

# Chapitre 17 Python

Le mode **Python** fournit un environnement d'exécution pour le langage de programmation Python. Vous pouvez utiliser le mode **Python** pour créer, enregistrer, modifier et exécuter des fichiers Python.

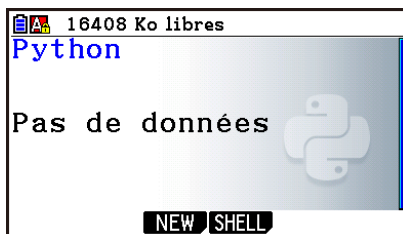
## Important !

- Le mode **Python** prend en charge la version 1.9.4 de MicroPython, qui a été adaptée pour fonctionner sur cette calculatrice. Notez que MicroPython est généralement différent de la version de Python s'exécutant sur un ordinateur. De plus, le mode **Python** ne prend pas en charge tous les modules, fonctions, commandes et bibliothèques de MicroPython.
- MicroPython est un projet open-source. Pour obtenir des informations sur la licence, consultez la section « MicroPython license information » (page  $\gamma-1$ ).
- Le mode **Python** exécute des instructions à l'aide du système de traitement MicroPython. Pour cette raison, les résultats de calcul et les autres données produites par ce mode peuvent différer des résultats d'exécution d'autres modes de fonctionnement.
- Python est une marque déposée de Python Software Foundation. Les symboles de marque de commerce (™) et de marque déposée (®) ne sont pas utilisés dans ce manuel.

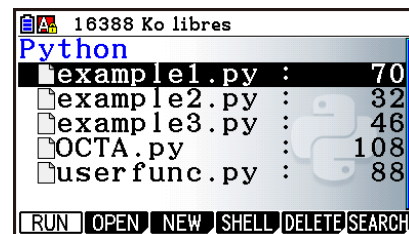
## 1. Aperçu du mode Python

### ■ Écran de liste des fichiers

La première chose qui apparaît lorsque vous sélectionnez le mode **Python** dans le menu principal est l'écran de la liste des fichiers.



Lorsqu'il n'y a aucun fichier py\* ou dossier dans la mémoire



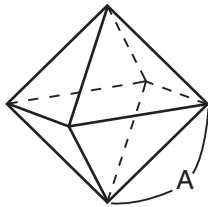
Lorsqu'il y a des fichiers py ou des dossiers dans la mémoire

\* Dans ce manuel, un fichier créé dans le mode **Python** (extension de nom de fichier py) est appelé un fichier « py ».

## ■ Séquence depuis la création du fichier py jusqu'à l'exécution du fichier

L'exemple ci-dessous explique la séquence de fonctionnement depuis la création d'un nouveau fichier py jusqu'à son exécution.

**Exemple :** Créer un fichier py qui calcule la superficie et le volume d'un octaèdre régulier et l'exécuter pour calculer la superficie et le volume lorsque la longueur d'un côté est égale à 10. Le nom du fichier est OCTA.

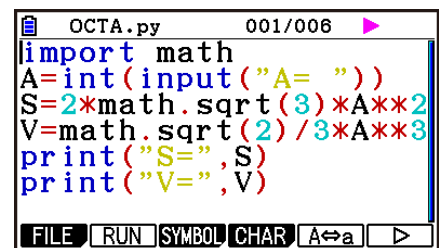


Vous pouvez calculer la superficie (S) et le volume (V) d'un octaèdre régulier lorsque la longueur d'un côté (A) est connue en utilisant les formules ci-dessous.

$$S = 2\sqrt{3} A^2, \quad V = \frac{\sqrt{2}}{3} A^3$$

Nous allons écrire un programme qui, lorsque le fichier py est exécuté, demande la saisie de A, qui est ensuite utilisé dans les formules ci-dessus pour générer les résultats des calculs. Dans ce manuel, un programme écrit en Python (et stocké dans un fichier py) est appelé un script « py ».

Dans cet exemple, nous allons saisir un script py comme celui qui est montré sur la capture d'écran à droite.

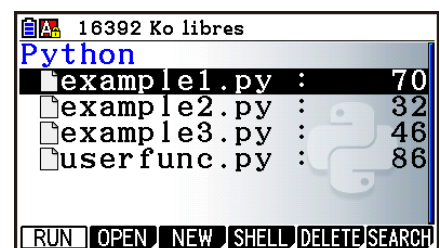


```
OCTA.py 001/006
import math
A=int(input("A= "))
S=2*math.sqrt(3)*A**2
V=math.sqrt(2)/3*A**3
print("S=",S)
print("V=",V)
```

### Procédure

1. À partir du menu principal, entrez dans le mode **Python**.

- Cela affiche l'écran de la liste des fichiers.
- Les noms des fichiers sont énumérés par ordre alphabétique.
- Les valeurs sur le côté droit de la liste de fichiers indiquent le nombre d'octets utilisés par chaque fichier py.



2. Créez un nouveau fichier et enregistrez-le sous un nom de fichier\*.

Ici, nous allons utiliser la procédure ci-dessous pour créer un nouveau fichier py nommé « OCTA ».

**F3** (NEW) **F5** (A↔a) **9** (O) **In** (C) **÷** (T) **X,0,T** (A) **EXE**

- Cela affiche l'écran de l'éditeur de script.

#### \* Noms de fichiers

- Vous pouvez saisir jusqu'à huit lettres (huit octets) pour définir un nom de fichier.
- Un nom de fichier peut être composé de caractères alphanumériques d'un octet (de A à Z, de a à z, et de 0 à 9). Cette calculatrice ne fait pas de distinction entre les majuscules et les minuscules.

#### **Important !**

Notez qu'un fichier avec un nom qui commence par un chiffre ou un nom qui est un mot réservé de Python ne sera pas exécuté.

