



Guide de programmation TI-Basic pour la TI-82 Advanced Édition Python

Pour en savoir plus sur la technologie TI, consultez l'aide en ligne à l'adresse education.ti.com/eguide.

Informations importantes

Sauf disposition contraire stipulée dans la licence qui accompagne un programme, Texas Instruments n'émet aucune garantie expresse ou implicite, y compris sans s'y limiter, toute garantie implicite de valeur marchande et d'adéquation à un usage particulier, concernant les programmes ou la documentation, ceux-ci étant fournis "tels quels" sans autre recours. En aucun cas, Texas Instruments ne peut être tenue responsable vis à vis de quiconque pour quelque dommage de nature spéciale, collatérale, fortuite ou indirecte occasionné à un tiers, en rapport avec ou découlant de l'achat ou de l'utilisation desdits matériels, la seule et exclusive responsabilité de Texas Instruments, pour quelque forme d'action que ce soit, ne pouvant excéder le montant indiqué dans la licence du programme. Par ailleurs, la responsabilité de Texas Instruments ne saurait être engagée pour quelque réclamation que ce soit en rapport avec l'utilisation desdits matériels par toute autre tierce partie.

© 2006 - 2021 Texas Instruments Incorporated

Sommaire

Introduction au TI-Basic sur votre TI-82 Advanced Édition Python	1
Qu'est-ce qu'un programme ?	1
Premières activités :	2
Programmer la formule donnant le volume d'un cylindre à partir de son rayon et de sa hauteur	2
Création d'un NOUVEAU programme	2
Nommer le programme	3
Entrer des commandes	4
Afficher le Volume calculé	5
Exécution d'un programme	5
Trouver le volume	6
Créer et supprimer des programmes	7
Versions du système d'exploitation et Programmation	7
Création d'un nouveau programme	7
Gestion de la mémoire et suppression d'un programme	8
Augmenter la mémoire disponible	8
Entrer des lignes de commande et exécuter des programmes	10
Entrée d'une ligne de commande d'un programme	10
Exécution d'un programme	10
Interruption d'un programme	11
Modification de programmes	12
Modification d'un programme	12
Fonction d'édition pour CE OS 5.3 et plus tard	12
Copier et renommer les programmes	14
Copier et renommer un programme	14
Faire défiler les menus PGRM EXEC et PRGM EDIT	14
Instruction PGRM CTL (Control)	15
Menu PRGM CTL	15
If	17
If-Then	17
If-Then-Else	18
For(.....	19
While	19
Repeat	20

End	20
Pause	21
Lbl, Goto	22
Wait	22
IS>(.....	23
DS<(.....	23
Menu(.....	24
prgm	24
Return	24
Stop	25
DelVar	25
GraphStyle(.....	25
CouleurGraph	26
OpenLib(.....	26
ExecLib(.....	27
Instructions PRGM E/S (Entrée/Sortie)	28
Menu PRGM E/S	28
Affichage d'un graphique avec Input	29
Stocker une Valeur de variable avec Input	30
Input [variable]	30
Prompt	31
Disp	31
DispGraph	32
DispTable	32
Output(.....	32
getKey	33
Diagramme des codes des touches	34
ClrHome, ClrTable	34
GetCalc(.....	34
eval(.....	35
expr(.....	35
toString(.....	35
Chaîne4Équ(.....	36
Instructions PRGM COULEUR	37
Menu PRGM COULEUR	37
Instructions PRGM EXÉC	38
Appel d'autres programmes comme sous-programmes	38
Appel d'un programme à partir d'un autre programme	38

Informations générales	40
Aide en ligne	40
Contacter l'assistance technique TI	40
Informations sur le service et la garantie	40

Introduction au TI-Basic sur votre TI-82 Advanced Édition Python

Vous pouvez utiliser TI-Basic pour créer un programme sur votre calculatrice graphique. Vous pouvez créer un programme qui contrôlera une sortie souhaitée ou contrôler une expérience, comme un jeu.

Qu'est-ce qu'un programme ?

Un programme est un ensemble d'une ou plusieurs lignes de commande, contenant chacune une ou plusieurs instructions. Lorsque vous exécutez un programme, la TI-82 Advanced Édition Python de calculatrices graphiques exécute chaque instruction sur chaque ligne de commande dans l'ordre dans lequel vous les avez entrées. Le nombre et la taille des programmes que la TI-82 Advanced Édition Python de calculatrices graphiques peut stocker sont limités uniquement par la mémoire disponible.

Pour créer un programme, entrez simplement des lignes de commande en utilisant l'éditeur de programme. Le programme sera exécuté depuis l'écran de calcul. Utilisez ce guide pour apprendre comment créer, modifier et supprimer des programmes.

Astuce : Utilisez l'aide du Catalogue en appuyant sur [+] sur la plupart des commandes pour vous aider à remplir les arguments corrects pour les commandes avant de les copier dans l'éditeur de programme.

Tandis que vous progressez dans la programmation, un éditeur de programme TI-Basic est également disponible dans le logiciel TI Connect™ CE. Vous pouvez utiliser l'espace de travail de l'éditeur de programme dans TI Connect™ CE pour créer des programmes, pour envoyer des programmes vers une calculatrice connectée par USB, pour tester vos programmes et pour sauvegarder des programmes dans votre ordinateur. L'espace de travail de l'éditeur de programme dans TI Connect™ CE permet de copier, couper, coller et annuler des commandes. Please use the TI-83 Premium CE catalog in the Program Editor in TI Connect CE v5.6.3 or higher.

Please select only the TI-82 Advanced Python Edition commands and functions as described in the TI-82 Advanced Edition Python calculator catalog or Reference Guide. Running unsupported commands and functions will result in an Invalid error on the TI-82 Advanced Edition Python.

Note: The TI-Basic program editor on the calculator contains editing features in the [alpha] [f5] menu such as undo, copy, cut and paste.

Premières activités :

Programmer la formule donnant le volume d'un cylindre à partir de son rayon et de sa hauteur

A partir du rayon et de la hauteur d'un cylindre, vous pouvez calculer son volume à l'aide de cette formule. Cette activité vous permet d'écrire un programme pour demander les valeurs du rayon et de la hauteur d'un cylindre, afin de vous permettre ensuite d'en calculer le volume.

La formule pour le volume d'un cylindre est

$$= \pi R^2 H \text{ unités cubiques}$$

Où

V = Volume

R = Rayon de la base

H = Hauteur du cylindre

Ce programme peut être utile pour diverses activités comme :

- Fournir un tableau avec de nombreuses valeurs de Rayon et de Hauteur et demander aux élèves de remplir la colonne du Volume
- Exécuter un programme pour remplir les valeurs de volume dans le tableau

Quelques questions à examiner :

- (si les élèves ne connaissent pas la formule), quel modèle voyez-vous dans les valeurs de Volume pour leur faire deviner la formule ?
- Quel est le plus grand Volume trouvé ?
- De combien le Volume augmente-t-il si la Hauteur augmente d'une unité ?
- De combien le Volume augmente-t-il si le Rayon augmente d'une unité ?

Exécuter un programme plusieurs fois comme outil permet une analyse rapide de problèmes plus complexes.

Création d'un NOUVEAU programme

1. Appuyez sur `prgm` `▶` `▶` pour afficher le menu **PRGM NEW**.

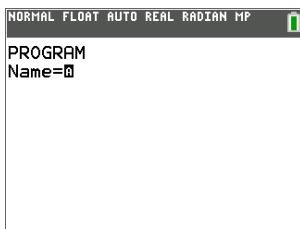
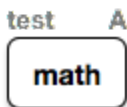


Nommer le programme

1. Appuyez sur **[entrer]** pour sélectionner **1:Create New**.

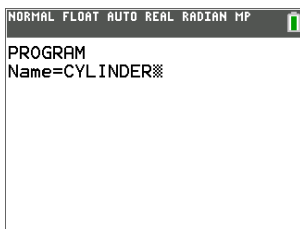
L'invite **Name=** est affichée et [2nde] [verr A] (alpha-lock) est activé.

