC renvoie 0. Après le 128e caractère est écrit, LOC renvoie 1.</li> <li>• En mode INPUT, avant toute lecture, LOC renvoie 1. Une fois le 129e caractère lu, LOC renvoie 2.</li> <li>• Si le codage de texte est défini, les caractères peuvent être codés par des séquences de plus de un octet. LOC renverra le nombre d'octets plutôt que le nombre d'encodés personnages.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• num_fichier n'est pas dans [-32768—32767] : Overflow.</li> <li>• file_num n'est pas dans [0—255] : Illegal function call.</li> <li>• num_fichier n'est pas un fichier ouvert : Bad file number.</li> <li>• file_num est ouvert sur un périphérique LPT : Bad file mode.</li> </ul>		



<b>LOF</b>	length = LOF(file_num)	Renvoie le nombre d'octets dans le fichier ouvert sous file_num.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• file_num est une expression numérique dans la plage [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si file_num est ouvert sur un périphérique COM:, LOF renvoie le nombre d'octets libres dans le tampon d'entrée.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• num_fichier n'est pas dans [-32768—32767] : Overflow.</li> <li>• file_num n'est pas dans [0—255] : Illegal function call.</li> <li>• num_fichier n'est pas un fichier ouvert : Bad file number.</li> <li>• file_num est ouvert sur un périphérique LPT : Bad file mode.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si le codage de texte est défini, les caractères peuvent être codés par des séquences de plus de un octet. LOF renverra le nombre d'octets plutôt que le nombre de caractères encodés.</li> </ul>		
<b>LOG</b>	$y = \text{LOG}(x)$	Renvoie le logarithme népérien de x.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• x est une expression numérique supérieure à zéro.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur de simple précision.</li> <li>• LOG(x) peut différer de GW-BASIC de 1 dans le chiffre le moins significatif.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> <li>• x est nul ou négatif : Illegal function call.</li> </ul>		

<b>LPOS</b>	position = LPOS(printer_number)	Renvoie la position de la colonne pour une imprimante.
<p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• printer_number est une expression numérique dans [0—3]. Si c'est 0 ou 1, la position pour LPT1 : est renvoyé. Si c'est 2, LPT2 : ; 3, LPT3 :.</li> </ul> <p><b>Remarques</b></p> <ul style="list-style-type: none"> <li>• Lors du passage en mode direct, LPT1 : (mais pas les autres imprimantes) est vidé et sa position est remise à 1.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• printer_number a une valeur de chaîne : Type mismatch.</li> <li>• printer_number n'est pas dans [-32768—32767] : Overflow.</li> <li>• printer_number n'est pas dans [0—3] :Illegal function call.</li> </ul>		
<b>MID\$ (function)</b>		substring = MID\$(string, position [,length])
<p>Renvoie une sous-chaîne de chaîne commençant à la position , en comptant à partir de 1. La sous-chaîne a la longueur de length si spécifié. Si la longueur n'est pas spécifiée, la sous-chaîne s'étend jusqu'à la fin de la chaîne.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• string est une expression de chaîne.</li> <li>• position est une expression numérique comprise entre 1 et la longueur de la chaîne, inclus.</li> <li>• length est une expression numérique dans [0—255].</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• la chaîne a une valeur numérique ou la position ou la longueur a des valeurs de chaîne : Type mismatch.</li> <li>• la position ou la longueur ne sont pas dans [-32768—32767] : Overflow.</li> <li>• la position n'est pas dans [1—255] : Illegal function call.</li> <li>• la longueur n'est pas dans [0—255] : Illegal function call.</li> </ul>		

<b>MKD\$</b>	bytes = MKD\$(double)	Renvoie la représentation interne au format binaire Microsoft sur 8 octets d'un nombre à double précision.
<b>Les erreurs</b> <ul style="list-style-type: none"> <li>• double a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>MKI\$</b>	bytes = MKI\$(int)	Renvoie la représentation little-endian interne de 2 octets d'un entier.
<b>Les erreurs</b> <ul style="list-style-type: none"> <li>• int a une valeur de chaîne : Type mismatch.</li> <li>• int n'est pas dans [-32768—32767] : Overflow.</li> </ul>		
<b>MKS\$</b>	bytes = MKS\$(single)	Renvoie la représentation interne au format binaire Microsoft sur 8 octets d'un nombre simple précision.
<b>Les erreurs</b> <ul style="list-style-type: none"> <li>• single a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>OCT\$</b>	octal = OCT\$(x)	Renvoie une chaîne avec la représentation octale de x.
Paramètres <ul style="list-style-type: none"> <li>• x est une expression numérique dans [-32768—65535].</li> </ul> Les valeurs pour x négatif sont affichées comme complément à deux.  <b>Les erreurs</b> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> <li>• x n'est pas dans [-32768—65535] : Overflow.</li> </ul>		

<b>PEEK</b>	value = PEEK(address)	
<p>Renvoie la valeur de la mémoire au segment * 16 + adresse où segment est le segment courant défini avec DEF SEG.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• adresse est une expression numérique dans [-32768—65535]. Les valeurs négatives sont interprété comme leur complément à deux.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• La mémoire n'est que partiellement émulée en PC-BASIC. Voir Modèle de mémoire pour les adresses prises en charge. En dehors des zones émulées, PEEK renvoie 0.</li> <li>• Les valeurs d'une adresse mémoire particulière peuvent être prédéfinies sur la ligne de commande à l'aide de l'option peek. Cela peut être utilisé pour la compatibilité avec les anciens programmes. Ces valeurs remplaceront les valeurs de vidéo ou de segment de données, si elles se trouvent à ces emplacements.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• address a une valeur de chaîne : Type mismatch.</li> <li>• address n'est pas dans [-32768—65535] : Overflow.</li> </ul>		
<b>PEN (function)</b>	x = PEN(mode)	Lit le crayon lumineux. Ce que cette fonction retourne dépend de mode :
mode	Valeur de retour	
0	booléen ; si le crayon lumineux est tombé depuis le dernier poll.	
1	Coordonnée x de la dernière position du stylo vers le bas	
2	coordonnée y de la dernière position du stylo vers le bas	
3	booléen ; si le stylet est actuellement baissé	
4	Coordonnée x de la position actuelle du stylet	
5	coordonnée y de la position actuelle du stylet	
6	coordonnée de rangée de caractères de la dernière position du stylo vers le bas	
7	coordonnée de la colonne de caractères de la dernière position du stylo vers le bas	
8	coordonnée de ligne de caractères de la position actuelle du stylet	
9	coordonnée de la colonne de caractères de la position actuelle du stylet	

### Paramètres

- mode est une expression numérique dans [0—9].

### Remarques

- Dans PC-BASIC, pour lire le stylet enfoncé, appuyez sur le bouton de la souris.  
Pour la position du stylo, lit la position du pointeur de la souris.

### Les erreurs

- mode a une valeur de chaîne : Type mismatch.
- le mode n'est pas en [-32768—32767] : Overflow.
- le mode n'est pas dans [0—9] : Illegal function call.

<b>PLAY (function)</b>	length = PLAY(voice)	
------------------------	----------------------	--

Renvoie le nombre de notes dans la file d'attente de la musique de fond. La valeur de retour est dans [0—32].

### Paramètres

- voice est une expression numérique dans [0—255]. Si syntax={pcjr|tandy} , indique pour quel canal de voix de tonalité le nombre de notes doit être retourné.

Si la voice n'est pas dans [0—2] , la file d'attente pour voice 0 est renvoyée.

Pour les autres choix de syntaxe , la valeur vocale n'a aucun effet.

### Remarques

- Il y a au maximum 32 notes dans la file d'attente musicale. Cependant, à moins que l'articulation ne soit réglé sur legato, il y a de courts écarts entre chaque note; ceux-ci sont comptés comme notes séparées dans la file d'attente. Effectivement, la longueur de la file d'attente est donc de 16 pour la valeur par défaut et staccato et 32 pour legato.

### Les erreurs

- voice a une valeur de chaîne : Type mismatch.
- la voix n'est pas dans [0—255] : Illegal function call.
- la voix n'est pas dans [-32768—32767] : Overflow.

## PMAP

transformed\_coord =  
PMAP(original\_coord, fn)

Cartes entre la fenêtre d'affichage et les coordonnées logiques ( WINDOW ).

Si aucune VUE n'a été définie, les coordonnées de la fenêtre sont des coordonnées physiques.

En fonction de la valeur de fn , PMAP passe des coordonnées logiques aux coordonnées de la fenêtre ou vice versa :

fn

0 renvoie viewport x donnée logique x

1 renvoie viewport y donnée logique y

2 retour logique x fenêtre viewport x

3 retour logique y donné viewport y

Paramètres

- fn est une expression numérique dans [0—3].

Remarques

- Initialement, en mode texte, PMAP renvoie 0.
- Dans GW-BASIC, PMAP se comporte anormalement lors des changements d'ÉCRAN, où il renvoie parfois des résultats comme si le dernier paramètre WINDOW avait persisté. Ce comportement n'est pas implémenté dans PC-BASIC.

Les erreurs

- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.
- Une coordonnée physique n'est pas dans [-32768—32767] : Overflow.
- fn n'est pas dans [-32768—32767] : Overflow.
- fn n'est pas dans [0—3] : Illegal function call.

**POINT** (current  
coordinate)

coord = POINT(fn)

Renvoie une coordonnée actuellement active de l'écran graphique.

Il s'agit généralement de la dernière position à laquelle un pixel a été tracé, du deuxième coin indiqué dans une commande LINE ou du centre de la fenêtre si rien n'a été tracé.

fn est une expression numérique dans [0—3].

La coordonnée renvoyée dépend de la valeur de fn :

fn

0 viewport x

1 viewport y

2 logical x

3 logical y

Paramètres

- fn est une expression numérique dans [0—3].

Remarques

- En mode texte, renvoie la coordonnée active de tout mode graphique précédent ; si aucun mode graphique n'a été actif, renvoie 0.

**Les erreurs**

- fn a une valeur de chaîne : Type mismatch.
- fn n'est pas dans [-32768—32767] : Overflow.
- fn n'est pas dans [0—3] : Illegal function call.

<b>POINT (pixel attribute)</b>		attrib = POINT(x, y)
<p>Renvoie l'attribut du pixel à la coordonnée logique x , y.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• x , y sont des expressions numériques dans [-32768—32767].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si x , y est en dehors de l'écran, renvoie -1.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• La fonction est appelée en mode texte : Illegal function call.</li> <li>• x ou y a une valeur de chaîne : Type mismatch.</li> <li>• x ou y ou les coordonnées physiques qu'ils traduisent ne sont pas dans [-32768—32767] : Overflow.</li> </ul>		
<b>POS</b>	pos = POS(dummy)	Renvoie la position actuelle de la colonne du curseur, dans la plage [1—80].
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• dummy est une expression valide de n'importe quel type ; sa valeur n'a aucun effet.</li> </ul>		
<b>RIGHT\$</b>	child = RIGHT\$(parent,num_chars)	
<p>Renvoie les num_chars caractères les plus à droite du parent. Si num_chars est égal à zéro ou parent est vide, RIGHT\$ renvoie une chaîne vide. Si num_chars est supérieur à la longueur de parent , renvoie le parent.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• parent est une expression de chaîne.</li> <li>• num_chars est une expression numérique dans [0—255].</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• num_chars a une valeur de chaîne : Type mismatch.</li> <li>• num_chars n'est pas dans [-32768—32767] : Overflow.</li> <li>• num_chars n'est pas dans [0—255] : Illegal function call.</li> </ul>		



<b>RND</b>	random = RND[(x)]	Renvoie un nombre pseudo-aléatoire dans l'intervalle [0—1)
<p><b>Paramètres</b></p> <p>x est une expression numérique.</p> <ul style="list-style-type: none"> <li>• Si x est zéro, RND répète le dernier nombre pseudo-aléatoire.</li> <li>• Si x est supérieur à zéro, un nouveau nombre pseudo-aléatoire est renvoyé.</li> <li>• Si x est négatif, x est converti en une valeur à virgule flottante simple précision et la graine de nombre aléatoire est définie sur la valeur absolue de sa mantisse.</li> </ul> <p>La fonction génère alors un nouveau nombre pseudo-aléatoire avec cette graine. Puisque le seul mantisse de x est utilisée, deux valeurs quelconques dont le rapport est une puissance de 2 produiront la même graine.</p> <p>Notez que cette procédure pour générer une nouvelle graine diffère de celle utilisée par RANDOMIZE.</p> <p><b>Remarques</b></p> <ul style="list-style-type: none"> <li>• La fonction RND de PC-BASIC génère des nombres pseudo-aléatoires via une générateur congruentiel avec modulo 224, multiplicateur 214013 et incrément 2531011. Cela reproduit exactement les nombres aléatoires du RND de GW-BASIC.</li> <li>• Il convient de noter, cependant, qu'il s'agit d'un très mauvais générateur de nombres aléatoires : impliquent une période de récurrence de 224, ce qui signifie qu'après moins de 17 millions d'appels RND s'enroulera et commencera à parcourir exactement le même série de chiffres à nouveau. RND ne doit pas être utilisé pour la cryptographie, simulations scientifiques ou toute autre chose sérieuse à distance.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> </ul>		

<b>SCREEN (function)</b>	value = SCREEN(row, column [, fn])	
<p>Renvoie le point de code ou l'attribut de couleur du caractère à la position row , col.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• row est une expression numérique dans la plage [1—25].</li> <li>• col est une expression numérique comprise entre 1 et la largeur de l'écran (40 ou 80).</li> <li>• fn est une expression numérique dans [0—255]. S'il est nul ou non spécifié, le code point du caractère est renvoyé. S'il est différent de zéro, en mode texte l'attribut est renvoyé; dans les autres écrans, 0 est renvoyé.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Tout paramètre a une valeur de chaîne : Type mismatch.</li> <li>• fn n'est pas dans [0—255] : Illegal function call.</li> <li>• fn n'est pas dans [-32768—32767] : Overflow.</li> <li>• la ligne n'est pas à l'intérieur de la zone VIEW PRINT actuelle : appel de fonction illégal.</li> <li>• KEY ON et ligne=25 : Illegal function call.</li> <li>• col n'est pas dans [1, largeur] : Illegal function call.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, la fonction SCREEN a un comportement d'erreur anormal : constructions qui, pour d'autres fonctions, soulèverait une erreur de syntaxe ou un opérande manquant à la place raise Illegal function call. Ce comportement n'est pas reproduit dans PC-BASIC.</li> </ul>		
<b>SGN</b>	sign = SGN(number)	Renvoie le signe du nombre : 1 pour positif, 0 pour zéro et -1 pour négatif.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• nombre est une expression numérique.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• number a une valeur de chaîne : Type mismatch.</li> </ul>		

<b>SIN</b>	sine = SIN(angle)	Renvoie le sinus de l'angle.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• angle est une expression numérique donnant l'angle en <b>radians</b>.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur de simple précision.</li> <li>• Le sinus renvoyé diffère généralement de la valeur renvoyée par GW-BASIC dans le chiffre le moins significatif.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• angle a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>SPACE\$</b>	spaces = SPACE\$(number)	Renvoie une chaîne d'espaces numériques.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• number est une expression numérique dans [0—255].</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• number a une valeur de chaîne : Type mismatch.</li> <li>• le numéro n'est pas dans [-32768—32767] : Overflow.</li> <li>• le numéro n'est pas dans [0—255] : Illegal function call.</li> </ul>		
<b>SQR</b>	root = SQR(number)	Renvoie la racine carrée du nombre.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• nombre est une expression numérique.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur de simple précision.</li> </ul>		

## Les erreurs

- le nombre a une valeur de chaîne : Type mismatch

## STICK

pos = STICK(axis)

Renvoie une coordonnée d'un axe du joystick. Toutes les coordonnées renvoyées sont dans la plage [1—254] avec 128 indiquant la position neutre.

axe Valeur de retour

- 0 1er manette x coordonnée
- 1 1er manette y coordonné
- 2 2ème manette x coordonnée
- 3 2ème manette y coordonnée

Paramètres

- axis est une expression numérique dans [0—3] et indique quel axe lire.

## Les erreurs

- l'axe a une valeur de chaîne : Type mismatch
- l'axe n'est pas dans [-32768—32767] : Overflow.
- l'axe n'est pas dans [0—3] : Illegal function call.

## STR\$

repr = STR\$(number)

Paramètres

- nombre est une expression numérique.

## Les erreurs

- number a une valeur de chaîne : Type mismatch.

<b>STRIG (function)</b>	result = STRIG(mode)	Renvoie l'état des boutons de déclenchement du joystick.
-------------------------	----------------------	--

STRIG renvoie les résultats suivants, tous sont des valeurs booléennes :

- 0 1ère manette, la 1ère gâchette a été enfoncée depuis la dernière interrogation.
- 1 1ère manette, la 1ère gâchette est actuellement enfoncée.
- 2 2ème manette, 1ère gâchette a été enfoncée depuis la dernière interrogation.
- 3 2ème manette, la 1ère gâchette est actuellement enfoncée.
- 4 1ère manette, le 2e déclencheur a été enfoncé depuis la dernière interrogation.
- 5 1ère manette, la 2ème gâchette est actuellement enfoncée.
- 6 2e manette, 2e gâchette a été enfoncée depuis la dernière interrogation.
- 7 2e manette, 2e gâchette est actuellement enfoncée.

#### Paramètres

- mode est une expression numérique dans [0—7].

#### Remarques

- La fonction STRIG renvoie des résultats corrects quel que soit l'état STRIG ON ou si STRIG(0) a été appelé en premier.

#### Les erreurs

- mode a une valeur de chaîne : Type mismatch.
- le mode n'est pas en [-32768—32767] : Overflow.
- le mode n'est pas dans [0—7] : Illegal function call.

<b>STRING\$</b>	string = STRING\$(length, char)	Renvoie une chaîne dont la longueur est multipliée par le caractère char.
-----------------	---------------------------------	---

#### Paramètres

- Si char est une expression numérique, elle doit être comprise entre [0—255] et est interprétée comme point de code du caractère.
- Si char est une expression de chaîne, son premier caractère est utilisé.

#### Les erreurs

- length a une valeur de chaîne : Type mismatch.
- char est la chaîne vide : Illegal function call.
- car ou longueur n'est pas dans [-32768—32767] : Overflow.
- car ou longueur n'est pas dans [0—255] : Illegal function call.

<b>TAN</b>	tangent = TAN(angle)	Renvoie la tangente de l'angle.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• angle est une expression numérique donnant l'angle en <b>radians</b>.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• À moins que PC-BASIC ne soit exécuté avec l'option double, cette fonction renvoie un valeur simple précision.</li> <li>• La tangente renvoyée diffère généralement de la valeur renvoyée par GW-BASIC dans le chiffre le moins significatif.</li> <li>• Pour des angles proches de multiples de <math>\pi/2</math>, la tangente est divergente ou proche de zéro.</li> </ul> <p>Les valeurs renvoyées auront une très faible précision dans ces cas.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• angle a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>TIMER (function)</b>	seconds = TIMER	Renvoie le nombre de secondes depuis minuit sur l'horloge BASIC interne.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• Mises à jour TIMER en tiques de 1/20 seconde.</li> <li>• Les deux octets les moins significatifs de TIMER sont souvent utilisés comme graine pour le générateur de nombres pseudo-aléatoires via RANDOMIZE TIMER.</li> </ul> <p>Étant donné que ces octets ne prennent que des valeurs d'un ensemble limité, ce n'est en fait pas particulièrement un bon générateur de hasard. Cependant, le générateur de nombres pseudo-aléatoires inclus avec GW-BASIC et PC-BASIC est si faible qu'il ne devrait de toute façon pas être utilisé pour quelque chose de sérieux.</p> <ul style="list-style-type: none"> <li>• Cette fonction ne prend aucun argument.</li> </ul>		

<b>USR</b>	value = USR[n](expr)	Raises Appel de fonction illégal.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• n est un chiffre [0—9].</li> <li>• expr est une expression.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, appelle une fonction de code machine et renvoie sa valeur de retour.</li> <li>• Cette fonction n'est pas implémentée dans PC-BASIC.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• n n'est pas un chiffre [0—9] : Syntax error.</li> </ul>		
<b>VAL</b>	value = VAL(string)	
<p>Renvoie la valeur numérique de l'expression de chaîne chaîne. L'analyse s'arrête dès que le premier caractère rencontré ne peut pas faire partie d'un nombre.</p> <p>Si aucun caractère n'est analysé, VAL renvoie zéro. Voir la section sur les littéraux numériques pour les formats numériques reconnus.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Les espaces avant ou même à l'intérieur d'un nombre sont ignorés : VAL(" 1 0") renvoie 10.</li> <li>• Si la chaîne contient l'un des caractères séparateurs ASCII CHR\$(28) (fichier séparateur), CHR\$(29) (séparateur de groupe) ou CHR\$(31) (séparateur d'unité), VAL renvoie zéro.</li> </ul> <p>Ce n'est pas le cas avec CHR\$(30) (séparateur d'enregistrement). Ce comportement est conforme à GW-BASIC.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• chaîne a une valeur numérique : Type mismatch.</li> </ul>		

<b>VARPTR</b>	pointer = VARPTR({name #file_num})	
<p>Renvoie l'adresse mémoire du nom de variable ou du bloc de contrôle de fichier du numéro de fichier num_fichier.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• name est une variable précédemment définie ou un élément de tableau entièrement indexé.</li> <li>• file_num est un numéro de dossier légal.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• VARPTR peut être utilisé avec PEEK pour lire la représentation interne d'une variable.</li> <li>• Si nom est un élément de tableau indéfini, il sera implicitement alloué de la même manière que s'il avait été utilisé dans une expression.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• nom est un scalaire qui n'a pas été défini auparavant : Illegal function call.</li> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• file_num n'est pas dans [1, max_files] , où max_files est le nombre maximum de nombre de fichiers défini par l'option max-files : Bad file number.</li> </ul>		
<b>VARPTR\$</b>	pointer = VARPTR\$(name)	
<p>Renvoie l'adresse mémoire de la variable nom sous la forme d'une chaîne de 3 octets.  name est un élément de tableau entièrement indexé (qui peut ou non avoir été défini) ou une variable scalaire définie précédemment. Le premier octet est la longueur de l'enregistrement vers lequel pointe le pointeur :</p> <p>2 pour les entiers  3 pour les chaînes (longueur + pointeur vers l'espace de la chaîne)  4 pour floteurs simple précision  8 pour floteurs double précision</p> <p>Les deux derniers octets sont l'adresse du pointeur (telle que renvoyée par VARPTR ) dans l'ordre little-endian.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si nom est un élément de tableau indéfini, il sera implicitement alloué de la même manière que s'il avait été utilisé dans une expression.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• nom est un scalaire qui n'a pas été défini auparavant : Illegal function call.</li> </ul>		



## Tableau des scancodes du clavier

Key	Cod e	Key	Cod e	Key	Cod e	Key	Cod e
Esc	01	U	16	/	2B	F7	41
1/!	02	I	17	Z	2C	F8	42
2/@	03	O	18	X	2D	F9	43
3/#	04	P	19	C	2E	F10	44
4/\$	05	[/{	1A	V	2F	NUM LOCK	45
5/%	06	]}/	1B	B	30	SCROLL LOCK	46
6/^	07	ENTER	1C	N	31	7/HOME	47
7/&	08	CTRL	1D	M	32	8/CURSORS UP	48
8/*	09	A	1E	,/<	33	9/PAGE UP	49
9/(	0A	S	1F	//?	35	-	4A
0/)	0B	D	20	RIGHT SHIF T	36	4/CURSORS LEFT	4B
-/_	0C	F	21	*/PRTSCR	37	5	4C
=/+	0D	G	22	ALT	38	6/CURSORS RIGH T	4D
BACKSPAC E	0E	H	23	SPACE	39	+	4E
TAB	0F	J	24	CAPS LOCK	3A	1/END	4F
Q	10	K	25	F1	3B	2/CURSORS DOW N	50
W	11	L	26	F2	3C	3/PAGE DN	51
E	12	;/:	27	F3	3D	0/INS	52
R	13	'/"	28	F4	3E	./DEL	53
T	14	'/~	29	F5	3F	SysReq	54
Y	15	LEFT SHIF T	2A	F6	40		

## Autres

⌘ (Non-US 102-key)	56		?? Henkan (Japanese 106-key)	79
F11	57		??? Muhenkan (Japanese 106-key)	7B
F12	58		??/? Hankaku/Zenkaku (Japanese 106-key)	29
Left Logo (Windows 104-key)	5B		¥   (Japanese 106-key)	7D
Right Logo (Windows 104-key)	5C		?? Hanja (Korean 103-key)	F1
Menu (Windows 104-key)	5D		?? Han/Yeong (Korean 103-key)	F2
????/???? Hiragana/Katakana (Japanese 106-key)	70		\ ? ° (Brazilian ABNT2)	73
\ _ (Japanese 106-key)	73		keypad. (Brazilian ABNT2)	7E

## Annexe des erreurs

### 1 NEXT without FOR

Une instruction NEXT a été rencontrée pour laquelle aucun FOR correspondant n'a pu être trouvé.

### 2 Syntax error

La syntaxe BASIC est incorrecte. Une instruction ou une expression a été mal saisie ou appelée de l'une des nombreuses manières incorrectes. Cette erreur est également déclenchée sur une ligne DATA si une instruction READ qui rencontre une entrée de données d'un format incorrect.

### 3 RETURN without GOSUB

Une instruction RETURN a été rencontrée pour laquelle aucun appel GOSUB n'a été effectué.

### 4 Out of DATA

Une instruction READ tente de lire plus d'entrées de données qu'il n'est possible d'en trouver à partir de l'emplacement DATA actuel.

### 5 Illegal function call

Une instruction, une fonction ou un opérateur a été appelé avec des paramètres en dehors de la plage acceptée. Cette erreur est également générée pour une grande variété d'autres conditions - vérifiez la référence de l'instruction ou de la fonction appelée.

### 6 Overflow

Un résultat d'expression numérique ou une valeur intermédiaire est trop grand pour le format numérique requis.

### 7 Out of memory

Il n'y a pas assez de mémoire BASIC libre pour terminer l'opération. Trop de mémoire est consommée par le programme ; variables, tableaux et chaînes, ou piles d'exécution pour les boucles, les sous-programmes ou les fonctions définies par l'utilisateur.

### 8 Undefined line number

Une référence est faite à un numéro de ligne qui n'existe pas dans le programme.

### 9 Subscript out of range

Un index de tableau (indice) est utilisé en dehors de la plage réservée pour ce tableau par l'instruction DIM.

### 10 Duplicate Definition

Une instruction DIM est utilisée sur un tableau qui a été dimensionné auparavant (implicitement ou explicitement) ou OPTION BASE est appelée d'une manière qui entre en conflit avec une définition implicite ou explicite antérieure de l'index de départ.

## 11 Division by zero

Une tentative est faite pour diviser un nombre par zéro ou par un nombre trop petit pour être distingué de zéro dans la précision du format numérique.

## 12 Illegal direct

Une instruction DEF FN est utilisée en mode direct.

## 13 Type mismatch

L'expression utilisée est d'un type qui ne peut pas être converti dans le type requis pour la fonction ou l'instruction. Le plus souvent, ceci est déclenché si un argument de chaîne est fourni à une instruction ou une fonction qui attend un nombre, ou vice versa.

## 14 Out of string space

Il n'y a pas assez de mémoire BASIC libre pour stocker la variable de chaîne.

## 15 String too long

Un résultat d'expression de chaîne ou une valeur intermédiaire comporte plus de 255 caractères.

## 16 String formula too complex

## 17 Can't continue

L'instruction CONT est utilisée dans des circonstances où la poursuite de l'exécution du programme n'est pas possible.

## 18 Undefined user function

La fonction FN est appelée avec un nom de fonction pour lequel aucune définition n'a été faite par une instruction DEF FN.

## 19 No RESUME

Le programme se termine à l'intérieur d'une routine d'interception d'erreurs qui n'a pas été fermée avec RESUME ou END.

## 20 RESUME without error

Une instruction RESUME est rencontrée alors que le programme n'exécute pas de routine d'interception d'erreurs.

## 21 inutilisés

## 22 Missing operand

Une expression d'opérateur manque un opérande ou une fonction ou une instruction n'est pas fournie avec des paramètres suffisants.

### 23 Line buffer overflow

Une instruction INPUT ou LINE INPUT a rencontré une chaîne d'entrée de plus de 255 caractères ou le fichier de programme en texte brut chargé par LOAD, CHAIN ou MERGE contient une ligne de plus de 255 caractères. La tentative de chargement d'un fichier texte contenant des fins de ligne LF plutôt que CR LF peut provoquer cette erreur.

### 24 Device Timeout

L'établissement de liaison a échoué sur un périphérique série ou un périphérique à bande a atteint la fin de la bande.

### 25 Device Fault

### 26 FOR without NEXT

Une instruction FOR a été rencontrée pour laquelle aucune instruction NEXT correspondante n'a été trouvée.

### 27 Out of paper

Tentative d'écriture sur une imprimante qui n'a plus de papier ou sur un autre périphérique parallèle qui a déclenché une condition de manque de papier.

### 28 inutilisés

### 29 WHILE without WEND

Une instruction WHILE a été rencontrée pour laquelle aucune instruction WEND correspondante n'a été trouvée.

### 30 WEND without WHILE

Une instruction WEND a été rencontrée pour laquelle aucune instruction WHILE correspondante n'a été trouvée.

### 31—49 inutilisé

### 50 FIELD overflow

Tentative de lecture, d'écriture ou de définition d'une variable FIELD au-delà de la longueur du tampon de fichier à accès aléatoire.

### 51 Internal error

L'instruction TERM est exécutée mais aucun programme de gestionnaire de terminal n'a été défini.

### 52 Bad file number

Un numéro de fichier auquel aucun fichier n'est ouvert est accédé, ou le numéro de fichier utilisé dans une instruction OPEN est en dehors de la plage des numéros de fichier autorisés, ou (ce qui prête à confusion) la spécification de fichier est vide, mal formée ou contient des caractères illégaux.

### 53 File not found

Un fichier nommé sur un périphérique de disque est introuvable.

### 54 Bad file mode

Le mode de fichier demandé dans une instruction OPEN n'existe pas ou n'est pas pris en charge pour le périphérique donné, ou la fonction de fichier appelée n'est pas prise en charge pour ce périphérique, ou la fonction ou l'instruction appelée nécessite un fichier ouvert pour RANDOM et le fichier ne l'est pas.

### 55 File already open

Tentative d'ouverture d'un fichier vers un numéro de fichier déjà utilisé ; ou une tentative est faite pour ouvrir un fichier pour OUTPUT ou APPEND sur un périphérique série, disque ou cassette lorsqu'un fichier (ou, sur un périphérique disque, un fichier portant le même nom) est déjà ouvert pour OUTPUT ou APPEND sur ce périphérique ; ou une instruction KILL ou NAME est exécutée sur un fichier disque lorsqu'un fichier portant le même nom est ouvert sur le même périphérique.

### 56 inutilisés

### 57 Device I/O error

Une erreur d'E/S s'est produite lors de l'entrée/sortie d'un périphérique. Cela inclut les erreurs de familiarisation, les échecs de vérification CRC et les fins de bande inattendues sur les périphériques de cassette.

### 58 File already exists

Le nouveau nom proposé pour un fichier disque dans une instruction NAME est déjà utilisé.

### 59—60 inutilisé

### 61 Disk full

L'espace libre sur le périphérique de disque est insuffisant pour terminer l'opération.