3. Effectuez les opérations de touches ci-dessous pour saisir chaque ligne du script py.

- Vous pouvez utiliser la fonction de catalogue de **Python** (page 17-9) pour saisir plus efficacement les fonctions et les commandes. Dans les opérations de touches ci-dessous, les chaînes de texte qui sont soulignées et incluses entre parenthèses indiquent des noms de fonction et de commande saisies à l'aide de la fonction de catalogue.

| Effectuer cette opération de touches :                                                                                                                                                                                                                                                                                                             | Pour saisir ceci :    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| [SHIFT] [4] (CATALOG) [F6] (CAT) [3] (math) [C] (I)(import math) [EXE]<br>[EXE]                                                                                                                                                                                                                                                                    | import math           |
| [ALPHA] [F5] (A↔a) [X,θ,T] (A) [SHIFT] [=] (=)<br>[SHIFT] [4] (CATALOG) [F6] (CAT) [2] (Built-in) [C] (I) [8] (N) [÷] (T)<br>(int()) [EXE]<br>[SHIFT] [4] (CATALOG) [▲] [▲] (input()) [EXE]<br>[ALPHA] [x10 <sup>3</sup> ] (") [ALPHA] [F5] (A↔a) [X,θ,T] (A) [SHIFT] [=] (=) [ALPHA] [ ] (SPACE)<br>[ALPHA] [x10 <sup>3</sup> ] (") [▶] [▶] [EXE] | A=int(input("A= "))   |
| [ALPHA] [F5] (A↔a) [X] (S) [SHIFT] [=] (=) [2] [X]<br>[SHIFT] [4] (CATALOG) [F6] (CAT) [3] (math) [7] (M)(math.) [EXE]<br>[SHIFT] [x <sup>2</sup> ] (√) [3] [▶] [X] [ALPHA] [F5] (A↔a) [X,θ,T] (A) [x <sup>2</sup> ] [EXE]                                                                                                                         | S=2*math.sqrt(3)*A**2 |
| [ALPHA] [F5] (A↔a) [2] (V) [SHIFT] [=] (=)<br>[SHIFT] [4] (CATALOG)(math.) [EXE]<br>[SHIFT] [x <sup>2</sup> ] (√) [2] [▶] [÷] [3] [X] [ALPHA] [F5] (A↔a) [X,θ,T] (A) [▲] [3] [EXE]                                                                                                                                                                 | V=math.sqrt(2)/3*A**3 |
| [SHIFT] [4] (CATALOG) [F6] (CAT) [2] (Built-in) [4] (P) [6] (R)(print()) [EXE]<br>[ALPHA] [F5] (A↔a) [x10 <sup>3</sup> ] (") [ALPHA] [X] (S) [SHIFT] [=] (=) [ALPHA] [x10 <sup>3</sup> ] (")<br>[▶] [ALPHA] [F5] (A↔a) [X] (S) [▶] [EXE]                                                                                                           | print("S=",S)         |
| [SHIFT] [4] (CATALOG)(print()) [EXE]<br>[ALPHA] [x10 <sup>3</sup> ] (") [ALPHA] [F5] (A↔a) [2] (V) [SHIFT] [=] (=) [ALPHA] [x10 <sup>3</sup> ] (")<br>[▶] [ALPHA] [F5] (A↔a) [2] (V)                                                                                                                                                               | print("V=",V)         |

4. Effectuez l'opération de touches ci-dessous pour exécuter le script py actuellement affiché.

[F2] (RUN) [F1] (Oui)

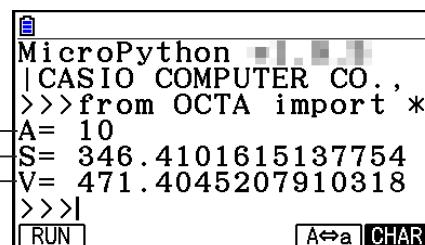
(Enregistre le script dans un fichier avant de l'exécuter.)

Les opérations ci-dessous sont exécutées une fois que le script est en cours d'exécution.

[1] [0] (Saisit la valeur de A)

[EXE]

Valeur de A saisie  
 Résultat de l'exécution  
 (valeur de S)  
 Résultat de l'exécution  
 (valeur de V)



- À la suite de l'opération ci-dessus, vous pouvez exécuter à nouveau le même script py en effectuant l'opération ci-dessous.

1. Appuyez sur [EXIT] pour revenir à l'écran de l'éditeur de script.
2. Appuyez sur [F2] (RUN).

---

## ■ Écran SHELL

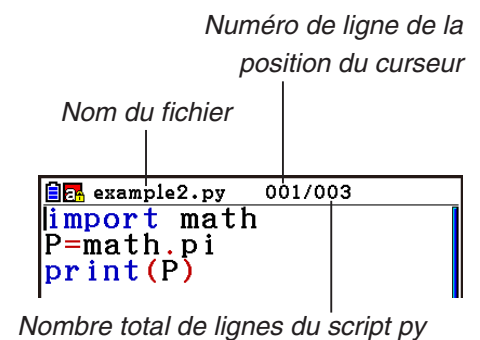
Le fait d'appuyer sur **F2** (RUN) à l'étape 4 de la procédure ci-dessus démarre le SHELL du mode **Python** qui peut être utilisé pour exécuter les scripts py. L'écran qui apparaît alors est appelé l'« écran SHELL ». L'écran SHELL vous permet non seulement d'exécuter les scripts py qui ont été enregistrés en tant que fichiers, mais également de saisir directement des commandes et des expressions et les exécuter une ligne à la fois. Pour plus d'informations à propos du SHELL, voir « Utilisation du SHELL » (page 17-14).

- Si un script py ne fonctionne pas normalement en raison d'une erreur de saisie, l'exécution du script à l'étape 4 affiche un message d'erreur en lettres rouges. Appuyez sur **EXIT** pour revenir à l'écran de l'éditeur de script à partir de l'écran SHELL. Pour plus d'informations sur la façon de corriger un script py, voir « Modification d'un fichier py » (page 17-18).

---

## ■ Écran de l'éditeur de script

Vous pouvez utiliser l'écran de l'éditeur de script qui apparaît à l'étape 2 ci-dessus pour saisir jusqu'à 300 lignes, chacune pouvant contenir jusqu'à 255 caractères. La barre d'état de l'écran de l'éditeur affiche le nom du fichier py actuellement ouvert, le nombre total de lignes du script py, et le numéro de ligne de la position actuelle du curseur.



Pour plus d'informations sur la façon d'ouvrir un fichier py et de vérifier son contenu, et sur la façon de déboguer un script py et de le modifier, voir « Modification d'un fichier py » (page 17-18).

## 2. Menu de fonctions de Python

---

### ■ Menu de fonctions de l'écran de la liste des fichiers

S'il n'existe pas de fichiers py dans la mémoire, seules les options {NEW} et {SHELL} sont disponibles dans le menu ci-dessous.