Astuce: Les caractères alphabétiques se trouvent en haut à droite des touches du clavier et sont copiés lorsque vous appuyez sur [alpha] or [2nde] [verr A] avant d'appuyer sur la touche principales.



2. Appuyez sur **CY L I N D E R**, puis appuyez sur **[entrer]** pour nommer le programme **CYLINDER**.

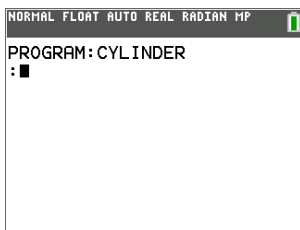
Astuce: Les noms de programme peuvent être composés de huit caractères maximum. Le premier caractère doit être une lettre. Veuillez remarquer le curseur du damier sur l'écran lorsque le maximum est atteint.



3. Appuyez sur **[entrer]** et vous êtes maintenant dans l'éditeur de programme.

Les deux points (:) dans la première colonne de la deuxième ligne indique le début d'une ligne de commande.

Remarque : Sur la calculatrice, les lignes de commande ne sont pas numérotées comme lors de l'utilisation de l'éditeur de Programme TI Connect™ CE.

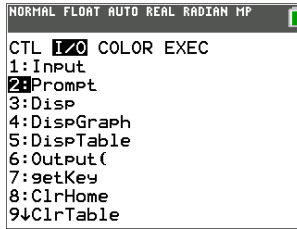


Entrer des commandes

La personne utilisant votre programme devra entrer les valeurs de Rayon et de Hauteur. Vous utiliserez la commande **Prompt**.

1. Appuyez sur `[prgm]` `[▶]` pour accéder au menu E/S (Entrée/Sortie).
2. Appuyez sur `[▼]` pour mettre la commande **Prompt** en surbrillance.

Remarque : Pour cette exemple, vous utiliserez l'aide du Catalogue pour illustrer l'aide à la syntaxe des arguments intégrée dans la calculatrice. Si vous connaissez déjà les arguments d'une commande, vous pouvez choisir un élément et le copier dans l'éditeur de Programme sans utiliser l'aide du Catalogue.



```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL 1/0 COLOR EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
8:CirHome
9↓CirTable
```

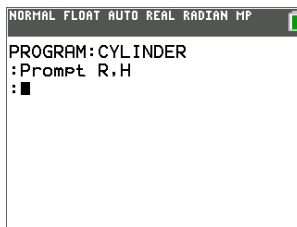
3. Le numéro d'élément de menu de **Prompt** est mis en surbrillance, appuyez sur `[+]`. Utilisez l'aide du Catalogue pour l'éditeur de syntaxe (si nécessaire) La syntaxe pour les arguments de Prompt est affichée sous la ligne d'édition tandis que les variables sont séparées par des virgules. Tout argument entre crochets [] est optionnel, Prompt a donc besoin d'au moins un nom de variable.



```
NORMAL FLOAT AUTO REAL RADIAN MP
CATALOG HELP
Prompt R,H
variableA[,variableB
.....variable n]
PASTE| ESC
```

4. Appuyez sur `[alpha]` `R` `[.]` `[alpha]` `H` pour entrer les noms de variables pour le Rayon et la Hauteur.
5. Appuyez sur `[PASTE]` (`[trace]`) pour copier la commande avec les arguments dans l'éditeur de Programme. Appuyez sur `[ESC]` (`[graphe]`) pour retourner au dernier emplacement du curseur sans copier.

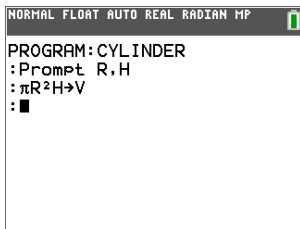
6. De retour dans l'éditeur de Programme, appuyez sur `[entrer]` pour déplacer le curseur vers la ligne de commande suivante.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:CYLINDER
:Prompt R,H
:█
```

Stockez la formule du volume d'un cylindre.

- Pour entrer l'expression $[\pi] R^2 H$ et stocker la valeur dans la variable V, appuyez sur $[2\text{nde}] [\pi] [\alpha] R [x^2]$ $[\alpha] H [\text{sto}\rightarrow] [\alpha] V [\text{entrer}]$.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CYLINDER
:Promp t R,H
:πR²H→V
:█
```

Afficher le Volume calculé

Créez une ligne de commande pour afficher le volume calculé :

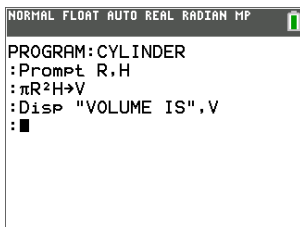
- Appuyez sur $[\text{prgm}] [\rightarrow] 3$ pour sélectionner 3:Disp depuis le menu **PRGM E/S**.

Disp est copié sur la ligne de commande.

Astuce : Souvenez-vous que vous pouvez appuyer sur $[+]$ sur la plupart des commandes pour utiliser l'aide du Catalogue pour l'éditeur de syntaxe pour voir les arguments corrects pour les commandes.

- Appuyez sur $[2\text{nde}] [\text{verr A}] ["]$ **VOLUME EST** $["] [\alpha] [.] [\alpha] V [\text{entrer}]$

Ceci affichera le texte **VOLUME EST** sur une ligne et la valeur calculée de V sur la ligne suivante de l'écran de calcul lorsque vous exécutez le programme.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CYLINDER
:Promp t R,H
:πR²H→V
:DISP "VOLUME IS",V
:█
```

Exécution d'un programme

Votre programme est terminé ! Vous pouvez maintenant exécuter le programme à partir de l'écran de calcul

- Appuyez sur $[2\text{nde}] [\text{quitter}]$ pour afficher l'écran de calcul.
- Appuyez sur $[\text{prgm}]$ pour afficher le menu **PRGM EXEC**.

Les éléments de ce menu sont les noms des programmes stockés.



```
NORMAL FLOAT AUTO REAL RADIAN MP
EXEC EDIT NEW
1:CYLINDER
:█
```

- Appuyez sur `⏎` pour copier **prgm** **CYLINDER** à l'emplacement actuel du curseur. (Si **CYLINDER** n'est pas l'élément 1 de votre menu **PRGM EXEC**, déplacez le curseur vers **CYLINDER** avant d'appuyer sur `⏎`.)

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmCYLINDER█
```

Trouver le volume

Pour trouver le volume du cylindre de 1,5 cm de rayon et de 3 cm de hauteur, suivez les étapes suivantes.

- Appuyez sur `⏎` pour (exécuter) le programme.
- Entrez **1,5** pour **R** et appuyez sur `⏎`
- Entrez **3** pour **H** et appuyez sur `⏎`.

Le texte **VOLUME EST**, la valeur de **V** et **Fait** s'affichent.

Le volume du cylindre s'affiche à 8 décimales près
21. 20575041 cm cube.