### 62 Input past end

Une tentative est faite pour récupérer l'entrée d'un fichier qui a dépassé sa fin de fichier.

### 63 Bad record number

Un numéro d'enregistrement de fichier à accès aléatoire est référencé en dehors de la plage autorisée.

### 64 Bad file name

Le nom de fichier ou une autre chaîne de paramètres de périphérique dans une spécification de fichier est incorrect ou contient des caractères non autorisés.

### 65 inutilisé

#### 66 Direct statement in file

Une ligne sans numéro de ligne est rencontrée dans un fichier de programme en texte brut.

#### 67 Too many files

#### 68 Device Unavailable

Tentative d'accès à un périphérique qui n'existe pas ou qui n'est pas activé.

#### 69 Communication buffer overflow

Un périphérique série reçoit plus de données que sa mémoire tampon n'en contient.

#### 70 Permission Denied

L'accès demandé à un fichier n'est pas accordé en raison de restrictions LOCK, d'un verrouillage du système d'exploitation ou d'autorisations de fichier du système d'exploitation insuffisantes.

#### 71 Disk not Ready

Le périphérique de disque n'est pas prêt pour l'accès. Par exemple, il n'y a pas de disquette dans un lecteur de disquette ou le verrou du lecteur est ouvert.

#### 72 Disk media error

#### 73 Advanced Feature

#### 74 Rename across disks

Tentative d'utilisation de l'instruction NAME pour déplacer un fichier d'un périphérique de disque à un autre.

#### 75 Path/File access error

Tentative de création d'un répertoire qui existe déjà ou de suppression d'un répertoire qui n'est pas vide.

#### 76 Path not found

Une instruction OPEN, MKDIR, RMDIR ou CHDIR est exécutée en référence à un chemin (parent) qui n'existe pas sur le périphérique de disque.

#### 77 Deadlock

Tout code d'erreur auquel aucun message n'est associé générera le message Erreur non imprimable.

Si une erreur se produit en mode direct, le message d'erreur est imprimé comme ci-dessus.

Si l'erreur se produit dans un programme, le message est complété par le numéro de la ligne dans laquelle l'erreur s'est produite.

Par exemple, Appel de fonction illégal dans 100

indique que l'appel de fonction illégal a eu lieu dans la ligne numéro 100.

Si une erreur de syntaxe se produit pendant l'exécution du programme, le message d'erreur est suivi d'une liste de la ligne de programme dans laquelle l'erreur s'est produite, avec le curseur positionné à l'endroit où l'erreur a été déclenchée.

Une erreur de division par zéro ou, dans un calcul en virgule flottante, un débordement, n'interrompra pas l'exécution à moins qu'elle ne se produise dans une routine de gestion des erreurs. Le message d'erreur sera imprimé sur la console et le résultat du calcul incriminé sera considéré comme la valeur maximale qui tient dans la variable à virgule flottante appropriée. Le débordement dans un calcul d'entier interrompra toujours l'exécution comme les autres erreurs.

## **Autres messages**

### **Break**

L'exécution d'une instruction composée ou d'un programme a été interrompue par une instruction CONT ou par une interruption du clavier de l'utilisateur (telle que Ctrl + Break ).

Si l'interruption se produit dans un programme, le message Break sera complété par le numéro de la ligne dans laquelle l'interruption s'est produite.

### **?Redo from start**

L'entrée fournie sur la console pour une instruction INPUT ne correspond pas au format attendu. Le nombre ou le type d'entrées n'est pas correct. Entrez à nouveau toutes les entrées.

### **Undefined line ref\_num in line\_num**

L'instruction RENUM a rencontré une référence au numéro de ligne ref\_num qui n'est pas défini dans le programme. La référence se produit sur la ligne numéro line\_num. La référence de numéro de ligne indéfinie ne sera pas renumérotée.

### **filename Found.**

Un fichier correspondant à la spécification demandée a été trouvé sur le périphérique de cassette. Ce message n'apparaît qu'en mode direct.

### **filename Skipped.**

Un fichier ne correspondant pas à la spécification demandée a été rencontré sur le périphérique de cassette. Ce message n'apparaît qu'en mode direct.



## PC-BASIC Instructions

### Déclarations

Une ligne de programme est composée d'un numéro de ligne et d'une ou plusieurs instructions.

Si plusieurs instructions sont placées sur une même ligne, elles doivent être séparées par des deux-points :

Les déclarations peuvent être vides. Chaque instruction a sa propre syntaxe idiosyncratique.

De nombreux ouvrages de référence sur GW-BASIC distinguent les commandes et les instructions ; cette distinction découle de la conception originale de Dartmouth du langage BASIC, dans laquelle les commandes ne faisaient pas partie du langage et ne pouvaient pas être utilisées dans les programmes, mais étaient plutôt utilisées pour contrôler l'interpréteur lui-même.

Cependant, dans GW-BASIC, cette distinction est moins utile et, par conséquent, cette référence inclut ce qui est traditionnellement considéré comme des commandes dans la catégorie des instructions.

<b>AUTO</b>	AUTO [line_number.] [, [increment]]
<p>Démarre la numérotation automatique des lignes.</p> <p>Les numéros de ligne sont générés automatiquement lorsque vous appuyez sur Entrée. Si une ligne de programme existe à un numéro de ligne généré, un * s'affiche après le numéro de ligne.</p> <p>Pour éviter d'écraser cette ligne, laissez-la vide et appuyez sur Entrée. Pour arrêter la numérotation automatique des lignes, appuyez sur Ctrl + Break ou Ctrl + C.</p> <p>La ligne en cours d'édition à ce point n'est pas enregistré. Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.</p> <p>Paramètres</p> <ul style="list-style-type: none"><li>• La numérotation des lignes commence à line_number, si est spécifié, la numérotation des lignes commence à la dernière ligne de programme enregistrée. Sinon, la numérotation des lignes commence à 10.</li><li>• Chaque numéro de ligne suivant est incrémenté par incrément, increment si est spécifié. Si une virgule est utilisée sans spécifier d'incrément, le dernier incrément spécifié dans une commande AUTO est utilisé. Si ce n'est pas le cas, l'incrément est par défaut de 10.</li></ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"><li>• numéro_ligne n'est pas une valeur entière non signée dans [0—65529] : Syntax error.</li><li>• Lorsque la numérotation automatique des lignes est activée et que la touche Entrée est enfoncée sur une ligne</li></ul>	

vide avec un numéro supérieur à 65519 : Undefined line number.

- l'incrément vaut 0 : Illegal function call.

## **BEEP**

Bip le haut-parleur à 800 Hz pendant 0,25 s.

- Si une erreur de syntaxe est levée, le bip est toujours produit.

## **BEEP (switch)**

BEEP {ON|OFF}

Active ou désactive le haut-parleur interne.

## Remarques

- Uniquement légal avec l'option syntax={pcjr|tandy}.
- Sur PC-BASIC, les haut-parleurs interne et externe sont émulés via le même système audio.

## **BLOAD**

BLOAD file\_spec [, offset]

Charge un fichier d'image mémoire dans la mémoire.

## Paramètres

- L'expression de chaîne file\_spec est une spécification de fichier valide indiquant le fichier à partir duquel lire l'image mémoire.
- offset est une expression numérique dans la plage [-32768—65535]. Il indique un décalage dans le segment DEF SEG actuel où le fichier doit être stocké. S'il n'est pas spécifié, le décalage stocké dans le fichier BSAVE sera utilisé. S'il est négatif, son complément à deux sera utilisé.

## **Les erreurs**

- Le fichier chargé n'est pas au format BSAVE : Bad file mode.
- spéc\_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ; Bad file name (sur les périphériques de disque).
- file\_spec a une valeur numérique : Type mismatch.
- le décalage n'est pas dans la plage [-32768—65535] : Overflow.

<b>BSAVE</b>	BSAVE file_spec, offset, length	Enregistre une zone de mémoire dans un fichier image.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• L'expression de chaîne file_spec est une spécification de fichier valide indiquant le fichier dans lequel écrire.</li> <li>• offset est une expression numérique dans la plage [-32768—65535] indiquant le décalage dans le segment DEF SEG actuel à partir duquel commencer la lecture.</li> <li>• length est une expression numérique dans la plage [-32768—65535] indiquant le nombre d'octets à lire.</li> <li>• Si l'offset ou la longueur sont négatifs, leur complément à deux sera utilisé.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_spec a une valeur numérique : Type mismatch.</li> <li>• spéc_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ; Bad file name (sur les périphériques de disque).</li> <li>• le décalage n'est pas dans la plage [-32768—65535] : Overflow.</li> <li>• la longueur n'est pas comprise dans la plage [-32768—65535] : Overflow.</li> </ul>		
<b>CALL and CALLS</b>	{CALL CALLS} address_var [( p0 [, p1]... )]	
<p>Ne fait rien.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, CALL ou CALLS exécute un sous-programme en langage machine.</li> <li>• Cette instruction n'est pas implémentée dans PC-BASIC.</li> </ul> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• address_var est un nom de variable numérique.</li> <li>• p0, p1,... sont des noms de variables ou des éléments de tableau.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• address_var est une variable de chaîne : Type mismatch.</li> <li>• var_adresse est un littéral ou une expression : Syntax error.</li> </ul>		

<b>CHAIN</b>	CHAIN [MERGE] file_spec [, [line_number_expr] [, ALL] [, DELETE range [, ign]]]
--------------	---

Charge un programme d'un fichier dans la mémoire et l'exécute, en transférant éventuellement des variables.

- Si ALL est spécifié, toutes les variables sont transférées. Sinon, les variables spécifiées dans l'instruction COMMON est transférée.
- Si MERGE est spécifié, le programme chargé est fusionné avec le programme existant. Pour être en mesure de l'utiliser, le fichier de programme indiqué par file\_spec doit être en texte brut format.
- Si DELETE est spécifié, la plage de numéros de ligne est supprimée du code avant la fusion. C'est inutile sans MERGE.

#### Paramètres

- L'expression de chaîne file\_spec est une spécification de fichier valide indiquant le fichier à lire le programme à partir de.
- line\_number\_expr est une expression numérique. Il sera interprété comme un numéro de ligne dans le nouveau programme et l'exécution commencera à partir de ce numéro de ligne. Si line\_number\_expr est négatif, il sera interprété comme son complément à deux.
- range est une plage de numéros de ligne dont le numéro de ligne de fermeture est spécifié et existe avant la fusion.
- ign est facultatif et ignoré.

#### Remarques

- CHAIN préserve le réglage OPTION BASE.
- Uniquement si ALL est spécifié, les définitions DEF FN sont conservées.
- Uniquement si MERGE est spécifié, les définitions DEFINT, DEFSTR, DEFSNG, DEFDBL sont conservé.
- Si spécifié, ALL doit précéder DELETE ; si non spécifié, aucune virgule ne doit être insérée sa place et seulement deux virgules doivent précéder DELETE.

#### Les erreurs

- file\_spec a une valeur numérique : Type mismatch.

- spéc\_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ;  
Bad file name (sur les périphériques de disque).
- Le fichier spécifié dans spéc\_fichier est introuvable : File not found.
- MERGE est spécifié et le programme chargé n'a pas été enregistré en mode texte : Bad file mode.
- Un numéro de ligne dans range est supérieur à 65529 : Syntax error.
- Si une erreur de syntaxe est levée par CHAIN, aucune ligne n'est supprimée et le nouveau programme n'est pas chargé.
- Le numéro de ligne de fermeture dans la plage n'existe pas : Illegal function call.
- Si line\_number\_expr ne correspond pas à un numéro de ligne existant dans le nouveau programme, un appel de Illegal function call est déclenché mais le chargement ou la fusion est en cours effectué.
- Un fichier texte chargé contient des lignes sans numéro de ligne : Direct statement in file.
- Un fichier texte chargé contient des lignes de plus de 255 caractères : Line buffer overflow.

Tentative de chargement d'un fichier texte contenant une ligne LF plutôt que CR LF les terminaisons peuvent provoquer cette erreur.

<b>CHDIR</b>	CHDIR dir_spec	
--------------	----------------	--

Remplacez le répertoire actuel sur un périphérique de disque par dir\_spec. Chaque périphérique de disque a son propre répertoire actuel.

#### Paramètres

- L'expression de chaîne dir\_spec est une spécification de fichier valide indiquant un répertoire sur un périphérique de disque.

#### Les erreurs

- Aucun chemin correspondant n'a été trouvé : Path not found.
- dir\_spec a une valeur numérique : Type mismatch.
- dir\_spec est vide : Bad file name.

<b>CIRCLE</b>	CIRCLE [STEP] (x, y), radius [, [colour] [, [start] [, [end] [, aspect]]]
<p>Dessinez une ellipse ou un secteur d'ellipse.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• Le milieu de l'ellipse est en (x, y). Si STEP est spécifié, le milieu est (x, y) est loin de la position actuelle.</li> <li>• radius est le rayon, en pixels, le long de l'axe longitudinal.</li> <li>• colour est l'attribut de couleur.</li> <li>• Si start et end sont spécifiés, un secteur de l'ellipse est tracé à partir de start radian à end radian, avec zéro radian l'intersection avec le x est l'axe droite. Si une valeur négative est spécifiée, le secteur d'arc est relié par une ligne au point médian.</li> <li>• aspect spécifie le rapport entre le rayon y et le rayon x. Si ce n'est pas spécifiée, la valeur standard du mode SCREEN est utilisée, de manière à faire apparaître l'ellipse comme un cercle sur le matériel d'origine.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Pour aspect <math>\diamond 1</math>, l'algorithme du point médian utilisé ne reproduit pas parfaitement les pixels des ellipses de GW-BASIC.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• L'instruction est exécutée en mode texte : Illegal function call.</li> <li>• début ou fin n'est pas dans <math>[0—2\pi]</math> : Illegal function call.</li> <li>• L'instruction se termine par une virgule : Missing operand.</li> </ul>	
<b>CLEAR</b>	CLEAR [expr] [, [mem_limit] [, [stack_size] [, video_memory]]]
<p>Efface toutes les variables, les tableaux, les fonctions utilisateur DEF FN et les définitions de type DEFtype.</p> <p>Ferme tous les fichiers. Coupe tous les sons. Réinitialise l'état PLAY et place la musique au premier plan.</p> <p>Efface tous les pièges ON ERROR. Réinitialise ERR et ERL à zéro.</p>	

Désactive tous les évènements. Désactive PEN et STRIG.

Réinitialise le générateur de nombres aléatoires. Efface la pile de boucles. Réinitialise l'état DRAW et la position graphique actuelle.

#### Paramètres

- `mem_limit` spécifie la limite supérieure de mémoire utilisable. La valeur par défaut est la précédente taille mémoire. La taille de mémoire par défaut est 65534.
- `stack_size` spécifie la quantité de mémoire disponible pour la pile BASIC. La valeur par défaut est la taille de pile précédente. La taille de pile par défaut est 512.
- `video_memory` spécifie la quantité de mémoire disponible pour la carte vidéo. Ce paramètre n'est légal qu'avec l'une des options `syntax={pcjr, tandy}`. Au lieu d'utiliser CLEAR, l'option vidéo-mémoire peut également être utilisée pour définir la vidéo taille mémoire.

#### Remarques

- Le but de `expr` est inconnu.
- S'il est appelé à l'intérieur d'une boucle FOR — NEXT ou WHILE — WEND, une erreur sera déclenchée à l'instruction NEXT ou WEND, car les piles de boucles ont été effacées.

#### Les erreurs

- L'un des arguments a une valeur de chaîne : Type mismatch.
- `mem_limit`, `stack_size` n'est pas dans [-32768—65535] : Overflow.
- `mem_limit` ou `stack_size` équivaut à 0 : Illegal function call.
- `mem_limit` équivaut à -1 or 65535 : Out of memory.
- `mem_limit` ou `expr` est trop petit: Out of memory.
- `expr` n'est pas dans [-32768—32767] : Overflow.
- `expr` est négatif: Illegal function call.

<b>CLOSE</b>	CLOSE [[#] file_0 [, [#] file_1]...]	
<p>Ferme les fichiers. Si aucun numéro de dossier n'est spécifié, tous les dossiers ouverts sont fermés. Le dièse ( # ) est facultatif et n'a aucun effet.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• file_1, file_2,... sont des expressions numériques donnant des numéros de fichier.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Aucune erreur n'est générée si les numéros de fichiers spécifiés n'étaient pas ouverts.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_1, file_2,... ne sont pas dans [-32768—32767] : Overflow.</li> <li>• file_1, file_2,... ne sont pas dans [0—255] : Illegal function call.</li> <li>• file_1, file_2,... ont une valeur de chaîne : Type mismatch.</li> <li>• L'instruction se termine par une virgule,: Missing operand.</li> <li>• Si une erreur se produit, seuls les fichiers avant la valeur erratique sont fermés.</li> </ul>		
<b>CLS</b>	CLS [x][,]	
<p>Efface l'écran ou une partie de celui-ci. Si x n'est pas spécifié, dans SCREEN 0, la zone d'affichage du texte est effacé ; dans d'autres écrans, la zone d'affichage graphique est effacée. La virgule est facultative et n'a aucun effet.</p> <p>Paramètres</p> <p>x est une expression à valeur numérique.</p> <p>0 Efface l'écran de tous les textes et graphiques  1 Efface uniquement la fenêtre graphique  2 Efface uniquement la fenêtre de texte</p> <p>L'argument facultatif x n'est pas disponible avec syntax={pcjr tandy}.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• x a une valeur de chaîne : Type mismatch.</li> <li>• x n'est pas dans [-32768—32767] : Overflow.</li> <li>• x n'est pas dans [0, 1, 2] : Illegal function call.</li> <li>• Aucune virgule n'est spécifiée mais du texte suit : Illegal function call.</li> <li>• Une virgule est spécifiée suivie de plus : Syntax error.</li> </ul>		



- syntax=pcjr est défini et un argument est spécifié : Syntax error.
- syntax=tandy est défini et un argument est spécifié : Illegal function call.
- Si une erreur se produit, l'écran n'est pas effacé.

**COLOR (text mode)**

COLOR [foreground] [,  
[background] [, border]]

Modifie les attributs actuels de premier plan et d'arrière-plan. Tous les nouveaux caractères imprimés seront selon les attributs nouvellement définis. Les caractères existants à l'écran ne sont pas affectés.

**Paramètres**

- foreground est une expression numérique dans [0—31]. Cela précise le nouveau attribut de premier plan. Les attributs 16 à 31 sont des versions clignotantes des attributs 0 à 15.
- background est une expression numérique dans [0—15]. Cela précise le nouveau attribut de fond. Il est pris MOD 8 : Les valeurs 8—15 produisent la même couleur comme 0—7.
- border est une expression numérique dans [0—15] spécifiant l'attribut de bord.

**Attributs du mode texte (couleur)**

		Background attribute							
		0	1	2	3	4	5	6	7
FG	0	00	10	20	30	40	50	60	70
	1	01	11	21	31	41	51	61	71
	2	02	12	22	32	42	52	62	72
	3	03	13	23	33	43	53	63	73
	4	04	14	24	34	44	54	64	74
	5	05	15	25	35	45	55	65	75
	6	06	16	26	36	46	56	66	76
	7	07	17	27	37	47	57	67	77
	8	08	18	28	38	48	58	68	78
	9	09	19	29	39	49	59	69	79
	10	0a	1a	2a	3a	4a	5a	6a	7a
	11	0b	1b	2b	3b	4b	5b	6b	7b
	12	0c	1c	2c	3c	4c	5c	6c	7c
	13	0d	1d	2d	3d	4d	5d	6d	7d
	14	0e	1e	2e	3e	4e	5e	6e	7e
	15	0f	1f	2f	3f	4f	5f	6f	7f
	16	80	90	a0	b0	c0	d0	e0	f0
	17	81	91	a1	b1	c1	d1	e1	f1
	18	82	92	a2	b2	c2	d2	e2	f2
	19	83	93	a3	b3	c3	d3	e3	f3
	20	84	94	a4	b4	c4	d4	e4	f4
	21	85	95	a5	b5	c5	d5	e5	f5
	22	86	96	a6	b6	c6	d6	e6	f6
	23	87	97	a7	b7	c7	d7	e7	f7
	24	88	98	a8	b8	c8	d8	e8	f8
	25	89	99	a9	b9	c9	d9	e9	f9
	26	8a	9a	aa	ba	ca	da	ea	fa
	27	8b	9b	ab	bb	cb	db	eb	fb
	28	8c	9c	ac	bc	cc	dc	ec	fc
	29	8d	9d	ad	bd	cd	dd	ed	fd
	30	8e	9e	ae	be	ce	de	ee	fe
	31	8f	9f	af	bf	cf	df	ef	ff

Attributs du mode texte (monochrome)

		Background attribute															
		0	1	2	3	4	5	6	7								
FG	0	00	██	10	XX	20	XX	30	XX	40	XX	50	XX	60	XX	70	XX
	1	01	XX	11	XX	21	XX	31	XX	41	XX	51	XX	61	XX	71	XX
	2	02	XX	12	XX	22	XX	32	XX	42	XX	52	XX	62	XX	72	XX
	3	03	XX	13	XX	23	XX	33	XX	43	XX	53	XX	63	XX	73	XX
	4	04	XX	14	XX	24	XX	34	XX	44	XX	54	XX	64	XX	74	XX
	5	05	XX	15	XX	25	XX	35	XX	45	XX	55	XX	65	XX	75	XX
	6	06	XX	16	XX	26	XX	36	XX	46	XX	56	XX	66	XX	76	XX
	7	07	XX	17	XX	27	XX	37	XX	47	XX	57	XX	67	XX	77	XX
	8	08	██	18	XX	28	XX	38	XX	48	XX	58	XX	68	XX	78	XX
	9	09	XX	19	XX	29	XX	39	XX	49	XX	59	XX	69	XX	79	XX
	10	0a	XX	1a	XX	2a	XX	3a	XX	4a	XX	5a	XX	6a	XX	7a	XX
	11	0b	XX	1b	XX	2b	XX	3b	XX	4b	XX	5b	XX	6b	XX	7b	XX
	12	0c	XX	1c	XX	2c	XX	3c	XX	4c	XX	5c	XX	6c	XX	7c	XX
	13	0d	XX	1d	XX	2d	XX	3d	XX	4d	XX	5d	XX	6d	XX	7d	XX
	14	0e	XX	1e	XX	2e	XX	3e	XX	4e	XX	5e	XX	6e	XX	7e	XX
	15	0f	XX	1f	XX	2f	XX	3f	XX	4f	XX	5f	XX	6f	XX	7f	XX
	16	80	██	90	X	a0	X	b0	X	c0	X	d0	X	e0	X	f0	X
	17	81	X	91	X	a1	X	b1	X	c1	X	d1	X	e1	X	f1	X
	18	82	X	92	X	a2	X	b2	X	c2	X	d2	X	e2	X	f2	X
	19	83	X	93	X	a3	X	b3	X	c3	X	d3	X	e3	X	f3	X
	20	84	X	94	X	a4	X	b4	X	c4	X	d4	X	e4	X	f4	X
	21	85	X	95	X	a5	X	b5	X	c5	X	d5	X	e5	X	f5	X
	22	86	X	96	X	a6	X	b6	X	c6	X	d6	X	e6	X	f6	X
	23	87	X	97	X	a7	X	b7	X	c7	X	d7	X	e7	X	f7	X
	24	88	██	98	X	a8	X	b8	X	c8	X	d8	X	e8	X	f8	X
	25	89	X	99	X	a9	X	b9	X	c9	X	d9	X	e9	X	f9	X
	26	8a	X	9a	X	aa	X	ba	X	ca	X	da	X	ea	X	fa	X
	27	8b	X	9b	X	ab	X	bb	X	cb	X	db	X	eb	X	fb	X
	28	8c	X	9c	X	ac	X	bc	X	cc	X	dc	X	ec	X	fc	X
	29	8d	X	9d	X	ad	X	bd	X	cd	X	dd	X	ed	X	fd	X
	30	8e	X	9e	X	ae	X	be	X	ce	X	de	X	ee	X	fe	X
	31	8f	X	9f	X	af	X	bf	X	cf	X	df	X	ef	X	ff	X

### Remarques

- La syntaxe et l'effet de COULEUR sont différents dans les différents modes ÉCRAN : COULEUR (mode texte), COULEUR (ÉCRAN 1), (ÉCRAN 3—9).
- Au moins un paramètre doit être fourni et l'instruction ne doit pas se terminer par une virgule.

## Les erreurs

- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.
- Aucun des paramètres n'est dans [-32768—32767] : Overflow.
- le premier plan n'est pas dans [0—31], l'arrière-plan n'est pas dans [0—15] ou la bordure n'est pas dans [0—15] : Illegal function call.
- L'instruction est utilisée dans SCREEN 2 : Illegal function call.

### COLOR (SCREEN 1)

COLOR [palette\_0] [, palette [, override]]

Attribue de nouvelles couleurs à la palette d'attributs.

- palette\_0 est une expression numérique dans [0—255]. Cela définit la couleur de la palette associée à l'attribut 0 ; par défaut, le fond a cet attribut. Tous les pixels avec cet attribut changeront de couleur.

La valeur de la couleur de la palette est tirée de la l'ensemble de 64 couleurs. palette\_0 est prise MOD 64.














- palette est une expression numérique dans [0—255] qui spécifie la palette :
  - palette impair définit la palette CGA standard (cyan, magenta, gris).
  - palette définit même la palette alternative (vert, rouge, marron).

Tous les pixels avec les attributs 1,2,3 changeront de couleur vers la nouvelle palette.

- override est une expression numérique dans [0—255].

Si override est spécifié, palette est défini comme ci-dessus mais en utilisant override au lieu de palette. palette est alors ignorée.

### Palettes CGA

Attribute	Palette 0			Palette 1			Alternate palette		
	Colour	Lo	Hi	Colour	Lo	Hi	Colour	Lo	Hi
0	Black			Black			Black		
1	Green			Cyan			Cyan		
2	Red			Magenta			Red		
3	Brown			White			White		

### Remarques

- La syntaxe et l'effet de COULEUR sont différents dans les différents modes ÉCRAN : COULEUR (mode texte), COULEUR (ÉCRAN 1), (ÉCRAN 3—9).
- Au moins un paramètre doit être fourni et l'instruction ne doit pas se terminer par une virgule.

## Les erreurs

- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.
- Aucun des paramètres n'est dans [-32768—32767] : Overflow.
- L'un des paramètres n'est pas dans [0—255] : Illegal function call.

## COLOR (SCREEN 3—9)

COLOR [foreground] [, palette\_0 [, dummy]]

Modifie l'attribut de premier plan actuel et la couleur de l'attribut 0.

















### Paramètres

- foreground est une expression numérique dans [1—15] Cela définit le nouvel attribut de premier plan. Cela s'applique uniquement aux nouveaux caractères imprimés ou pixels tracés.
- palette\_0 est une expression numérique dans [0—15] Cela définit la couleur associée à l'attribut 0 ; par défaut, le fond a cet attribut. Tous les pixels avec cet attribut changeront de couleur.










Dans SCREEN 7 et 8, la couleur palette\_0 est prise parmi les 8 premiers du set EGA 16 couleurs. palette\_0 est MOD 8. À L'ÉCRAN 9, la valeur de couleur est tirée du jeu de 64 couleurs.

- dummy est une expression numérique avec une valeur dans [0—255] La valeur de dummy est ignorée.

### Palette par défaut EGA

Attribute	Colour	
0	Black	
1	Blue	
2	Green	
3	Cyan	
4	Red	
5	Magenta	
6	Brown	
7	Low-intensity white	
8	Grey	
9	Light Blue	
10	Light Green	
11	Light Cyan	
12	Light Red	
13	Light Magenta	
14	Light Yellow	
15	High-intensity white	

## Liste des couleurs EGA

0		8		16		24		32		40		48		56	
1		9		17		25		33		41		49		57	
2		10		18		26		34		42		50		58	
3		11		19		27		35		43		51		59	
4		12		20		28		36		44		52		60	
5		13		21		29		37		45		53		61	
6		14		22		30		38		46		54		62	
7		15		23		31		39		47		55		63	

## Remarques

- La syntaxe et l'effet de COULEUR sont différents dans les différents modes ÉCRAN : COULEUR (mode texte), COULEUR (ÉCRAN 1), (ÉCRAN 3—9).
- Au moins un paramètre doit être fourni et l'instruction ne doit pas se terminer par une virgule.

## Les erreurs

- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.
- Aucun des paramètres n'est dans [-32768—32767] : Overflow.
- le premier plan n'est pas dans [1—15] ; l'arrière-plan n'est pas dans [0—15] ; ou dummy n'est pas dans [0—255] : Appel de fonction illégal.

## COM

COM(port) {ON|OFF|STOP}

- ON : active l'interception d'événements ON COM(port) du port série émulé.
- OFF : désactive le piégeage.
- STOP : arrête le trapping jusqu'à ce que COM(port) ON soit utilisé.

Les événements qui se produisent alors que le piégeage est arrêté se déclenchent immédiatement lorsque le piégeage est réactivé.

## Paramètres

- port est une expression numérique avec une valeur de 1 ou 2. Cela spécifie quel port série ( COM1: ou COM2: ) est piégé. Si port est égal à 0, cette instruction ne fait rien.

## Les erreurs

- port une valeur de chaîne : Type mismatch.

- le port n'est pas dans [-32768—32767] : Overflow.
- le port n'est pas dans [0—3] : Illegal function call.

COMMON

COMMON [var\_0 [( [index\_0] )] [, [var\_1 [( [index\_1] )] ] ]...]

Spécifie les variables à transmettre en tant que variables communes au programme appelé avec CHAIN .

#### Paramètres

- var\_0, var\_1,... sont des noms de variables scalaires ou tableaux.
- index\_0, index\_1,... sont des nombres littéraux facultatifs ; ils sont ignorés.

#### Remarques

- Les éléments de tableau avec des crochets et un index ne provoquent pas d'erreur, mais sont ignorés.
- Les instructions COMMON ne sont pas exécutées au moment de l'exécution ; au lieu de cela, lorsqu'une commande CHAIN est rencontrée où ALL n'est pas spécifié, toutes les déclarations COMMON du programme sont analysées. En conséquence, le DEFSTR, DEFINT, DEFSNG ou les paramètres DEFDBL utilisés sont ceux qui sont actifs au moment de l'exécution de l'instruction CHAIN.
- Les déclarations COMMON n'ont pas besoin d'être accessibles dans le déroulement du programme pour être utilisées. Ils peuvent apparaître n'importe où avant ou après l'instruction CHAIN qui les utilise.
- Les variables peuvent être répétées ou apparaître dans plusieurs déclarations COMMON.
- Si le mot clé COMMON n'est pas le premier élément de l'instruction, la déclaration sera ignorée.

En particulier, toute déclaration COMMON qui se produit directement après un mot clé THEN ou ELSE ne sera pas utilisée.

COMMON dans la deuxième instruction ou les instructions suivantes d'une instruction composée après THEN ou ELSE sera utilisé quelle que soit la valeur de la condition IF.

<b>CONT</b>	CONT [anything]	
-------------	-----------------	--

Reprend l'exécution d'un programme qui a été interrompu par STOP, END, Ctrl + C ou Ctrl + Pause.

S'il est exécuté à partir d'un programme, arrête l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.

Remarques

- Tout ce qui suit le mot-clé CONT est ignoré.
- Cette instruction ne peut être utilisée qu'en mode direct.
- Si une interruption est rencontrée dans la routine GOSUB appelée à partir d'une ligne directe continue (par ex. GOSUB 100:PRINT A\$ ), CONT écrasera la ligne directe en cours d'exécution.