- **{RUN}/{OPEN}** ... exécute ou ouvre un fichier py enregistré afin de le modifier
- **{NEW}** ... affiche un écran d'enregistrement de nom de fichier lors de la création d'un nouveau fichier py
- **{SHELL}** ... exécute le shell et affiche l'écran SHELL
- **{DELETE}** ... supprime le fichier py spécifié
- **{SEARCH}** ... recherche un nom de fichier

---

## ■ Menu de fonctions pour enregistrer le nom d'un nouveau fichier py

- **{A↔a}** ... bascule entre la saisie en majuscules et minuscules

---

## ■ Menu de fonctions de l'écran de l'éditeur de script

- **{FILE}**
  - **{SAVE}** ... écrase le fichier py actuellement ouvert
  - **{SAVE • AS}** ... enregistre le fichier py actuellement ouvert sous un nom différent
- **{RUN}** ... affiche l'écran SHELL et exécute le script py actuellement affiché
- **{SYMBOL}** ... affiche un menu de fonctions de saisie de symboles
- **{CHAR}** ... affiche un menu de saisie de caractères alphanumériques, de symboles et d'opérateurs
- **{A↔a}** ... bascule entre la saisie en majuscules et minuscules
- **{COMMAND}** ... affiche un menu de commandes de branche conditionnelle et de boucle  
Voir « Utilisation du menu de fonctions pour saisir des commandes (branches conditionnelles ou boucles) en tant que blocs d'instructions » (page 17-8).
- **{OPERAT}** ... affiche un menu de saisie d'opérateurs (= != > < % | ^ & ~)
- **{JUMP}** ... affiche un menu de fonctions de saut à la ligne
  - **{TOP}** ... saute à la première ligne d'un script py
  - **{BOTTOM}** ... saute à la dernière ligne d'un script py
  - **{LINE}** ... affiche une boîte de dialogue pour spécifier une ligne et saute à la ligne spécifiée du script py
- **{SEARCH}** ... recherche la chaîne spécifiée

---

## ■ Menu de fonctions de l'écran SHELL

- **{RUN}** ... exécute l'expression ou la commande saisie dans la dernière ligne (ligne d'invite) de l'écran SHELL
- **{A↔a}** ... bascule entre la saisie en majuscules et minuscules
- **{CHAR}** ... affiche un menu de saisie de caractères alphanumériques, de symboles et d'opérateurs

### 3. Saisie de texte et de commandes

Il y a trois façons de saisir du texte et des commandes dans le mode **Python**.

- Utilisation du clavier pour saisir des caractères, des symboles et des fonctions (Voir la procédure ci-dessous.)
- Saisie via le menu de fonctions
  - Saisie de caractères alphanumériques, de symboles et d'opérateurs (page 17-7)
  - Saisie de commandes de branche conditionnelle et de commandes de boucle (page 17-8)
- Utilisation du catalogue (liste de fonctions ou de commandes) pour sélectionner un élément et le saisir (page 17-9)

#### ■ Utilisation du clavier pour saisir directement des commandes

À partir de l'écran de l'éditeur de script ou de l'écran SHELL, vous pouvez utiliser le clavier de la calculatrice pour saisir des chiffres, des caractères alphanumériques et des fonctions ( $\sqrt{\quad}$ , log, etc.) qui sont affectés à chaque touche.

#### • Utilisation des touches pour saisir des chiffres, des opérateurs, des parenthèses et des fonctions

Le tableau ci-dessous montre la saisie faite (nombre, opérateur, parenthèses ou fonction) lorsque vous appuyez sur une touche ou appuyez sur **SHIFT** puis sur une touche.

| Effectuer cette opération de touches : | Saisit cela : |
|----------------------------------------|---------------|
| <b>0</b> à <b>9</b>                    | 0 à 9         |
| <b>x<sup>2</sup></b>                   | **2           |
| <b>^</b>                               | **            |
| <b>X,θ,T</b>                           | x             |
| <b>log</b>                             | log10()       |
| <b>In</b>                              | log()         |
| <b>sin</b>                             | sin()         |
| <b>cos</b>                             | cos()         |
| <b>tan</b>                             | tan()         |
| <b>(</b>                               | (             |
| <b>)</b>                               | )             |
| <b>.</b>                               | .             |
| <b>,</b>                               | ,             |
| <b>×</b>                               | *             |
| <b>÷</b>                               | /             |

| Effectuer cette opération de touches :               | Saisit cela : |
|------------------------------------------------------|---------------|
| <b>+</b>                                             | +             |
| <b>-</b>                                             | -             |
| <b>x10<sup>n</sup></b>                               | e             |
| <b>SHIFT</b> <b>x<sup>2</sup></b> ( $\sqrt{\quad}$ ) | sqrt()        |
| <b>SHIFT</b> <b>In</b> ( $e^x$ )                     | exp()         |
| <b>SHIFT</b> <b>sin</b> (Asn)                        | asin()        |
| <b>SHIFT</b> <b>cos</b> (Acs)                        | acos()        |
| <b>SHIFT</b> <b>tan</b> (Atn)                        | atan()        |
| <b>SHIFT</b> <b>)</b> ( $x^{-1}$ )                   | ** -1         |
| <b>SHIFT</b> <b>×</b> ( { )                          | {             |
| <b>SHIFT</b> <b>÷</b> ( } )                          | }             |
| <b>SHIFT</b> <b>+</b> ( [ )                          | [             |
| <b>SHIFT</b> <b>-</b> ( ] )                          | ]             |
| <b>SHIFT</b> <b>0</b> (i)                            | 1j            |
| <b>SHIFT</b> <b>.</b> (=)                            | =             |
| <b>SHIFT</b> <b>x10<sup>n</sup></b> ( $\pi$ )        | pi            |

## Important !

Parmi les chaînes de texte ci-dessus qui sont saisies à l'aide d'opérations de touches, la fonction  $\log()$  et d'autres fonctions qui sont suivies par des parenthèses,  $e$  (base d'un logarithme naturel) et  $\pi$  sont des fonctions du module `math`. Pour utiliser ces fonctions, vous devez d'abord importer le module `math`.\* Pour plus d'informations, voir « Catégories de commandes » (page 17-10) et « Exemple d'opération : Pour utiliser les fonctions du module `math` » (page 17-13).

\* Si vous utilisez `import` à la place de `from` pour charger le module, vous devez ajouter « `math.` » avant chaque fonction que vous utilisez. Voir « Utilisation des modules (`import`) » (page 17-12) pour plus d'informations.

---

### • Saisie de touches alphabétiques

Si la première saisie est effectuée immédiatement après avoir appuyé sur `[ALPHA]`, ou si elle a été effectuée en verrouillage alpha en appuyant sur `[SHIFT][ALPHA]` (page 1-2), alors une pression sur une touche permet de saisir le caractère indiqué en rouge sur la touche, un espace ou des guillemets (").