- À ce stade, pour relancer le programme, appuyez sur `⏎` et répétez pour les différentes valeurs de **R** et de **H**.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmCYLINDER
R=?1.5
H=?3
VOLUME IS          21.20575041
.....Done
█
```

Créer et supprimer des programmes

Cette section décrit comment créer des programmes et comment supprimer des programmes.

Versions du système d'exploitation et Programmation

- L'exécution des programmes peut s'effectuer en mode classique ou en mode MathPrint™.
- Les menus de raccourcis sont disponibles partout où le menu MATH est accessible.
- Les modèles MathPrint™ ne sont pas disponibles pour les programmes. Toutes les saisies et l'affichage des résultats s'effectuent en mode classique.
- Vous pouvez utiliser les fractions dans les programmes, mais il est recommandé de tester le programme afin de s'assurer que les résultats obtenus correspondent à vos attentes.
- L'espacement de l'affichage peut être légèrement différent en mode MathPrint™ par rapport au mode classique. Si vous préférez l'espacement du mode classique, réglez le mode à l'aide d'une commande dans votre programme. Les captures d'écran des exemples présentés dans ce chapitre ont été réalisées en mode MathPrint™.

Remarque : Appuyez sur $\boxed{+}$ lorsqu'une commande est sélectionnée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

Création d'un nouveau programme

Pour créer un nouveau programme, suivez ces étapes.

1. Appuyez sur $\boxed{\text{prgm}}$ $\boxed{\leftarrow}$ pour afficher le menu **PRGM NOUVEAU**.



2. Appuyez sur $\boxed{\text{entrer}}$ pour sélectionner **1:Create New**. L'invite **Name=** est affichée et l'alpha lock est activé.
3. Appuyez sur une lettre de A à Z ou θ pour entrer le premier caractère du nouveau nom de programme.

Remarque : le nom du programme peut comporter de 1 à 8 caractères. Le premier caractère doit être une lettre de A à Z ou θ . Les caractères suivants peuvent être des lettres, des chiffres ou θ .

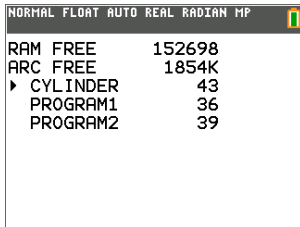
4. Entrez de 0 à 7 lettres, chiffres ou θ pour compléter le nom du nouveau programme.

- Appuyez sur **[enter]**. L'éditeur de programme est affiché.
- Entrez une ou plusieurs commandes de programme.
- Appuyez sur **[2nde]** **[quitter]** pour quitter l'éditeur de programme et retourner à l'écran de calcul.

Gestion de la mémoire et suppression d'un programme

Pour vérifier si la mémoire disponible est suffisante pour un programme que vous souhaitez entrer :

- Appuyez sur **[2nde]** **[mém]** pour afficher le menu **MEMORY (MÉMOIRE)**.
- Choisissez **2:Mem Management/Delete** pour afficher le menu **MEMORY MANAGEMENT/DELETE**.
- Choisissez **7:Prgm** pour afficher l'éditeur **PRGM**.



La TI-82 Advanced Édition Python de calculatrice graphiques indique la quantité de mémoire en octets.

Augmenter la mémoire disponible

Vous pouvez augmenter la mémoire disponible de deux façons. Vous pouvez supprimer un ou plusieurs programme ou vous pouvez archiver certains programmes.

Pour augmenter la mémoire disponible en supprimant un programme spécifique :

- Appuyez sur **[2nde]** **[mém]** puis sélectionnez **2:Mem Management/Delete** depuis le menu **MEMORY**.



- Choisissez **7:Prgm** pour afficher les fichiers de type programme.

NORMAL FLOAT AUTO REAL RADIAN MP	
RAM FREE	152698
ARC FREE	1854K
▶ CYLINDER	43
PROGRAM1	36
PROGRAM2	39

- Appuyez sur \leftarrow et \downarrow [alpha] pour déplacer le curseur de sélection (▶) près du programme que vous voulez supprimer et appuyez sur [suppr]. Le programme est supprimé de la mémoire.

Remarque : Vous recevrez un message vous demandant de confirmer cette action de suppression. Choisissez **2:yes** pour continuer.

Pour quitter l'écran d'édition **PRGM** sans supprimer de programme, appuyez sur [2nde] [quitter], qui affiche l'écran de calcul.

Pour augmenter la mémoire disponible en archivant un programme :

- Appuyez sur [2nde] [mém] puis sélectionnez **2:Mem Management/Delete** depuis le menu **MEMORY**.
- Choisissez **2:Mem Management/Delete** pour afficher le menu **MEMORY MANAGEMENT/DELETE**.
- Choisissez **7:Prgm...** pour afficher les fichiers de type programme.

NORMAL FLOAT AUTO REAL RADIAN MP	
RAM FREE	152726
ARC FREE	1854K
▶*CYLINDER	43
PROGRAM1	36
PROGRAM2	39

- Appuyez sur [entrer] pour archiver le programme. Une astérisque apparaîtra à gauche du programme pour indiquer que ce programme est archivé.

Pour désarchiver un programme dans cet écran, placez le curseur près du programme archivé et appuyez sur [entrer]. L'astérisque disparaît.

Remarque : Il vous est impossible de modifier ou d'exécuter les programmes archivés. Pour modifier ou exécuter un programme archivé, vous devez d'abord le désarchiver.

Entrer des lignes de commande et exécuter des programmes

Cette section décrit comment entrer des lignes de commande et exécuter des programmes.

Entrée d'une ligne de commande d'un programme

Sur une ligne de commande, vous pouvez entrer toute commande, instruction ou expression que vous pouvez exécuter depuis l'écran de calcul. Dans l'éditeur de programme, chaque nouvelle ligne de commande commence avec deux-points. Pour entrer plus d'une instruction ou expression sur une ligne de commande, séparez-les avec deux-points.

Remarque : Une ligne de commande peut être plus longue que la largeur de l'écran.

Dans l'éditeur de programme, vous pouvez afficher et sélectionner depuis les menus. Vous pouvez retourner à l'éditeur de programme depuis un menu de deux façons.

- Sélectionnez une option de façon à insérer l'élément dans la ligne de commande courante.
— ou —
- Appuyez sur `[annul]`.

Pour terminer une ligne de commande, appuyez sur `[entrer]`. Le curseur se positionne sur la ligne de commande suivante.

Les programmes peuvent accéder aux variables, aux listes, aux matrices et aux chaînes de caractères sauvegardées dans la mémoire. Si un programme stocke une nouvelle valeur dans une variable, une liste, une matrice ou une chaîne de caractères, le programme change la valeur dans la mémoire pendant l'exécution.

Vous pouvez appeler un autre programme comme sous-programme.

Exécution d'un programme

Pour exécuter un programme, commencez sur une ligne vierge de l'écran de calcul et suivez ces étapes.

1. Appuyez sur `[prgm]` pour afficher le menu **PRGM EXEC**.
2. Sélectionnez un nom de programme dans le menu **PRGM EXEC**. `prgmname` est collé dans l'écran de calcul (par exemple, `prgmCYLINDER`).
3. Appuyez sur `[entrer]` pour exécuter le programme. Pendant que le programme s'exécute, l'indicateur d'occupation est activé.

Last Answer (**Ans**) est mis à jour pendant l'exécution du programme. Last Entry n'est pas mis à jour pendant que chaque commande est exécutée.