Comme le sous-programme RETURN est placé après le GOSUB dans l'ancienne ligne directe, des choses étranges peuvent se produire si des commandes sont données après CONT.

Dans GW-BASIC, cela peut entraîner des erreurs étranges dans des lignes de programme inexistantes car l'analyseur exécute des octets qui ne font pas partie d'une ligne de programme.

En PC-BASIC, si la nouvelle ligne directe est plus courte, l'exécution s'arrête après RETURN ; mais si la ligne directe est prolongée au-delà de l'ancienne position de retour, l'analyseur essaie de reprendre à cette position de retour, avec des effets étranges.

### **Les erreurs**

- Aucun programme n'est chargé, un programme n'a pas été exécuté, après modification d'une ligne de programme ou après CLEAR : Can't continue.
- La rupture s'est produite en ligne directe : Can't continue.
- CONT est utilisé dans un programme : Can't continue.



<b>DATA</b>	DATA [const_0] [, [const_1]]...	Spécifie les données qui peuvent être lues par une instruction READ.
-------------	---------------------------------	--

#### Paramètres

- const\_0, const\_1,... sont des chaînes et des nombres littéraux ou peuvent être vides. Les chaînes de caractères et les littéraux peuvent être donnés avec ou sans guillemets.

Si les guillemets sont omis, les espaces de début et de fin sont ignorés et les virgules ou les deux-points doivent terminer l'instruction de données.

#### Remarques

- Les déclarations DATA n'ont pas besoin d'être accessibles dans le déroulement du programme pour être utilisées.

Ils peuvent apparaître n'importe où avant ou après l'instruction READ qui les utilise.

- Si le mot clé DATA n'est pas le premier élément de l'instruction, la déclaration sera ignorée. En particulier, toute déclaration DATA qui se produit directement après un mot-clé THEN ou ELSE ne sera pas utilisée.

DATA dans la deuxième instruction ou les instructions suivantes d'une instruction composée après THEN ou ELSE sera utilisée quelle que soit la valeur de la condition IF.

#### Les erreurs

- Si le type du littéral ne correspond pas à celui de l'instruction READ correspondante, Syntax error se produit sur l'instruction DATA.

<b>DATE\$ (statement)</b>	DATE\$ = date	
---------------------------	---------------	--

Définit la date système.

Date est une expression de chaîne qui représente une date dans l'un des formats suivants :

"mm{-/}jj{-/}aa" ou "mm{-/}jj{-/}aaaa".

Parmi ceux-ci,

- mm peut comporter un ou deux caractères et doit être entre [1—12].
- dd peut comporter un ou deux caractères et doit figurer dans [1—31].
- yyyy doit être en [1980—2099].
- yy peut comporter un ou deux caractères et doit se trouver dans l'une des plages suivantes :
  - [0—77], interprété comme 2000—2077 ; ou
  - [80—99], interprété comme 1980—1999.

## Remarques

- La date système n'est pas réellement modifiée ; à la place, PC-BASIC mémorise le décalage par rapport à la véritable date système. Cela évite de demander l'autorisation de l'utilisateur pour modifier l'heure du système.
- GW-BASIC semble accepter des dates non valides telles que "02-31-2000". PC-BASIC déclenche un Illegal function call pour ceux-ci.

## les erreurs

- date a une valeur numérique : Type mismatch.
- la date n'est pas au format spécifié ci-dessus : Illegal function call.

## DEF FN

DEF FN[ ]name [(arg\_0 [, arg\_1]...)] = expression

Définit une fonction appelée FNname (ou FN name : les espaces entre FN et name sont facultatifs). Lors de l'appel de FNname(...), l'expression est évaluée avec les paramètres fournis substitués.

Tous les noms de variables utilisés dans la fonction qui ne figurent pas dans la liste des arguments font référence aux variables globales correspondantes. Le résultat de l'évaluation est la valeur de retour de FNname. Le type de la valeur de retour doit être compatible avec le type indiqué par name.

## Remarques

- Cette instruction ne peut être utilisée que sur une ligne de programme.
- Comme la fonction doit être une expression unique et que PC-BASIC n'a pas d'opérateur ternaire, il n'y a aucun moyen de définir une fonction récursive qui se termine réellement.

## Paramètres

- name doit être un nom de variable légal.
- arg\_0, arg\_1,... doivent être des noms de variables légaux. Ce sont les paramètres de la fonction. Les variables du même nom peuvent ou non exister dans le programme ; leur valeur n'est pas affectée ou utilisée par la fonction définie.
- expression doit être une expression PC-BASIC légale.

## Les erreurs

- L'instruction est exécutée directement et non dans une ligne de programme : Illegal direct.

- Si le type de la valeur de retour est incompatible avec le type de name, aucune erreur n'est générée au niveau de l'instruction DEF FN ; cependant, une incompatibilité de type sera déclenchée au premier appel de FNname.

DEFINT, DEFDBL, DEFSNG, DEFSTR

{DEFINT|DEFDBL|DEFSNG|DEFSTR}  
first\_0[- last\_0] [, first\_1[- last\_1]]...

Définit le type supposé si aucun sceau n'est spécifié lorsqu'un nom de variable est utilisé. L'instruction définit le type par défaut pour les variables commençant par une lettre parmi les plages spécifiées.

Le type par défaut est défini sur :

DEFINT entier ( % )

DEFDBL double ( # )

DEFSNG single ( ! )

DEFSTR Chaîne ( \$ )

Paramètres

- first\_0, last\_0,... sont des lettres de l'alphabet. Paires de lettres reliées par un tiret - indique des plages inclusives.

Remarques

- DEFSNG A-Z est le réglage par défaut.

**DEF SEG**

DEF SEG [= address]

Définit le segment de mémoire à utiliser par BLOAD, BSAVE, ALL, PEEK, POKE et USR.

Paramètres

- adresse est une expression numérique dans [-32768—65535].

Remarques

- Si l'adresse est négative, elle est interprétée comme son complément à deux.
- Si l'adresse n'est pas spécifiée, le segment est défini sur le segment de données GW-BASIC.

**Les erreurs**

- address a une valeur de chaîne : Type mismatch.
- adresse n'est pas dans [-32768—65535] : Overflow.

<b>DEF USR</b>	DEF USR[n] = address	Ne fait rien.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• n est un chiffre compris entre 0 et 9 inclus.</li> <li>• adresse est une expression numérique dans [-32768—65535].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, cette instruction définit l'adresse de début d'une fonction en langage assembleur.</li> <li>• Cette instruction n'est pas implémentée dans PC-BASIC.</li> <li>• Si l'adresse est négative, elle est interprétée comme son complément à deux.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• n n'est pas un chiffre dans [0—9] : Syntax error.</li> <li>• address a une valeur de chaîne : Type mismatch.</li> <li>• l'adresse n'est pas dans [-32768—65535] : Overflow.</li> </ul>		
<b>DELETE</b>	DELETE [line_number_0 .] [-[line_number_1 .] ]	
<p>Supprime une plage de lignes du programme. Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• line_number_0 et line_number_1 sont des numéros de ligne dans la plage [0—65529], spécifiant la plage inclusive de numéros de ligne à supprimer.</li> <li>• A. indique la dernière ligne éditée.</li> <li>• Si le point de départ est omis, la plage commencera au début du programme.</li> <li>• Si le point final est omis, la plage se terminera à la fin du programme.</li> <li>• Si aucune plage n'est spécifiée, tout le programme sera supprimé.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• line_number_0 ou line_number_1 est supérieur à 65529 : Syntax error.</li> <li>• La plage spécifiée n'inclut aucune ligne de programme stockée : Illegal function call.</li> </ul>		

**DIM**

DIM name [{" limit\_0 [, limit\_1]... }]]} [,... ]

Alloue de la mémoire pour une ou plusieurs taies.

L'instruction DIM fixe également le nombre d'indices du tableau. Un tableau ne peut être alloué qu'une seule fois ; pour réallouer un tableau, ERASE ou CLEAR doivent d'abord être exécutés.

Si un tableau est utilisé pour la première fois sans instruction DIM, il est automatiquement alloué avec ses indices maximaux fixés à 10 pour chaque position d'index utilisée.

Une entrée DIM sans crochets ni index n'effectue aucune opération. Les parenthèses vides ne sont pas autorisées. Le plus petit index autorisé est déterminé par OPTION BASE.

**Paramètres**

- name,... sont des noms de variable légaux spécifiant les tableaux à allouer.
- limit\_0, limit\_1,... sont des expressions numériques qui spécifient le plus grand index autorisé à cette position.

**Remarques**

- Les parenthèses mixtes sont autorisées.
- La taille des matrices est limitée par la mémoire BASIC disponible.
- Le nombre maximum d'indices est théoriquement de 255. En pratique, il est limité par la limite de 255 octets sur la longueur des lignes de programme.

**Les erreurs**

- le name a déjà été coté : Duplicate definition.
- Un index est vide : Syntax error.
- Un index manque à la fin : Missing operand.
- limit\_0, limit\_1,... ont une valeur de chaîne : Type mismatch.
- limit\_0, limit\_1,... ne sont pas compris entre [-32768—32767] : Overflow.
- limit\_0, limit\_1,... sont négatifs : Illegal function call.
- Le tableau dépasse la taille de l'espace variable disponible : Out of memory.

**DRAW****DRAW gml\_string**

Dessine la forme spécifiée par gml\_string, une expression de chaîne dans Graphics Macro Language (GML).

Référence du langage de macro graphique

Commandes de mouvement

Mouvement [B][N]

où la valeur par défaut est de se déplacer et de dessiner ; les préfixes facultatifs signifient :

B se déplace mais ne trace pas

N retour au point d'origine après le déplacement

et le mouvement est l'un des suivants :

U[n] monte de n

L[n] gauche de n

D[n] descend de n

R[n] droite de n

E[n] vers le haut et à droite de n

F[n] vers le bas et vers la droite de n

G[n] vers le bas et vers la gauche de n

H[n] vers le haut et vers la gauche de n

M{+|-}x,[+|-]y déplace de (x, y)

Mx,y déplace pour afficher les coordonnées de la région (x, y)

où n est un entier dans [-32768—32767] et x, y sont des entiers dans [0—9999].

Lorsqu'il est facultatif, n prend par défaut la valeur 1.

Commandes d'échelle

Sn définit la taille du pas sur n/4. La taille de pas par défaut est de 1 pixel. n est un entier dans [1—255]

TAn réglez l'angle sur n degrés. L'angle par défaut est de 0 degré. n est un entier dans [-360—360]

An définit l'angle sur 0 pour n=0, 90 pour n=1, 180 pour n=2, 270 pour n=3. n est un entier dans [0—3]

Commandes de couleur

Cn définit l'attribut de premier plan sur n, où n est un entier dans [-32768—32767] Voir COLOR.

Pn,b remplissage inondé avec l'attribut n et l'attribut de limite b, où n, b sont des entiers dans [0—9999]  
Voir PAINT.

Commande de sous-programme

Xs exécutent une sous-chaîne

s est l'un des suivants :

- un nom de variable de chaîne suivi d'un point-virgule ( ; )
- le résultat de VARPTR\$() sur une variable de chaîne

Les variables numériques n, x, y, b dans les commandes ci-dessus peuvent être :

- un littéral entier, par ex. DRAW "U100"
- un nom de variable numérique ou un élément de tableau var précédé de = et suivi de ;  
Par exemple, DRAW "U=VAR ;" ou DESSINER "U=A(1);"
- le résultat de VARPTR\$(var) précédé de =  
Par exemple, DESSINEZ "U=" + VARPTR\$(VAR)

Remarques

- L'instruction CLS réinitialise la taille du pas à 1 pixel, l'angle à 0 degré et la position au centre de la zone de visualisation.
- La valeur n dans les commandes TA, A et C peut être omise mais uniquement si la commande se termine par un point-virgule. n est par défaut à 0.
- Dans GW-BASIC, les arguments numériques de U, L, D, R, E, F, G, H et C peuvent être dans la plage [-

99999—99999] ; cependant, les résultats pour de grands nombres sont imprévisibles. Ceci n'est pas implémenté dans PC-BASIC.

### Les erreurs

- gml\_string a une valeur numérique : Type mismatch.
- gml\_string contient des erreurs dans le GML : Illegal function call.
- Une variable référencée dans la chaîne GML est de type incorrect : Type mismatch.

<b>EDIT</b>	EDIT {line_number .}	
-------------	----------------------	--

Affiche la ligne de programme spécifiée avec le curseur positionné pour l'édition.

line\_number doit être une ligne qui existe dans le programme, ou un point (.) pour indiquer la dernière ligne stockée.

Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.

### Les erreurs

- Aucun numéro de ligne n'est spécifié : Undefined line number.
- D'autres caractères sont écrits après le numéro de ligne : Illegal function call.
- line\_number n'est pas dans [0—65529] : Illegal function call.
- Le numéro de ligne spécifié n'existe pas : Undefined line number.

<b>ELSE</b>	ELSE [anything]	
-------------	-----------------	--

À moins qu'il ne fasse partie d'une instruction IF sur la même ligne, tout ce qui se trouve après ELSE est ignoré de la même manière qu'après ' ou :REM.

Pas de deux-points : avant l'instruction ELSE est nécessaire.

Voir IF pour une utilisation normale.

<b>END</b>	END	
------------	-----	--

Ferme tous les fichiers, arrête l'exécution du programme et rend le contrôle à l'utilisateur. Aucun message n'est imprimé. Il est possible de reprendre l'exécution à l'instruction suivante en utilisant CONT.



<b>ENVIRON</b>	ENVIRON command_string	Définit une variable d'environnement shell.
<p>Paramètres</p> <p>command_string est une expression de chaîne de l'une des formes suivantes :</p> <p>"VARIABLE=VALEUR" pour définir VARIABLE sur VALUE ;</p> <p>"VARIABLE="</p> <p>pour désactiver VARIABLE.</p> <p>VARIABLE doit être une chaîne ASCII et sera convertie en majuscules sur les systèmes sensibles à la casse.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• command_string a une valeur numérique : Type mismatch.</li> <li>• command_string n'est pas au format requis : Illegal function call.</li> <li>• VARIABLE contient des caractères hors ASCII : Illegal function call.</li> </ul>		
<b>ERASE</b>	ERASE array_0 [, array_1]...	
<p>Désalloue les tableaux. Les données stockées dans les tableaux sont perdues.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• array_0, array_1... sont des noms de tableaux existants. Les noms doivent être spécifiés sans crochets.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Aucun nom de tableau n'est donné :Syntax error.</li> <li>• array_0, array_1... n'existent pas : Illegal function call.</li> <li>• Si une erreur se produit, les tableaux nommés avant que l'erreur ne se produise sont effacés.</li> </ul>		
<b>ERROR</b>	ERROR error_number	
<p>Déclenche l'erreur avec le numéro error_number. Voir la liste des numéros d'erreur et des messages à la fin de cette section.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• error_number est une expression avec une valeur numérique.</li> </ul> <p>les erreurs</p>		

- error\_number a une valeur de chaîne : Type mismatch.
- error\_number n'est pas dans [-32768—32767] : Overflow.
- error\_number n'est pas dans [1—255] : Illegal function call.

## FIELD

FIELD [#] file\_number [, width\_0 AS name\_0 [, width\_1 AS name\_1]...]

Affecte des variables au tampon d'enregistrement à accès aléatoire. Le tampon d'enregistrement est une région de mémoire de longueur définie par l'instruction OPEN ; la longueur d'enregistrement par défaut est de 128 octets.

L'instruction FIELD affecte une partie de cette région à une ou plusieurs variables de chaîne de longueur fixe, de sorte que la valeur de ces chaînes soit celle qui se trouve dans le tampon d'enregistrement à cet emplacement.

### Remarques

- Une instruction FIELD sans aucune variable spécifiée n'a aucun effet.
- Une autre instruction FIELD sur le même fichier spécifiera un mappage alternatif du même tampon de fichier ; tous les mappages seront en vigueur simultanément.
- Une affectation ultérieure ou une instruction LET ou MID\$ sur name\_0, name\_1... dissociera la variable de chaîne du tampon de champ.
- Utilisez LSET, RSET ou MID\$ pour copier les valeurs dans un tampon FIELD.
- Utilisez GET pour lire les valeurs du fichier dans le tampon de champ, en modifiant les variables.
- Utilisez PUT pour écrire le tampon de champ dans le fichier.

### Paramètres

- numéro\_fichier est une expression numérique qui donne le numéro d'un fichier à accès aléatoire ouvert. Le # est facultatif et n'a aucun effet.
- width\_0, width\_1,... sont des expressions numériques donnant la longueur des variables de chaîne
- name\_0, name\_1... sont des variables de chaîne.

### Les erreurs

- file\_number n'est pas dans [0—255] : Illegal function call.

- file\_number n'est pas le numéro d'un fichier ouvert : Bad file number.
- file\_number est ouvert sous un mode autre que RANDOM : Bad file mode.
- L'instruction se termine par une virgule : Missing operand.
- Aucun numéro de fichier n'est spécifié : Missing operand.
- Les longueurs d'une instruction FIELD s'ajoutent à un nombre supérieur à la longueur d'enregistrement du tampon de champ : Field overflow.
- name\_0, name\_1... spécifiez une variable non-chaîne : Type mismatch.

## FILES

FILES [filter\_spec]

Affiche les fichiers correspondant au filtre spécifié dans le répertoire spécifié sur un périphérique de disque. Si filter\_spec n'est pas spécifié, affiche tous les fichiers du répertoire de travail actuel.

### Paramètres

- filter\_spec est une expression de chaîne qui ressemble beaucoup à une spécification de fichier, mais qui permet éventuellement à la partie du nom de fichier de contenir des caractères génériques.

### Remarques

- Le filtre de nom de fichier peut contenir les caractères génériques suivants :
  - ? Correspond à n'importe quel caractère de nom de fichier légal.
  - \* Correspond à n'importe quelle série de caractères de nom de fichier légal.
- Le filtre ne correspondra qu'aux noms de fichiers de style MS-DOS.
- Les séries de caractères correspondantes ne s'étendent pas sur les séparateurs de répertoire \ ou les séparateurs d'extension. Pour faire correspondre tous les fichiers avec toutes les extensions, utilisez \*.\*.
- Sinon, si tous les fichiers d'un répertoire spécifié sont requis, terminez le nom du répertoire par une barre oblique inverse \.

### Les erreurs

- filter\_spec a une valeur numérique : Type mismatch.
- filter\_spec est la chaîne vide : Bad file name.
- Le filtre spécifié ne correspond à aucun fichier : File not found.

<b>FOR</b>	FOR loop_var = start TO stop [STEP step]
<p>Initie une boucle FOR—NEXT.</p> <p>Initialement, loop_var est défini sur start. Ensuite, les instructions entre l'instruction FOR et l'instruction NEXT sont exécutées et loop_var est incrémenté de step (si step n'est pas spécifié, de 1).</p> <p>Ceci est répété jusqu'à ce que loop_var soit devenu supérieur à stop. L'exécution continue ensuite à l'instruction suivant NEXT. La valeur de loop_var est égale à stop+step après la boucle.</p> <p>start, stop et step ne sont évalués qu'une seule fois et les valeurs résultantes sont utilisées tout au long de la boucle.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• loop_var est un nombre entier ou une variable simple précision.</li> <li>• start, stop et step sont des expressions numériques.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Aucune instruction NEXT ne correspond à l'instruction FOR : FOR without NEXT se produit à l'instruction FOR.</li> <li>• loop_var est une variable chaîne ou start, stop ou end a une valeur chaîne : Type mismatch.</li> <li>• loop_var est une variable à double précision : Type mismatch.</li> <li>• loop_var est un élément de tableau : Syntax error.</li> <li>• loop_var est une variable entière et un start, stop ou step est en dehors de la plage [-32768, 32767] : Overflow.</li> </ul>	
<b>GET (files)</b>	GET [#] file_number [, record_number]
<p>Lit un enregistrement du fichier à accès aléatoire numéro_fichier à la position numéro_enregistrement. L'enregistrement est accessible via les variables FIELD ou via INPUT\$, INPUT ou LINE INPUT.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• numéro_fichier est une expression numérique qui donne le numéro d'un fichier à accès aléatoire ouvert. Le # est facultatif et n'a aucun effet.</li> <li>• numéro_enregistrement est une expression numérique dans [1—33554432] ( <math>2^{25}</math> ) et est interprété comme le numéro d'enregistrement.</li> </ul> <p>Remarques</p>	

- Si le numéro d'enregistrement est au-delà de la fin du fichier, le tampon du fichier est rempli d'octets nuls.
- Le numéro d'enregistrement est stocké en simple précision ; cette précision n'est pas assez élevée pour distinguer des enregistrements uniques proches de la valeur maximale de  $2^{25}$ .

### Les erreurs

- numéro\_enregistrement n'est pas dans [1—33554432] : Bad record number.
- file\_number n'est pas dans [0—255] : Illegal function call.
- file\_number n'est pas le numéro d'un fichier ouvert : Bad file mode.
- file\_number est ouvert sous un mode autre que RANDOM : Bad file mode.
- numéro\_fichier n'est pas spécifié : Missing operand.

**GET** (communications)

GET [#] com\_file\_number [,  
number\_bytes]

Lit number\_bytes octets du tampon de communication ouvert sous le numéro de fichier com\_file\_number. L'enregistrement est accessible via les variables FIELD ou via INPUT\$, INPUT ou LINE INPUT.

### Paramètres

- numéro\_fichier est une expression numérique qui donne le numéro d'un fichier ouvert sur un périphérique COM. Le # est facultatif et n'a aucun effet.
- number\_bytes est une expression numérique comprise entre 1 et la longueur du tampon COM, inclus.

### Remarques

- Si octets est égal ou supérieur à 32 768, GW-BASIC se bloque. Cette fonctionnalité n'est pas implémentée dans PC-BASIC.
- Dans GW-BASIC, une erreur d'E/S de périphérique est déclenchée pour une erreur de dépassement, une erreur de trame et une interruption d'interruption.

Le périphérique par défaut est déclenché si le DSR est perdu pendant les E/S. Une erreur de parité est déclenchée si la parité est activée et qu'une parité incorrecte est rencontrée. C'est selon le manuel; il n'est pas testé.

### Les erreurs

- octets est inférieur à 1 : Bad record number.
- octets est inférieur à 32768 et supérieur à la longueur du tampon COM : illégal function call.
- com\_file\_number n'est pas spécifié : Missing operand.

- `com_file_number` n'est pas dans [0—255] : illégal function call.
- `com_file_number` n'est pas le numéro d'un fichier ouvert : Bad file number.
- Si le tampon d'entrée série est plein, c'est-à-dire  $LOF(\text{com\_file\_number}) = 0$ , et  $LOC(\text{com\_file\_number}) = 255$  : Communication buffer overflow.
- Si la porteuse tombe pendant GET, se bloque jusqu'à ce que la touche Ctrl + Break soit enfoncée.

### GET (graphics)

GET (x0, y0) - [STEP] (x1, y1),  
array\_name

Stocke une zone rectangulaire de l'écran graphique dans un tableau. La zone stockée est un rectangle parallèle aux bords de l'écran, délimité par les coordonnées en haut à gauche et en bas à droite x0, y0 et x1, y1.

Si STEP est spécifié, x1, y1 est un décalage par rapport à x0, y0. La zone est telle que ces points d'angle sont à l'intérieur.

L'image stockée dans le tableau peut ensuite être affichée à l'écran à l'aide de PUT.

Pour les besoins de GET, tout tableau est considéré comme une chaîne d'octets. La taille en octets d'un tableau peut être calculée comme  $\text{number\_elements} * \text{byte\_size}$  avec `byte_size` égal à 2 pour les entiers ( % ), 4 pour simple ( ! ) et 8 pour double ( # ).

La taille en octets du tableau pour la chaîne est de 3, mais les tableaux de chaînes ne sont pas autorisés dans GET.

Pour calculer le nombre d'éléments, gardez à l'esprit que OPTION BASE 0 est la valeur par défaut ; auquel cas un tableau avec un indice maximum de 10 a 11 éléments. Cela fonctionne dans les tableaux multidimensionnels.

Le format du tableau est le suivant :

L'octet contient

- 0, 1 Nombre de x pixels, entier non signé. Dans SCREEN 1, cette valeur est doublée.
- 2, 3 Nombre de pixels y, entier non signé.
- 4— Données de pixels.

Les données sont organisées en mots de 2 octets. Le premier mot de 16 bits contient le bit 0 des 16 premiers pixels de la rangée supérieure. Le deuxième mot contient le deuxième bit, etc.

Les données sont alignées sur les mots à la fin de chaque ligne. Ainsi, dans un mode écran à 4 bits par pixel, la première ligne occupe au moins 8 octets (4 mots), même si elle n'est constituée que d'un seul pixel. Le nombre de bits par pixel dépend du mode SCREEN.

## Paramètres

- array\_name est le nom d'un tableau numérique dimensionné avec suffisamment d'espace pour stocker la zone.
- x0, y0, x1, y1 sont des expressions numériques.

## Remarques

- En mode PCjr/Tandy, dans SCREEN 6, GET stocke une zone de deux fois la largeur du rectangle spécifié.

## Les erreurs

- Le tableau n'existe pas : Illegal function call.
- array\_name fait référence à un tableau de chaînes : Type mismatch.
- La zone est trop grande pour le tableau : Illegal function call.
- x0,... y1 sont des expressions de chaîne : Type mismatch.
- x0,... y1 ne sont pas dans [-32768—32767] : Overflow.
- x0,... y1 sont en dehors de la VUE ou de la FENÊTRE courante : Illegal function call.

## GOSUB

GO[ ]SUB line\_number [anything]

Passes à une sous-routine à line\_number. L'instruction RETURN suivante revient à l'instruction après GOSUB. Tout ce qui se trouve après line\_number jusqu'à la fin de l'instruction est ignoré. S'il est exécuté à partir d'une ligne directe, GOSUB exécute le sous-programme et le RETURN suivant renvoie l'exécution à la ligne directe.

## Paramètres

- line\_number est un littéral de numéro de ligne existant.
- Les autres caractères de la ligne sont ignorés jusqu'à la fin de l'instruction.

## Remarques

- Si aucun RETURN n'est rencontré, pas de problème.
- Un espace facultatif est autorisé entre GO et SUB ; il ne sera pas retenu dans le programme.

## Les erreurs

- Si numéro\_ligne n'existe pas : Undefined line number.
- Si line\_number est supérieur à 65529, seuls les 4 premiers caractères sont lus (par exemple 6553 )

<b>GOTO</b>	GO[ ]TO line_number [anything]	
<p>Saute à line_number. Tout ce qui se trouve après line_number jusqu'à la fin de l'instruction est ignoré. S'il est exécuté à partir d'une ligne directe, GOTO lance l'exécution du programme à la ligne spécifiée.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• line_number est un littéral de numéro de ligne existant.</li> <li>• Les autres caractères de la ligne sont ignorés jusqu'à la fin de l'instruction.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• N'importe quel nombre d'espaces facultatifs est autorisé entre GO et TO, mais ils ne seront pas conservés dans le programme.</li> <li>• Si line_number est supérieur à 65529, seuls les 4 premiers caractères sont lus (par exemple, GOTO 65530 est exécuté comme GOTO 6553 )</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• line_number n'existe pas : Undefined line number.</li> </ul>		
<b>IF</b>	<p>IF truth_value [,] {THEN GOTO} [compound_statement_true line_number_true [anything]]</p> <p>[ELSE [compound_statement_false line_number_false [anything]]]</p>	
<p>If <b>truth_value</b> est différent de zéro, exécute <b>compound_statement_true</b> ou saute à <b>line_number_true</b>.</p> <p>Si c'est zéro, exécute <b>compound_statement_false</b> ou saute à <b>line_number_false</b>.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• truth_value est une expression numérique.</li> <li>• line_number_false et line_number_true sont des numéros de ligne existants.</li> <li>• compound_statement_false et compound_statement_true sont des instructions composées, constituées d'au moins une instruction, éventuellement suivie d'autres instructions séparées par des deux-points :</li> </ul> <p>Les instructions composées peuvent contenir des instructions IF—THEN—ELSE imbriquées.</p> <p>Remarques</p>		



- La virgule est facultative et ignorée.
- Les clauses ELSE sont facultatives ; ils sont liés à l'instruction IF libre la plus interne si elles sont imbriquées. Les clauses ELSE supplémentaires qui n'ont pas de IF correspondant sont ignorées.
- Toutes les clauses doivent être sur la même ligne de programme.
- THEN et GOTO sont interchangeables ; celui qui est choisi est indépendant du fait qu'une déclaration ou un numéro de ligne soit donné. GOTO PRINT 1 est correct.
- Comme dans GOTO, tout ce qui suit le numéro de ligne est ignoré.

les erreurs

- Si truth\_value a une valeur de chaîne : Type mismatch.
- truth\_value est égal à 0 et numéro\_ligne\_faux est un numéro de ligne inexistant, ou truth\_value est différent de zéro et line\_number\_true est un numéro de ligne inexistant : Undefined line number.

**INPUT (console)**

INPUT [;] [prompt {;,}] var\_0 [, var\_1]...

Affiche l'invite à l'écran et attend que l'utilisateur saisisse les valeurs des variables spécifiées.

Le point-virgule avant l'invite, s'il est présent, arrête l'impression d'une nouvelle ligne après la saisie des valeurs.

Si l'invite est suivie d'un point-virgule, elle est imprimée avec un ?. Si l'invite est suivie d'une virgule, aucun point d'interrogation n'est ajouté.

Paramètres

- prompt est un littéral de chaîne.
- var\_0, var\_1,... sont des noms de variables ou des éléments de tableau entièrement indexés.

Remarques

- Les valeurs saisies doivent être séparées par des virgules. Les espaces de début et de fin sont ignorés.
- Les valeurs de chaîne peuvent être saisies avec ou sans guillemets ( " ).
- Si une chaîne avec une virgule, un espace blanc de début ou de fin est nécessaire, les guillemets sont le seul

moyen de la saisir.

- Entre un guillemet fermant et la virgule à la fin de l'entrée, seul un espace blanc est autorisé.
- Si des guillemets sont nécessaires dans la chaîne elle-même, le premier caractère ne doit être ni un guillemet ni un espace. Il n'est pas possible d'entrer une chaîne qui commence par un guillemet via INPUT.
- Si un var\_n donné est une variable numérique, la valeur saisie doit être un littéral numérique.
- Les caractères au-delà du 255e caractère de la ligne d'écran sont ignorés.
- Si l'entrée de l'utilisateur est interrompue par Ctrl + Break, CONT réexécute l'instruction INPUT.

### Les erreurs

- Si la valeur saisie pour une variable numérique n'est pas un littéral numérique valide ou si le nombre de valeurs saisies ne correspond pas au nombre de variables dans l'instruction, ?Redo from start est imprimé et toutes les valeurs doivent être saisies à nouveau.
- Une erreur de syntaxe qui se produit après l'impression de l'invite n'est déclenchée qu'après la saisie de la valeur rasée. Aucune valeur n'est stockée.

<b>INPUT (files)</b>	INPUT # file_num, var_0 [, var_1]...
----------------------	--------------------------------------

Lit des variables de chaîne ou numériques à partir d'un fichier texte ou du tampon FIELD d'un accès aléatoire.

### Paramètres

- file\_num est le numéro d'un fichier ouvert en mode INPUT ou un fichier à accès aléatoire ouvert en mode ALÉATOIRE.
- var\_0, var\_1,... sont des noms de variables ou des éléments de tableau entièrement indexés.