- Choisir `{NEW}` ou `{OPEN}` sur l'écran de la liste des fichiers va afficher l'écran d'éditeur de script et verrouiller automatiquement la saisie en minuscules alphanumériques.

---

### • Mise en retrait automatique lorsqu'un retour à la ligne est saisi

Sur l'écran de l'éditeur de script du mode **Python**, appuyez sur `[EXE]` pour saisir un retour à la ligne.

- Le fait d'appuyer sur `[EXE]` après une ligne qui se termine par un deux-points (:) mettra automatiquement la nouvelle ligne en retrait de deux espaces de plus que la ligne précédente (mise en retrait automatique).
- Le fait d'appuyer sur `[EXE]` lorsque le curseur est placé sur une ligne en retrait mettra en retrait la ligne après le retour à la ligne du même nombre de caractères que la ligne en retrait au-dessus d'elle.
- Pour saisir un retour à la ligne sans mettre la nouvelle ligne en retrait, appuyez sur `[SHIFT][EXE]`.

Les codes de retour à la ligne ne sont pas affichés dans le mode **Python**.

---

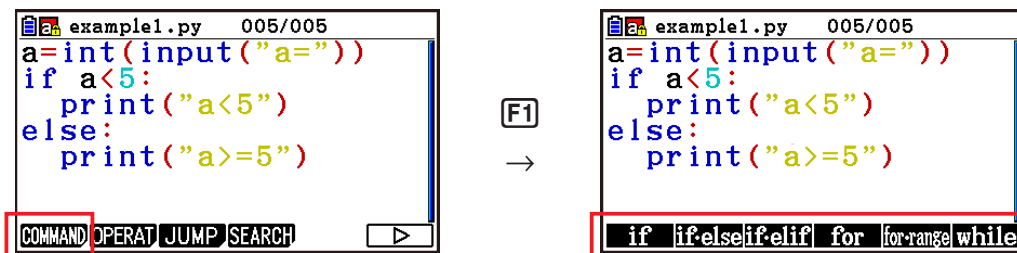
## ■ Utilisation du menu de fonctions pour saisir du texte (caractères alphanumériques, symboles, opérateurs)

Utilisez les menus de fonctions donnés dans le tableau ci-dessous pour saisir des caractères alphanumériques, des symboles ou des opérateurs.

| Opération de touches                                                        |                                                                             | Caractères pouvant être saisis (alphanumériques, symboles, opérateurs)                                                                                                                                   |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Écran de l'éditeur de script                                                | Écran SHELL                                                                 |                                                                                                                                                                                                          |
| <code>[F3]</code> (SYMBOL) (symboles)                                       | —                                                                           | , () [] : ; # ' " \ _                                                                                                                                                                                    |
| <code>[F4]</code> (CHAR) (caractères alphanumériques, symboles, opérateurs) | <code>[F6]</code> (CHAR) (caractères alphanumériques, symboles, opérateurs) | ! " # \$ % & ' ( ) * + , - . / 0 1 2 3<br>4 5 6 7 8 9 : ; < = > ? @ A B C D E F<br>G H I J K L M N O P Q R S T U V W X Y<br>Z [ \ ] ^ _ ` a b c d e f g h i j k l<br>m n o p q r s t u v w x y z {   } ~ |
| <code>[F6]</code> (>) <code>[F2]</code> (OPERAT) (opérateurs)               | —                                                                           | = != > < %   ^ & ~                                                                                                                                                                                       |

## ■ Utilisation du menu de fonctions pour saisir des commandes (branches conditionnelles ou boucles) en tant que blocs d'instructions

À partir de l'écran de l'éditeur de script, vous pouvez utiliser le menu {COMMAND} du menu de fonctions pour saisir des blocs d'instructions de commande de branche conditionnelle ou de commande de boucle.



| Effectuer cette opération de touches :                   | Pour saisir ce bloc d'instructions :*     | Effectuer cette opération de touches :                     | Pour saisir ce bloc d'instructions :* |
|----------------------------------------------------------|-------------------------------------------|------------------------------------------------------------|---------------------------------------|
| <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F1</b> (if)      | if□ :<br>□□                               | <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F4</b> (for)       | for□i□in□ :<br>□□                     |
| <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F2</b> (if·else) | if□ :<br>□□<br>else:<br>□□                | <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F5</b> (for·range) | for□i□in□range( ):<br>□□              |
| <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F3</b> (if·elif) | if□ :<br>□□<br>elif:<br>□□<br>else:<br>□□ | <b>F6</b> (▷) <b>F1</b> (COMMAND)<br><b>F6</b> (while)     | while□ :<br>□□                        |

\* Les symboles (□) de boîte dans les tableaux ci-dessus représentent des espaces vides. Les symboles de boîte n'apparaissent pas sur l'écran. Les lignes verticales (|) correspondent à l'emplacement du curseur immédiatement après la saisie. Le caractère de ligne verticale (|) n'est pas inséré.

- En plus des six blocs d'instructions ci-dessus, vous pouvez également utiliser le catalogue (page 17-9) pour saisir les blocs d'instructions illustrés ci-dessous.
  - **for:range(,)**
  - **for:range(,,)**
  - **if·and:else**
  - **if·or:else**
  - **def:return**
- L'écran SHELL ne permet de saisir qu'une seule une ligne, donc la saisie de bloc d'instructions n'est pas autorisée. À partir de l'écran SHELL, le fait de choisir un menu qui insère des blocs d'instructions ne fera qu'insérer la première ligne du bloc.

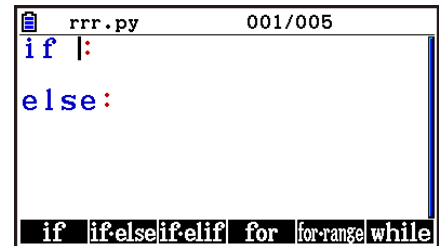


---

## • Exemple : Pour saisir une instruction if...else

1. Sur l'écran de l'éditeur de script, déplacez le curseur sur la ligne où vous voulez insérer le bloc d'instructions, puis appuyez sur **F6**(▷) **F1**(COMMAND) **F2**(if·else).