La TI-82 Advanced Édition Python de calculatrices graphiques cherche les erreurs pendant l'exécution d'un programme. Elle ne cherche pas les erreurs pendant que vous entrez un programme.

Remarque: CE OS 5.3 et plus tard

- Les programmes peuvent être exécutés à partir de RAM ou d'archives. Les programmes ne peuvent pas être modifiés si dans Archive.
- Édition de MENU (**[alpha]** **[f5]**) dans l'éditeur de programme
 - **1: Exécuter le programme**
 - Le programme en cours d'édition sera exécuté directement à partir de l'éditeur de programme.
 - **7: Insérer un commentaire ci-dessus (**[alpha]** **[f5]** **7**)**
 - **Insérer un commentaire** Au-dessus, colis le jeton de la citation au début d'une nouvelle ligne de commande au-dessus de l'emplacement actuel du curseur. Lorsque cette ligne de commande est exécutée, une variable String du texte de commentaire est créée et est stockée dans la variable Ans. Programme et utilisation des variables, si les commentaires sont inclus en utilisant le jeton de citation, planifiez que la chaîne de commentaires sera stockée dans Ans lors de l'exécution.

Interruption d'un programme

Pour annuler l'exécution d'un programme, appuyez sur **[on]**. Le menu **ERR:BREAK** s'affiche.

- Pour revenir dans l'écran de calcul, sélectionnez **1:Quitter**.
- Pour aller à l'endroit où l'interruption s'est produite, sélectionnez **2:Goto**.

Modification de programmes

Dans cette section, vous suivrez les étapes suivantes pour modifier un programme. Cette section décrit comment insérer et supprimer une ligne de commande.

Modification d'un programme

Pour modifier un programme stocké, suivez ces étapes.

1. Appuyez sur `[prgm]` `[▶]` pour afficher le menu **PRGM EDIT**.
2. Sélectionnez un nom de programme dans le menu **PRGM EDIT**. Les neuf premières lignes du programme sont affichées.

Remarque : L'éditeur de programme n'affiche pas de ↓ pour indiquer qu'un programme continue en dehors de l'écran.

3. Modifiez les lignes de commande du programme.
 - Déplacez le curseur à l'emplacement approprié, puis supprimez, écrasez ou insérez.
 - Appuyez sur `[annul]` pour effacer toutes les commandes de programme sur la ligne de commande (les deux points d'entrée restent), puis entrez une nouvelle commande de programme.

Remarque : Pour déplacer le curseur au début d'une ligne de commande, appuyez sur `[2nde]` `[←]`; pour le déplacer à la fin, appuyez sur `[2nde]` `[▶]`. Pour faire défiler le curseur vers le bas de sept lignes de commande, appuyez sur `[alpha]` `[↓]`. Pour faire défiler le curseur vers le haut de sept lignes de commande, appuyez sur `[alpha]` `[↑]`.

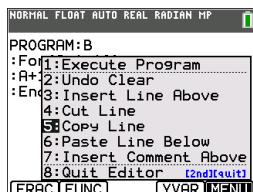
Insérer et Supprimer des Lignes de Commande

Pour insérer une nouvelle ligne de commande n'importe où dans le programme, placez le curseur à l'endroit où vous voulez insérer la nouvelle ligne, appuyez sur `[2nde]` `[insérer]`, puis appuyez sur `[entrer]`. Deux points indiquent une nouvelle ligne.

Pour supprimer une ligne de commande, placez le curseur sur la ligne, appuyez sur `[annul]` pour effacer toutes les instructions sur la ligne, puis appuyez sur `[suppr]` pour supprimer la ligne de commande, y compris les deux points.

Fonction d'édition pour CE OS 5.3 et plus tard

Les fonctions d'édition ont été ajoutées à l'éditeur de programme. Ceux-ci incluent annuler, insérer et couper / copier / coller. Appuyez sur `[prgm]` pour EDITER votre programme. En appuyant sur `[alpha]` `[f5]` ouvre le nouveau MENU d'édition.



Astuce: MENU Élément 7: Insérer un commentaire ci-dessus(alpha) [f5] 7)

Insérer un commentaire Au-dessus, colis le jeton de la citation au début d'une nouvelle ligne de commande au-dessus de l'emplacement actuel du curseur. Lorsque cette ligne de commande est exécutée, une variable String du texte de commentaire est créée et est stockée dans la variable Ans. Programme et utilisation des variables, si les commentaires sont inclus en utilisant le jeton de citation, planifiez que la chaîne de commentaires sera stockée dans Ans lors de l'exécution.

Copier et renommer les programmes

Cette section décrit comment copier et renommer un programme et comment parcourir les menus.

Copier et renommer un programme

Pour copier toutes les lignes de commande d'un programme dans un nouveau programme, suivez les étapes 1 à 5 pour Créer un Nouveau programme, puis suivez ces étapes.

1. Appuyez sur **[2nde]** **[rappel]**. **rappel** s'affiche sur la dernière ligne de l'éditeur de programme dans le nouveau programme.
2. Appuyez sur **[prgm]** pour afficher le menu **PRGM EXEC**.
3. Sélectionnez un nom dans le menu. **prgmnom** est collé sur la dernière ligne de l'éditeur de programme.
4. Appuyez sur **[entrer]**. Toutes les lignes de commande du programme choisi sont copiées dans le nouveau programme.

Copier les programmes a au moins deux applications pratiques.

- Vous pouvez créer un modèle pour les groupes d'instructions que vous utilisez fréquemment.
- Vous pouvez renommer un programme en copiant son contenu dans un nouveau programme.

Remarque : Vous pouvez également copier toutes les lignes de commande d'un programme existant dans un autre programme en utilisant **rappel (RCL)**.

Faire défiler les menus PGRM EXEC et PRGM EDIT

La TI-82 Advanced Édition Python de calculatrice graphiques trie les éléments de menu **PRGM EXEC** et **PRGM EDIT** automatiquement par ordre alphanumérique. Chaque menu marque uniquement les 10 premiers éléments à l'aide des touches 1 à 9, puis 0.

Pour passer au premier nom de programme commençant par un caractère particulier alpha ou θ , appuyez sur **[alpha]** [lettre de A à Z ou θ].

Remarque : Depuis le haut du menu **PRGM EXEC** ou **PRGM EDIT**, appuyez sur **[▲]** pour vous déplacer à la fin. À partir du bas du menu, appuyez sur **[▼]** pour en afficher le début. Pour faire défiler le curseur vers le haut dans les sept éléments du menu, appuyez sur **[alpha]** **[▼]**. Pour faire défiler le curseur vers le bas dans les sept éléments du menu, appuyez sur **[alpha]** **[▲]**.

Instruction PGRM CTL (Control)

Cette section les instructions **PGRM CTL** (Control).

Menu **PGRM CTL**

Pour afficher le menu **PGRM CTL**, appuyez sur **[prgm]** seulement depuis l'éditeur de programme.

Astuce importante : Pour retrouver rapidement une commande, utilisez **[alpha]** **[↑]** ou **[alpha]** **[↓]** pour vous déplacer entre les écrans.

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7:End
8:Pause
9↓Lb1
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
9↑Lb1
0:Goto
A:Wait
B:IS>(
C:DS<(
D:Menu(
E:prgm
F:Return
G:Stop
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
D↑Menu(
E:prgm
F:Return
G:Stop
H:DelVar
I:GraphStyle(
J:GraphColor(
K:OpenLib(
L:ExecLib
```