### Remarques

- Le # est obligatoire. Il peut y avoir ou non des espaces entre INPUT et #.
- Les valeurs de chaîne peuvent être saisies avec ou sans guillemets ( " ).
- Les valeurs numériques se terminent par LF, CR,,.
- Les chaînes sans guillemets se terminent par LF, CR,,.
- Les chaînes entre guillemets se terminent par le guillemet fermant.
- Toute entrée se termine par le caractère EOF ou son 255<sup>ème</sup> caractère.
- Les espaces de début et de fin sont supprimés.

- Si l'entrée ne peut pas être convertie dans le type demandé, une valeur nulle est renvoyée.
- Si file\_num est ouvert sur KYBD :, INPUT# lit à partir du clavier jusqu'à ce qu'un retour ou une virgule soit rencontré (comme dans un fichier).

Les touches fléchées et la suppression sont transmises comme caractères de contrôle (pas de scancodes !) précédés de CHR\$(&hFF).

### Les erreurs

- L'entrée est demandée après que la fin d'un fichier texte a été atteinte ou qu'un caractère EOF a été rencontré : Input past end.
- Le dernier caractère du tampon de champ est lu : Field overflow.
- file\_num a une valeur de chaîne : Type mismatch.
- num\_fichier est supérieur à 32767 : Overflow.
- file\_num est inférieur à zéro : Illegal function call.
- num\_fichier n'est pas un fichier ouvert : Bad file number.
- num\_fichier n'est pas ouvert pour INPUT ou RANDOM : Bad file mode.
- file\_num est ouvert sur un port COM et il s'agit du premier appel INPUT, LINE INPUT ou INPUT\$ sur ce port depuis que le tampon s'est complètement rempli (c'est-à-dire que LOF(file\_num) est devenu zéro) : Communication buffer overflow.

### IOCTL

IOCTL [#] file\_num, control\_string

Appel de fonction illégal.

### Remarques

- Dans GW-BASIC, IOCTL envoie une chaîne de contrôle à un appareil.
- Cette instruction n'est pas implémentée dans PC-BASIC.

### Les erreurs

- file\_num a une valeur de chaîne : Type mismatch.
- num\_fichier n'est pas dans [-32768—32767] : Overflow.
- num\_fichier n'est pas un fichier ouvert : Bad file number.

- Sinon : Illegal function call

**KEY (macro list)**

KEY {ON|OFF|LIST}

Active ou désactive la liste des macros de touches de fonction en bas de l'écran.

Si LIST est spécifié, imprime une liste des 10 (ou 12 avec syntax=tandy ) touches de fonction avec les macros de touches de fonction définies pour ces touches sur la console.

La plupart des caractères sont représentés par leur symbole équivalent dans la page de code actuelle.

Cependant, certains caractères obtiennent une représentation différente, qui est une représentation symbolique de leur effet en tant que caractères de contrôle à l'écran.

Code point	Replacement	Usual glyph
&h07	&h0E	♪
&h08	&hFE	■
&h09	&h1A	→
&h0A	&h1B	←
&h0B	&h7F	△
&h0C	&h16	—
&h0D	&h1B	←
&h1C	&h10	▶
&h1D	&h11	◀
&h1E	&h18	↑
&h1F	&h19	↓

**KEY (macro definition)**

KEY key\_id, string\_value

Définit la macro de chaîne pour la touche de fonction key\_id.

Seuls les 15 premiers caractères de string\_value sont stockés.

Paramètres

- key\_id est une expression numérique dans la plage [1—10] (ou [1—12] lorsque syntax=tandy ).
- string\_value est une expression de chaîne.

Remarques

- Si key\_id n'est pas dans la plage prescrite, l'instruction est interprétée comme une instruction KEY de capture

d'événement.

- Si `string_value` est la chaîne vide ou si le premier caractère de `string_value` est `CHR$(0)`, la macro de touche de fonction est désactivée et la capture ultérieure de la touche de fonction associée avec `INKEY$` est activée.

### Les erreurs

- `key_id` n'est pas dans [-32768—32767] : Overflow.
- `key_id` n'est pas dans [1—255] : Illegal function call.
- `key_id` a une valeur de chaîne : Type mismatch.

<b>KEY (event switch)</b>	KEY (key_id) {ON OFF STOP}	
---------------------------	----------------------------	--

Contrôle l'interception d'évènements de la clé avec l'identifiant `key_id`.

Le piégeage d'évènement est activé ou désactivé. `STOP` suspend l'interception d'évènements jusqu'à ce qu'un `KEY() ON` soit exécuté.

Jusqu'à un évènement peut être déclenché pendant la suspension, à condition que la gestion des évènements ait été activée avant la suspension. L'évènement déclenché pendant la suspension est traité immédiatement après la prochaine instruction `KEY() ON`. L'interception d'évènements ne peut être active que pendant l'exécution d'un programme, elle ne fonctionne pas en mode direct.

### Paramètres

`key_id` est une expression numérique dans [1—20]. Les clés sont :

- 1 F1
- 2 F2
- 3 F3
- 4 F4
- 5 F5
- 6 F6
- 7 F7
- 8 F8
- 9 F9
- 10 F10
- 11 ↑
- 12 ←
- 13 →
- 14 ↓

Les clés 15 à 20 sont définies à l'aide de l'instruction de définition `KEY` de capture d'évènement.

## Remarques

- Avec syntax=tandy, la clé 11 est F11 et la clé 12 est F12. Les touches prédéfinies 11—14 passent à 13—16.

## Les erreurs

- key\_id n'est pas dans [-32768—32767] : Overflow.
- key\_id n'est pas dans [0—20] : Illegal function call.
- key\_id a une valeur de chaîne : Type mismatch.

<b>KEY(event definition)</b>	KEY key_id, two_char_string	Définit la clé à intercepter pour key_id.
------------------------------	-----------------------------	---

## Paramètres

- key\_id est une expression numérique dans [15—20] (ou [17—20] lorsque syntax=tandy ).
- two\_char\_string est une expression de chaîne de longueur 2. Le premier caractère est interprété comme un modificateur tandis que le deuxième caractère est interprété comme un scancode.

Le caractère modificateur est une combinaison OU au niveau du bit des indicateurs suivants :

CHR(&h80) Étendu (ignoré)  
CHR(&h40) Verrouillage des majuscules  
CHR(&h20) Verr Num  
CHR(&h10) non utilisé  
CHR(&h08) Autre  
CHR(&h04) Ctrl  
CHR(&h02) Décalage (de chaque côté)  
CHR(&h01) Décalage (de chaque côté)

Pour la clé non modifiée, le caractère modificateur est CHR\$(0).

## Remarques

- Si key\_id n'est pas dans la plage prescrite, aucune erreur n'est générée ; ces valeurs sont ignorées. Dans GW-BASIC, des choses étranges peuvent se produire dans ce cas : des anomalies d'écran et des plantages suggérant un accès mémoire involontaire.
- Si key\_id est dans [1—10] (ou [1—12] lorsque syntax=tandy ), l'instruction est interprétée comme une définition de macro de touche de fonction.
- Le modificateur étendu &h80 fait référence aux touches supplémentaires qui ont été introduites avec le clavier

modèle M à 101 touches dans la zone située entre le clavier principal et le pavé numérique. Ce sont les touches fléchées, Home, End, PgUp, PgDn, Ins, Del comme distinctes du pavé numérique. PC-BASIC ignore ce modificateur ; dans une définition de clé, cela ne fait aucune différence qu'elle soit définie ou non.

### Les erreurs

- key\_id n'est pas dans [-32768—32767] : Overflow.
- key\_id n'est pas dans [1—255] : Illegal function call.
- key\_id a une valeur de chaîne : Type mismatch.
- two\_char\_string est plus long que deux : Illegal function call.
- two\_char\_string a une valeur numérique : Type mismatch.

<b>KILL</b>	KILL filter_spec	
-------------	------------------	--

Supprime un ou plusieurs fichiers sur un périphérique de disque.

### Paramètres

- L'expression de chaîne filter\_spec est une spécification de fichier valide indiquant les fichiers à supprimer. Les caractères génériques sont autorisés. Voir FICHIERS pour une description des wildcards.

### Remarques

- Soyez très prudent avec l'utilisation de caractères génériques dans cette instruction : les règles de correspondance DOS peuvent ne pas être les mêmes que celles habituelles sur votre système d'exploitation, ce qui pourrait entraîner la suppression de fichiers inattendus.
- Cette déclaration ne peut pas supprimer les fichiers cachés et les fichiers qui n'ont pas de noms abrégés qui sont des noms DOS légaux. Cependant, ce comportement n'est pas garanti, vous ne devez donc pas en dépendre.

### Les erreurs

- filter\_spec est une valeur numérique : Type mismatch.
- Un fichier dont le nom de base est égal à celui d'un fichier correspondant à filter\_spec est ouvert : File already open
- Aucun fichier ne correspond à filter\_spec : File not found
- L'utilisateur n'a pas d'autorisation en écriture : Permission denied
- Si une erreur de syntaxe se produit après le guillemet fermant, le fichier est quand même supprimé.

<b>LCOPY</b>	LCOPY [num]	Ne fait rien.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• num est une expression numérique dans [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Cette instruction ne fait rien dans GW-BASIC. Vraisemblablement, il s'agit d'un reste d'une déclaration dans les anciennes versions de MS Basic qui copierait l'écran sur l'imprimante.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• num n'est pas dans [-32768—32767] : Overflow.</li> <li>• num n'est pas dans [0—255] : Illegal function call.</li> <li>• num a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>LET</b>	[LET] name = expression	
<p>Attribue la valeur de expression à la variable ou à l'élément de tableau name.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• name est une variable qui peut ou non déjà exister.</li> <li>• Le type d'expression correspond à celui de name : c'est-à-dire que tous les types numériques peuvent être affectés les uns aux autres mais que les chaînes ne peuvent être affectées qu'à des chaînes.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• le nom et l'expression ne sont pas de types correspondants : Type mismatch.</li> </ul>		
<b>LINE</b>	LINE [[STEP] (x0, y0)] - [STEP] (x1, y1) [, [attr] [, [ <b>B</b> [F]] [, pattern]]]	
<p>Dessine une ligne ou une boîte en mode graphique.</p> <p>Si B n'est pas spécifié, une ligne est tracée de (x0, y0) à (x1, y1), extrémités incluses.</p> <p>Si B est spécifié, un rectangle est dessiné avec des côtés parallèles à l'écran et deux coins opposés spécifiés par (x0, y0) et (x1, y1).</p> <p>Si le point de départ n'est pas donné, la position graphique actuelle est utilisée comme point de départ.</p>		



Si STEP est spécifié, (x0, y0) est un décalage par rapport à la position actuelle et (x1, y1) est un décalage par rapport à (x0, y0).

LIGNE déplace la position graphique actuelle vers le dernier point final donné.

Si F est spécifié avec B, le rectangle est rempli avec l'attribut spécifié. F et B peuvent être séparés par zéro ou plusieurs espaces.

#### Paramètres

- attr est une expression numérique dans [0—255], qui spécifie l'attribut de couleur de la ligne. S'il n'est pas donné, l'attribut courant est utilisé.
- motif est une expression numérique dans [-32768—32767]. Ceci est interprété comme un masque de motif binaire de 16 bits appliqué à des pixels consécutifs dans la ligne : un bit 1 indique un pixel tracé ; un bit 0 indique un pixel laissé intact.

Le modèle commence au bit le plus significatif, qui est appliqué au point de terminaison le plus élevé.

Si une boîte est dessinée, le motif est appliqué dans la séquence contre-intuitive suivante : (x1,y1)—(x0, y1), (x1, y0)—(x0, y0), puis (x1, y0)—(x1, y1), (x0, y0)—(x0,y1) si  $y_0 < y_1$  et  $y_0, y_1$  inversé si  $y_1 < y_0$ .

Lors du dessin d'une boîte remplie, LINE ignore le motif.

#### Remarques

- Si une coordonnée est en dehors des limites de l'écran, elle est remplacée par -1 (si inférieure à 0) ou la dimension de l'écran (si supérieure à la dimension de l'écran).

#### Les erreurs

- L'instruction se termine par une virgule et c'est la première ou la troisième : Missing operand.  
Si c'est le second : Syntax error.
- L'une des coordonnées n'est pas dans [-32768—32767] : Overflow.
- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.

<b>LINE INPUT (console)</b>	LINE INPUT [;] [prompt_literal {; ,}] string_name
<p>Affiche l'invite de donnée dans prompt_literal et lit les entrées de l'utilisateur à partir du clavier, en le stockant dans la variable string_name.</p> <p>Toutes les entrées sont lues jusqu'à ce que la touche Entrée soit enfoncée ; les 255 premiers caractères sont stockés.</p> <p>Si ; est donné juste après LINE INPUT, l'entrée de l'utilisateur Enter ending n'est pas renvoyée à l'écran.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• prompt_literal est un littéral de chaîne. Peu importe qu'il soit suivi d'une virgule ou d'un point-virgule.</li> <li>• string_name est une variable de chaîne ou un élément de tableau.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si l'entrée de l'utilisateur est interrompue par Ctrl + Break, CONT réexécute l'instruction LINE INPUT.</li> <li>• Contrairement à INPUT, LINE INPUT ne termine pas l'invite par ?.</li> </ul>	
<b>LINE INPUT (files)</b>	LINE INPUT # file_num, string_name
<p>Lit des variables de chaîne ou numériques à partir d'un fichier texte ou du tampon FIELD d'un fichier à accès aléatoire.</p> <p>Toutes les entrées sont lues jusqu'à ce que la touche Entrée soit enfoncée ; les 255 premiers caractères sont stockés. file_num doit être le numéro d'un fichier ouvert en mode INPUT ou d'un fichier à accès aléatoire ouvert en mode RANDOM.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• string_name est une variable de chaîne ou un élément de tableau.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Le # est obligatoire. Il peut y avoir ou non des espaces entre INPUT et #.</li> <li>• L'entrée se termine uniquement par un CR.</li> <li>• Si file_num est ouvert sur KYBD :, LINE INPUT# lit à partir du clavier jusqu'à ce qu'un retour ou une virgule soit rencontré (comme dans un fichier). Les touches fléchées et la suppression sont transmises comme caractères de contrôle (pas de scancodes !) précédés de CHR\$(&amp;hFF).</li> </ul>	

## Les erreurs

- L'entrée est demandée après que la fin d'un fichier texte a été atteinte ou qu'un caractère EOF a été rencontré : Input past end.
- Le dernier caractère du tampon de champ est lu : Field overflow.
- num\_fichier n'est pas un fichier ouvert : Bad file number.
- file\_num est inférieur à zéro : Illegal function call.
- num\_fichier n'est pas dans [-32768—32767] : Overflow.
- num\_fichier n'est pas ouvert pour INPUT ou RANDOM : Illegal function call.
- file\_num a une valeur de chaîne : Type mismatch.
- file\_num est ouvert sur un port COM et il s'agit du premier appel INPUT, LINE INPUT ou INPUT\$ sur ce port depuis que le tampon s'est complètement rempli (c'est-à-dire que LOF(file\_num) est devenu zéro) : Communication buffer overflow.

## LIST

LIST [line\_number\_0|.] [-[line\_number\_1|.]] [, file\_spec [anything]]

Imprime le programme à l'écran ou dans un fichier, en commençant par line\_number\_0 jusqu'à et y compris line\_number\_1.

Arrête également l'exécution du programme et rend le contrôle à l'utilisateur.

Si l'instruction LIST se termine par une spécification de fichier, tout ce qui suit est ignoré. Dans tous les cas, toute instruction supplémentaire dans un composé après LIST sera ignorée, à la fois dans un programme et en mode direct.

Lors du listage à l'écran, les mêmes caractères de contrôle sont reconnus que dans l'instruction PRINT.

### Remarques

- Dans GW-BASIC 3.23, LIST n'affichera pas les numéros de ligne 65531 — 65535 inclus. Par défaut, la LISTE de PC-BASIC affiche ces lignes. Cependant, leur affichage peut être désactivé avec l'option hide-listing=65530.

### Paramètres

- line\_number\_0 et line\_number\_1 sont des numéros de ligne dans la plage [0—65529] ou un. pour indiquer la

dernière ligne éditée. Les numéros de ligne n'ont pas besoin d'exister ; elles ou ils spécifient une plage. Si la plage est vide, rien n'est imprimé.

- L'expression de chaîne file\_spec est une spécification de fichier valide indiquant le fichier à répertorier. Si ce fichier existe déjà, il sera écrasé.

### Les erreurs

- Un numéro de ligne est supérieur à 65529 : Syntax error.
- file\_spec a une valeur numérique : Type mismatch.
- spéc\_fichier se termine par deux-points mais n'est pas un nom de périphérique ou une lettre de lecteur : Bad file number.
- spéc\_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ;  
Bad file name (sur les périphériques de disque).

<b>LLIST</b>	LLIST [line_number_0 .] [-[line_number_1 .]]
--------------	--

Imprime le programme sur l'imprimante ligne LPT1 : en commençant par line\_number\_0 jusqu'à et y compris line\_number\_1. Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur une ligne après LLIST sera ignorée, à la fois dans un programme et en mode direct.

### Remarques

- Dans GW-BASIC 3.23, LLIST n'affichera pas les numéros de ligne 65531 — 65535 inclus. Par défaut, la LLIST de PC-BASIC affiche ces lignes. Cependant, leur affichage peut être désactivé avec l'option hide-listing=65530.

### Paramètres

- line\_number\_0 et line\_number\_1 sont des numéros de ligne dans la plage [0—65529]. ou un.(point) pour indiquer la dernière ligne éditée. Les numéros de ligne n'ont pas besoin d'exister ; ils spécifient une plage. Si la plage est vide, rien n'est imprimé.

### Les erreurs

- Un numéro de ligne est supérieur à 65529 : Syntax error.

<b>LOAD</b>	LOAD file_spec [, R]	
<p>Charge le programme stocké dans un fichier en mémoire. Les variables existantes seront effacées et tout programme en mémoire sera effacé.</p> <p>LOAD implique CLEAR.</p> <p>Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.</p> <p>Si,R est spécifié, garde tous les fichiers de données ouverts et exécute le fichier spécifié.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• L'expression de chaîne file_spec est une spécification de fichier valide indiquant le fichier à partir duquel lire le programme.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_spec a une valeur numérique : Type mismatch.</li> <li>• file_spec contient des caractères interdits : Bad file number (sur CAS1 : ) ; Bad file name (sur les périphériques de disque).</li> <li>• Le fichier spécifié dans file_spec est introuvable : File not found.</li> <li>• Un fichier texte chargé contient des lignes sans numéro de ligne : Direct statement in file.</li> <li>• Un fichier texte chargé contient des lignes de plus de 255 caractères : Line buffer overflow.</li> </ul> <p>La tentative de chargement d'un fichier texte contenant des fins de ligne LF plutôt que CR LF peut provoquer cette erreur.</p>		
<b>LOCATE</b>	LOCATE [row] [, [col] [, [cursor_visible] [, [start_line] [, [stop_line] [,]]]]	
<p>Positionne le curseur sur row, col sur l'écran et modifie la forme et la visibilité du curseur.</p> <p>cursor_visible peut être 0 ou 1. Si cursor_visible est 0, cela rend le curseur invisible ; si c'est 1, rend le curseur visible.</p> <p>Cela ne fonctionne que lorsqu'un programme est en cours d'exécution. La forme du curseur est ajustée dans une cellule de caractère pour commencer à start_line et se terminer à end_line où start_line et end_line sont dans</p>		

[0—31].

Si start\_line ou end\_line est supérieur à la hauteur de la cellule de caractère (15), remplacez par 15.

#### Remarques

- Sur les cartes VGA émulées, les paramètres de forme du curseur sont interprétés d'une manière compliquée destinée à maintenir la compatibilité fonctionnelle avec CGA.
- Dans GW-BASIC, la forme du curseur est conservée après avoir appuyé deux fois sur Ins. Le curseur du mode d'insertion est différent du demi-bloc habituel. Dans PC-BASIC, le mode d'insertion réinitialise la forme du curseur par défaut.
- Les options de forme et de visibilité du curseur n'ont aucun effet en mode graphique.
- Locate accepte une 5e virgule à la fin, qui est ignorée.

#### Les erreurs

- Tout les paramètre a une valeur de chaîne : Type mismatch.
- Tout les paramètre n'est pas dans [-32768—32767] : Overflow.
- la ligne est en dehors de la zone d'affichage actuelle : Illegal function call.
- col est supérieur à la largeur actuelle : Illegal function call.
- cursor\_visible n'est pas dans [0, 1] ( [0—255] sur Tandy/PCjr) : Illegal function call.

#### **LOCK**

LOCK [#] file\_number [, record\_0]

LOCK [#] file\_number, [record\_0] TO record\_1

Verrouille un fichier ou une partie de fichier contre l'accès par d'autres utilisateurs.

Sur un fichier RANDOM, record\_0 est le premier enregistrement verrouillé et record\_1 est le dernier enregistrement verrouillé.

Sur tout autre type de fichier, record\_0 et record\_1 n'ont aucun effet. Si record\_0 n'est pas spécifié, il est supposé être 1.

Si aucun enregistrement n'est spécifié, le fichier entier est verrouillé.

#### Paramètres

- file\_number est une expression numérique dans [0—255].
- record\_0 et record\_1 sont des expressions numériques dans [1—2<sup>25</sup>-2].

## Remarques

- Dans GW-BASIC sous MS-DOS, la commande LOCK nécessite le chargement de SHARE.EXE. Le nombre maximal de verrous est spécifié dans la commande MS-DOS SHARE. Si SHARE n'a pas été activé ou si tous les verrous sont utilisés, LOCK lève la permission refusée. PC-BASIC se comporte comme si SHARE avait été activé avec des verrous illimités.
- Si numéro\_fichier est ouvert pour RANDOM, les instructions LOCK et UNLOCK doivent correspondre en termes d'enregistrement\_0 et d'enregistrement\_1. Un UNLOCK non correspondant augmentera la permission refusée.
- Pour vérifier si un autre fichier ouvert est le même fichier, PC-BASIC ne regarde que le nom de base du fichier, c'est-à-dire son nom DOS sans répertoires. Par conséquent, si un fichier "test.txt" est ouvert et verrouillé, une tentative de verrouillage d'un fichier "dir\test.txt" échouera, même s'il s'agit de fichiers différents. Inversement, si deux noms de fichiers sont différents mais pointent vers le même fichier dans le système de fichiers (par exemple en raison de liens de système de fichiers), ils seront alors considérés comme des fichiers différents par BASIC.

## Les erreurs

- Tout paramètre a une valeur de chaîne : Type mismatch.
- num\_fichier n'est pas dans [-32768—32767] : Overflow.
- file\_num n'est pas dans [0—255] : Illegal function call.
- num\_fichier n'est pas un fichier ouvert : Bad file number.
- LOCK (une partie de) un fichier portant le même nom qu'un fichier déjà verrouillé : Permission denied.
- record\_0 ou record\_1 n'est pas dans [1—2^25-2] : Bad record number.

<b>LPRINT</b>	Voir PRINT	
<b>LSET</b>	LSET string_name = expression	

Copie une valeur de chaîne dans une variable de chaîne ou un élément de tableau existant. La valeur sera justifiée à gauche et tous les caractères restants seront remplacés par des espaces.

## Paramètres

- string\_name est une variable de chaîne ou un élément de tableau.
- expression est une expression de chaîne.

## Remarques

- Si expression a une valeur plus longue que la longueur de la variable cible, elle est tronquée à la fin à la

longueur de la variable cible.

- Si string\_name n'a pas été alloué auparavant, cette instruction n'a aucun effet.
- Utilisez LSET, RSET ou MID\$ pour copier les valeurs dans un tampon FIELD.
- Si LET est utilisé sur une variable FIELD au lieu de L|RSET, la variable est détachée du champ et une nouvelle variable de chaîne normale est allouée.

### Les erreurs

- string\_name n'est pas une variable de chaîne : Type mismatch.
- expression n'a pas de valeur de chaîne : Type mismatch.

<b>MERGE</b>	MERGE file_spec	
--------------	-----------------	--

Superpose les lignes d'un programme à partir d'un fichier de programme en texte brut dans le programme existant. Les lignes chargées écrasent les lignes existantes si elles ont le même numéro de ligne.

Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.

### Paramètres

- L'expression de chaîne file\_spec est une spécification de fichier valide indiquant le fichier à partir duquel lire le programme.

### Les erreurs

- spéc\_fichier introuvable : File not found.
- spéc\_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ; Bad file name (sur les périphériques de disque).
- file\_spec n'a pas été enregistré en tant que texte brut : Bad file mode.
- Un fichier texte chargé contient des lignes sans numéro de ligne : Direct statement in file.
- Un fichier texte chargé contient des lignes de plus de 255 caractères : Line buffer overflow.

La tentative de chargement d'un fichier texte contenant des fins de ligne LF plutôt que CR LF peut provoquer cette erreur.



<b>MID\$ (statement)</b>	MID\$(string_name, position [, length]) = substring	
<p>Remplace une partie de string_name par substring.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• string_name est un nom de variable de chaîne valide.</li> <li>• position est une expression numérique comprise entre 1 et la longueur de la chaîne, inclus.</li> <li>• length est une expression numérique dans [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Aucun espace n'est autorisé entre MID\$ et (.</li> <li>• Si substring est plus long que length, seuls les premiers caractères de longueur sont utilisés.</li> <li>• Si substring est plus court que length, seuls les caractères LEN(substring) sont remplacés.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• position est supérieure à la longueur de string_name : Illegal function call, sauf si la longueur est spécifiée comme 0.</li> <li>• la position n'est pas dans [1—255] : Illegal function call.</li> <li>• la longueur n'est pas dans [0—255] : Illegal function call.</li> <li>• la position ou la longueur ne sont pas dans [-32768—32767] : Overflow.</li> </ul>		
<b>MKDIR</b>	MKDIR dir_spec	Crée un nouveau répertoire sur un périphérique de disque.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• L'expression de chaîne dir_spec est une spécification de fichier valide qui spécifie le chemin du nouveau répertoire sur un périphérique de disque.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• dir_spec n'est pas une chaîne : Type mismatch.</li> <li>• Le répertoire parent n'existe pas : Path not found.</li> <li>• Le nom du répertoire existe déjà sur ce chemin : Path/File access error.</li> <li>• L'utilisateur n'a pas d'autorisation en écriture : Permission denied.</li> </ul>		

<b>MOTOR</b>	MOTOR [num]	Ne fait rien.
<p>Paramètres</p> <ul style="list-style-type: none"> <li>• num est une expression numérique dans [0—255].</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, cette instruction active le moteur de la cassette si num est différent de zéro ou omis, et l'éteint si num est égal à zéro. Ceci n'est pas implémenté dans PC-BASIC.</li> </ul> <p>Les erreurs</p> <ul style="list-style-type: none"> <li>• num a une valeur de chaîne : Type mismatch.</li> <li>• num n'est pas dans [-32768—32767] : Overflow.</li> <li>• num n'est pas dans [0—255] : Illegal function call.</li> </ul>		
<b>NAME</b>	NAME old_name AS new_name	
<p>Renomme le fichier disque old_name en new_name.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• Les expressions de chaîne old_name et new_name sont des spécifications de fichier valides donnant respectivement le chemin d'accès sur un périphérique de disque à l'ancien et au nouveau nom de fichier.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• new_name sera modifié au format 8.3 tout en majuscules.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• old_name ou new_name ont des valeurs numériques : Type mismatch.</li> <li>• old_name n'existe pas : File not found.</li> <li>• Un fichier dont le nom de base est égal à old_name ou new_name est ouvert : File already open.</li> <li>• new_name existe : File already exists.</li> </ul>		
<b>NEW</b>	NEW	
<p>Arrête l'exécution d'un programme, supprime le programme en mémoire, exécute CLEAR et RESTORE et rend le contrôle à l'utilisateur.</p>		

<b>NEXT</b>	NEXT [var_0 [, var_1]...]	
<p>Itère une boucle FOR—NEXT : incrémente la variable de boucle et saute à l'instruction FOR.</p> <p>Si aucune variable n'est spécifiée, next correspond à l'instruction FOR la plus récente. Plusieurs instructions NEXT imbriquées peuvent être regroupées en une seule à l'aide de la liste de variables.</p> <p>Si une ou plusieurs variables sont spécifiées, leur ordre doit correspondre à l'ordre des instructions FOR précédentes.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• var_0, var_1,... sont des variables numériques qui sont des compteurs de boucle dans une instruction FOR.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Aucune instruction FOR ne correspond à l'instruction et aux variables NEXT : NEXT without FOR.</li> <li>• var_0, var_1,... sont des variables de chaîne :NEXT without FOR.</li> <li>• La variable de boucle (implicite ou explicite) est une variable entière et est prise en dehors de l'intervalle [-32768, 32767] lorsqu'elle est incrémentée après l'itération finale : Overflow.</li> </ul>		
<b>NOISE</b>	NOISE source, volume, duration	
<p>Génère différents types de bruit.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• source est une expression numérique dans [0—7]. Il indique le type de bruit :</li> </ul> <p>type de source haut de la bande de fréquence (Hz)</p> <p>0 périodique 6991</p> <p>1 périodique 3495</p> <p>2 périodiques 1747</p> <p>3 dernières tonalités périodiques jouées sur la voix 2</p> <p>0 bruit blanc 6991</p> <p>1 bruit blanc 3495</p> <p>2 bruit blanc 1747</p> <p>3 bruit blanc dernier ton joué sur la voix 2</p> <ul style="list-style-type: none"> <li>• volume est une expression numérique entre [0—15].</li> <li>• la durée est une expression numérique.</li> </ul> <p>Le volume et la durée sont déterminés de la même manière que pour l'instruction SOUND.</p>		

## Remarques

- Cette instruction n'est disponible que si syntax={pcjr|tandy} est défini.

## Les erreurs

- SOUND ON n'a pas été exécuté : Illegal function call.
- la durée n'est pas dans [-65535—65535] : Illegal function call.
- le volume n'est pas dans [0—15] : Illegal function call.
- la source n'est pas dans [0—7] : Illegal function call.

## ON (calculated jump)

ON n {GOTO|GOSUB} line\_number\_0 [, line\_number\_1]...

Saute au n<sup>ième</sup> numéro de ligne spécifié dans la liste. Si n est 0 ou supérieur au nombre de numéros de ligne dans la liste, aucun saut n'est effectué. Si GOTO est spécifié, le saut est incondtionnel ; si GOSUB est spécifié, saute à un sous-programme.

## Paramètres

- n est une expression numérique dans [0—255]. L'expression ne doit pas commencer par les mots clés de fonction STRIG, PEN, PLAY ou TIMER ; si vous avez besoin de ces fonctions, l'expression doit être entre crochets.
- line\_number\_0, line\_number\_1,... sont des numéros de ligne existants dans le programme.

## Les erreurs

- n a une valeur de chaîne : Type mismatch.
- n n'est pas dans [-32768—32767], Overflow.
- n n'est pas dans [0—255] : Illegal function call.
- Le numéro de ligne sauté n'existe pas : Undefined line number.