- Cela insère le bloc d'instructions if...else, avec le curseur placé pour la saisie de la condition « if ».
- Les lignes 2 et 4 sont mises en retrait de deux espaces automatiquement.



```
rrr.py 001/005
if :
else:

```

if if-else if-elif for for-range while

---

## ■ Saisie d'une commande depuis le catalogue (fonction de catalogue)

Le catalogue est une liste de fonctions et de commandes. Vous pouvez effectuer la saisie en affichant l'écran du catalogue, puis en sélectionnant la fonction ou la commande souhaitée. Cette opération est possible à la fois sur l'écran de l'éditeur de script et sur l'écran SHELL.\*

\* Uniquement lorsque le curseur est sur la ligne d'invite.

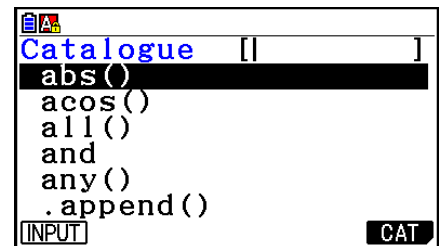
Dans les opérations de cette section, les fonctions, les commandes et les autres éléments qui peuvent être saisis depuis le catalogue sont collectivement appelés des « commandes ».

---

### • Pour entrer une commande depuis le catalogue

1. Sur l'écran de l'éditeur de script ou sur l'écran SHELL, appuyez sur **SHIFT** **4** (CATALOG).

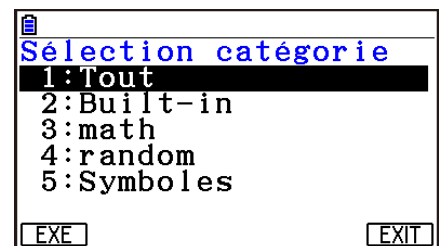
- Cela affiche l'écran de la liste des commandes du catalogue.
- Si vous voulez sélectionner une commande sur cet écran afin de la saisir, allez à l'étape 4 de cette procédure. Si vous voulez sélectionner une catégorie, allez à l'étape 2 de cette procédure.



```
Catalogue [ | ]
abs()
acos()
all()
and
any()
.append()
INPUT CAT
```

2. Appuyez sur **F6**(CAT).

- Cela affiche la liste des catégories.
- Pour plus d'informations sur chaque catégorie, voir « Catégories de commandes » (page 17-10).



```
Sélection catégorie
1: Tout
2: Built-in
3: math
4: random
5: Symboles
EXE EXIT
```

3. Appuyez sur une touche numérique (de **1** à **5**) qui correspond à la catégorie que vous souhaitez sélectionner.

Vous pouvez aussi utiliser ▲ et ▼ pour changer la catégorie en surbrillance, puis appuyer sur **EXE**.

- L'écran de la liste de commande est alors à nouveau affiché, montrant uniquement les commandes à l'intérieur de la catégorie choisie.

4. Utilisez ▲ et ▼ pour sélectionner la commande que vous voulez saisir.
- Vous pouvez faire défiler les écrans en appuyant sur [SHIFT] ▲ ou [SHIFT] ▼.
  - Vous pouvez aussi saisir une chaîne jusqu'à huit caractères pour rechercher les commandes qui commencent par les caractères que vous saisissez. Pour plus de détails sur la façon de saisir ces caractères, voir « Utilisation du catalogue pour rechercher et saisir une commande » (page 17-11).
5. Après avoir sélectionné la commande que vous voulez saisir, appuyez sur [F1] (INPUT) ou [EXE].
- Lorsque vous entrez dans le mode **Python** et que vous affichez le catalogue, la commande qui a été choisie la dernière fois que vous avez affiché le catalogue s'affiche en premier.

### Catégories de commandes

Le contenu de chaque catégorie du catalogue du mode **Python** est décrit dans le tableau ci-dessous.

| Nom de la catégorie | Description                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tout                | Affiche une liste de toutes les fonctions et commandes incluses dans le catalogue du mode <b>Python</b> .                                                                               |
| Built-in            | Affiche une liste des fonctions et commandes intégrées de Python. Les fonctions et les commandes incluses dans cette catégorie peuvent être utilisées sans importer le moindre module.* |
| math                | Affiche une liste des commandes qui importent le module math* de Python (les fonctions mathématiques) et les fonctions incluses dans le module math.                                    |
| random              | Affiche une liste des commandes qui importent le module random* de Python (fonctions de nombres aléatoires) et les fonctions incluses dans le module random.                            |
| Symboles            | Affiche une liste des symboles et des opérateurs.                                                                                                                                       |

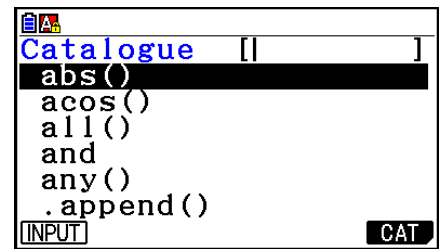
- \* Pour plus de détails sur les modules, voir « Utilisation des modules (*import*) » (page 17-12).
- La recherche de commandes ne peut pas être effectuée lorsque la liste de commandes produite en sélectionnant « Symboles » est affichée.
  - Contrairement aux catalogues d'autres modes (page 1-12), il n'y a pas de fonction de l'historique des commandes ou de fonction de QR Code dans le mode **Python**.

---

## • Utilisation du catalogue pour rechercher et saisir une commande

1. Sur l'écran de l'éditeur de script ou sur l'écran SHELL, appuyez sur **[SHIFT] [4]** (CATALOG).

- Cela affiche l'écran de la liste des commandes du catalogue. La commande actuellement sélectionnée est en surbrillance.
- Au besoin, effectuez les étapes 2 et 3 de la section « Pour entrer une commande depuis le catalogue » (page 17-9) pour sélectionner une catégorie (en dehors de « Symboles »).



2. Saisissez la première lettre ou jusqu'aux huit premières lettres de la commande que vous voulez trouver.

- Chaque fois que vous saisissez un caractère, la première commande qui correspond aux lettres que vous avez saisies jusqu'à présent est mise en surbrillance.

3. Utilisez **[↑]** et **[↓]** si nécessaire pour mettre la commande désirée en surbrillance, puis appuyez sur **[F1]** (INPUT) ou **[EXE]**.

---

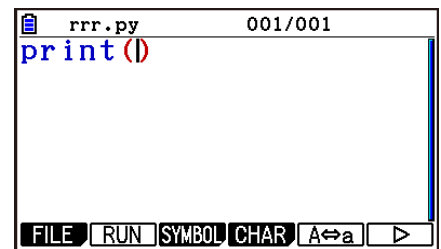
## • Exemple de saisie 1 : Pour utiliser la fonction de catalogue afin de saisir « print() »

1. Sur l'écran de l'éditeur de script, déplacez le curseur sur la ligne où vous voulez insérer la commande, puis appuyez sur **[SHIFT] [4]** (CATALOG).