CTRL E/S COULEUR EXEC HUB

	Description
1: If	Crée un test conditionnel :
2: Then	Exécute les commandes lorsque If est vraie.
3: Else	Exécute les commandes lorsque If est fausse.
4: For(Crée une boucle incrémentée.
5: While	Crée une boucle conditionnelle.
6: Repeat	Crée une boucle conditionnelle.

7: End	Signifie la fin d'un bloc :
8: Pause	Met en pause l'exécution d'un programme.
9: Lbl	Définit un label.
0: Goto	Va vers un label.
A: Wait	Suspend l'exécution d'un programme pour un temps donné.
B: IS>(Incrémente et saute si plus grand que.
C: DS<(Décrémente et saute si plus petit que.
D: Menu(Définit les éléments de menu et les branches.
E: prgm	Exécute un programme comme sous-programme.
F: Return	Revient depuis une sous-programme.
G: Stop	Arrête l'exécution.
H: DelVar	Efface le contenu d'une variable depuis un programme.
I: GraphStyle(Désigne le style de graphique à tracer.
J: CouleurGraph(Désigne la couleur du graphique à tracer.
K: OpenLib(Extension de TI-Basic (non disponible)
L: ExecLib(Extension de TI-Basic (non disponible)

Remarque : Appuyez sur $\boxed{+}$ lorsqu'une commande est surlignée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

Ces éléments de menu dirigent le flux d'un programme en cours d'exécution. Ils facilitent la répétition ou le saut d'un groupe de commande pendant l'exécution du programme. Lorsque vous sélectionnez un élément depuis le menu, le nom est collé à l'emplacement du curseur sur la ligne de commande dans le programme.

Pour revenir à l'éditeur de programmes sans sélectionner la rubrique, appuyez sur $\boxed{\text{annul}}$.

Contrôle du flux d'un programme

Les instructions de contrôle d'un programme indiquent à la Calculateur graphique TI CE quelle prochaine commande exécuter dans un programme. **If**, **While**, et **Repeat** vérifient une condition définie pour déterminer quelle commande exécuter ensuite. Les conditions utilisent fréquemment des tests relationnels ou booléen, comme dans :

If A<7:A+1>A

or

If N=1 et M=1:Goto Z

If

Utilisez **If** pour des tests et des branchements. Si *condition* est fausse (0), alors la *commande* suivant immédiatement **If** est sautée. Si *condition* est vraie (non nulle), alors la *commande* suivante est exécutée. Les instructions **If** peuvent être imbriquées.

```
:if condition
:commande (si vraie)
:commande
```

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: COUNT
:0→A
:Lb1 Z
:A+1→A
:Disp "A IS ",A
:If A≥2
:Stop
:Goto Z
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mCOUNT
A IS 1
A IS 2
..... Done.
```

If-Then

Then suivant un **If** exécute un groupe de *commandes* si *condition* est vraie (non nulle). **End** indique la fin du groupe de *commandes*.

```
:if condition
:Then
:commande (si vraie)
:commande (si vraie)
:End
:commande
```

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TEST
:1→X:10→Y
:If X<10
:Then
:2X+3→X
:2Y-3→Y
:End
:Disp X,Y
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mTEST
..... 5
..... 17
..... Done.
```

If-Then-Else

Else suivant **If-Then** exécute un groupe de *commandes* si *condition* est fausse (nulle).
End indique la fin du groupe de *commandes*.

```
:if condition
:Then
:commande (si vraie)
:commande (si vraie)
:Else
:commande (si vraie)
:commande (si vraie)
:End
:commande
```

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:TESTELSE
:Input "X=",X
:If X<0
:Then
: X^2→Y
:Else
: X→Y
:End
:Disp {X,Y}
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mTESTELSE
X=5
{5 5}
Done
Pr9mTESTELSE
X=-5
{-5 25}
Done
```

Remarque : Appuyez sur pour répéter le programme

For(

For(effectue des boucles et incrémente Il incrémente la *variable* de *début* à *fin* par *incrément*. *incrément* est facultatif (la valeur par défaut est 1) et peut être négatif ($fin < début$). *fin* est une valeur maximum ou minimum à ne pas dépasser **End** indique la fin de la boucle. Les boucles **For(** peuvent être imbriquées.

```
:For(variable,début,fin[,incrément])  
:commande (alors que fin non dépassée)  
:commande (alors que fin non dépassée)  
:End  
:commande
```

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: SQUARE
:For(A,0,8,2)
:Disp A^2
:End
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mSQUARE
0
4
16
36
64
.....Done.
```

While

While exécute un groupe de *commandes* tandis que *condition* est vraie. *condition* est souvent un test relationnel. *condition* est testée lorsque **While** est rencontré. Si *condition* est vraie (non nulle), le programme exécute un groupe de *commandes*. **End** indique la fin du groupe. Lorsque *condition* est fausse (nulle), le programme exécute chaque *commande* suivant **End**. Les instructions **While** peuvent être imbriquées.

```
:While condition  
:commande (alors que condition est vraie)  
:commande (alors que condition est vraie)  
:End  
:commande
```


Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: LOOP
:0→I
:0→J
:While I<6
:J+1→J
:I+1→I
:End
:Disp "J=",J
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Prgm LOOP
J=
.....6
Done
```

Repeat

While répète un groupe de *commandes* tandis que *condition* est vraie (non nulle). Elle est similaire à **While**, mais *condition* est testée lorsque **End** est rencontré; le groupe de *commandes* est donc exécuté au moins une fois. Les instructions **Repeat** peuvent être imbriquées.

:Repeat condition

:commande (jusqu'à ce que *condition* soit vraie)

:commande (jusqu'à ce que *condition* soit vraie)

:End

:commande

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: RLOOP
:0→I
:0→J
:Repeat I≥6
:J+1→J
:I+1→I
:End
:Disp "J=",J
:■
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Prgm RLOOP
J=
.....6
Done
```

End

End indique la fin d'un groupe de *commandes*. Vous devez inclure une instruction **End** à la fin de chaque boucle **For**, **While**, ou **Repeat**. Vous devez également inclure une instruction **End** à la fin de chaque groupe **If-Then** et de chaque groupe **If-Then-Else**.

Pause

Pause suspend l'exécution du programme, afin que vous puissiez voir les réponses ou les graphiques. Pendant la pause, l'indicateur de pause est situé dans le coin supérieur droit.

- **Pause** sans argument interrompt temporairement le programme. Si l'instruction **DispGraph** ou **Disp** a été exécutée, l'écran approprié apparaît. Appuyez sur **[entrer]** pour reprendre l'exécution.
- **Pause** avec *valeur* affiche *valeur* sur l'écran de calcul courant. Vous pouvez faire défiler *valeur*. **Pause** [*valeur*] Appuyez sur **[entrer]** pour reprendre l'exécution.
- **Pause** avec *valeur* et *temps* affiche la valeur sur l'écran de calcul actuel et l'exécution du programme continue pour la période spécifiée. Pour le temps seulement, utilisez **Pause " ",time** où la valeur est une chaîne de caractères vide. Temps en secondes. **Pause** *valeur,temps*.

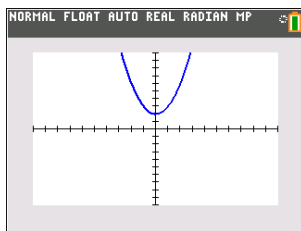
Remarque : Lorsque vous utilisez l'Éditeur de programme de TI Connect™ CE, Pause doit avoir un espace après la commande même si aucun argument n'est entré.

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: PAUSE
:10→X
:"X^2+2"→Y1
:Disp "X=",X
:Pause
:DispGraph
:Pause
:Disp
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmPAUSE
X= 10
```



```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmPAUSE
X= 10
..... Done
```

Lbl, Goto

Lbl

Lbl (étiquette) et **Goto** (aller à) sont utilisés ensemble pour le branchement.

Lbl spécifie le *label* pour une commande. *label* peut avoir un ou deux caractères (A à Z, 0 à 99, ou θ).

Lbl étiquette

Goto

Goto branche le programme à *label* lorsque **Goto** est rencontré.

Gotoétiquette

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: CUBE
:Lbl 99
:Input A
:If A≥100
:Stop
:Disp A³
:Pause
:Goto 99
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
Pr9mCUBE
?2
?3
?105
..... Done.
```

Wait

Wait suspend l'exécution d'un programme pour un temps donné. Le temps maximum est de 100 secondes. Pendant le temps de pause, l'indicateur d'occupation est situé dans le coin supérieur droit.