<b>ON (event trapping)</b>	ON {COM(n) KEY(n) STRIG(n) PEN PLAY(n) TIMER(x)} GOSUB line_number
<p>Définit une sous-routine de capture d'évènement. Le type d'évènement est donné par l'un des mots clés suivants :</p> <p><b>COM(n)</b> L'évènement est déclenché si des données sont présentes dans le tampon d'entrée de COMn : n est le numéro de port dans [1,2].</p> <p><b>KEY(n)</b> L'évènement est déclenché si la touche n est enfoncée. n est le numéro de clé [1—20] défini dans l'instruction KEY.</p> <p><b>STRIG(n)</b> Cet évènement est déclenché si le bouton de tir n est enfoncé. n dans [0,2,4,6] fait référence aux deux déclencheurs de tir sur deux joysticks.</p> <p><b>STYLO</b> L'évènement est déclenché si le crayon lumineux est sur l'écran. (Dans PC-BASIC, le stylet lumineux est émulé par défaut par le bouton droit de la souris).</p> <p><b>PLAY(n)</b> L'évènement est déclenché s'il reste exactement n notes dans la file d'attente de musique de fond. n est une expression numérique dans [1—32].</p> <p><b>TIMER(x)</b> L'évènement est déclenché toutes les x secondes après l'instruction TIMER ON. x est une expression numérique dans [1—86400].</p> <p><b>Remarques</b></p> <ul style="list-style-type: none"> <li>• Le piégeage d'évènements pour l'évènement que vous avez choisi doit d'abord être activé à l'aide de l'une des instructions : COM(n) ON, KEY(n) ON, STRIG(n) ON, PEN ON, PLAY ON, TIMER ON</li> <li>• Les évènements ne sont interceptés que lorsqu'un programme est en cours d'exécution.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• n ou x a une valeur de chaîne : Type mismatch.</li> <li>• n n'est pas dans [-32768—32767] : Overflow.</li> <li>• n ou x est en dehors de la plage spécifiée : Illegal function call.</li> </ul>	

<b>ON ERROR</b>	ON ERROR GOTO {line_number 0}
<p>Active ou désactive l'interception d'erreurs. Lorsque line_number est défini, toute erreur provoque l'appel de la routine de gestion des erreurs commençant à ce numéro de ligne ; aucun message n'est imprimé et l'exécution du programme n'est pas arrêtée.</p> <p>La routine de gestion des erreurs se termine par une instruction RESUME. Dans une routine de gestion d'erreurs, les évènements sont mis en pause et l'interception d'erreurs est désactivée.</p> <p>Après l'instruction RESUME, tous les évènements déclenchés sont récupérés dans l'ordre suivant : KEY, TIMER, PLAY - l'ordre des autres est inconnu.</p> <p>Contrairement à l'interception d'évènements, l'interception d'erreurs reste active lorsqu'aucun programme n'est qu'en cours d'exécution. ON ERROR GOTO 0 désactive l'interception d'erreurs.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>line_number est un numéro de ligne existant dans le programme.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>Il n'est pas possible de démarrer le gestionnaire d'erreurs à la ligne numéro 0.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>line_number n'existe pas : numéro de ligne indéfini.</li> </ul>	
<b>OPEN</b>	<pre>OPEN mode_char, [#] file_num, file_spec [, rec_len]  OPEN file_spec [FOR {INPUT OUTPUT APPEND RANDOM}] [ACCESS {READ WRITE READ WRITE}] [SHARED LOCK {READ WRITE READ WRITE}] AS [#] file_num [LEN = rec_len]</pre>
<p>Ouvre un fichier de données sur un appareil.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>L'expression de chaîne file_spec est une spécification de fichier valide.</li> <li>file_num est une expression numérique dans [1—max_files], où max_files est le nombre maximal de fichiers (3 par défaut).</li> <li>rec_len est une expression numérique dans [1—128] : la longueur de l'enregistrement.</li> </ul>	

- mode\_char est une expression de chaîne dont le premier caractère est l'un des ["I", "O", "A", "R"].

## Modes d'accès

Les modes FOR ou mode\_char sont les suivants :

### "I" pour INPUT

Ouvre un fichier texte en lecture et positionne le pointeur de fichier au début.

### "O" pour OUTPUT

Tronque un fichier texte au début et l'ouvre en écriture. Toutes les données précédemment présentes dans le fichier seront supprimées.

### "A" pour APPEND

Ouvre un fichier texte pour l'écriture à la fin de toute donnée existante.

### "R" pour RANDOM

Ouvre un fichier pour un accès aléatoire ; le fichier est divisé en enregistrements de longueur rec\_len.

Si LEN n'est pas spécifié, la longueur d'enregistrement par défaut est 128. Le contenu du fichier est accessible à l'aide de GET et PUT du tampon FIELD ; le tampon FIELD est accessible via les variables FIELD ou via les instructions PRINT# et INPUT#.

Si aucun mode FOR ou mode\_char n'est spécifié, le fichier est ouvert pour RANDOM.

Si FOR et ACCESS sont spécifiés, n'importe quel mode ACCESS est autorisé pour RANDOM mais pour les autres modes, l'accès doit correspondre comme suit :

```
INPUT    READ
OUTPUT   WRITE
APPEND   READ WRITE READ
RANDOM    READ WRITE all
```

## Partage et verrouillages

Si ni SHARED ni LOCK ne sont spécifiés. Dans ce processus, un fichier peut être ouvert plusieurs fois pour INPUT ou RANDOM mais une seule fois pour OUTPUT ou APPEND, tant qu'il est à nouveau ouvert en mode par défaut. Il ne peut pas être ouvert en mode PARTAGÉ ou en mode VERROUILLÉ.

Si SHARED, LOCK READ, LOCK WRITE ou LOCK READ WRITE est spécifié, le fait que deux instructions OPEN puissent accéder au même fichier dépend de l'état LOCK de l'une et de l'état ACCESS de l'autre et vice

versa.

Pour deux instructions OPEN comme suit :

OUVRIR verrou "fichier"\_1 AS 1

OUVRIR "fichier" ACCÈS acc\_2 PARTAGÉ EN TANT QUE 2

les combinaisons suivantes sont autorisées :

Access allowed		acc_2		
		READ	WRITE	READ WRITE
lock_1	SHARED	yes	yes	yes
	LOCK READ	no	yes	no
	LOCK WRITE	yes	no	no
	LOCK READ WRITE	no	no	no

Dans GW-BASIC sous MS-DOS avec SHARE.EXE actif, ces verrous doivent être appliqués sur un réseau ainsi qu'à l'intérieur d'un seul processus BASIC.

Sans SHARED et LOCK, le fichier est verrouillé exclusivement pour être utilisé par le processus GW-BASIC. En revanche, dans PC-BASIC, les verrous ne sont implémentés qu'en interne. Le fait que d'autres processus puissent accéder au fichier dépendra du système d'exploitation hôte.

Pour vérifier si un autre fichier ouvert est le même fichier, PC-BASIC ne regarde que le nom de base du fichier, c'est-à-dire son nom DOS sans répertoires. Par conséquent, si un fichier "test.txt" est ouvert et verrouillé, une tentative de verrouillage d'un fichier "dir\test.txt" échouera, même s'il s'agit de fichiers différents.

Inversement, si deux noms de fichiers sont différents mais pointent vers le même fichier dans le système de fichiers (par exemple en raison de liens de système de fichiers), ils seront alors considérés comme des fichiers différents par BASIC.

### Spécifications du fichier

Une spécification de fichier file\_spec est une expression de chaîne non vide de la forme "[device:]parameters", où device est un périphérique PC-BASIC et la forme des paramètres est spécifique au type de périphérique. Si le périphérique est omis, le périphérique actuel (l'un des périphériques de disque ou CAS1 : ) est utilisé.

Unités de disque A : — Z : et @ :

Les paramètres doivent spécifier un chemin de fichier valide sous la forme [ \ ] nom\_répertoire \ ] ... nom\_fichier.

PC-BASIC suit les conventions du système de fichiers DOS. Les noms de répertoire sont séparés par des barres



obliques inverses \ (même si le système d'exploitation hôte sépare les chemins par des barres obliques). Les noms de fichiers et de répertoires se composent d'un nom à 8 caractères et d'une extension à 3 caractères. Les noms sont insensibles à la casse.

Les caractères autorisés pour le nom de fichier et l'extension sont les caractères ASCII imprimables dans la plage &h20 – &h7E à l'exclusion des caractères " \* +, / : ; < = > ? \ [ ] |.

Les espaces sont autorisés mais les espaces de début et de fin sont ignorés. les noms AUX, CON, PRN et NUL sont réservés en tant qu'alias de périphérique et ne sont pas des noms légaux pour les fichiers ou les répertoires sur un périphérique de disque.

Un chemin commençant par une barre oblique inverse est interprété comme un chemin absolu, commençant à la racine du périphérique de disque spécifié. Sinon, le chemin est interprété comme relatif au répertoire courant sur le périphérique spécifié.

Le nom de répertoire spécial.. fait référence au répertoire parent d'un chemin précédent, ou au répertoire parent du répertoire courant si aucun chemin n'est donné. Le nom du répertoire spécial. fait référence au même répertoire que celui donné par le chemin précédent, ou au répertoire courant si aucun chemin précédent n'est donné.

Si le nom de fichier fourni ne contient aucun point, les instructions LOAD, SAVE, BLOAD, BSAVE, CHAIN , MERGE, RUN et LIST ajoutent l'extension par défaut.BAS. Pour faire référence à un nom de fichier sans extension, la spécification du fichier doit se terminer par un point.. Pour les autres instructions, l'ajout d'un point est autorisé mais pas obligatoire.

#### Notes de compatibilité

Contrairement à PC-BASIC, certaines versions de MS-DOS autorisent certains caractères dans la plage &h7F – &hFF. Cependant, leur admissibilité et leur interprétation dépendent de la page de code de la console, qui peut être différente de la page de code d'affichage qui affecte GW-BASIC.

En fonction de sa page de code de console, MS-DOS remplacera les lettres accentuées par leur variante majuscule non accentuée. Certaines implémentations DOS supprimeront les espaces des noms de fichiers ; c'est notamment le cas sur DOSBox.

Afin de permettre l'accès aux fichiers dont le nom sur le système hôte n'est pas conforme aux normes DOS tout en maintenant la compatibilité avec GW-BASIC, PC-BASIC suivra ces étapes pour faire correspondre les noms de fichiers de style DOS aux noms de fichiers hôtes :

1. Recherchez un fichier portant le nom fourni. Il peut s'agir d'un long nom de fichier qui peut contenir des caractères non autorisés et qui sera sensible à la casse si votre système de fichiers l'est.
2. Si un tel fichier n'est pas trouvé, il tronquera le nom fourni au format 8.3 tout en majuscules et recherchera

une correspondance exacte. Le nom tronqué se compose des 8 premiers caractères avant le premier point, suivis des trois premiers caractères après le premier point. Si le nom de fichier résultant contient des caractères non autorisés, une erreur sera générée.

3. Recherchez les noms 8.3 en casse mixte qui correspondent au nom fourni sans tenir compte de la casse. Ces fichiers sont recherchés dans l'ordre lexicographique. Les noms de fichiers supérieurs à 8.3 ne seront pas mis en correspondance, à moins que leur nom ne soit entré exactement.

Sous Windows, le nom correspondant peut être un nom de fichier court ainsi qu'un nom de fichier long à condition qu'il soit de longueur 8.3 — il peut, par exemple, contenir des espaces et donc ne pas être un nom de fichier court Windows valide.

Si le nom de fichier fourni se termine par un seul point et ne contient aucun autre point, PC-BASIC fera d'abord correspondre le nom tel qu'il est fourni ; s'il n'est pas trouvé, il correspondra au nom fourni mais sans le point unique. Le format 8.3 d'un tel nom de fichier correspondra aux noms de fichiers avec et sans le point, dans l'ordre lexicographique.

Si aucun fichier correspondant n'est trouvé pour un nom de fichier de sortie, un nouveau fichier sera créé avec un nom de fichier 8.3 entièrement en majuscules.

Appareil à cassettes CAS1 :

Les paramètres peuvent être un nom de fichier de huit caractères maximum. Les noms de fichier de cassette sont sensibles à la casse, n'ont ni chemin ni extension, peuvent être vides et n'ont pas besoin d'être uniques. Ils peuvent contenir n'importe quel caractère dans la plage &h20 – &hFF.

Sur l'appareil à cassette, lorsqu'il est appelé en mode direct, OPEN, CHAIN, MERGE, LOAD et BLOAD imprimeront un message à la console pour chaque fichier trouvé lors de l'enroulement de la bande. Le message se compose du nom de fichier suivi d'un point et du type de fichier et se termine par un message d'état. Le type de fichier est l'un des suivants :

A fichier programme au format texte

B Fichier de programme au format tokenisé

D Fichier de données

M Image mémoire BSAVE

P Fichier programme au format protégé

Si le fichier ne correspond pas à la spécification de fichier et au type de fichier requis, l'état est Ignoré ; si le fichier correspond, le statut est Found.

Lorsqu'elles sont appelées à partir d'un programme, ces instructions n'impriment pas de messages sur la console.

Si le périphérique a été spécifié explicitement, les paramètres peuvent également être vides. Dans ce cas, le

premier fichier du type approprié est ouvert.

Console et périphériques parallèles SCRN :, KYBD : et LPTn :

Ces appareils ne permettent pas d'autres paramètres d'appareil.

Périphériques série COMn :

Lors de l'ouverture d'un port COM, le file\_spec a la forme

```
"COMn:[speed[,parity[,data[,stop[,RS][,CS[n]][,DS[n]][,CD[n]] [,LF][,PE]]]]]"
```

Les quatre premiers paramètres après les deux-points du périphérique doivent être donnés dans l'ordre spécifié, mais les paramètres nommés peuvent être donnés dans n'importe quel ordre. La signification des paramètres est:

Paramètre	Par défaut	Signification
speed	300	Débit en bauds (bps) pour la connexion. la vitesse est l'un des [75, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600].
parity	E	Convention de bit de parité. la parité est l'une des [S, M, O, E, N]. S SPACE Bit de parité toujours mis à 0. M MARK Bit de parité toujours mis à 1. O ODD Bit de parité défini de sorte que la parité des caractères soit impaire. E EVEN Bit de parité défini pour que la parité des caractères soit paire. N NONE Aucun bit de parité transmis ou reçu.
data	?	Bits de données par octet. les données doivent être l'une des [4, 5, 6, 7, 8]. Un octet se compose des bits de données plus le bit de parité, le cas échéant. La taille des octets doit être comprise dans la plage [5—8] : si les données sont 4, la parité ne doit pas être N ; si les données sont 8, la parité doit être N.
stop	1	Le nombre de bits d'arrêt. stop doit être 1 ou 2. La valeur par défaut est 2 si la vitesse est 75 ou 110 ; 1 sinon.
RS	no	Supprimer la demande d'envoi.
CS[n]	CS1000	Définit le délai d'expiration Clear To Send sur n millisecondes. Si n vaut 0 ou n'est pas donné, désactivez la vérification CTS. La valeur par défaut est CS0 si RS est défini ; CS1000 sinon.
DS[n]	DS1000	Définit le délai d'expiration de l'ensemble de données prêt sur n millisecondes. Si n vaut 0 ou n'est pas donné, désactive la vérification DSR.
CD[n]	CD0	Définit le délai de détection de porteuse sur n millisecondes. Si n vaut 0 ou n'est

		pas donné, désactive la vérification du CD.
LF	no	Envoie un saut de ligne après chaque retour de charriot.
PE	no	Activer le contrôle de parité (ce paramètre est ignoré par PC-BASIC).

### Remarques

- Si un port COM est ouvert pour RANDOM, l'accès s'effectue octet par octet plutôt que via les enregistrements FIELD ; PRINT# et INPUT# accèdent directement au port. rec\_len définit le nombre d'octets lus par les instructions GET et PUT.
- Pour les modes INPUT, OUTPUT et APPEND, LEN peut être spécifié mais est ignoré.
- Si une tentative d'E/S contrevient au mode FOR spécifié, les touches PRINT ou INPUT instruction déclenchera Bad file mode.
- Si une E/S ALÉATOIRE est tentée en violation du mode d'ACCÈS spécifié, l'instruction PUT ou GET génère une erreur Path/File access error.
- Le # est facultatif et n'a aucun effet.

### Les erreurs

- file\_spec est vide ou périphérique inexistant : Bad file number.
- FOR APPEND ACCESS WRITE est spécifié : Path/File access error.
- FOR et ACCESS ne correspondent pas d'une autre manière : Syntax error.
- Les paramètres COM : file\_spec ne suivent pas la spécification : Bad file name.
- Le CAS1 : file\_spec contient des caractères interdits : Bad file number.
- Un fichier portant le même nom est déjà ouvert pour OUTPUT ou APPEND : File already open. Ceci n'est déclenché que pour COMn:, CASn: et les périphériques de disque.
- rec\_len ou file\_num ont des valeurs de chaîne : Type mismatch.
- file\_spec ou mode\_char ont des valeurs numériques : Type mismatch.
- num\_fichier n'est pas dans [-32768—32767] : Overflow.

- file\_num n'est pas dans [0—255] : Illegal function call.
- file\_num n'est pas dans [1—max\_files] : Bad file number.
- rec\_len n'est pas dans [-32768—32767] : Overflow.
- rec\_len n'est pas dans [1—128] : Illegal function call.
- mode\_char est vide ou le premier caractère n'est pas dans ["I", "O", "A", "R"] : Bad file mode.

## OPTION BASE

OPTION BASE n

Définit l'index de départ de tous les tableaux sur n.

### Paramètres

- n est un chiffre littéral 0 ou 1. Les expressions ne sont pas autorisées.

### Remarques

- Si OPTION BASE n'a pas été appelée, la première allocation de tableau prend par défaut l'index de départ 0.

### Les erreurs

- n n'est pas un chiffre 0 ou 1 : Syntax error.
- L'OPTION BASE 1 est appelée mais un tableau a déjà été alloué auparavant : Duplicate definition.
- OPTION BASE est appelée plus d'une fois avec un index de départ différent : Duplicate definition.

## OUT

OUT port, value

Envoie un octet à un port de machine émulé.

Les ports machine suivants sont émulés dans PC-BASIC :

port Effet &h201 réinitialise le port de jeu (port joystick)

&h3C5 définit le masque d'écriture pour les plans de couleur SCREEN 7, 8, 9. masque de bits = 2 ^ valeur.

&h3CF définit le plan de couleur lu sur valeur.

&h3D8 si valeur = &h1A, active le colorburst composite.

si valeur = &h1E, désactive le colorburst composite.

Nécessite vidéo={cga, tandy, pcjr}.

### Remarques

- Seul un nombre limité de ports machine sont émulés.

- Dans GW-BASIC sous MS-DOS, la séquence nécessaire pour définir le masque de plan de couleur est:

`SORTIE &h3C4, 2`

`OUT &h3C5, 2 ^ avion`

La séquence nécessaire pour définir le plan de couleur est :

`SORTIE &h3CE, 4`

`OUT &h3CF, avion`

Les instructions `OUT` initiales n'ont actuellement aucun effet dans `PC-BASIC`.

#### Paramètres

- port est une expression numérique dans [-32768—65535].
- valeur est une expression numérique dans [0—255].

#### Les erreurs

- port ou value a une valeur de chaîne : Type mismatch.
- le port n'est pas dans [-32768—65535] : Overflow.
- la valeur n'est pas dans [-32768—32767] : Overflow.
- la valeur n'est pas dans [0—255] : Illegal function call.

### **PAINT**

`PAINT [STEP] (x, y) [, attrib [, border [, background]]]`

Remplit l'écran avec une couleur ou un motif, à partir du point de départ donné.

#### Paramètres

- x, y sont des expressions numériques dans la plage [-32768—32767] Si `STEP` est spécifié, x y sont des décalages par rapport à la position actuelle. Si le point d'origine se trouve en dehors de la zone d'écran visible, aucun remplissage plein n'est effectué.
- attrib est une expression qui spécifie l'attribut ou le motif de remplissage. S'il n'est pas spécifié, l'attribut de premier plan actuel est utilisé.
- Si attrib a une valeur numérique, il doit être dans [0—255] ; il spécifie l'attribut de couleur utilisé pour remplir.
- Si attrib a une valeur de chaîne, il spécifie un motif de mosaïque (voir ci-dessous).
- border est une expression numérique dans [0—255]. Il spécifie l'attribut de la limite de remplissage (voir ci-dessous).
- background est une expression de chaîne qui représente un motif de mosaïque d'arrière-plan à ignorer lors de

la détermination des limites (voir ci-dessous).

## Motifs de carreaux

Un modèle de mosaïque peut être spécifié par une chaîne de 255 caractères maximum. L'interprétation de la chaîne dépend du nombre de bits par pixel et du mode d'écran en cours.

1 bit par pixel (par exemple SCREEN 2 )

Voici un exemple:

76543210 Valeur d'octet

\*..... &h80

. \*..... &h40

.. \*..... &h20

... \*.... &h10

.... \*... &h08

..... \*.. &h04

..... \*.. &h02

Ce motif à rayures diagonales peut ainsi être réalisé avec

PEINTURE (0, 0), CHR\$(128)+CHR\$(64)+CHR\$(32)+CHR\$(16)+CHR\$(8)+CHR\$(4)+CHR\$(2)

## ÉCRAN 7, 8, 9

Le motif de mosaïque a toujours une largeur de 8 pixels. Le premier caractère de la chaîne de motif contient le premier bit de chacun de ces 8 pixels, le second caractère contient les seconds bits, etc. Par exemple, en mode 2 bits par pixel, quatre attributs de couleur peuvent être utilisés dans le motif. Pour créer un motif à rayures diagonales de la même forme, dans l'attribut &h03, nous avons maintenant besoin d'une chaîne de tuiles deux fois plus longue :

Attribute bit	76543210	Byte value
0	*.....	&h80
1	*.....	&h80
0	.*.....	&h40
1	.*.....	&h40
0	..*.....	&h20
1	..*.....	&h20
0	...*....	&h10
1	...*....	&h10
0	....*...	&h08
1	....*...	&h08
0	.....*..	&h04
1	.....*..	&h04
0	.....*.	&h02
1	.....*.	&h02

Si la chaîne de modèle est tronquée avant que tous les bits de la dernière ligne aient été définis, les bits restants seront à zéro.

### ÉCRAN 1, 3, 4, 5, 6

Chaque ligne du motif de mosaïque représente une ligne d'écran. Les couleurs sont codées en bits consécutifs ; plus il y a de bits par pixel, plus le motif est étroit. Pour 2 bits par pixel, le motif a une largeur de 4 pixels ; pour 4 bits par pixel, il fait 2 pixels de large. La chaîne de motif suivante encode une bande pointillée diagonale en deux couleurs :

3210	76543210	Byte value
2000	*.....	&h80
1000	.*.....	&h40
0200	..*.....	&h20
0100	...*....	&h10
0020	....*...	&h08
0010	.....*..	&h04
0002	.....*.	&h02

Le motif de carreaux est ancré à l'écran ; imaginez une grille commençant à (0,0) et recouvrant l'écran. Chaque fois qu'une zone est remplie de tuiles, les tuiles sont placées dans cette grille. De cette façon, les zones adjacentes auront un carrelage continu même si elles ont été remplies à partir de différents points de départ.



## Les frontières

Un remplissage uni plein s'arrête aux pixels qui ont le même attribut que le remplissage ou qui ont l'attribut de bordure spécifié, le cas échéant.

Un remplissage inondé de mosaïque s'arrête à l'attribut de bordure spécifié ; si aucun attribut de bordure n'est spécifié, il s'arrête à l'attribut de premier plan actuel.

Un remplissage en mosaïque s'arrête également à des intervalles de ligne de balayage identiques au motif de mosaïque de cette ligne, sauf si un motif d'arrière-plan est spécifié et que l'intervalle est également égal au motif d'arrière-plan de cette ligne.

Le motif de carreaux d'arrière-plan est construit exactement comme le motif de carreaux. Cependant, seule la première ligne de la tuile de fond est prise en compte ; le reste est ignoré. La tuile d'arrière-plan ne doit pas correspondre à la tuile d'attribut, ou à plus de deux lignes consécutives de celle-ci.

## Les erreurs

- Si plus de deux lignes consécutives de la tuile d'attribut (ou toutes les lignes, s'il y a moins supérieur à trois) est égal à la première ligne de la vignette d'arrière-plan : Illegal function call.
- background a une valeur numérique : Illegal function call.
- border, x ou y ont une valeur de chaîne : Type mismatch.
- border, x ou y ne sont pas dans [-32768—32767] : Overflow.
- la bordure n'est pas dans [0—255] : Illegal function call.
- attrib est numérique et non dans [-32768—32767] : Overflow.
- attrib est numérique et pas dans [0—255] : Illegal function call.

<b>PALETTE</b>	PALETTE [attrib, colour]	
<p>Asigne une couleur à un attribut. Tous les pixels avec cet attribut changeront de couleur immédiatement. Si aucun paramètre n'est spécifié, PALETTE revient au réglage initial.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• attrib est une expression numérique comprise entre 0 et la taille actuelle de la palette, moins un.</li> <li>• couleur est une expression numérique comprise entre -1 et le nombre maximal de couleurs pour le mode d'écran actuel, moins une. Si la couleur est égale à -1, la palette reste inchangée.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• attrib ou color a une valeur de chaîne : Type mismatch.</li> <li>• l'attribut ou la couleur n'est pas dans [-32768—32767] : Overflow.</li> <li>• l'attribut ou la couleur n'est pas dans la plage : Illegal function call.</li> </ul>		
<b>PALETTE USING</b>	PALETTE USING int_array_name {( [ start_index ) }	
<p>Attribue de nouvelles couleurs à tous les attributs.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• int_array_name est un tableau unidimensionnel ou multidimensionnel d'entiers ( % ) qui fournira les nouvelles valeurs de la palette.</li> <li>• start_index est une expression numérique qui indique à quel index du tableau commencer le mappage vers la palette.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Les valeurs de tableau sont affectées aux entrées de la palette dans l'ordre dans lequel elles sont stockées en mémoire. Voir Tableaux pour plus de détails sur la disposition des tableaux en mémoire.</li> <li>• Si une entrée de tableau a la valeur -1, l'attribut correspondant reste inchangé.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• int_array_name n'a pas été alloué : Illegal function call. Le tableau ne sera pas automatiquement alloué.</li> <li>• int_array_name n'est pas un tableau d'entiers : Type mismatch.</li> <li>• int_array_name est trop court : Illegal function call.</li> <li>• start_index a une valeur de chaîne : Type mismatch.</li> </ul>		

- start\_index n'est pas dans [-32768—32767] : Overflow.
- start\_index est en dehors des dimensions du tableau : Subscript out of range.

## PCOPY

PCOPY src, dst

Copie la page écran src vers dst. Tout le texte et les graphiques sur dst sont remplacés par ceux de src.

### Paramètres

- src et dst sont des expressions numériques comprises entre 0 et le nombre de pages du mode vidéo courant, moins une.

### Les erreurs

- src ou dst a une valeur de chaîne : Type mismatch.
- src ou dst n'est pas dans [-32768—32767] : Overflow.
- src ou dst est hors limites : Illegal function call.

## PEN (statement)

PEN {ON|OFF|STOP}

Contrôle le piégeage des événements et l'accès en lecture du stylet lumineux (émulé via la souris dans PC-BASIC).

PEN ON active la lecture et le recouvrement du stylet. PEN OFF l'éteint. PEN STOP suspend l'interception d'évènements PEN jusqu'à ce que PEN ON soit exécuté.

Jusqu'à 1 évènement peut être déclenché pendant la suspension, à condition que la gestion des évènements ait été activée avant la suspension. L'évènement déclenché pendant la suspension est traité immédiatement après la prochaine instruction PEN ON.

## PLAY (event switch)

PLAY {ON|OFF|STOP}

- ON : active le piégeage d'évènement ON PLAY de la file d'attente musicale.
- OFF : désactive le piégeage.
- STOP : arrête le recouvrement jusqu'à ce que PLAY ON soit utilisé.

Les évènements qui se produisent alors que le piégeage est arrêté se déclenchent immédiatement lorsque le piégeage est réactivé.

**PLAY**  
**(music statement)**

PLAY [mml\_string\_0] [, [mml\_string\_1] [, mml\_string\_2]]

Joue la mélodie définie par les chaînes du langage de macro de musique mml\_string\_0, sauf si syntax={tandy | pcjr} est défini, seule la syntaxe à voix unique est disponible. Les trois chaînes MML distinctes correspondent aux trois voix de l'adaptateur de son PCjr/Tandy. Les notes de ces chaînes sont jouées de manière synchrone.

Paramètres

- mml\_string\_0, mml\_string\_1, mml\_string\_2 sont des expressions de chaîne dans MML.
- Au moins un paramètre doit être fourni et l'instruction ne doit pas se terminer par une virgule.

Référence du langage macro musical

Notes et pauses

A-G [#,+,-]

A-G sont des notes. # ou + après une note produit un dièse ; - produit un plat.

Toute note suivie de #, + ou - doit faire référence à une touche noire sur un piano.

L(n)

Définit la longueur de chaque note. L4 est une noire, L1 est une ronde, et ainsi de suite. n peut être compris entre 1 et 64.

La longueur peut également suivre la note pour modifier la longueur de cette note uniquement. A16 est équivalent à L16A.

MF Musique de premier plan. Les instructions PLAY et SOUND doivent être exécutées au premier plan. C'est-à-dire que chaque note ou son suivant n'est pas démarré tant que la note ou le son précédent n'est pas terminé. Il s'agit de la valeur par défaut initiale.

Fond de musique MB. Les instructions PLAY et SOUND doivent s'exécuter en arrière-plan. C'est-à-dire que chaque note ou son est placé dans un tampon permettant au programme BASIC de continuer son exécution pendant que la musique joue en arrière-plan. Jusqu'à 32 notes (ou silences) peuvent être jouées en arrière-plan en même temps.

MN Musique normale. Chaque note joue sept huitièmes du temps déterminé par L (longueur).

ML Music legato. Chaque note joue la période complète définie par L.

MS Musique saccadée. Chaque note joue les trois quarts du temps déterminé par L.

N(n) Joue la note n. n peut varier de 0 à 84. Dans les 7 octaves possibles, il y a 84 notes. n mis à 0 indique un silence.

O(n) Octave 0 définit l'octave actuelle. Il y a 7 octaves (0 à 6). La valeur par défaut est 4. Le do médian est au

début de l'octave 3.

P(n) Pause. P peut varier de 1 à 64.

T(n) Tempo. T définit le nombre de L4 en une minute. n peut varier de 32 à 255. La valeur par défaut est 120.

. (période)

Une période après une note augmente le temps de jeu de la note de 3/2 fois la période déterminée par L (longueur de note) fois T (tempo). Plusieurs périodes peuvent apparaître après une note, et le temps de jeu est mis à l'échelle en conséquence.

Par exemple, A. fera jouer la note A une fois et demie le temps de jeu déterminé par L (longueur de la note) fois T (le tempo); deux points placés après A (A..) feront jouer la note à 9/4 fois sa valeur attribuée ; un A à trois points (A...) à 27/8, etc.

Des points peuvent également apparaître après un P (pause) et augmenter la durée de la pause comme décrit ci-dessus.

Xstring ;

Exécute une sous-chaîne, où chaîne est une variable affectée à une chaîne de commandes PLAY.

En raison de la lenteur du taux d'interruption de l'horloge, certaines notes ne jouent pas à des tempos plus élevés ; par exemple, 1,64 à T255. Ces combinaisons note/tempo doivent être déterminées par l'expérimentation.

>n Un symbole supérieur à précédant la note n joue la note dans l'octave immédiatement supérieure.

<n Un symbole inférieur à précédant la note n joue la note dans l'octave inférieure suivante.