2. Appuyez sur **[F6]** (CAT) pour afficher l'écran de catégorie, puis appuyez sur **[2]** (Built-in).

3. Appuyez sur **[4]** (P) **[6]** (R) pour rechercher des commandes qui commencent par « pr ».

4. Après avoir vérifié que « print() » est sélectionné, appuyez sur **[EXE]**.



- Pour obtenir un exemple utilisant « print() », voir les exemples au début de ce chapitre et les exemples 1, 2 et 4 dans la section « Exemples de scripts » (page 17-24).

---

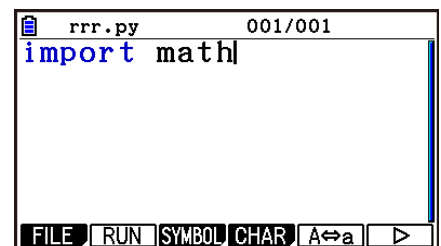
## • Exemple de saisie 2 : Pour utiliser la fonction de catalogue pour saisir « import math »

1. Sur l'écran de l'éditeur de script, déplacez le curseur sur la ligne où vous voulez insérer la commande, puis appuyez sur **[SHIFT] [4]** (CATALOG).

2. Appuyez sur **[F6]** (CAT) pour afficher l'écran de catégorie, puis appuyez sur **[3]** (math).

3. Appuyez sur **[I]** (l) pour rechercher des commandes qui commencent par « i ».

4. Après avoir vérifié que « import math » est sélectionné, appuyez sur **[EXE]**.



- Pour plus d'informations à propos d'« import math », voir « Utilisation des modules (import) » (page 17-12).

## ■ Utilisation des modules (*import*)

Dans le mode **Python**, vous pouvez utiliser les fonctions intégrées de Python ainsi que les fonctions des modules `math` et `random`. Cependant, pour utiliser une fonction contenue dans un module, vous devez d'abord importer (*import*) le module.

| Syntaxe de <i>import</i>                                                               | Description                                                                                          |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <code>import &lt;nom du module&gt;</code>                                              | Importe le module (fichier py) spécifié par <nom du module>.                                         |
| <code>from &lt;nom du module&gt; import *</code>                                       | Importe tous les éléments* inclus dans le module spécifié par <nom du module>.                       |
| <code>from &lt;nom du module&gt; import &lt;élément&gt;<br/>[, &lt;élément&gt;]</code> | Importe les éléments spécifiés (fonctions, etc.) inclus dans le module spécifié par <nom du module>. |

\* Un élément avec un nom qui commence par un caractère de soulignement (`_`) ne peut pas être importé.

- Un fichier py unique écrit par un script py est appelé un « module ». Les fichiers py sont importés en utilisant la même syntaxe que *import*.
- Pour un exemple d'importation utilisant un fichier py, voir « Exemple 4 : Importation d'un fichier py » dans la section « Exemples de scripts » (page 17-26).

### Exemples de syntaxe :

```
import math (Importe le module math.)
```

```
from math import pi, sqrt (Du module math, importe pi et sqrt seulement.)
```

- Si vous utilisez *import* pour importer un module, vous devez inclure le « <nom du module>. » avant le nom d'une fonction donnée pour utiliser cette fonction. Par exemple, pour utiliser *pi* du module `math`, vous devriez écrire « `math.pi` ».
- Lorsque vous utilisez *from* pour importer un module, n'utilisez pas la syntaxe ci-dessous.

```
<nom du module>.<nom de la fonction>
```

Exécuter une opération d'importation avec cette syntaxe provoquera une erreur.

## • Exemple d'opération : Pour utiliser les fonctions du module math

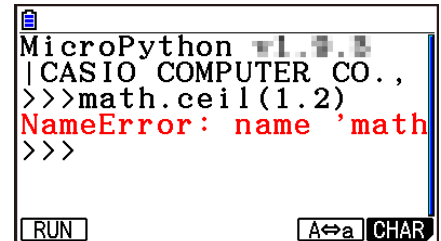
1. À partir de l'écran de la liste des fichiers, appuyez sur **F4** (SHELL).
2. Appuyez sur **SHIFT 4** (CATALOG) pour afficher le catalogue.
3. Appuyez sur **F6** (CAT) pour afficher l'écran de catégorie, puis appuyez sur **3** (math).
4. Effectuez la séquence d'opération de touches ci-dessous.

**7** (M) **EXE** (saisit « math. »)

**SHIFT 4** (CATALOG) **In** (C) **EXE** (saisit « ceil() »)

5. Appuyez sur **1** **◦** **2** **EXE**.

- Le module math n'est pas importé, donc la fonction « ceil() » du module math provoque une erreur.

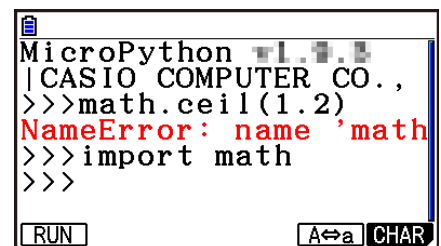


```
MicroPython 1.17.0
|CASIO COMPUTER CO.,
>>>math.ceil(1.2)
NameError: name 'math'
>>>
```

6. Appuyez sur **SHIFT 4** (CATALOG) pour afficher le catalogue de nouveau, appuyez sur **(I)** pour sélectionner « import math », puis appuyez sur **EXE**.

7. Appuyez une nouvelle fois sur **EXE** pour exécuter « import math ».

- Cela importe le module math.



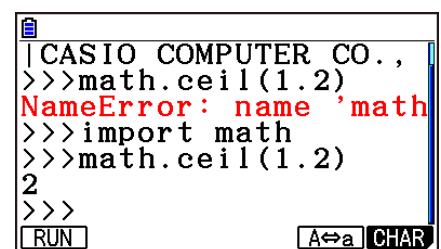
```
MicroPython 1.17.0
|CASIO COMPUTER CO.,
>>>math.ceil(1.2)
NameError: name 'math'
>>>import math
>>>
```

8. Utilisez **▲** pour sélectionner la ligne « math.ceil(1.2) » que vous avez saisie à l'étape 5 ci-dessus, puis appuyez sur **EXE**.

- Cela permet de copier la ligne sélectionnée dans la ligne d'invite.

9. Appuyez sur **EXE**.

- Cela affiche le résultat de l'exécution de « math.ceil(1.2) ».