Wait temps

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: WAIT
:ClrDraw
:AxesOff:FnOff
:TextColor(MAGENTA)
:Text(2,2,"HELLO WORLD"
:Wait 5
:TextColor(GREEN)
:Text(24,2,"BYE!"
:
```

Output : "Bye!" s'affiche après 5 secondes.

```
NORMAL FLOAT AUTO REAL RADIAN MP
HELLO WORLD
BYE!
```

IS>(

IS>((incrémente et saute) ajoute 1 à *variable*. Si la réponse est $>$ *valeur* (qui peut être une expression), la *commande* suivante est sautée; si la réponse est \leq *valeur*, la *commande* suivante est exécutée. *variable* ne peut pas être une variable système

:IS>(variable,valeur)

:commande (si la réponse *valeur*)

:commande (si la réponse $>$ *valeur*)

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: ISKIP
:7→A
:IS>(A,6)
:DISP "NOT > 6"
:DISP "> 6"
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgrmISKIP
> 6
..... Done
█
```

Remarque : **IS>(** n'est pas une instruction de boucle.

DS<(

DS<((décrémente et saute) soustrait 1 de *variable*. Si la réponse est $>$ *valeur* (qui peut être une expression), la *commande* suivante est sautée; si la réponse est \leq *valeur*, la *commande* suivante est exécutée. *variable* ne peut pas être une variable système

:DS>(variable,valeur)

:commande (si la réponse *valeur*)

:commande (si la réponse $>$ *valeur*)

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: DSKIP
:1→A
:DS<(A,6)
:DISP "> 6"
:DISP "NOT > 6"
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgrmDSKIP
NOT > 6
..... Done
█
```

Remarque : **DS<(** n'est pas une instruction de boucle.

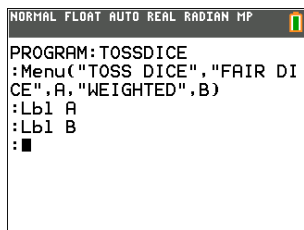
Menu()

Menu() définit un branchement dans un programme. Si **Menu()** est rencontré pendant l'exécution d'un programme, l'écran du menu est affiché avec les éléments de menu spécifiés, l'indicateur de pause est activé et l'exécution est interrompue jusqu'à ce qu'un élément de menu soit sélectionné.

Le *titre* du menu est entre guillemets ("). Jusqu'à neuf paires d'éléments de menu sont autorisées. Chaque paire est constituée d'un élément *texte* (également entre guillemets) à afficher comme menu de sélection et d'un élément *label* auquel brancher si vous sélectionnez la sélection de menu correspondante.

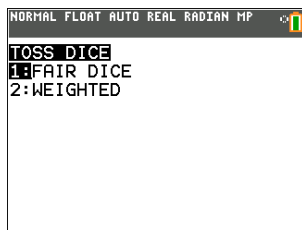
Menu("titre","texte1",label1,"texte2",label2,...)

Programme



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TOSSDICE
:Menu("TOSS DICE", "FAIR DICE",
A, "WEIGHTED", B)
:Lb1 A
:Lb1 B
:.
```

Sortie



```
NORMAL FLOAT AUTO REAL RADIAN MP
TOSS DICE
1:FAIR DICE
2:WEIGHTED
```

Le programme ci-dessus s'interrompt jusqu'à ce que **1** ou **2** soit sélectionné. Si vous sélectionnez **2**, par exemple, le menu disparaît et le programme continue l'exécution à **Lbl B**.

prgm

Utilisez **prgm** pour exécuter d'autres programmes comme sous-programmes. Lorsque vous sélectionnez **prgm**, il est collé à l'emplacement du curseur. Entrez les caractères pour écrire le *nom* d'un programme. Utiliser **prgm** est équivalent à sélectionner des programmes existants depuis le menu **PRGM EXEC**; cependant, il vous permet également d'entrer le nom d'un programme que vous n'avez pas encore créé.

prgmnom

Remarque : Vous ne pouvez pas entrer directement le nom du sous-programme lorsque vous utilisez **RCL**. Vous devez copier le nom depuis le menu **PRGM EXEC**.

Return

Return sort d'un sous-programme et l'exécution retourne au programme appelant, même s'il est rencontré avec des boucles imbriquées. Toutes les boucles sont terminées. Un **Return** existe à la fin de tout programme appelé comme sous-

programme. Dans le programme principal, **Return** arrête l'exécution et retourne à l'écran de calcul.

Stop

Stop arrête l'exécution d'un programme et retourne à l'écran de calcul. **Stop** est facultatif à la fin d'un programme.

DelVar

DelVar supprime de la mémoire le contenu de *variable*.

EffVar *variable*



GraphStyle(

GraphStyle(désigne le style du graphique à tracer. *function#* est le numéro du nom de la fonction Y= dans le mode graphique courant. *graphstyle* est un chiffre de 1 à 7 correspondant au style du graphique, comme indiqué ci-dessous

- | | |
|---|---|
| 1 =  (Thin) | 5 =  (Path) |
| 2 =  (Thick) | 6 =  (Animate) |
| 3 =  (Shade above) | 7 =  (Dot-Thick) |
| 4 =  (Shade below) | 8 =  (Dot-Thin) |

GraphStyle(*function#*,*graphstyle*)

Par exemple, **GraphStyle(1,5)** en mode **Func** définit le style du graphique pour Y1 à  (chemin; 5).

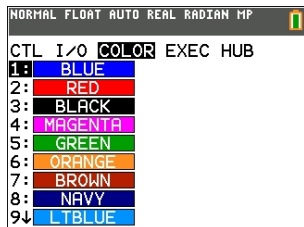
Tous les styles ne sont pas disponibles dans les modes graphiques.

CouleurGraph

GraphColor(désigne la couleur du graphique à tracer. *function#* est le numéro du nom de la fonction Y= dans le mode graphique courant. *color#* est un chiffre de 10 à 24 correspondant à la couleur du graphique, comme indiqué ci-dessous :

Numéro de couleur	Nom de couleur
10	BLEU
11	ROUGE
12	NOIR
13	MAGENTA
14	VERT
15	ORANGE
16	MARRON
17	BLEU MRN
18	BLEU CLR
19	JAUNE
20	BLANC
21	GRIS CLR
22	GRIS MOY
23	GRIS
24	GRIS FON

Vous pouvez également choisir une couleur dans le menu **var** (sous-menu **couleur**).



CouleurGraph(*n°fonction*,*n°couleur*)

Par exemple, **CouleurGraph(2, 4)** ou **CouleurGraph(2, MAGENTA)**.

OpenLib(

Extension de TI-Basic (non disponible)

ExecLib

Extension de TI-Basic (non disponible)

Instructions PRGM E/S (Entrée/Sortie)

Cette section les instructions **PRGM E/S** (Entrée/Sortie).

Menu PRGM E/S

Pour afficher le menu **PRGM E/S** (programme entrée/sortie), appuyez sur `[prgm]` `[>]` seulement depuis l'éditeur de programme.

Astuce importante : Pour retrouver rapidement une commande, utilisez `[alpha]` `[↑]` or `[alpha]` `[↓]` pour vous déplacer entre les écrans.