Contrôle du volume

Le contrôle du volume est disponible sur syntax={tandy | pcjr} uniquement :

Effet de commande

Vn Réglez le volume sur n, dans la plage [-1—15]. -1 signifie plein volume. Si SOUND ON n'a pas été exécuté, cela n'a aucun effet.

Paramètres MML

Les variables numériques n dans les commandes ci-dessus peuvent être :

- un littéral entier, par ex. JOUER "L4G"
- un nom de variable numérique ou un élément de tableau var précédé de = et suivi de ;. Par exemple, PLAY "L=VAR;G" ou PLAY "L=A(1);G"
- le résultat de VARPTR\$(var) précédé de =. Par exemple, PLAY "L=" + VARPTR\$(VAR) + "G"

Notez que seuls les littéraux numériques peuvent suivre les notes nommées et que le déréréfencement des

variables ou des tableaux n'est pas autorisé ici. C'est une erreur d'écrire `PLAY "G=VAR;"` ou `PLAY "G=" + VARPTR$(VAR)`.

Utilisez `PLAY "G4"` ou `PLAY "L=VAR;G"` ou `PLAY "L=" + VARPTR$(VAR) + "G"` à la place.

Les commandes peuvent éventuellement être séparées par un point-virgule `;`, mais pas par plus d'un.

`PLAY "MBABCDABCDABCD"`

`PLAY "L4 ABGD"`

### Les erreurs

- `mml_string` a une valeur numérique : Type mismatch.
- `mml_string` contient des erreurs dans le MML : appel de fonction illégal.
- Une variable dans une chaîne MML est de type incorrect : Type mismatch.
- Aucune chaîne MML n'est spécifiée : opérande manquant.
- Si `SOUND ON` n'a pas été exécuté, l'utilisation de la syntaxe à trois voix génère une erreur de syntaxe.

<b>POKE</b>	POKE address, value	
-------------	---------------------	--

Définit la valeur de l'octet de mémoire au segment \* 16 + adresse à valeur, où segment est le segment actuel défini avec `DEF SEG`.

### Paramètres

- adresse est une expression numérique dans [-32768—65535]. Les valeurs négatives sont interprétées comme leur complément à deux.
- valeur est une expression numérique dans [0—255].

### Remarques

- La mémoire n'est que partiellement émulée en PC-BASIC. Voir Modèle de mémoire pour les adresses prises en charge. En dehors des zones de mémoire émulées, cette instruction n'a aucun effet.

### Les erreurs

- l'adresse ou la valeur a une valeur de chaîne : Type mismatch.
- l'adresse n'est pas dans [-32768—65535] : Overflow.
- la valeur n'est pas dans [-32768—32767] : Overflow.
- la valeur n'est pas dans [0—255] : Illegal function call.

## **PSET and PRESET**

{ PSET | PRESET } [STEP] (x, y) [, attrib]

Change l'attribut d'un pixel sur l'écran à la position (x, y). Si STEP est spécifié, (x,y) est un décalage par rapport à la position actuelle.

Si attrib est compris entre 0 et la taille de la palette du mode écran, le pixel est remplacé par l'attribut attrib.

Si l'attribut est supérieur à la taille de la palette, l'attribut du pixel est remplacé par la valeur d'attribut légale la plus élevée.

Si attrib n'est pas spécifié, PSET remplace l'attribut par l'attribut de premier plan actuel tandis que PRESET le remplace par zéro.

### **Paramètres**

- x, y sont des expressions numériques dans [-32768—32767].
- attrib est une expression numérique dans [0—255].

### **Les erreurs**

- x ou y a une valeur de chaîne : Type mismatch.
- attrib, x ou y ou les coordonnées physiques dans lesquelles ils se traduisent ne sont pas dans [-32768—32767] : Overflow.
- attrib n'est pas dans [0—255] : Illegal function call.

**PRINT et LPRINT**

```
{LPRINT|{PRINT|?} [# file_num,]}
[expr_0|;|,|SPC(n)|TAB(n)]...
[USING format; uexpr_0 [{;|,} uexpr_1]... [;|,]]
```

Écrit des expressions sur l'écran, l'imprimante ou le fichier.

Si LPRINT est utilisé, la sortie passe à LPT1 :

Si file\_num est spécifié, la sortie va au fichier ouvert sous ce numéro.

? est un raccourci pour PRINT.

Lors de l'écriture d'une expression de chaîne à l'écran, les caractères de contrôle suivants ont une signification particulière. Les autres caractères sont affichés sous la forme de leur glyphe correspondant dans la page de code actuelle.

Code	Car	Effet
&h07	BEL	Bip le haut-parleur.
&h08	BS	Efface le caractère de la colonne précédente et déplacez le curseur vers l'arrière.
&h09	HT	Passe au prochain taquet de tabulation à 8 cellules.
&h0A	LF	Aller à la colonne la plus à gauche de la rangée suivante ; connecte les rangées à une ligne
&h0B	VT	logique.
&h0C	FF	Déplace le curseur en haut à gauche de l'écran.
&h0D	CR	Efface l'écran.
&h1C	FS	Aller à la colonne la plus à gauche de la ligne suivante.
&h1D	GS	Déplace le curseur d'une colonne vers la droite.
&h1E	RS	Déplace le curseur d'une colonne vers la gauche.
&h1F	US	Déplace le curseur d'une ligne vers le haut.
		Déplace le curseur d'une ligne vers le bas.

Les expressions peuvent éventuellement être séparées par un ou plusieurs des mots clés suivants :

;

Attache deux expressions étroitement ensemble ; les chaînes seront imprimées sans aucun espace entre elles, les nombres auront un espace les séparant, en plus de l'espace ou du signe moins qui indique le signe du nombre.

,

L'expression après sera positionnée à la prochaine zone disponible. Le fichier de sortie est divisé en zones de 14 caractères ; si la largeur du fichier n'est pas un multiple de 14, les espaces restants ne sont pas utilisés et la première zone de la ligne suivante est utilisée à la place. Si le fichier a une largeur inférieure à 14 caractères, les zones sont déterminées comme si le fichier était en boucle continue.



## SPC(n)

Produit n espaces, où n est une expression numérique. si n est inférieur à zéro, sa valeur par défaut est zéro. Si n est supérieur à la largeur du fichier, il est pris modulo la largeur du fichier.

## TAB(n)

Passé à la colonne n, où n est une expression numérique. si n est inférieur à zéro, sa valeur par défaut est zéro. Si n est supérieur à la largeur du fichier, il est pris modulo la largeur du fichier. Si la colonne actuelle est supérieure à n, TAB passe à la colonne n sur la ligne suivante.

Si l'instruction print ne se termine pas par l'un de ces quatre jetons de séparation, une nouvelle ligne est imprimée après la dernière expression. Les expressions de chaîne peuvent être séparées par un ou plusieurs espaces, ce qui a le même effet que la séparation par des points-virgules.

## Formater la syntaxe de la chaîne

Une déclaration USING se produit à la fin d'une instruction [L]PRINT[#] et écrit une chaîne formatée sur l'écran, l'imprimante ou le fichier. Les tableaux suivants répertorient les jetons de format pouvant être utilisés dans la chaîne de format.

–

Caractère d'échappement; entraîne l'impression du caractère suivant dans la chaîne de format tel quel plutôt qu'interprété comme un jeton de format.

Pour les expressions de chaîne :

! Imprime le premier caractère d'une chaîne.

\\ Imprime 2 caractères ou plus d'une chaîne.

Un plus grand nombre de caractères est sélectionné en séparant les \s par des espaces.

& Imprime toute la chaîne.

Pour les expressions numériques, la chaîne de format spécifie une largeur et un alignement.

# Indique une position pour un chiffre.

. Indiquez la virgule décimale.

, Avant la virgule : provoque le regroupement des chiffres par trois séparés par des virgules.

Après la virgule décimale, ce n'est pas un jeton. Fournit une position de chiffre.

Le nombre de caractères dans le champ ne doit pas dépasser 24.

Jetons précédant le champ numérique :

+ Provoque l'impression du signe pour les nombres positifs et négatifs. Le signe doit être imprimé à gauche du numéro.

\*\* Provoque le remplacement de tous les espaces de début par \*s. Fournit des positions à deux chiffres. \$\$ Provoque l'impression d'un \$ à gauche du nombre. Fournit une position de chiffre.

Jetons à la fin du champ numérique :

+ Provoque l'impression du signe pour les nombres positifs et négatifs. Le signe sera imprimé à droite du numéro.

- Provoque l'impression du signe des nombres négatifs à droite du nombre. Notez que - précédant le champ n'est pas un jeton mais imprimé littéralement.

^^^^ Précise que la notation scientifique doit être utilisée, avec une mantisse et un exposant représenté par E+00. En notation scientifique, les séparateurs de milliers ne seront pas affichés et tout caractère, a le même effet que #.

Les expressions numériques sont toujours entièrement imprimées, même si elles ne tiennent pas dans les positions spécifiées.

Si le nombre ne tient pas dans l'espace autorisé, un % le précédant est imprimé.

- S'il y a plus d'expressions que de champs de format, la chaîne de format est bouclée.
- Les expressions peuvent être séparées par des points-virgules ou des virgules ; l'effet est le même.
- Si la déclaration USING se termine par une virgule ou un point-virgule, aucune nouvelle ligne n'est imprimée à la fin.
- Après une déclaration USING, les autres éléments de la syntaxe PRINT tels que SPC(et TAB( ne peuvent pas être utilisés.

Paramètres

- `expr_0`, `expr_1`,... sont des expressions de tout type.
- `format` est une expression de chaîne qui spécifie le format de sortie.

- uexpr\_0, uexpr\_1,... sont des expressions correspondant à un jeton dans la chaîne de format.

#### Remarques

- Si une erreur est générée, la sortie avant que l'erreur ne se produise est imprimée normalement.
- Dans GW-BASIC, lors du formatage d'un nombre avec un signe dollar, si le nombre est dans la plage [-10000—32767] et ne tient pas dans la largeur du champ numérique, le signe moins est omis. Ceci n'est pas implémenté dans PC-BASIC.

#### Les erreurs

- n a une valeur de chaîne : Type mismatch.
- n n'est pas dans [-32768—65535] : Overflow.
- La chaîne de format ne contient aucun jeton : Illegal function call.
- Une expression ne correspond pas au type de jeton de format correspondant : Type mismatch.
- Un champ numérique dans la chaîne de format dépasse 24 caractères : Illegal function call.
- Un champ numérique dans la chaîne de format ne contient aucun caractère # : Illegal function call.

#### PUT (files)

PUT [#] file\_number [, record\_number]

Écrit un enregistrement dans le fichier à accès aléatoire file\_number à la position record\_number.

#### Paramètres

- file\_number est une expression numérique qui donne le numéro d'un fichier à accès aléatoire ouvert. Le # est facultatif et n'a aucun effet.
- record\_number est une expression numérique dans [1—33554432] ( $2^{25}$ ) et est interprété comme le numéro d'enregistrement.

#### Remarques

- Le numéro d'enregistrement est stocké en simple précision ; cette précision n'est pas assez élevée pour distinguer des enregistrements uniques proches de la valeur maximale de  $2^{25}$ .

#### Les erreurs

- record\_number n'est pas dans [1—33554432] : Bad record number.
- file\_number n'est pas dans [0—255] : Illegal function call.
- file\_number n'est pas le numéro d'un fichier ouvert : Bad file mode.
- file\_number est ouvert sous un mode autre que RANDOM : Bad file mode.
- record\_number n'est pas spécifié : Missing operand.

<b>PUT</b> <b>(communications)</b>	PUT [#] com_file_number [, number_bytes]
<p>Écrit number_bytes octets dans le tampon de communication ouvert sous le numéro de fichier com_file_number. number_bytes est une expression numérique comprise entre 1 et la longueur du tampon COM, inclus.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Dans GW-BASIC, une erreur d'E/S de périphérique est déclenchée pour une erreur de dépassement, une erreur de trame et une interruption d'interruption. Le périphérique de défaut est déclenché si le DSR est perdu pendant les E/S. Une erreur de parité est déclenchée si la parité est activée et qu'une parité incorrecte est rencontrée. C'est selon le manuel; il n'est pas testé.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• octets est inférieur à 1 : Bad record number.</li> <li>• octets est inférieur à 32768 et supérieur à la longueur du tampon COM : Illegal function call.</li> <li>• com_file_number n'est pas spécifié : Missing operand.</li> <li>• com_file_number n'est pas dans [0—255] : Illegal function call.</li> <li>• com_file_number n'est pas le numéro d'un fichier ouvert : Bad file number.</li> <li>• Le buffer d'entrée série est plein, c'est-à-dire LOF(com_file_number) = 0 et LOC(com_file_number)=255 : Communication buffer overflow.</li> </ul>	
<b>PUT (graphics)</b>	PUT (x0, y0), array_name [, {PSET PRESET AND OR XOR}]
<p>Affiche un tableau dans une zone rectangulaire de l'écran graphique. Habituellement, PUT est utilisé avec des tableaux qui ont été stockés à l'aide de GET. Voir GET pour le format du tableau.</p> <p>Les mots clés ont l'effet suivant :</p> <p>PSET Écrase l'emplacement de l'écran avec la nouvelle image</p> <p>PRESET Écrase l'emplacement de l'écran avec l'image inverse</p> <p>AND Combine les anciens et les nouveaux attributs avec bitwise AND</p> <p>OR Combine les anciens et nouveaux attributs avec bitwise OR</p> <p>XOR Combine les anciens et nouveaux attributs avec bitwise XOR</p> <p>Paramètres</p>	

- array\_name est un tableau numérique.
- x0, y0 sont des expressions numériques.

### Les erreurs

- Le tableau n'existe pas : Illegal function call.
- array\_name fait référence à un tableau de chaînes : Type mismatch.
- x0, y0 sont des expressions de chaîne : Type mismatch.
- x0, y0 ne sont pas dans [-32768—32767] : Overflow.
- x0, y0 est en dehors de la VUE ou de la FENÊTRE actuelle : Illegal function call.

### RANDOMIZE

RANDOMIZE [expr]

Initialise le générateur de nombres aléatoires avec expr.

Si aucune graine n'est spécifiée, RANDOMIZE demandera à l'utilisateur d'entrer une graine aléatoire. La valeur fournie par l'utilisateur est arrondie à un nombre entier.

La graine aléatoire est formée des deux derniers octets de cet entier ou expr.

Si expr est un flottant (4 ou 8 octets), ceux-ci sont XOR avec les 2 précédents. Les 4 premiers octets d'un double sont ignorés.

La même graine aléatoire conduira à la même séquence de nombres pseudo-aléatoires générés par la fonction RND.

### Paramètres

- expr est une expression numérique.

### Remarques

- Pour la même graine, PC-BASIC produit les mêmes nombres pseudo-aléatoires que GW-BASIC 3.23.
- Le générateur de nombres aléatoires est très pauvre et ne doit pas être utilisé à des fins sérieuses. Voir RND pour plus de détails.

### Les erreurs

- expr a une valeur de chaîne : Illegal function call.
- L'utilisateur fournit une graine en dehors de [-32768—32767] à l'invite : Overflow.

### READ

READ var\_0 [, var\_1]...

Affecte des données d'une instruction DATA à des variables. La lecture commence à la position DATA actuelle, qui est l'entrée DATA immédiatement après la dernière lue par les instructions READ précédentes. La position DATA est réinitialisée au début par les instructions RUN et RESTORE.

#### Paramètres

- var\_0, var\_1 sont des variables ou des éléments de tableau.

#### Les erreurs

- Il n'y a pas assez de données dans les instructions DATA : Out of DATA.
- Le type de la variable n'est pas compatible avec celui de la donnée en cours de lecture : une erreur de syntaxe se produit sur la ligne DATA.

**REM** ou '

{REM|'} [anything]

Ignore tout jusqu'au bout de la ligne.

La déclaration REM est destinée aux commentaires.

Tout ce qui suit REM sera stocké dans le programme sans modification ni interprétation.

' (apostrophe) est un alias pour :REM' ; il peut être placé à n'importe quel point de la ligne de programme et garantira que le reste de la ligne est ignoré.

Notez qu'un deux-points : ne termine pas l'instruction REM ; les deux-points et tout ce qui suit seront traités comme faisant partie du commentaire.

**RENUM**

RENUM [new|.] [, [old|.] [, increment]]

Remplace les numéros de ligne dans le programme par une énumération systématique commençant par new et augmentant par incrément.

Si old est spécifié, les numéros de ligne inférieurs à ancien restent inchangés. new, old sont des numéros de ligne ; le point. signifie la dernière ligne éditée.

incrément est un numéro de ligne mais ne doit pas être un point ou zéro. Arrête également l'exécution du programme et rend le contrôle à l'utilisateur. Toute instruction supplémentaire sur la ligne sera ignorée, également en mode direct.

#### Remarques

- Les numéros de ligne après les mots clés suivants seront renumérotés : AUTO, EDIT, ELSE, ERL, DELETE,

GOSUB, GOTO, LIST, LLIST, RENUM, RESTORE, RESUME, RETURN, RUN, THEN.

- Aucun numéro de ligne dans les instructions CHAIN ne sera renuméroté ; notez que ces numéros de ligne font référence à un autre programme.
- Tous les arguments des instructions RENUM ou AUTO dans un programme seront renumérotés, y compris tous les décalages ou incréments de numéro de ligne, même si cela n'a pas beaucoup de sens.
- Un numéro de ligne zéro suivant les mots clés ERROR GOTO ne sera pas renuméroté.
- Si un numéro de ligne référencé n'existe pas dans le programme, un message Réf. ligne indéfinie dans old\_line est imprimé. Ici, old\_line est le numéro de ligne avant la renumérotation. Le numéro de ligne référencé restera inchangé, mais l'ancien numéro de ligne de la ligne sera renuméroté.

Les erreurs

- Aucun des paramètres n'est dans [0—65529] : Syntax error.
- L'un des numéros de ligne nouvellement générés est supérieur à 65529 : Illegal function call.  
Les numéros de ligne jusqu'à l'erreur n'ont pas été modifiés.
- l'incrément est vide ou nul : Illegal function call.
- old est spécifié et new est inférieur ou égal à un numéro de ligne existant inférieur à old : Illegal function call.

<b>RESET</b>	RESET	Ferme tous les fichiers ouverts.
--------------	-------	----------------------------------

Remarques

- La documentation officielle GW-BASIC et de nombreuses autres sources indiquent que RESET ferme tous les fichiers sur les périphériques de disque. Cependant, en réalité, GW-BASIC 3.23 ferme également les fichiers sur bande et tout autre périphérique, rendant cette instruction identique à CLOSE sans arguments. PC-BASIC suit ce comportement.

<b>RESTORE</b>	RESTORE [line]	
<p>Réinitialise le pointeur DATA. ligne est un numéro de ligne.</p> <p>Si ligne n'est pas spécifiée, le pointeur DATA est réinitialisé à la première entrée DATA du programme.</p> <p>S'il est spécifié, le pointeur DATA est réinitialisé à la première entrée DATA dans ou après la ligne.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• la ligne n'est pas un numéro de ligne existant : Undefined line number.</li> </ul>		
<b>RESUME</b>	RESUME [0 NEXT line]	
<p>Continue l'exécution normale après une routine de gestion des erreurs.</p> <p>Si 0 ou aucune option n'est spécifiée, réexécute l'instruction à l'origine de l'erreur.</p> <p>Si NEXT est spécifié, exécute l'instruction suivant celle qui a causé l'erreur.</p> <p>Si line est spécifié, il doit s'agir d'un numéro de ligne valide.</p> <p>les erreurs</p> <ul style="list-style-type: none"> <li>• RESUME est rencontré en dehors d'une routine d'interception d'erreurs : RESUME without error.</li> <li>• Le programme se termine dans une routine d'interception d'erreurs sans instruction RESUME ou END : No RESUME.</li> <li>• la ligne n'est pas un numéro de ligne existant : Undefined line number.</li> </ul>		
<b>RETURN</b>	RETURN [line]	
<p>Retourne d'un sous-programme GOSUB.</p> <p>Si la ligne n'est pas spécifiée, RETURN revient à l'instruction après le GOSUB qui a sauté dans le sous-programme. Si line est spécifié, il doit s'agir d'un numéro de ligne valide. RETURN saute à cette ligne (et fait apparaître la pile GOSUB). Lors du retour d'une routine d'interception d'erreurs, RETURN réactive l'interception d'évènements qui a été arrêtée lors de l'entrée dans la routine d'interruption.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• la ligne n'est pas un numéro de ligne existant : Undefined line number.</li> </ul>		



<b>RMDIR</b>	RMDIR dir_spec	
<p>Supprime un répertoire vide sur un périphérique de disque.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• L'expression de chaîne dir_spec est une spécification de fichier valide qui spécifie le chemin et le nom du répertoire.</li> </ul> <p>les erreurs</p> <ul style="list-style-type: none"> <li>• dir_spec a une valeur numérique : Type mismatch.</li> <li>• dir_spec est une chaîne vide : Bad file name.</li> <li>• Aucun chemin correspondant n'a été trouvé : Path not found.</li> <li>• Le répertoire à supprimer n'est pas vide : Path/File access error.</li> </ul>		
<b>RSET</b>	RSET string_name = expression	
<p>Copie une valeur de chaîne dans une variable de chaîne ou un élément de tableau existant. La valeur sera justifiée à droite et tous les caractères restants seront remplacés par des espaces.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• string_name est une variable de chaîne ou un élément de tableau.</li> <li>• expression est une expression de chaîne.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Si expression a une valeur plus longue que la longueur de la variable cible, elle est tronquée à la fin à la longueur de la variable cible.</li> <li>• Si string_name n'a pas été alloué auparavant, cette instruction n'a aucun effet.</li> <li>• Utilisez LSET, RSET ou MID\$ pour copier les valeurs dans un tampon FIELD.</li> <li>• Si LET est utilisé sur une variable FIELD au lieu de L RSET, la variable est détachée du champ et une nouvelle variable de chaîne normale est allouée.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• string_name n'est pas une variable de chaîne : Type mismatch.</li> <li>• expression n'a pas de valeur de chaîne : Type mismatch.</li> </ul>		

<b>RUN</b>	RUN [line_number [anything]]file_spec [, R]
<p>Exécute un programme.</p> <p>Les variables existantes seront effacées. RUN implique CLEAR.</p> <p>Si file_spec est donné, tout programme en mémoire sera effacé.</p> <p>Si R est spécifié après file_spec, les fichiers restent ouverts ; sinon, tous les fichiers sont fermés.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• line_number est un numéro de ligne valide dans le programme en cours. Si spécifié, l'exécution commence à partir de ce numéro de ligne. Le reste de l'instruction RUN est ignoré dans ce cas.</li> <li>• L'expression de chaîne file_spec, si elle est spécifiée, est une spécification de fichier valide indiquant le fichier à partir duquel lire le programme.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• numéro_ligne n'est pas un numéro de ligne dans le programme en cours : Undefined line number.</li> <li>• spéc_fichier introuvable : File not found.</li> <li>• spéc_fichier est une chaîne vide : Bad file number.</li> <li>• Un fichier texte chargé contient des lignes sans numéro de ligne : Direct statement in file.</li> </ul>	
<b>SAVE</b>	SAVE file_spec [, {A P}]
<p>Enregistre le programme en cours dans un fichier.</p> <ul style="list-style-type: none"> <li>• Si ,A est spécifié, le programme sera enregistré au format texte brut. Dans ce cas, l'exécution du programme s'arrêtera et le contrôle sera rendu à l'utilisateur. Toute autre déclaration sur la ligne sera ignorée, également en mode direct.</li> <li>• Si ,P est spécifié, le programme sera enregistré dans un format protégé. Lorsqu'un programme protégé est chargé dans GW-BASIC, il ne peut pas être LISTED ou SAVED dans un format non protégé.</li> <li>• Si aucun n'est spécifié, le programme sera enregistré au format tokenisé.</li> </ul> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• L'expression de chaîne file_spec est une spécification de fichier valide indiquant le fichier dans lequel stocker.</li> </ul> <p><b>Les erreurs</b></p>	

- file\_spec a une valeur numérique : Type mismatch.
- spéc\_fichier est une chaîne vide : Bad file number.
- spéc\_fichier contient des caractères interdits : Bad file number (sur CAS1 : ) ;  
Bad file name (sur les périphériques de disque).
- hide-protected est activé, le programme en cours est protégé et ,P n'est pas spécifié : Illegal function call.

**SCREEN (statement)**

SCREEN [mode] [, [colorburst] [, [apage] [, [vpage] [,  
erase]]]]

Modifiez le mode vidéo, composite colorburst, la page active et la page visible.

Les modes vidéo sont décrits dans la section Modes vidéo.

#### Paramètres

- mode est une expression numérique qui définit le mode d'écran.
- colorburst est une expression numérique. Voir les notes ci-dessous.
- apage est une expression numérique qui définit la page active.
- vpage est une expression numérique qui définit la page visible.
- erase est une expression numérique dans la plage [0, 1, 2]. Ce n'est légal qu'avec syntax={pcjr, tandy}.

Voir les notes ci-dessous.

#### Modes vidéo

Les modes vidéo sont les suivants :

##### ÉCRAN 0 Mode texte

80x25 ou 40x25 caractères de 8x16 pixels

16 attributs choisis parmi 64 couleurs

Les attributs 16-31 sont des versions clignotantes de 0-15

4 pages ega

##### ÉCRAN 1 CGA couleur

320x200 pixels

40x25 caractères de 8x8 pixels

4 attributs choisis parmi 16 couleurs ; 2 bits par pixel

1 page ega 2 pages pcjr tandy

##### ÉCRAN 2 CGA monochrome

640x200 pixels  
80x25 caractères de 8x8 pixels  
2 attributs choisis parmi 16 couleurs ; 1 bit par pixel  
1 page ega 2 pages pcjr tandy

ÉCRAN 3 PCJR tandy 16 couleurs basse résolution  
160x200 pixels  
20x25 caractères de 8x8 pixels  
16 attributs choisis parmi 16 couleurs ; 4 bits par pixel  
2pages

ÉCRAN 3 Hercule monochrome hercule  
720x348 pixels  
80x25 caractères de 9x14 pixels (avec ligne du bas tronquée de 2 px)  
2 attributs ; 1 bit par pixel  
2pages

ÉCRAN 3—255 Altissima risoluzione olivetti  
640x400 pixels  
80x25 caractères de 8x16 pixels  
2 attributs dont un choisi parmi 16 couleurs ; 2 bits par pixel  
1 pages

ÉCRAN 4 Med-res 4 couleurs pcjr tandy  
320x200 pixels  
40x25 caractères de 8x8 pixels  
4 attributs choisis parmi 16 couleurs ; 2 bits par pixel  
2pages

ÉCRAN 5 Med-res 16 couleurs pcjr tandy  
320x200 pixels  
40x25 caractères de 8x8 pixels  
16 attributs choisis parmi 16 couleurs ; 4 bits par pixel  
1 pages

Remarque : un minimum de 32 768 octets de mémoire vidéo doit être réservé pour utiliser ce mode vidéo.  
Utilisez l'instruction CLEAR ,,32768 ! ou l'option video-memory=32768.

ÉCRAN 6 Tandy pcjr 4 couleurs haute résolution  
640x200 pixels  
80x25 caractères de 8x8 pixels  
4 attributs choisis parmi 16 couleurs ; 2 bits par pixel  
1 pages

Remarque : un minimum de 32 768 octets de mémoire vidéo doit être réservé pour utiliser ce mode vidéo.

Utilisez l'instruction CLEAR „,32768 ! ou l'option video-memory=32768.

ÉCRAN 7 EGA couleur ega

320x200 pixels

40x25 caractères de 8x8 pixels

16 attributs choisis parmi 16 couleurs ; 4 bits par pixel

8pages

ÉCRAN 8 EGA couleur ega

640x200 pixels

80x25 caractères de 8x8 pixels

16 attributs choisis parmi 16 couleurs ; 4 bits par pixel

4pages

ÉCRAN 9 EGA couleur ega

640x350 pixels

80x25 caractères de 8x14 pixels

16 attributs choisis parmi 64 couleurs ; 4 bits par pixel

2pages

ÉCRAN 10 EGA monochrome moniteur ega=mono

640x350 pixels

80x25 caractères de 8x14 pixels

4 attributs choisis parmi 9 pseudo-couleurs ; 2 bits par pixel

2pages

Rafale de couleurs composite NTSC

Sur CGA, Tandy et PCjr, colorburst a les effets suivants, selon le type de moniteur - RVB (par défaut) ou composite :

<i>mode</i>	<i>colorburst</i>	<b>CGA mode</b>	<b>Effect (composite)</b>	<b>Effect (RGB)</b>
0	0	0, 2	greyscale	default palette
0	1	1, 3	colour	default palette
1	0	4	colour	default palette
1	1	5	greyscale	alternate palette

Sur SCREEN 2 , colorburst n'a aucun effet ; sur un moniteur composite, les artéfacts de couleur peuvent être activés sur cet écran via OUT (voir ici). Sur SCREEN 3 et plus, colorburst n'a pas effet.

Erase

Par défaut, si le mode change ou si le colorburst passe de zéro à non nul, l'ancienne page et la nouvelle page de

l'écran sont effacées. Sur syntax={pcjr, tandy} , le paramètre erase peut être utilisé pour modifier ce comportement.

Ses valeurs sont les suivantes :

0 N'efface aucune page d'écran

1 (défaut)

Si le mode change ou si la couleur change entre zéro et non zéro, l'ancienne page et la nouvelle page de l'écran sont effacées.

2 Si le mode change ou si la save de couleur passe de zéro à non zéro, toutes les pages de l'écran sont effacées.

Remarques

- Au moins un paramètre doit être spécifié.
- Les artéfacts de couleur composite ne sont émulés que grossièrement dans PC-BASIC, et pas du tout dans SCREEN 1.

### **Les erreurs**

- Aucun paramètre n'est spécifié : Missing operand.
- Tout paramètre a une valeur de chaîne : Type mismatch.
- Tout paramètre n'est pas dans [-32768—32767] : Overflow.
- mode n'est pas un numéro de mode vidéo disponible pour le réglage de votre carte vidéo : Illegal function call.
- vpage , apage ne sont pas compris entre 0 et le nombre de pages pour le mode vidéo choisi moins un : Illegal function call.
- colorburst n'est pas dans [0—255] : Illegal function call.
- erase n'est pas dans [0, 1, 2] : Illegal function call.