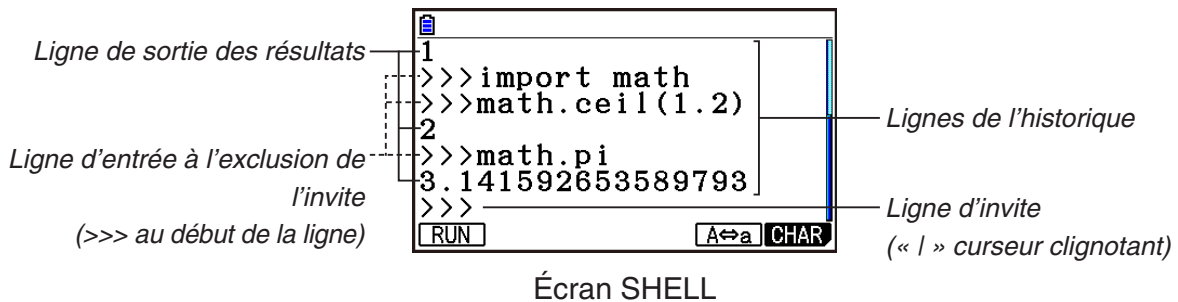


```
|CASIO COMPUTER CO.,
>>>math.ceil(1.2)
NameError: name 'math'
>>>import math
>>>math.ceil(1.2)
2
>>>
```

- L'opération ci-dessus peut être effectuée en exécutant directement les commandes sur l'écran SHELL. Pour plus d'informations à propos du SHELL, voir « Utilisation du SHELL » (page 17-14).
- Pour utiliser une fonction du module math et/ou du module random dans un script py, la commande d'importation de module applicable doit être écrite une fois sur une ligne avant la première utilisation de la fonction.

## 4. Utilisation du SHELL

L'écran SHELL fournit une ligne de commande interactive qui peut être utilisée pour saisir des expressions et des commandes et afficher leurs résultats. Vous pouvez saisir directement une expression ou une commande sur l'écran SHELL et l'exécuter pour obtenir un résultat. L'exécution d'un fichier py affichera son résultat sur l'écran SHELL.



- Jusqu'à 255 caractères peuvent être saisis dans une ligne d'invite, sans compter les caractères >>> au début de la ligne.
- Une ligne de sortie de résultats peut contenir jusqu'à 509 caractères.
- Toutes les lignes au-dessus de la ligne d'invite indiquent l'historique des entrées/sorties. Jusqu'à 200 lignes d'historique, qui sont constituées de lignes de sortie de résultats et de lignes d'entrée à l'exclusion de l'invite, sont conservées sur l'écran SHELL. Cela est à ajouter à la ligne d'invite. Il est possible de faire défiler les lignes de l'historique de la plus récente à la plus ancienne, et une ligne de l'historique peut être copiée sur la ligne d'invite. Le contenu de l'historique est conservé\* même si vous changez d'écran dans le mode **Python**. Ce contenu est supprimé lorsque vous entrez dans un mode de fonction autre que le mode **Python**.

\* Même si le contenu de l'historique est conservé lorsque vous passez de l'écran SHELL à un autre écran du mode **Python**, les variables utilisées par le SHELL sont initialisées à chaque fois que vous passez à un autre écran. Voir « Initialisation du SHELL » (page 17-17) pour plus d'informations.

---

### ■ Opérations de base de l'écran SHELL

---

#### • Pour afficher l'écran SHELL

Le fait d'effectuer n'importe laquelle des opérations ci-dessous entraînera l'affichage de l'écran SHELL.

- À partir de l'écran de la liste des fichiers, appuyez sur **F4** (SHELL).
- Sur l'écran de la liste des fichiers, sélectionnez un fichier py et appuyez sur **F1** (RUN) ou sur **EXE**. Cette commande exécute le fichier py et affiche l'écran SHELL.
- À partir de l'écran de l'éditeur de script, appuyez sur **F2** (RUN). Cela exécute le script py affiché et affiche l'écran SHELL.

---

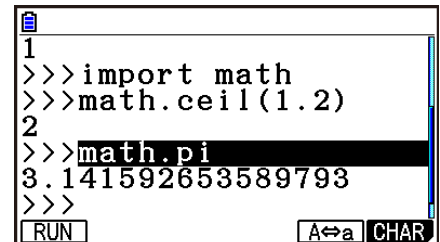
## • Pour exécuter une commande depuis l'écran SHELL

Voir « Saisie d'une commande directement sur l'écran SHELL et exécution de la commande » (page 17-16).

---

## • Pour faire défiler l'écran SHELL verticalement (pour afficher les lignes de l'historique)

Appuyez sur ▲ ou ▼. La ligne de l'historique actuellement sélectionnée est celle qui est en surbrillance.



```
1
>>>import math
>>>math.ceil(1.2)
2
>>>math.pi
3.141592653589793
>>>
>>>
```

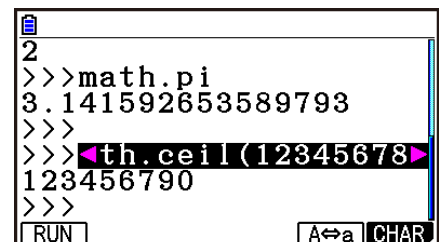
---

## • Pour faire défiler horizontalement une ligne sur l'écran SHELL (ligne de l'historique ou ligne d'invite)

1. Utilisez ▲ et ▼ pour mettre en surbrillance la ligne que vous souhaitez faire défiler.

2. Appuyez sur ◀ ou ▶.

- Une ligne de l'historique qui est trop longue pour être affichée complètement est indiquée par des flèches (◀ et ▶) qui montrent dans quelle direction il y a d'autres caractères à afficher. Les flèches (◀ et ▶) ne sont pas affichées dans la ligne d'invite même si son contenu est trop long pour être affiché complètement.



```
2
>>>math.pi
3.141592653589793
>>>
>>>◀th.ceil(12345678▶
123456790
>>>
>>>
```

---

## • Pour copier une ligne de l'historique de l'écran SHELL sur la ligne d'invite

Utilisez ▲ et ▼ pour mettre en surbrillance la ligne que vous souhaitez copier puis appuyez sur [EXE].

Comme exemple d'opérations actuelles, voir l'étape 8 sous « Exemple d'opération : Pour utiliser les fonctions du module math » (page 17-13).

---

## • Pour revenir à l'écran de la liste des fichiers depuis l'écran SHELL

Appuyez sur [EXIT].