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7:getKey
8:ClrHome
9↓ClrTable
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
8↑ClrHome
9:ClrTable
0:GetCalc(
A:eval(
B:expr(
C:toString(
D:String→Equ(
```

CTRL `[E/S]` **COULEUR EXEC HUB**

	Description
1: Input	Entre une valeur ou utilise le curseur.
2: Prompt	Invite pour l'entrée de valeurs de variable.
3: Disp	Affiche le texte, la valeur ou l'écran de calcul.
4: DispGraph	Affiche le graphique courant.
5: DispTable	Affiche le tableau de valeurs courant.
6: Output(Affiche le texte à une position spécifiée.
7: getKey	Contrôle la frappe sur le clavier.
8: EffÉcran	Efface l'affichage.
9: EffTable	Efface le tableau courant.
0: GetCalc(Obtient une variable à partir d'une autre TI-82 Advanced Édition Python
A: eval(Retourne une expression évaluée sous forme de chaîne avec 8 chiffres significatifs.

- B: `expr(` Convertit la chaîne de caractères contenue dans *chaîne* en une expression et l'évalue.
- C: `toString(` Convertit une valeur en une chaîne de caractères où *valeur* peut être un réel, un complexe, une expression évaluée, une liste ou une matrice.
- D: `Chaîne↵Équ(` Converts a string into an equation and stores it in $Y=$ var. The string can be a string or string variable.

Remarque : Appuyez sur $\boxed{+}$ lorsqu'une commande est surlignée dans un menu pour utiliser l'aide à la syntaxe pour votre programmation.

Ces instructions contrôlent l'entrée et la sortie depuis un programme pendant l'exécution. Elles vous permettent d'entrer des valeurs et affiche les réponses pendant l'exécution du programme.

Pour revenir à l'écran précédent sans sélectionner d'élément, appuyez sur $\boxed{\text{annul}}$.

Affichage d'un graphique avec Input

Input sans variable affiche le graphique courant. Vous pouvez déplacer le curseur libre, qui met à jour X et Y (et R et θ pour le format **PolarGC**). L'indicateur pause est activé. Appuyez sur $\boxed{\text{entrer}}$ pour reprendre l'exécution du programme.

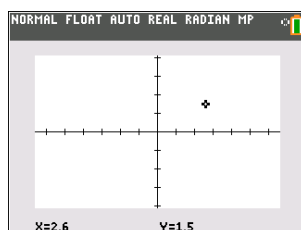
Input

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:GINPUT
:FnOff
:ZDecimal
:Input
:Disp X,Y
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmGINPUT
```



Programme

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmGINPUT
                                2.6
                                1.5
                                Done
█
```

Stocker une Valeur de variable avec Input

Input avec *variable* affiche une invite ? (point d'interrogation) pendant l'exécution. *variable* peut être un nombre réel, un nombre complexe, une liste, une matrice, une chaîne de caractères ou une fonction $Y=$. Pendant l'exécution du programme, entrez une valeur, pouvant être une expression, puis appuyez sur `[entrer]`. La valeur est évaluée et stockée dans *variable* et le programme reprend l'exécution.

Input [*variable*]

Vous pouvez afficher *text* ou le contenu de *Strn* (une variable de type chaîne) jusqu'à 26 caractères comme invite. Pendant l'exécution du programme, entrez une valeur après l'invite, puis appuyez sur `[entrer]`. La valeur est stockée dans *variable* et le programme reprend l'exécution.

Input ["*text*",*variable*]

Input [*Strn*,*variable*]

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:HINPUT
:Input A
:Input L1
:Input "Y1=",Y1
:Input "DATA=",LDATA
:Disp Y1(A)
:Disp Y1(L1)
:Disp Y1(LDATA)
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmHINPUT
?2
?{1,2,3}
Y1="2X+2"
DATA={4,5,6}
                                6
                                {4 6 8}
                                {10 12 14}
                                Done
.....
```

Remarque : Lorsqu'un programme invite à entrer des listes et des fonctions Yn pendant l'exécution, vous devez inclure des accolades ({ }) autour des éléments de la liste et des guillemets autour (") des expressions.

Prompt

Pendant l'exécution du programme, **Prompt** affiche chaque *variable*, une à la fois, suivie de =?. À chaque invite, entrez une valeur ou une expression pour chaque *variable*, puis appuyez sur **[Entrer]**. Les valeurs sont stockées et le programme reprend l'exécution.

Prompt *variableA[,variableB,...,variable n]*

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:WINDOW
:Prompt Xmin
:Prompt Xmax
:Prompt Ymin
:Prompt Ymax
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrømWINDOW
Xmin=?-10
Xmax=?10
Ymin=?-3
Ymax=?3
.....Done
█
```

Remarque : Les fonctions Y= ne sont pas valides avec **Prompt**.

Disp

Affichage de l'écran d'accueil

Disp (affichage) sans valeur affiche l'écran de calcul. Pour voir l'écran de calcul pendant l'exécution du programme, suivez l'instruction **Disp** avec une instruction **Pause**.

Afficher les valeurs et les messages

Disp avec une ou plusieurs *valeurs* affiche la valeur de chacune.

Disp [*valeur A,valeur B,valeur C,...,valeur n*]

- Si *valeur* est une variable, la valeur actuelle est affichée.
- Si *valeur* est une expression, elle est évaluée et le résultat est affiché sur le côté droit de la ligne suivante.
- Si *valeur* est un texte entre guillemets, il est affiché sur le côté droit de la ligne d'affichage actuelle. → n'est pas valide comme texte.

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:A
:DISP "THE ANSWER IS ",π/2
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmA
THE ANSWER IS
1.570796327
.....
Done
```

Si **Pause** est rencontré après **Disp**, le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Pour reprendre l'exécution, appuyez sur `[enter]`.

Remarque : Si une matrice ou une liste est trop longue pour être affichée entièrement, trois points (...) sont affichés dans la dernière colonne, mais on ne peut pas faire défiler la matrice ou la liste. Pour la faire défiler, utilisez **Pause valeur**.

DispGraph

DispGraph (affichage du graphe) affiche le graphique courant. Si **Pause** est rencontré après **DispGraph**, le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Appuyez sur `[enter]` pour reprendre l'exécution.

DispTable

DispTable (affichage du tableau) affiche le tableau courant. Le programme s'interrompt temporairement, afin de vous permettre d'examiner l'écran. Appuyez sur `[enter]` pour reprendre l'exécution.

Output(

Output(affiche *texte* ou *valeur* sur l'écran d'accueil commençant à *ligne* (1 à 10) et *colonne* (1 à 26), écrasant tout caractère existant.

Remarque : Vous voudrez peut-être faire précéder **Output(** de **ClrHome**.

Les expressions sont évaluées et les valeurs sont affichées selon les paramètres du mode courant. Les matrices sont affichées au format d'entrée et retournent à la ligne suivante. → n'est pas valide comme texte.

Output(ligne,colonne,"texte")

Output(ligne,colonne,valeur)

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: OUTPUT
:3+5→B
:ClrDraw
:Output(5,4,"ANSWER:"
:Output(5,12,B)
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmOUTPUT
.....Done.

ANSWER: 8
```

Pour **Output**(sur un écran partagé **Horiz**, la valeur maximum pour *ligne* est 4.

getKey

getKey retourne un chiffre correspondant à la dernière touche pressée, selon le diagramme de code de touches ci-dessous. Si aucune touche n'a été pressée, **getKey** retourne 0. Utilisez **getKey** dans des boucles pour transférer le contrôle, par exemple, lors de la création de jeux vidéo.

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: GETKEY
:While 1
:getKey→K
:While K=0
:getKey→K
:End
:Disp K
:If K=105
:Stop
:End█
```

Sortie

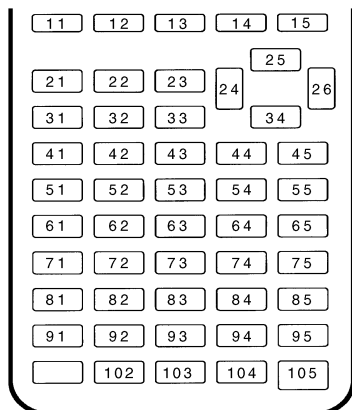
```
NORMAL FLOAT AUTO REAL RADIAN MP
prgmGETKEY

41
42
43
105
.....Done.
```

Remarque : **math**, **apps**, **prgm** et **entrer** ont été pressés pendant l'exécution du programme.