<b>SHELL</b>	SHELL [command]	
<p>Démarre un sous-shell du système d'exploitation sur la console. Si command est spécifié, la commande est exécutée sur le shell et l'exécution revient au programme.</p> <p>Pour activer cette instruction, l'option shell doit être définie sur un interpréteur de commandes valide.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• command est une expression de chaîne.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• <b>Soyez prudent lorsque vous activez cette commande, car elle permet au programme BASIC en cours d'exécution un accès complet à vos fichiers et à votre système d'exploitation.</b></li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• l'option shell n'est pas spécifiée : Illegal function call.</li> <li>• la commande a une valeur numérique : Type mismatch.</li> <li>• Toutes les sorties du sous-shell du système d'exploitation, y compris les messages d'erreur, s'affichent sur l'écran PC-BASIC.</li> </ul>		
<b>SOUND (tone)</b>	SOUND frequency, duration [, volume [, voice]]	
<p>Produit un son à la fréquence Hz pendant une durée/18,2 secondes. Sur PCjr et Tandy, le volume et le canal vocal peuvent également être spécifiés.</p> <p>Si PLAY "MB" a été exécuté, SOUND joue en arrière-plan. Si PLAY "MF" a été exécuté, le son joue au premier plan et l'interprète bloque jusqu'à ce que le son soit terminé.</p> <p>Le mode premier plan est par défaut. Contrairement à PLAY, le son joué par l'instruction SOUND la plus récente joue toujours en arrière-plan, même si PLAY "MF" a été entré. En arrière-plan, chaque instruction SOUND compte pour 1 dans la longueur de la file d'attente signalée par la fonction PLAY.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• fréquence est une expression numérique dans [37—32767] ou 0 (pour syntax={advanced   pcjr} ) ou dans [-32768—32767] (pour syntax=tandy ).</li> <li>• duration est une expression numérique dans [0—65535].</li> </ul>		

- volume est une expression numérique entre [-1—15]. 0 est silencieux, 15 est plein volume ; chaque pas en moins réduit le volume de 2 dB. -1 est également à plein volume. (Pour syntax={pcjr | tandy} ).
- voice est une expression numérique entre [0—2] , indiquant lequel des trois canaux vocaux est utilisé pour ce son. (Pour syntax={pcjr | tandy} )

#### Remarques

- Sur PCjr et Tandy, les fréquences inférieures à 110 Hz sont lues comme 110 Hz.
- Si la durée est nulle, tout son de fond actif est arrêté et la file d'attente sonore est vidée.
- Si la durée est de zéro, le volume et la voix ne doivent pas être spécifiés.
- Si la durée est inférieure à 0,022 mais différente de zéro, le son sera joué en arrière-plan et continuera indéfiniment jusqu'à ce qu'une autre instruction sonore soit exécutée. C'est également le comportement pour une durée négative.
- Si fréquence est égale à 32767 ou 0 , un silence de longueur durée est mis en file d'attente.

#### Les erreurs

- Tout argument a une valeur de chaîne : Type mismatch.
- la fréquence n'est pas dans sa plage autorisée et la durée n'est pas nulle : Illegal function call.
- la durée est nulle et plus de deux arguments sont spécifiés : Syntax error.
- syntaxe={ pcjr | tandy } n'est pas défini et plus de deux arguments sont spécifiés : Erreur de syntaxe.
- la fréquence n'est pas dans [-32768—32767] : Overflow.
- la durée n'est pas dans [-65535—65535] : Illegal function call.
- le volume n'est pas dans [0—15] :Illegal function call.
- la voix n'est pas dans [0—2] : Illegal function call.

<b>SOUND (switch)</b>	<b>SOUND {ON OFF}</b>	
-----------------------	-----------------------	--

Active ou désactive le haut-parleur externe et bascule la disponibilité des capacités sonores avancées sur PCjr et Tandy. Cela inclut le son à 3 voix, la génération de bruit et le contrôle du volume. Efface la file d'attente de la musique de fond.

#### Remarques

- Uniquement disponible avec syntaxe={pcjr | tandy}.
- Sur PC-BASIC, les haut-parleurs interne et externe sont émulés via le même système audio.

#### Les erreurs

- Cette instruction est utilisée et syntaxe={ pcjr | tandy } n'est pas défini : Syntax error.



<b>STOP</b>	STOP	
<p>Interrompt l'exécution du programme, imprime un message Break sur la console et rend le contrôle à l'utilisateur. Les fichiers ne sont pas fermés. Il est possible de reprendre l'exécution du programme à l'instruction suivante en utilisant CONT.</p>		
<b>STRIG (switch)</b>	STRIG {ON OFF}	
<p>N'a aucun effet.</p>		
<b>STRIG (event switch)</b>	STRIG[ ](button) {ON OFF STOP}	
<p>Active ou désactive le piégeage d'évènement du bouton de déclenchement du joystick.</p> <p>STRIG (bouton) STOP suspend le piégeage d'évènement jusqu'à ce que STRIG (bouton) ON soit exécuté. Jusqu'à un évènement peut être déclenché pendant la suspension, à condition que la gestion des évènements ait été activée avant la suspension. L'évènement déclenché pendant la suspension est traité immédiatement après la prochaine instruction STRIG (bouton) ON.</p> <p>valeur de retour du bouton</p> <p>0 1ère manette 1ère gâchette  2 2ème manette 1ère gâchette  4 1ère manette 2ème gâchette  6 2ème manette 2ème gâchette</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• button est une expression numérique dans [0, 2, 4, 6].</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• button a une valeur de chaîne : Type mismatch.</li> <li>• le bouton n'est pas dans [-32768—32767] : Overflow.</li> <li>• le bouton n'est pas dans [0, 2, 4, 6] : Illegal function call.</li> </ul>		

<b>SWAP</b>	SWAP var_0, var_1	Échange les variables var_0 et var_1.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• Les variables sont échangées par référence. Si, par exemple, var_0 est une variable FIELD et var_1 ne l'est pas, alors SWAP inversera ces rôles.</li> </ul> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• var_0 et var_1 sont des variables ou des éléments de tableau du même type. var_1 doit avoir été défini au préalable.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• var_1 n'est pas défini : Illegal function call.</li> </ul> <p>Notez qu'aucune erreur n'est générée si var_0 n'est pas défini, et qu'après cette erreur, les deux variables seront définies.</p> <ul style="list-style-type: none"> <li>• Les types de var_0 et var_1 ne sont pas identiques : Type mismatch.</li> </ul>		
<b>SYSTEM</b>	SYSTEM	Quitte l'interpréteur.
<p>Remarques</p> <ul style="list-style-type: none"> <li>• SYSTEM quitte immédiatement l'interpréteur PC-BASIC sans autre interaction. Tout programme ou donnée non enregistré sera perdu.</li> </ul>		
<b>TERM</b>	TERM	
<p>Charge et exécute le programme défini par l'option terme. Par défaut, comme sur IBM PCjr, il s'agit d'une application d'émulation de terminal série intégrée. Cette instruction n'est disponible qu'avec syntax={pcjr tandy}.</p> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Si term n'est pas défini, cette instruction génère une erreur interne.</li> <li>• Si la syntaxe n'est pas définie sur pcjr ou tandy , ce mot-clé n'est pas présent. L'appel de TERM générera une erreur de syntaxe.</li> </ul>		

<b>TIME\$ (statement)</b>	TIME\$ = time	
<p>Règle l'heure BASIC actuelle sur l'heure.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• Time est une expression de chaîne de la forme "HH{:.}mm{:.}ss" où <math>0 \leq \text{HH} &lt; 24</math>, <math>0 \leq \text{mm} &lt; 60</math> et <math>0 \leq \text{ss} &lt; 60</math>. Chaque position peut avoir un ou deux caractères.</li> </ul> <p>Remarques</p> <ul style="list-style-type: none"> <li>• PC-BASIC stocke un décalage par rapport à l'heure système et l'utilise pour les futurs appels aux fonctions TIME\$ et DATE\$ dans la même session d'interpréteur. L'heure système n'est pas modifiée, contrairement à GW-BASIC sous MS-DOS.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• time a une valeur numérique : Type mismatch.</li> <li>• l'heure n'est pas de la forme correcte : Illegal function call.</li> </ul>		
<b>TIMER (statement)</b>	TIMER {ON OFF STOP}	
<ul style="list-style-type: none"> <li>• ON : active le piégeage d'événement ON TIMER de l'horloge de la minuterie.</li> <li>• OFF : désactive le piégeage.</li> <li>• STOP : arrête le piégeage jusqu'à ce que TIMER ON soit utilisé. Les événements qui se produisent alors que le piégeage est arrêté se déclenchent immédiatement lorsque le piégeage est réactivé.</li> </ul>		
TRON and TROFF	{TRON TROFF}	Active ou désactive le suivi des numéros de ligne.
<p>Si le suivi des numéros de ligne est activé, BASIC imprime une balise [100] sur la console lorsque la ligne de programme 100 est exécutée, et ainsi de suite.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Le traçage est désactivé par les instructions NEW et LOAD.</li> </ul>		

<b>UNLOCK</b>	UNLOCK [#] file_number [, record_0]  UNLOCK [#] file_number, [record_0] TO record_1
<p>Déverrouille un fichier ou une partie de celui-ci qui a été précédemment verrouillé avec LOCK.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• file_number est une expression numérique dans [0—255].</li> <li>• record_0 et record_1 sont des expressions numériques dans [1—2^25-2].</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Tout paramètre a une valeur de chaîne : Type mismatch.</li> <li>• file_number n'est pas dans [-32768—32767] : Overflow.</li> <li>• file_number n'est pas dans [0—255] : Illegal function call.</li> <li>• numéro_fichier n'est pas un fichier ouvert : Bad file number.</li> <li>• Si numéro_fichier est ouvert pour RANDOM, les instructions LOCK et UNLOCK doivent correspondre en termes d'enregistrement_0 et d'enregistrement_1. Un UNLOCK non correspondant déclenchera une autorisation refusée.</li> <li>• record_0 ou record_1 n'est pas dans [1—2^25-2] :Bad record number.</li> </ul>	
<b>VIEW</b>	VIEW [[SCREEN] (x0, y0)-(x1, y1) [, [fill] [, border]]]
<p>Définit une fenêtre graphique. Les graphiques dessinés en dehors de la fenêtre ne seront pas affichés. (x0,y0) , (x1, y1) sont les coordonnées écran absolues de deux coins opposés de la zone.</p> <p>À moins que SCREEN ne soit spécifié, après une instruction VIEW, le système de coordonnées est décalé de sorte que (0, 0) devienne la coordonnée supérieure gauche de la fenêtre.</p> <p>Si VIEW est appelé sans arguments, la fenêtre est réinitialisée sur tout l'écran.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• fill est un attribut. La fenêtre sera remplie avec cet attribut.</li> <li>• border est un attribut. Une bordure sera dessinée juste à l'extérieur de la fenêtre avec cet attribut.</li> </ul> <p><b>Les erreurs</b></p>	

- N'importe lequel des paramètres a une valeur de chaîne : Type mismatch.
- L'une des coordonnées n'est pas dans [-32768—32767] : Overflow.
- L'une des paires de coordonnées se trouve en dehors de l'écran physique : Illegal function call.

## VIEW PRINT

VIEW PRINT top\_row TO bottom\_row

Définit la zone de défilement du texte de l'écran. Les instructions LOCATE, le mouvement du curseur et le défilement seront limités à la zone de défilement.

### Paramètres

- top\_row et bottom\_row sont des expressions numériques dans [1—24].

### Remarques

- Si syntaxe={pcjr | tandy} et KEY OFF est défini, bottom\_row peut être 25. Sinon, la ligne d'écran 25 ne peut pas faire partie de la zone de défilement.

### Les erreurs

- top\_row ou bottom\_row n'est pas dans [1—24] : Illegal function call.

## WAIT

WAIT port, and\_mask [, xor\_mask]

Attend que la valeur de (INP(port) XOR xor\_mask) AND and\_mask devienne non nulle. La gestion des évènements est suspendue jusqu'au retour de WAIT.

Si xor\_mask n'est pas spécifié, sa valeur par défaut est 0.

### Remarques

- Un nombre limité de ports machine sont émulsés dans PC-BASIC. Voir INP.

### Les erreurs

- Tout paramètre a une valeur de chaîne : Type mismatch.
- le port n'est pas dans [-32768—65535] : Overflow.
- and\_mask ou xor\_mask ne sont pas dans [0—255] : Type mismatch.

<b>WEND</b>	WEND	
<p>Itère une boucle WHILE—WEND : saute à l'instruction WHILE correspondante, où sa condition peut être vérifiée.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Les boucles WHILE—WEND peuvent être imbriquées. WEND saute à l'instruction WHILE la plus récente qui n'a pas été fermée par un autre WEND.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Toutes les instructions WHILE précédentes ont été fermées par un autre WEND ou aucune instruction WHILE n'a été exécutée auparavant : WEND without WHILE.</li> </ul>		
<b>WHILE</b>	WHILE expr	Initie une boucle WHILE—WEND.
<p>Si expr vaut zéro, WHILE saute à l'instruction immédiatement après le WEND correspondant. Sinon, l'exécution continue.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• expr est une expression numérique.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• Aucun WEND correspondant n'est trouvé : WHILE sans WEND.</li> <li>• expr a une valeur de chaîne : Type mismatch.</li> </ul>		
<b>WIDTH (console)</b>	WIDTH num_columns [, [num_rows] [,]]	
<p>Définit la largeur de l'écran sur 20, 40 ou 80 colonnes.</p> <p>Remarques</p> <ul style="list-style-type: none"> <li>• Lorsque vous modifiez la largeur de l'écran en mode graphique, le mode vidéo est modifié.</li> </ul> <p>Les modifications suivantes se produisent :</p> <p>ÉCRAN 1 (40) ←→ ÉCRAN 2 (80)  ÉCRAN 7 (40) ←→ ÉCRAN 8 (80)  ÉCRAN 7 (40) ← ÉCRAN 9 (80)</p> <ul style="list-style-type: none"> <li>• La valeur de largeur d'écran 20 n'est autorisée que sur Tandy et PCjr. Le passage à cette largeur passe à</li> </ul>		

## SCREEN 3.

En outre, les modifications suivantes se produisent :

ÉCRAN 3 (20) → ÉCRAN 1 (40)

ÉCRAN 3 (20) → ÉCRAN 2 (80)

ÉCRAN 4 (40) → ÉCRAN 2 (80)

ÉCRAN 5 (40) ↔ ÉCRAN 6 (80)

### Paramètres

- num\_columns est soit un littéral 20 , 40 ou 80 soit une expression numérique entre parenthèses. La virgule finale est facultative et n'a aucun effet.
- num\_rows est facultatif et doit être égal à 25. Si syntax={pcjr | tandy} est défini, num\_rows peut être compris entre [0—25] mais sa valeur est ignorée.

### Les erreurs

- num\_columns est une expression de chaîne : Type mismatch.
- num\_columns n'est pas dans [-32768—32767] : Overflow.
- num\_columns n'est pas dans [0—255] : Illegal function call.
- num\_columns n'est pas un littéral et n'est pas entre crochets : Illegal function call.
- num\_rows n'est pas dans sa plage acceptée : Illegal function call.

### **WIDTH** **(devices and files)**

WIDTH {#file\_num,|device\_name,|LPRINT} num\_columns

Définit la largeur de ligne d'un fichier ou d'un périphérique. Lorsqu'une opération d'écriture dépasse la largeur de colonne, une séquence CR LF est insérée.

Si un périphérique est spécifié, il n'est pas nécessaire qu'un fichier lui soit ouvert ; le paramètre de largeur sera la largeur par défaut la prochaine fois qu'un fichier sera ouvert sur ce périphérique.

Si device\_name est "LPT1:" ou si LPRINT est spécifié, le paramètre de largeur de périphérique affecte LPRINT et LLIST.

Si device\_name est "SCRN:" , "KYBD:" , ou omis, la largeur de l'écran est modifiée. Dans ce cas, num\_columns doit être l'un des 20, 40 ou 80. Voir les notes sur WIDTH (console) pour les effets secondaires.

### Paramètres

- file\_num est une expression numérique qui correspond au numéro d'un fichier ouvert.

- nom\_périphérique est une expression de chaîne parmi "KYBD :", "SCRN :", "LPT1 :", "LPT2 :", "LPT3 :", "COM1 :", "COM2 :", "CAS1 :"
- num\_columns est une expression numérique.

### Les erreurs

- nom\_périphérique n'est pas l'un des périphériques autorisés : Bad file name.
- nom\_périphérique est "SCRN :", "KYBD :" et num\_columns n'est pas 20, 40 ou 80 : Illegal function call.
- file\_num ou num\_columns sont des chaînes : Type mismatch.
- file\_num ou num\_columns ne sont pas dans [-32768—32767] : Overflow.
- file\_num ou num\_columns ne sont pas dans [0—255] : Illegal function call.
- num\_fichier n'est pas un fichier ouvert : Bad file mode.

### WINDOW

WINDOW [[SCREEN] (x0, y0)-(x1, y1)]

Définisse les coordonnées logiques de la fenêtre.

Si SCREEN n'est pas spécifié, le coin inférieur gauche de l'écran est mappé aux coordonnées inférieures ; le coin supérieur droit de l'écran correspond aux coordonnées supérieures.

Si SCREEN est spécifié, le coin supérieur gauche de l'écran est mappé aux coordonnées inférieures ; le coin inférieur droit de l'écran correspond aux coordonnées supérieures.

Si WINDOW est appelée sans arguments, les coordonnées logiques sont réinitialisées aux coordonnées de la fenêtre.

#### Paramètres

- x0 , y0 , x1 , y1 sont des expressions numériques.

### Les erreurs

- Toutes les coordonnées ont une valeur de chaîne : Type mismatch.
- x0 = x1 ou y0 = y1 : Illegal function call.



<b>WRITE</b>	WRITE [# file_num,] [expr_0 [{,;} expr_1]... ]
<p>Écrit des valeurs dans un fichier ou sur l'écran sous une forme lisible par machine. Les valeurs sont séparées par des virgules et la ligne se termine par une séquence CR LF. Les chaînes sont délimitées par des guillemets doubles ". Aucun espace de remplissage n'est inséré.</p> <p>Lors de l'écriture à l'écran, les mêmes caractères de contrôle sont reconnus que pour l'instruction PRINT.</p> <p>Paramètres</p> <ul style="list-style-type: none"> <li>• expr_0, expr_1,... sont des expressions dont la valeur doit être imprimée.</li> </ul> <p><b>Les erreurs</b></p> <ul style="list-style-type: none"> <li>• file_num a une valeur de chaîne : Type mismatch.</li> <li>• num_fichier est ouvert pour INPUT : Bad file mode.</li> </ul>	

## Annexe des erreurs

### 1 NEXT without FOR

Une instruction NEXT a été rencontrée pour laquelle aucun FOR correspondant n'a pu être trouvé.

### 2 Syntax error

La syntaxe BASIC est incorrecte. Une instruction ou une expression a été mal saisie ou appelée de l'une des nombreuses manières incorrectes. Cette erreur est également déclenchée sur une ligne DATA si une instruction READ qui rencontre une entrée de données d'un format incorrect.

### 3 RETURN without GOSUB

Une instruction RETURN a été rencontrée pour laquelle aucun appel GOSUB n'a été effectué.

### 4 Out of DATA

Une instruction READ tente de lire plus d'entrées de données qu'il n'est possible d'en trouver à partir de l'emplacement DATA actuel.

### 5 Illegal function call

Une instruction, une fonction ou un opérateur a été appelé avec des paramètres en dehors de la plage acceptée. Cette erreur est également générée pour une grande variété d'autres conditions - vérifiez la référence de l'instruction ou de la fonction appelée.

## 6 Overflow

Un résultat d'expression numérique ou une valeur intermédiaire est trop grand pour le format numérique requis.

## 7 Out of memory

Il n'y a pas assez de mémoire BASIC libre pour terminer l'opération. Trop de mémoire est consommée par le programme ; variables, tableaux et chaînes, ou piles d'exécution pour les boucles, les sous-programmes ou les fonctions définies par l'utilisateur.

## 8 Undefined line number

Une référence est faite à un numéro de ligne qui n'existe pas dans le programme.

## 9 Subscript out of range

Un index de tableau (indice) est utilisé en dehors de la plage réservée pour ce tableau par l'instruction DIM.

## 10 Duplicate Definition

Une instruction DIM est utilisée sur un tableau qui a été dimensionné auparavant (implicitement ou explicitement) ou OPTION BASE est appelée d'une manière qui entre en conflit avec une définition implicite ou explicite antérieure de l'index de départ.

## 11 Division by zero

Une tentative est faite pour diviser un nombre par zéro ou par un nombre trop petit pour être distingué de zéro dans la précision du format numérique.

## 12 Illegal direct

Une instruction DEF FN est utilisée en mode direct.

## 13 Type mismatch

L'expression utilisée est d'un type qui ne peut pas être converti dans le type requis pour la fonction ou l'instruction. Le plus souvent, ceci est déclenché si un argument de chaîne est fourni à une instruction ou une fonction qui attend un nombre, ou vice versa.

## 14 Out of string space

Il n'y a pas assez de mémoire BASIC libre pour stocker la variable de chaîne.

## 15 String too long

Un résultat d'expression de chaîne ou une valeur intermédiaire comporte plus de 255 caractères.

## 16 String formula too complex

## 17 Can't continue

L'instruction CONT est utilisée dans des circonstances où la poursuite de l'exécution du programme n'est pas possible.

## 18 Undefined user function

La fonction FN est appelée avec un nom de fonction pour lequel aucune définition n'a été faite par une instruction DEF FN.

## 19 No RESUME

Le programme se termine à l'intérieur d'une routine d'interception d'erreurs qui n'a pas été fermée avec RESUME ou END.

## 20 RESUME without error

Une instruction RESUME est rencontrée alors que le programme n'exécute pas de routine d'interception d'erreurs.

## 21 inutilisés

## 22 Missing operand

Une expression d'opérateur manque un opérande ou une fonction ou une instruction n'est pas fournie avec des paramètres suffisants.

## 23 Line buffer overflow

Une instruction INPUT ou LINE INPUT a rencontré une chaîne d'entrée de plus de 255 caractères ou le fichier de programme en texte brut chargé par LOAD, CHAIN ou MERGE contient une ligne de plus de 255 caractères. La tentative de chargement d'un fichier texte contenant des fins de ligne LF plutôt que CR LF peut provoquer cette erreur.

## 24 Device Timeout

L'établissement de liaison a échoué sur un périphérique série ou un périphérique à bande a atteint la fin de la bande.

## 25 Device Fault

## 26 FOR without NEXT

Une instruction FOR a été rencontrée pour laquelle aucune instruction NEXT correspondante n'a été trouvée.

## 27 Out of paper

Tentative d'écriture sur une imprimante qui n'a plus de papier ou sur un autre périphérique parallèle qui a déclenché une condition de manque de papier.

## 28 inutilisés

## 29 WHILE without WEND

Une instruction WHILE a été rencontrée pour laquelle aucune instruction WEND correspondante n'a été trouvée.

## 30 WEND without WHILE

Une instruction WEND a été rencontrée pour laquelle aucune instruction WHILE correspondante n'a été trouvée.

## 31—49 inutilisé

## 50 FIELD overflow

Tentative de lecture, d'écriture ou de définition d'une variable FIELD au-delà de la longueur du tampon de fichier à accès aléatoire.

## 51 Internal error

L'instruction TERM est exécutée mais aucun programme de gestionnaire de terminal n'a été défini.

## 52 Bad file number

Un numéro de fichier auquel aucun fichier n'est ouvert est accédé, ou le numéro de fichier utilisé dans une instruction OPEN est en dehors de la plage des numéros de fichier autorisés, ou (ce qui prête à confusion) la spécification de fichier est vide, mal formée ou contient des caractères illégaux.

## 53 File not found

Un fichier nommé sur un périphérique de disque est introuvable.

## 54 Bad file mode

Le mode de fichier demandé dans une instruction OPEN n'existe pas ou n'est pas pris en charge pour le périphérique donné, ou la fonction de fichier appelée n'est pas prise en charge pour ce périphérique, ou la fonction ou l'instruction appelée nécessite un fichier ouvert pour RANDOM et le fichier ne l'est pas.

## 55 File already open

Tentative d'ouverture d'un fichier vers un numéro de fichier déjà utilisé ; ou une tentative est faite pour ouvrir un fichier pour OUTPUT ou APPEND sur un périphérique série, disque ou cassette lorsqu'un fichier (ou, sur un périphérique disque, un fichier portant le même nom) est déjà ouvert pour OUTPUT ou APPEND sur ce périphérique ; ou une instruction KILL ou NAME est exécutée sur un fichier disque lorsqu'un fichier portant le même nom est ouvert sur le même périphérique.

## 56 inutilisés

#### 57 Device I/O error

Une erreur d'E/S s'est produite lors de l'entrée/sortie d'un périphérique. Cela inclut les erreurs de familiarisation, les échecs de vérification CRC et les fins de bande inattendues sur les périphériques de cassette.

#### 58 File already exists

Le nouveau nom proposé pour un fichier disque dans une instruction NAME est déjà utilisé.

#### 59—60 inutilisé

#### 61 Disk full

L'espace libre sur le périphérique de disque est insuffisant pour terminer l'opération.

#### 62 Input past end

Une tentative est faite pour récupérer l'entrée d'un fichier qui a dépassé sa fin de fichier.

#### 63 Bad record number

Un numéro d'enregistrement de fichier à accès aléatoire est référencé en dehors de la plage autorisée.

#### 64 Bad file name

Le nom de fichier ou une autre chaîne de paramètres de périphérique dans une spécification de fichier est incorrect ou contient des caractères non autorisés.

#### 65 inutilisé

#### 66 Direct statement in file

Une ligne sans numéro de ligne est rencontrée dans un fichier de programme en texte brut.

#### 67 Too many files

#### 68 Device Unavailable

Tentative d'accès à un périphérique qui n'existe pas ou qui n'est pas activé.

#### 69 Communication buffer overflow

Un périphérique série reçoit plus de données que sa mémoire tampon n'en contient.

#### 70 Permission Denied

L'accès demandé à un fichier n'est pas accordé en raison de restrictions LOCK, d'un verrouillage du système d'exploitation ou d'autorisations de fichier du système d'exploitation insuffisantes.

## 71 Disk not Ready

Le périphérique de disque n'est pas prêt pour l'accès. Par exemple, il n'y a pas de disquette dans un lecteur de disquette ou le verrou du lecteur est ouvert.

## 72 Disk media error

## 73 Advanced Feature

## 74 Rename across disks

Tentative d'utilisation de l'instruction NAME pour déplacer un fichier d'un périphérique de disque à un autre.

## 75 Path/File access error

Tentative de création d'un répertoire qui existe déjà ou de suppression d'un répertoire qui n'est pas vide.

## 76 Path not found

Une instruction OPEN, MKDIR, RMDIR ou CHDIR est exécutée en référence à un chemin (parent) qui n'existe pas sur le périphérique de disque.

## 77 Deadlock

Tout code d'erreur auquel aucun message n'est associé générera le message Erreur non imprimable.

Si une erreur se produit en mode direct, le message d'erreur est imprimé comme ci-dessus.

Si l'erreur se produit dans un programme, le message est complété par le numéro de la ligne dans laquelle l'erreur s'est produite.

Par exemple, Appel de fonction illégal dans 100

indique que l'appel de fonction illégal a eu lieu dans la ligne numéro 100.

Si une erreur de syntaxe se produit pendant l'exécution du programme, le message d'erreur est suivi d'une liste de la ligne de programme dans laquelle l'erreur s'est produite, avec le curseur positionné à l'endroit où l'erreur a été déclenchée.

Une erreur de division par zéro ou, dans un calcul en virgule flottante, un débordement, n'interrompra pas l'exécution à moins qu'elle ne se produise dans une routine de gestion des erreurs. Le message d'erreur sera imprimé sur la console et le résultat du calcul incriminé sera considéré comme la valeur maximale qui tient dans la variable à virgule flottante appropriée. Le débordement dans un calcul d'entier interrompra toujours l'exécution comme les autres erreurs.

## Autres messages

### Break

L'exécution d'une instruction composée ou d'un programme a été interrompue par une instruction CONT ou par une interruption du clavier de l'utilisateur (telle que Ctrl + Break ).

Si l'interruption se produit dans un programme, le message Break sera complété par le numéro de la ligne dans laquelle l'interruption s'est produite.

### ?Redo from start

L'entrée fournie sur la console pour une instruction INPUT ne correspond pas au format attendu. Le nombre ou le type d'entrées n'est pas correct. Entrez à nouveau toutes les entrées.

### Undefined line ref\_num in line\_num

L'instruction RENUM a rencontré une référence au numéro de ligne ref\_num qui n'est pas défini dans le programme. La référence se produit sur la ligne numéro line\_num. La référence de numéro de ligne indéfinie ne sera pas renumérotée.

### filename Found.

Un fichier correspondant à la spécification demandée a été trouvé sur le périphérique de cassette. Ce message n'apparaît qu'en mode direct.

### filename Skipped.

Un fichier ne correspondant pas à la spécification demandée a été rencontré sur le périphérique de cassette. Ce message n'apparaît qu'en mode direct.

## Référence technique

### Format de fichiers tokenisés

Un fichier de programme à jetons sur un périphérique de disque a le format suivant.

Octet magique FF

Lignes de programme

Chaque ligne est stockée comme suit :

Octets	Format	Signification
2	Entier little-endian 16 bits non signé.	Emplacement mémoire de la ligne suivant la ligne courante. Ceci est utilisé en interne par GW-BASIC mais ignoré lorsqu'un programme est chargé.
2	Entier little-endian 16 bits non signé.	Le numéro de ligne.
Variable	BASIC tokenisé, voir ci-dessous.	Le contenu de la ligne.
1	00 (octet NUL)	Marqueur de fin de ligne.

Marqueur de fin de fichier

Un 1A est écrit pour marquer la fin du fichier. Ceci est facultatif ; le fichier sera lu sans problème s'il est omis.

### BASIC tokenisé

Les caractères ASCII imprimables compris entre 20 et 7E sont utilisés pour les littéraux de chaîne, les commentaires, les noms de variable et les éléments de syntaxe d'instruction qui ne sont pas des mots réservés. Les mots réservés sont représentés par leurs jetons de mots réservés et les littéraux numériques sont représentés par des séquences de jetons numériques.



## Séquences de jetons numériques

Les littéraux numériques sont stockés dans des programmes tokenisés selon la représentation suivante.

Tous les nombres sont positifs ; les nombres négatifs sont stockés simplement en faisant précéder le nombre de EA , le jeton pour -.

Classe	Octets	Format
Numéros de ligne indirects	3	0E suivi d'un entier little-endian 16 bits non signé.
Entiers octaux	3	0B suivi d'un entier little-endian 16 bits non signé.
Entiers hexadécimaux	3	0C suivi d'un entier little-endian 16 bits non signé.
Entiers décimaux positifs inférieurs à 11	1	Les jetons 11—1B représentent 0—10.
Entiers décimaux positifs inférieurs à 256	2	0F suivi d'un entier 8 bits non signé.
Autres entiers décimaux	3	1C suivi d'un complément à deux signé 16 bits little-endian.  GW-BASIC reconnaîtra un nombre négatif rencontré de cette façon mais il ne stockera pas lui-même les nombres négatifs en utilisant le complément à deux, mais plutôt en faisant précéder le nombre positif avec EA.
Nombre à virgule flottante simple précision	5	1D suivi d'un single de quatre octets au format binaire Microsoft.
Nombre à virgule flottante double précision	9	1F suivi d'un double de huit octets au format binaire Microsoft.

Jetons de mot-clé La plupart des mots-clés dans PC-BASIC sont des mots réservés. Les mots réservés sont représentés dans un programme tokenisé par un jeton à un ou deux octets.

La liste complète est ci-dessous. Tous les noms de fonctions et opérateurs sont des mots réservés et toutes les instructions commencent par un mot réservé (qui dans le cas de LET est facultatif). Cependant, l'inverse n'est pas vrai : tous les mots réservés ne sont pas des instructions, des fonctions ou des opérateurs.

Par exemple, TO et SPC( n'apparaissent que dans le cadre d'une syntaxe d'instruction. De plus, certains mots clés qui font partie de la syntaxe d'une instruction ne sont pas des mots réservés : par exemple, AS , BASE et ACCESS.

Les mots-clés qui ne sont pas des mots réservés sont épelés en texte intégral dans la source tokenisée. Un nom de variable ou de fonction définie par l'utilisateur ne doit pas être identique à un mot réservé. La liste ci-dessous est une liste exhaustive de mots réservés qui peuvent être utilisés pour déterminer si un nom est légal.