Remarque : Vous pouvez appuyer sur **on** à tout moment pendant l'exécution pour interrompre le programme.

Diagramme des codes des touches



ClrHome, ClrTable

ClrHome (effacer l'écran de calcul) efface l'écran de calcul pendant l'exécution du programme.

ClrTable (effacer la table de valeurs) efface les valeurs de la table pendant l'exécution du programme.

GetCalc{

GetCalc{ obtient le contenu de *variable* à partir d'une autre TI-82 Advanced Édition Python et le place dans *variable* sur l'unité réceptrice TI-82 Advanced Édition Python. *variable* peut être un nombre réel ou complexe, un élément de liste, un nom de liste, un élément de matrice, un nom de matrice, une chaîne de caractères, une variable Y=, une base de données graphique ou une image.

GetCalc{variable[,portflag]}

Par défaut, la TI-82 Advanced Édition Python utilise le port USB s'il est connecté. Si le câble USB n'est pas connecté, elle utilise le port I/O. Si vous voulez spécifier le port USB ou I/O, utilisez les nombres suivants :

portflag=0 utilise le port USB, s'il est connecté ;

portflag=1 utilise le port I/O ;

portflag=2 utilise le port d'E/S.

Remarque : la TI-82 Advanced Édition Python ne lie que les liens vers un autre TI-82 Advanced Édition Python.

eval()

eval() retourne une expression évaluée comme une chaîne avec 8 chiffres significatifs. L'expression doit se simplifier à une expression réelle.

eval(*expression*)

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: EVALHOME
:5→A
:eval(2A+7)
:
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrmEVALHOME
17
■
```

expr()

Convertit la chaîne de caractères contenue dans *chaîne* en une expression et l'exécute. *chaîne* peut être une chaîne ou une variable de chaîne.

expr(*chaîne*)

Programme

```
NORMAL FLOAT DEC REAL RADIAN MP
PROGRAM: EXPR
:2→X
:"5X"→Str1
:Disp Str1
:expr(Str1)→A
:Disp "A=",A
:■
```

Sortie

```
NORMAL FLOAT DEC REAL RADIAN MP
PrmEXPR
5X
A=
..... 10
..... Done
.....
```

toString()

Convertit une valeur en une chaîne où *valeur* peut être réelle, complexe, une expression évaluée, une liste ou une matrice. La chaîne *valeur* s'affiche au *format* (0) classique suivant le paramètre de mode AUTO/DEC ou en *format* (1) décimal.

toString(*valeur* [, *format*])

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: TOSTR
:1/2→A
:Disp toString(A^2+2)
:Disp toString(A^2+2,0)
:Disp toString(A^2+2,1)
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmTOSTR
9/4
9/4
2.25
..... Done
█
```

Chaîne→Équ(

String→Equ(convertit *string* en une équation et stocke l'équation dans *Yn*. *string* peut être une chaîne ou une variable de type chaîne. **String→Equ(** est l'inverse de **Equ→String(**

String→Equ(*string*,*Yn*)

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: STREQU
:"2X"→Str2
:String→Equ(Str2,Y2)
:Disp "Y2(-10)=",Y2(-10)
:█
```

Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PrgmSTREQU
Y2(-10)=
..... -20
..... Done
█
```

Instructions PRGM COULEUR

Cette section décrit le menu **COULEUR** et les numéros de couleur à utiliser comme arguments lorsqu'une option propose de définir la couleur comme **GraphColor{**.

Vous pouvez copier le jeton de couleur, comme **BLEU**, ou utiliser le numéro de la couleur, comme **10**, affiché dans le tableau ci-dessous.

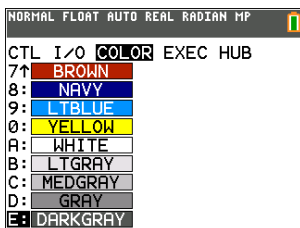
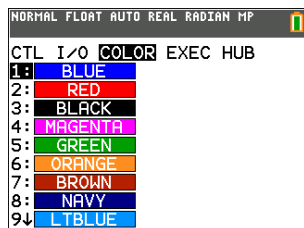
Menu PRGM COULEUR

Pour afficher le menu **PRGM COULEUR**, appuyez sur prgm ▶ seulement depuis l'éditeur de programme.

CTRL **E/S** **COULEUR** **EXEC** **HUB**

		Description
1:	BLEU	n°couleur : 10
2:	ROUGE	n°couleur : 11
3:	NOIR	n°couleur : 12
4:	MAGENTA	n°couleur : 13
5:	VERT	n°couleur : 14
6:	ORANGE	n°couleur : 15
7:	MARRON	n°couleur : 16
8:	BLEU MRN	n°couleur : 17
9:	BLEU CLR	n°couleur : 18
0:	JAUNE	n°couleur : 19
A:	BLANC	n°couleur : 20
B:	GRIS CLR	n°couleur : 21
C:	GRIS MOY	n°couleur : 22
D:	GRIS	n°couleur : 23
E:	GRIS FON	n°couleur : 24

Remarque : Vous pouvez également choisir un nom dans le menu var (sous-menu COULEUR).



Instructions PRGM EXÉC

Appel d'autres programmes comme sous-programmes

Sur la TI-82 Advanced Édition Python de calculatrices graphiques, tout programme stocké peut être appelé à partir d'un autre programme comme sous-programme. Entrez le nom du programme à utiliser comme sous-programme seul sur une ligne

Appel d'un programme à partir d'un autre programme

Vous pouvez entrer le nom d'un programme sur une ligne de commande des deux façons.

- Appuyez sur `[prgm]` `[↵]` pour afficher le menu **PRGM EXEC** et sélectionner le nom du programme `prgmnom` est collé sur l'emplacement actuel du curseur sur une ligne de commande.
- Sélectionnez `prgm` depuis le menu **PRGM CTL**, puis entrez le nom du programme.

`prgmnom`

Lorsque `prgmnom` est rencontré pendant l'exécution, la prochaine commande que le programme exécute est la première commande dans le deuxième programme. Il retourne à la commande suivante du premier programme lorsqu'il rencontre soit **Return** soit le **Return** implicite à la fin du deuxième programme.

Programme

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: VOLCYL
: Input "D=", D
: Input "H=", H
: PrgmAREACIR
: A*H→V
: Disp V
: █
```



Sortie

```
NORMAL FLOAT AUTO REAL RADIAN MP
PRgmVOLCYL
D=4
H=5
62.83185307
..... Done
█
```

Sous-programme ↓↑

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM: AREACIR
: D/2→R
: π*R²→A
: Return
```

Remarques à propos de l'appel de programmes

Les variables sont globales.

label utilisé avec **Goto** et **Lbl** est local au programme où il est situé. *label* dans un programme n'est pas reconnu par un autre programme. Vous ne pouvez pas utiliser **Goto** pour brancher vers un *label* dans un autre programme.

Return sort d'un sous-programme et retourne au programme appelant, même s'il est rencontré à l'intérieur d'une boucle.

Informations générales

Aide en ligne

education.ti.com/eguide

Sélectionnez votre pays pour obtenir d'autres informations relatives aux produits.

Contacter l'assistance technique TI

education.ti.com/ti-cares

Sélectionnez votre pays pour obtenir une assistance technique ou d'autres types de support.

Informations sur le service et la garantie

education.ti.com/warranty

Sélectionnez votre pays pour obtenir des informations sur la durée et les conditions de la garantie ou sur le service après-vente.

Garantie limitée. Cette garantie n'affecte pas vos droits statutaires.