81 END	82 FOR	83 NEXT
84 DATA	85 INPUT	86 DIM
87 READ	88 LET	89 GOTO
8A RUN	8B IF	8C RESTORE
8D GOSUB	8E RETURN	8F REM
90 STOP	91 PRINT	92 CLEAR
93 LIST	94 NEW	95 ON
96 WAIT	97 DEF	98 POKE
99 CONT	9C OUT	9D LPRINT
9E LLIST	A0 WIDTH	A1 ELSE
A2 TRON	A3 TROFF	A4 SWAP
A5 ERASE	A6 EDIT	A7 ERROR
A8 RESUME	A9 DELETE	AA AUTO
AB RENUM	AC DEFSTR	AD DEFINT
AE DEFSNG	AF DEFDBL	B0 LINE
B1 WHILE	B2 WEND	B3 CALL
B7 WRITE	B8 OPTION	B9 RANDOMIZE
BA OPEN	BB CLOSE	BC LOAD
BD MERGE	BE SAVE	BF COLOR
C0 CLS	C1 MOTOR	C2 BSAVE
C3 BLOAD	C4 SOUND	C5 BEEP
C6 PSET	C7 PRESET	C8 SCREEN
C9 KEY	CA LOCATE	CC TO
CD THEN	CE TAB(	CF STEP
D0 USR	D1 FN	D2 SPC(
D3 NOT	D4 ERL	D5 ERR
D6 STRING\$	D7 USING	D8 INSTR
D9 '	DA VARPTR	DB CSRLIN
DC POINT	DD OFF	DE INKEY\$
E6 >	E7 =	E8 <
E9 +	EA -	EB *
EC /	ED ^	EE AND
EF OR	F0 XOR	F1 EQV
F2 IMP	F3 MOD	F4 \
FD81 CVI	FD82 CVS	FD83 CVD
FD84 MKI\$	FD85 MKS\$	FD86 MKD\$
FD8B EXTERR	FE81 FILES	FE82 FIELD
FE83 SYSTEM	FE84 NAME	FE85 LSET
FE86 RSET	FE87 KILL	FE88 PUT
FE89 GET	FE8A RESET	FE8B COMMON

FE8C CHAIN	FE8D DATE\$	FE8E TIME\$
FE8F PAINT	FE90 COM	FE91 CIRCLE
FE92 DRAW	FE93 PLAY	FE94 TIMER
FE95 ERDEV	FE96 IOCTL	FE97 CHDIR
FE98 MKDIR	FE99 RMDIR	FE9A SHELL
FE9B ENVIRON	FE9C VIEW	FE9D WINDOW
FE9E PMAP	FE9F PALETTE	FEA0 LCOPY
FEA1 CALLS	FEA5 PCOPY	FEA7 LOCK
FEA8 UNLOCK	FF81 LEFT\$	FF82 RIGHT\$
FF83 MID\$	FF84 SGN	FF85 INT
FF86 ABS	FF87 SQR	FF88 RND
FF89 SIN	FF8A LOG	FF8B EXP
FF8C COS	FF8D TAN	FF8E ATN
FF8F FRE	FF90 INP	FF91 POS
FF92 LEN	FF93 STR\$	FF94 VAL
FF95 ASC	FF96 CHR\$	FF97 PEEK
FF98 SPACE\$	FF99 OCT\$	FF9B LPOS
FF9A HEX\$	FF9C CINT	FF9D CSNG
FF9E CDBL	FF9F FIX	FFA0 PEN
FFA1 STICK	FFA2 STRIG	FFA3 EOF
FFA4 LOC	FFA5 LOF	

Les mots réservés supplémentaires suivants sont activés par l'option `syntax={pcjr|tandy}`.

FEA4 NOISE    FEA6 TERM

Jetons à usage interne Les jetons 10, 1E et 0D sont connus pour être utilisés en interne par GW-BASIC. Ils ne doivent pas apparaître dans un fichier de programme à jetons correctement stocké.

### Format binaire Microsoft

Les nombres à virgule flottante dans GW-BASIC et PC-BASIC sont représentés au format binaire Microsoft (MBF), qui diffère de la norme IEEE 754 utilisée par pratiquement tous les logiciels et matériels modernes.

Par conséquent, les fichiers binaires générés par l'un ou l'autre BASIC sont entièrement compatibles entre eux et avec certaines applications contemporaines de GW-BASIC, mais ne sont pas facilement interchangeables avec d'autres logiciels. QBASIC, par exemple, utilise des flottants IEEE.

MBF diffère de l'IEEE par la position du bit de signe et par l'utilisation de seulement 8 bits pour l'exposant, à la fois en simple et en double précision. Cela rend la plage de nombres autorisés dans un nombre à double précision MBF plus petite, mais leur précision plus élevée, que pour un double IEEE :

un simple MBF a 23 bits de précision, tandis qu'un double MBF a 55 bits de précision. Les deux ont la même portée.

Contrairement à IEEE, le format binaire Microsoft ne prend pas en charge les zéros signés, les nombres inférieurs à la normale, les infinis ou les valeurs non numériques.

Les nombres à virgule flottante MBF sont représentés en octets comme suit :

Simple M3 M2 M1 E0

Double M7 M6 M5 M4 M3 M2 M1 E0

Ici, E0 est l'octet exposant et les autres octets forment la mantisse, dans l'ordre little-endian de sorte que M1 est l'octet le plus significatif.

Le bit le plus significatif de M1 est le bit de signe, suivi des bits les plus significatifs de la mantisse :

M1 = s0 f1 f2 f3 f4 f5 f6 f7.

Les autres octets contiennent les bits de mantisse les moins significatifs :

M2 = f8 f9 fA fB fC fD fE fF, et ainsi de suite.

La valeur du nombre à virgule flottante est  $v = 0$  si  $E0 = 0$  et  $v = (-1)^{s0} \times \text{mantisse} \times 2^{E0 - 128}$  sinon, où la mantisse est formée comme une fraction binaire  $\text{mantisse} = 0.1f1f2f3\dots$

### **Format de fichier protégé**

Le format protégé est une forme cryptée du format tokenisé. GW-BASIC refusera d'afficher le code source de ces fichiers. Ce schéma de protection pourrait facilement être contourné en changeant un indicateur en mémoire. Les programmes de déprotection ont largement circulé pendant des décennies et l'algorithme et les clés de décryptage ont été publiés dans un magazine mathématique.

Un fichier de programme protégé sur un périphérique de disque a le format suivant.

Octet magique  
FE

Charge utile

Contenu chiffré d'un fichier de programme à jeton, y compris son marqueur de fin de fichier mais à l'exclusion de son octet magique. Le chiffrement de chiffrement tourne à travers une clé de 11 octets et une clé de 13 octets afin que la transformation résultante soit la même après 143 octets.

Pour chaque octet

- Soustraire l'octet correspondant de la séquence de 11 octets  
0B 0A 09 08 07 06 05 04 03 02 01
- Exclusif-ou avec l'octet correspondant de la clé de 11 octets  
1E 1D C4 77 26 97 E0 74 59 88 7C
- Exclusif-ou avec l'octet correspondant de la clé de 13 octets  
A9 84 8D CD 75 83 43 63 24 83 19 F7 9A
- Ajouter l'octet correspondant de la séquence de 13 octets  
0D 0C 0B 0A 09 08 07 06 05 04 03 02 01

## Marqueur de fin de fichier

Un 1A est écrit pour marquer la fin du fichier. Ceci est facultatif ; le fichier sera lu sans problème s'il est omis. Étant donné que le marqueur de fin de fichier du programme tokenisé est inclus dans le contenu crypté, un fichier protégé est généralement plus long d'un octet que son équivalent non protégé.

## Format de fichier BSAVE

Un fichier de vidage de mémoire sur un périphérique de disque a le format suivant.

Octet magique

DF

Entête

Octets	Format	Signification
2	Entier Little-endian 16 bits non signé.	Segment du bloc mémoire.
2	Entier Little-endian 16 bits non signé.	Décalage du premier octet du bloc mémoire.
2	Entier Little-endian 16 bits non signé.	Longueur du bloc mémoire en octets.

Payload, Les octets du bloc mémoire.

Bas de page, Sur Tandy uniquement, l'octet magique et les six octets de l'entête sont répétés ici.

Ceci est facultatif ; le fichier sera lu sans problème s'il est omis.

## Marqueur de fin de fichier

Un 1A est écrit pour marquer la fin du fichier. Ceci est facultatif ; le fichier sera lu sans problème s'il est omis.

## Format de fichier de cassette

Les fichiers sur cassette sont stockés sous forme de son modulé en fréquence. Le format de charge utile des fichiers sur cassette est le même que pour les fichiers sur disque, mais les entêtes sont différents et les fichiers peuvent être divisés en morceaux.

## Modulation

Un 1 bit est représenté par une seule période d'onde de 1 ms (1000 Hz). Un bit 0 est représenté par une seule période d'onde de 0,5 ms (2000 Hz).

## Format d'octet

Un octet est envoyé sous la forme de 8 bits, le plus significatif en premier. Il n'y a pas de bit de démarrage ou d'arrêt.

## Format d'enregistrement

Un fichier est composé de deux enregistrements ou plus. Chaque enregistrement a le format suivant :

<b>Octets</b>	<b>Format</b>	<b>Signification</b>
256 octets	Tous les FF	Onde pilote de 2048 ms à 1000 Hz, utilisée pour l'étalonnage.
1 bit	0	Bit de synchronisation.
1 octet	16 (SYN)	Octet de synchronisation.
256 octets		Bloc de données.
2 octets	Entier big-endian 16 bits non signé	Somme de contrôle CRC-16-CCITT.
31 bits	30 1 suivis d'un 0.	Marqueur de fin d'enregistrement.

Les fichiers tokenisés, protégés et BSAVE consistent en un enregistrement d'entête suivi d'un seul enregistrement pouvant contenir plusieurs blocs de données de 256 octets, chacun suivi des 2 octets CRC.

Les fichiers de programme en texte brut et les fichiers de données se composent d'un enregistrement d'entête suivi de plusieurs enregistrements d'un seul bloc.

Format du bloc d'entête

<b>Octets</b>	<b>Format</b>	<b>Signification</b>
1	A5	Octet magique d'enregistrement d'entête
8	8 caractères	Nom de fichier.
1		Type de fichier. 00 pour fichier de données, 01 pour vidage mémoire, 20 ou A0 pour protégé, 40 pour le programme en texte brut, 80 pour le tokenisé programme.
2	Entier petit-boutiste 16 bits non signé	Longueur du prochain enregistrement de données, en octets.
2	Entier petit-boutiste 16 bits non signé	Segment d'emplacement de mémoire.
2	Entier petit-boutiste 16 bits non signé	Décalage de l'emplacement mémoire.
1	00	Fin des données d'entête
239	Tout 01	Remplissage

Format de bloc de données

<b>Octets</b>	<b>Format</b>	<b>Signification</b>
1	Entier non signé 8 bits	Nombre d'octets de charge utile dans le dernier enregistrement, plus un. Si zéro, l'enregistrement suivant n'est pas le dernier enregistrement.
255		Données de charge utile. S'il s'agit du dernier enregistrement, tous les octets inutilisés sont remplis en répétant le dernier octet de charge utile.

## Formats de fichier d'émulateur

PC-BASIC utilise un certain nombre de formats de fichiers pour prendre en charge son émulation de matériel hérité, qui sont documentés dans cette section. Ces formats de fichiers ne sont pas utilisés par GW-BASIC ou les logiciels contemporains.

### Format de fichier de police HEX

Le format de fichier HEX pour les bitfonts a été développé par Roman Czyborra pour le package GNU Unifont. PC-BASIC utilise une version étendue de ce format de fichier pour stocker ses polices.

Un fichier HEX est un fichier texte ASCII, composé de lignes terminées par LF. Chaque ligne de ce fichier est l'une des suivantes :

- Vide
- Un commentaire, commençant par un caractère #.
- Un ou plusieurs points de code Unicode hexadécimaux de 4 ou 6 caractères, séparés par des virgules, suivis de deux-points, suivis d'un nombre hexadécimal représentant le glyphe. Un nombre à 64 chiffres hexadécimaux ou plus représente un glyphe pleine largeur (16xN), chaque ligne de 16 pixels étant représentée par quatre chiffres hexadécimaux. Un nombre plus court représente un glyphe demi-largeur (8xN), chaque rangée de 8 pixels étant représentée par deux chiffres hexadécimaux.

### Format de fichier de page de code UCP

Les mappages Unicode-codepage sont stockés dans des fichiers UCP.

Un fichier UCP est un fichier texte ASCII, composé de lignes terminées par LF. Chaque ligne de ce fichier est l'une des suivantes :

- Vide
- Un commentaire, commençant par un caractère #.
- Un point de page de code hexadécimal à 2 ou 4 caractères, suivi de deux-points, suivi d'une liste séparée par des virgules de points de code Unicode hexadécimaux à 4 ou 6 caractères. Si plusieurs points de code Unicode sont fournis pour un point de page de code, les points de code se combinent en un seul glyphe.

### Format de fichier de bande CAS

Un fichier CAS est une représentation au niveau du bit des données de cassette introduites par l'émulateur PCE.

Les fichiers CAS produits par PC-BASIC commencent par les caractères PC-BASIC tapeEOF. Cette séquence est suivie de sept bits 0, suivis du contenu de la bande. Les sept bits zéro sont destinés à garantir que le contenu de la bande est aligné sur les octets ; le bit un est constitué par le bit de synchronisation suivant l'onde pilote.

Notez que PC-BASIC ne nécessite pas la séquence d'introduction pour lire correctement un fichier CAS, ni que le contenu d'un fichier CAS soit aligné sur les octets. Cependant, les nouveaux fichiers produits par PC-BASIC suivent cette convention.

## Codes de caractères

Selon le contexte, PC-BASIC traitera un point de code dans la plage des caractères de contrôle comme un caractère de contrôle ou comme un glyphe défini par la page de code active qui par défaut est la page de code 437. Les points de code de &h80 ou plus sont toujours interprétés comme un glyphe de page de code.

## ASCII

Ceci est une liste du code standard américain pour l'échange d'informations (ASCII). ASCII ne couvre que 128 caractères et définit les plages de points de code &h00 – &h1F et &h7F comme contrôle caractères auxquels aucun glyphe imprimable n'est attribué.

Cela inclut des valeurs telles que le caractère de retour chariot ( CR ) qui termine une ligne de programme. Dans le contexte de cette documentation, le caractère &h1A ( SUB ) sera généralement indiqué comme EOF puisqu'il joue le rôle de marqueur de fin de fichier sous DOS.

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1_	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2_		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

## Page de code 437

Ce tableau montre les caractères qui sont produits par les 256 points de code à un octet lorsque la page de codes DOS Latin USA 437 est chargée, qui est la valeur par défaut. D'autres pages de code peuvent être chargées pour affecter d'autres caractères à ces points de code.

- Le point de code &h00 ne peut pas être redéfini.
- La redéfinition des caractères dans la plage de points de code ASCII imprimables &h20 – &h7E entraînera l'affichage d'un glyphe différent à l'écran, mais le caractère continuera d'être traité comme le caractère ASCII correspondant. Il conservera sa valeur ASCII lorsqu'il sera transcodé en UTF-8. Cela se produit, par exemple, avec le signe Yen ( ¥ ) qui est affecté au point de code ASCII &h5C dans la page de codes 932 : dans cette page de codes, il est traité comme s'il s'agissait d'une barre oblique inverse ( \ ).
- Tous les autres caractères peuvent être redéfinis en chargeant une autre page de code avec l'option page de code. Cela affectera à la fois les glyphes visuels et les valeurs de caractère Unicode de ces caractères.



	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_		☉	☿	♥	♦	♣	♠	•	◻	○	◼	♂	♀	♊	♋	♌
1_	▶	◀	↑	⇄	¶	§	—	‡	↑	↓	→	←	↳	↔	▲	▼
2_		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
8_	ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	í	Ä	Å
9_	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ç	£	¥	¤	f
A_	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	¬	¼	½	¾	«	»
B_	⋮	≡	≡		†	‡	‡	‡	‡	‡	‡	‡	‡	‡	‡	‡
C_	L	⊥	⊥	⊥	—	†	‡	‡	‡	‡	‡	‡	‡	=	‡	⊥
D_	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	■	■	■	■	■
E_	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	∩
F_	≡	±	≥	≤	[	]	÷	≈	°	·	·	√	n	z	■	

### Codes des touches de clavier

#### Scancodes

PC-BASIC utilise des scancodes PC / XT, qui proviennent du clavier IBM Model F à 83 touches fourni avec l'IBM PC 5150.

La disposition de ce clavier était assez distincte des claviers standard modernes avec 101 touches ou plus, mais les touches sur un clavier moderne produit le même scancode que la touche avec la même fonction sur le modèle F.

Par exemple, la touche qui (sur un clavier américain) produit le \ était située à côté de la touche Maj gauche sur le clavier du modèle F et a le scancode &h2B. La barre oblique inverse (US) a toujours ce scancode, même s'il se trouve maintenant généralement au-dessus de la touche Entrée.

Pour compliquer davantage les choses, les claviers pour différents paramètres régionaux ont leur disposition remappée dans le logiciel plutôt que dans le matériel, ce qui signifie qu'ils produisent le même scancode que la touche qui, sur un clavier américain, se trouve au même emplacement, quel que soit le caractère qu'ils produisent réellement.

Par conséquent, le A sur un clavier français produira le même scancode que le Q sur un clavier britannique ou américain.

La touche US \ susmentionnée est identifiée avec la touche qui se trouve généralement en bas à gauche de la touche Entrée sur les claviers non américains.

Par exemple, sur mon clavier britannique, il s'agit de la touche #. Les claviers non américains ont une touche supplémentaire à côté de la touche Maj gauche qui, sur le clavier britannique, est le \. Par conséquent, bien que cette clé se trouve au même emplacement et ait la même fonction que le modèle F \, elle a un scancode différent.

Dans le tableau ci-dessous, les touches sont marquées par leur fonction sur un clavier américain, mais il faut garder à l'esprit que le scancode est lié à la position, et non à la fonction, de la touche.

Key	Scancode
Esc	01
1 !	02
2 @	03
3 #	04
4 \$	05
5 %	06
6 ^	07
7 &	08
8 *	09
9 (	0A
0 )	0B
- _	0C
= +	0D
Backspace	0E
Tab	0F
q Q	10
w W	11
e E	12
r R	13
t T	14
y Y	15
u U	16
i I	17
o O	18
p P	19
[ {	1A
] }	1B
Enter	1C
Ctrl	1D
a A	1E
s S	1F
d D	20
f F	21
g G	22

h H	23
j J	24
k K	25
l L	26
; :	27
' "	28
` ~	29
Left Shift	2A
\	2B
z Z	2C
x X	2D
c C	2E
v V	2F
b B	30
n N	31
m M	32
, <	33
. >	34
/ ?	35
Right Shift	36
keypad *    PrtSc	37
Alt	38
Space	39
Caps Lock	3A
F1	3B
F2	3C
F3	3D
F4	3E
F5	3F
F6	40
F7	41
F8	42
F9	43
F10	44
Num Lock	45

Scroll Lock	Pause	46
keypad 7	Home	47
keypad 8	↑	48
keypad 9	Pg Up	49
keypad -		4A
keypad 4	-	4B
keypad 5		4C
keypad 6	-	4D
keypad +		4E
keypad 1	End	4F
keypad 2	↓	50
keypad 3	Pg Dn	51
keypad 0	Ins	52
keypad .	Del	53
SysReq		54
\	(Non-US 102-key)	56
F11		57
F12		58
Left Logo	(Windows 104-key)	5B
Right Logo	(Windows 104-key)	5C
Menu	(Windows 104-key)	5D
????/????	Hiragana/Katakana (Japanese 106-key)	70
\ _	(Japanese 106-key)	73
??	Henkan (Japanese 106-key)	79
???	Muhenkan (Japanese 106-key)	7B
??/??	Hankaku/Zenkaku (Japanese 106-key)	29
¥	(Japanese 106-key)	7D
??	Hanja (Korean 103-key)	F1
??	Han/Yeong (Korean 103-key)	F2
\ ? °	(Brazilian ABNT2)	73
keypad .	(Brazilian ABNT2)	7E

## Codes e-ASCII

Outre les scancodes, la plupart des clés portent également une valeur de caractère que la documentation GW-BASIC appelle ASCII étendu. Comme il s'agit d'un terme plutôt surchargé, nous utiliserons l'abréviation e-ASCII exclusivement pour ces valeurs. Les valeurs renvoyées par la fonction INKEY\$ sont des valeurs e-ASCII.

Les codes e-ASCII ont une longueur d'un ou deux octets ; les codes à un octet sont simplement des codes ASCII tandis que les codes à deux octets consistent en un caractère NUL plus un code indiquant la touche enfoncée. Certains de ces codes, mais certainement pas tous, correspondent aux scancodes des clés.

Contrairement aux scancodes, les codes e-ASCII des touches non modifiées et ceux des touches modifiées par Shift, Ctrl ou Alt sont tous différents.

Les codes e-ASCII non modifiés, majuscules et Ctrl sont liés à la signification d'une touche, et non à son emplacement.

Par exemple, les e-ASCII pour Ctrl + a sont les mêmes sur un clavier français et américain. En revanche, les codes Alt ed sont connectés à l'emplacement de la clé, comme les scancodes.

La disposition du clavier américain est utilisée dans le tableau ci-dessous.

Key	e-ASCII	e-ASCII <b>Shift</b>	e-ASCII <b>Ctrl</b>	e-ASCII <b>Alt</b>
<b>Esc</b>	1B	1B	1B	
<b>1 !</b>	31	21		00 78
<b>2 @</b>	32	40	00 03	00 79
<b>3 #</b>	33	23		00 7A
<b>4 \$</b>	34	24		00 7B
<b>5 %</b>	35	25		00 7C
<b>6 ^</b>	36	5E	1E	00 7D
<b>7 &amp;</b>	37	26		00 7E
<b>8 *</b>	38	2A		00 7F
<b>9 (</b>	39	28		00 80
<b>0 )</b>	30	29		00 81
<b>- _</b>	2D	5F	1F	00 82
<b>= +</b>	3D	2B		00 83
<b>Backspace</b>	08	08	7F	00 8C
<b>Tab</b>	09	00 0F	00 8D	00 8E
<b>q Q</b>	71	51	11	00 10
<b>w W</b>	77	57	17	00 11
<b>e E</b>	65	45	05	00 12
<b>r R</b>	72	52	12	00 13
<b>t T</b>	74	54	14	00 14
<b>y Y</b>	79	59	19	00 15
<b>u U</b>	75	55	15	00 16
<b>i I</b>	69	49	09	00 17
<b>o O</b>	6F	4F	0F	00 18
<b>p P</b>	70	50	10	00 19
<b>[ (</b>	5B	7B	1B	
<b>] )</b>	5D	7D	1D	
<b>Enter</b>	0D	0D	0A	00 8F
<b>a A</b>	61	41	01	00 1E
<b>s S</b>	73	53	13	00 1F
<b>d D</b>	64	44	04	00 20
<b>f F</b>	66	46	06	00 21
<b>g G</b>	67	47	07	00 22
<b>h H</b>	68	48	08	00 23

j J	6A	4A	0A	00 24
k K	6B	4B	0B	00 25
l L	6C	4C	0C	00 26
; :	3B	3A		
' "	27	22		
~	60	7E		
\	5C	7C	1C	
z Z	7A	5A	1A	00 2C
x X	78	58	18	00 2d
c C	63	43	03	00 2E
v V	76	56	16	00 2F
b B	62	42	02	00 30
n N	6E	4E	0E	00 31
m M	6D	4D	0D	00 32
, <	2C	3C		
. >	2E	3E		
/ ?	2F	3F		
PrtSc			00 72	00 46
Space	20	20	20	00 20
F1	00 3B	00 54	00 5E	00 68
F2	00 3C	00 55	00 5F	00 69
F3	00 3D	00 56	00 60	00 6A
F4	00 3E	00 57	00 61	00 6C
F5	00 3F	00 58	00 62	00 6D
F6	00 40	00 59	00 63	00 6E
F7	00 41	00 5A	00 64	00 6F
F8	00 42	00 5B	00 65	00 70
F9	00 43	00 5C	00 66	00 71
F10	00 44	00 5D	00 67	00 72
F11 (Tandy)	00 98	00 A2	00 AC	00 B6
F12 (Tandy)	00 99	00 A3	00 AD	00 B7
Home	00 47	00 47	00 77	
End	00 4F	00 4F	00 75	
PgUp	00 49	00 49	00 84	
PgDn	00 51	00 51	00 76	



↑	00 48	00 48		
←	00 4B	00 87	00 73	
→	00 4D	00 88	00 74	
↓	00 50	00 50		
keypad 5	35	35		05
Ins	00 52	00 52		
Del	00 53	00 53		

## Modèle de mémoire

PC-BASIC (plutôt imparfaitement) émule la mémoire de MS-DOS en mode réel. Cela signifie que la mémoire peut être adressée par segments de 64 Ko. Chaque adresse mémoire est donnée par la valeur du segment et le décalage de 0 à 65 535 octets par rapport à ce segment.

Notez que les segments se chevauchent : l'adresse mémoire réelle est trouvée par  $\text{segment} * 16 + \text{offset}$ . La taille de mémoire maximale qui peut être adressée par ce schéma est donc de 1 Mo, qui était la taille de la mémoire conventionnelle et supérieure dans MS-DOS en mode réel.

### Aperçu

Les domaines de la mémoire avec une importance particulière sont :

Segment	Nom	Objectif
&h0000	Mémoire basse	Contient des informations sur la machine, entre autres
&h13AD (peut varier)	Segment de données	Code de programme, variables, tableaux, chaînes
&hA000 (EGA)	Segment vidéo	Textes et graphiques sur écrans visibles et virtuels
&hB000 (MDA)		
&hB800 (CGA)		
&hC000	--	Définition de la police RAM, entre autres
&hF000	Mémoire en lecture seule	Définition de la police ROM, entre autres

## Segment de données

Le segment de données est organisé comme suit. Les adresses peuvent varier en fonction des paramètres des différentes options ; ici sont les valeurs par défaut pour GW-BASIC 3.23.

Décalage	Taille (octets)	Fonction
&h0000	3429	
&h0D65	(max-files+1) *322	
&h126D	3 + c	
&h1270 + c	v	
&h1270 + c + v	a	
&hFDFC - s	a	
&hFDFC	512	
&hFFFE		

## Guide du développeur

Les fonctionnalités décrites dans ce guide sont destinées aux développeurs Python uniquement. Ils sont expérimentaux, peuvent ne pas fonctionner comme prévu et peuvent être supprimés des versions futures sans avertissement. Vous ne pourrez peut-être pas obtenir d'aide si vous rencontrez des problèmes. Heureusement, aucune des fonctionnalités décrites ici n'est nécessaire au fonctionnement normal de PC-BASIC.

### API de session

PC-BASIC peut être chargé en tant que package depuis Python, ce qui permet d'appeler du code BASIC directement depuis Python.

### Session de classe (\*\*kwargs)

Ouvrez une session PC-BASIC. L'objet de session contient l'état de l'interpréteur, par ex. la valeur des variables, le code de programme et les pointeurs, l'état de l'écran, etc. Notez que Session peut être utilisé comme gestionnaire de contexte avec l'instruction with.

Les arguments de mot-clé sont en grande partie (mais pas entièrement) analogues aux options de ligne de commande PC-BASIC.

Par défaut, l'objet Session saisit l'entrée et la sortie standard comme clavier et écran.

Cela peut être indésirable dans certaines applications ; dans de tels cas, définissez explicitement les arguments des mots-clés input\_streams et output\_streams (par exemple, sur None ).

### exécuter (code\_de\_base)

Exécutez le code BASIC. basic\_code peut être des commandes ou des lignes de programme, séparées par \n ou \r .

évaluer(basic\_expr)

Evalue une expression BASIC et renvoie sa valeur en tant que valeur Python. Pour les règles de conversion de type, voir get\_variable.

set\_variable(nom, valeur)

Définissez la valeur d'un scalaire ou d'un tableau sur une valeur Python.

name est un nom BASIC valide, y compris le sigil, et n'est pas sensible à la casse. Si la cible est un tableau, name doit se terminer par () .

value doit être d'un type compatible : int , bool ou float pour les variables numériques et bytes ou unicode pour les chaînes. Si la cible est un tableau, la valeur doit être une liste de ces valeurs. Les tableaux multidimensionnels doivent être spécifiés sous forme de listes imbriquées.

bool s sera représenté comme en BASIC, avec -1 pour True . unicode sera converti en fonction de la page de code active.

get\_variable(nom)

Récupère la valeur d'un scalaire ou d'un tableau en tant que valeur Python.

name est un nom BASIC valide, y compris le sigil, et n'est pas sensible à la casse. Si la cible est un tableau, name doit se terminer par () .

Les entiers seront renvoyés sous la forme int , les valeurs simple et double précision sous la forme float et la chaîne sous la forme bytes . Si la cible est un tableau, la fonction renvoie une liste (imbriquée) de ces valeurs.

close()

Fermer la session : ferme tous les fichiers ouverts et quitte PC-BASIC. S'il est utilisé comme gestionnaire de contexte, cette méthode est appelée automatiquement.

## Extensions

Il est possible d'activer vos propres instructions BASIC à l'aide d'extensions. Une extension est un objet ou un module Python chargé via l'option --extension ou via le paramètre d'extension de l'objet Session.

Les fonctions Python et autres objets appelables dans l'espace de noms de l'extension seront rendus accessibles via des instructions d'extension de base ou des fonctions dont le nom commence par un trait de soulignement \_ Pour que cela fonctionne, la fonction doit avoir un nom qui est également un nom de variable BASIC valide : alphanumérique uniquement, pas de traits de soulignement, pas égal à un mot-clé BASIC.

Le nom ne sera pas sensible à la casse en BASIC ; c'est-à-dire, def mytestfunc() : print 1 et def myTestFunc() : print 2 mappe à la fois sur l'instruction d'extension ou la fonction \_MYTESTFUNC . Laquelle de ces fonctions serait choisie n'est pas définie, évitez donc cette situation.

Tous les arguments fournis à l'instruction ou à la fonction d'extension sont fournis à la fonction Python en tant que type correspondant : les entiers BASIC deviennent des int s, les nombres simple et double précision deviennent des float s et les chaînes deviennent des octets (pas d'unicode et aucune conversion de page de code n'est appliquée) .

Par exemple, un appel à \_MYTESTFUNC 5, "test-string" s'attendrait à trouver une fonction Python mytestfunc(i, s) avec deux paramètres, et fournirait i=int(5) et a=bytes('test-string' ).

La même fonction Python peut également être appelée en tant que fonction d'extension, par ex. `A = _MYTESTFUNC(5, "test-string")` .

S'il est appelé en tant que fonction, `mytestfunc(i, s)` doit renvoyer une valeur parmi `int` , `float` , qui seront toutes deux converties en un float double précision BASIC ; `bool` , qui sera converti en un entier BASIC ; ou bytes ou unicode , qui sera converti en une chaîne BASIC.

Exemples

```
import pcbasic
import random
```

```
with pcbasic.Session(extension=random) as s:
```

```
    s.execute('a=1')
    print s.evaluate('string$(a+2, "@")')
    s.set_variable('B$', 'abcd')
    s.execute("""
        10 a=5
        20 print a
        run
        _seed(42)
        b = _uniform(a, 25.6)
        print a, b
    """)
```