- Si l'écran SHELL actuellement affiché l'a été en appuyant sur [F2] (RUN) pour exécuter un script py depuis l'écran de l'éditeur de script, la première fois que vous appuyez sur [EXIT], l'écran de l'éditeur de script est affiché à nouveau. Dans ce cas, appuyez sur [EXIT] pour revenir à l'écran de la liste des fichiers.
- Passer de l'écran SHELL à un autre écran du mode **Python** va initialiser toutes les variables utilisées par le SHELL. Voir « Initialisation du SHELL » (page 17-17) pour plus d'informations.

## ■ Saisie d'une commande directement sur l'écran SHELL et exécution de la commande

Vous pouvez saisir une expression ou une commande d'une seule ligne dans la ligne d'invite de l'écran SHELL et l'exécuter. Les opérations des exemples ci-dessous commencent toutes avec l'écran SHELL déjà affiché.

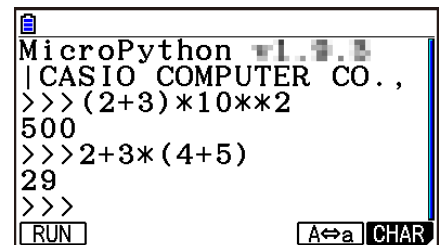
### • Exemple d'opérations 1 : Pour effectuer des opérations arithmétiques simples

$$(2+3) \times 10^2 = 500$$

`( 2 + 3 ) x 1 0 x2 EXE`

$$2+3 \times (4+5) = 29$$

`2 + 3 x ( 4 + 5 ) EXE`



```
MicroPython v1.17.0
|CASIO COMPUTER CO.,
>>>(2+3)*10**2
500
>>>2+3*(4+5)
29
>>>
[RUN] A↔a CHAR
```

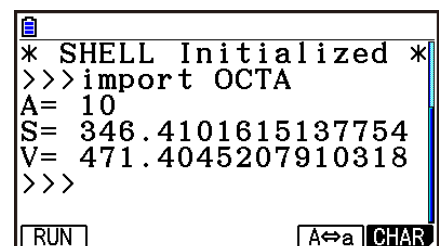
Notez les points importants ci-dessous.

- Utilisez la touche `=`, et non la touche `-` pour saisir un signe moins.
- La précision du calcul dans le mode **Python** est différente de celle des calculs effectués dans le mode **Exe-Mat**.

### • Exemple d'opérations 2 : Pour rappeler et exécuter un fichier py depuis l'écran SHELL

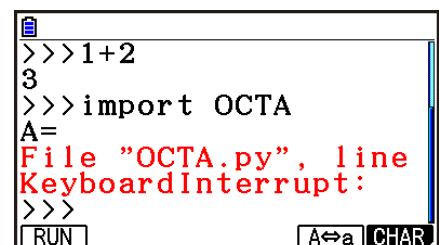
L'opération ci-dessous utilise le fichier « OCTA.py » créé à l'aide de l'exemple de la section « Séquence depuis la création du fichier py jusqu'à l'exécution du fichier » (page 17-2). Il suppose que le SHELL est déjà en cours d'exécution. Si vous souhaitez appeler le fichier « OCTA.py » depuis le SHELL, le SHELL doit avoir été démarré alors que la liste de fichiers contenant le fichier « OCTA.py » était à l'écran.

`SHIFT 4 (CATALOG) F6 (CAT) 2 (Built-in)`  
`(I) 7 (M) 4 (P)(import) EXE`  
`SHIFT ALPHA F5 (A↔a) 9 (O) In (C) ÷ (T) X,θ,T (A) EXE`  
`ALPHA 1 0 (Saisie de la valeur de A) EXE`



```
* SHELL Initialized *
>>>import OCTA
A= 10
S= 346.4101615137754
V= 471.4045207910318
>>>
[RUN] A↔a CHAR
```

- Pour arrêter l'exécution d'un script, appuyez sur `AC`. Cela va afficher le message « KeyboardInterrupt: » avec le curseur clignotant dans la ligne du bas (ligne d'invite) de l'écran.



```
>>>1+2
3
>>>import OCTA
A=
File "OCTA.py", line
KeyboardInterrupt:
>>>
[RUN] A↔a CHAR
```



## • Opération *input* dans le mode Python

*input* est une fonction intégrée de Python qui accepte une entrée de l'utilisateur alors qu'un script py est en cours d'exécution.

| Syntaxe de <i>input</i>                           | Description                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input([chaîne de texte de l'invite])</code> | Alors qu'un script py est en cours d'exécution, <i>input</i> écrit la [chaîne de texte de l'invite] de l'argument dans la ligne de sortie des résultats du SHELL et attend une saisie de l'utilisateur.<br>Un nom de variable de chaîne ou une chaîne de caractères entre guillemets (") ou entre guillemets simples (') peut être spécifié comme [chaîne de texte de l'invite]. |

Dans le cas d'une variable de chaîne faisant jusqu'à 16 caractères, tous les caractères du nom de la variable de chaîne spécifiée par *input* sont affichés comme invite lorsque la fonction est exécutée dans le mode **Python**. Dans le cas d'une variable de chaîne plus longue que 16 caractères, les 15 premiers caractères de la variable de chaîne suivie par le symbole de similitude (~) seront affichés comme invite.

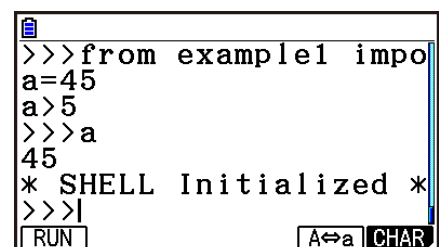
### Exemple d'exécution de *input*

|                                                                                                                     |                                                                                |
|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Chaîne de texte d'invite de 16 caractères au plus ("123?" a été saisi comme chaîne de texte d'invite.)              | <pre>&gt;&gt;&gt;a=input("123?") 123? </pre>                                   |
| Chaîne de texte d'invite de plus de 16 caractères ("12345678901234567" a été saisi comme chaîne de texte d'invite.) | <pre>&gt;&gt;&gt;a=input("1234567890123456789012345~") 123456789012345~ </pre> |

## ■ Initialisation du SHELL

Les fonctions et variables qui sont définies, les modules qui sont importés, et les résultats d'autres opérations du SHELL sont stockés dans la zone du tas du SHELL (zone de mémoire de stockage temporaire) durant l'exécution du SHELL. Chaque fois que le SHELL est quitté (en allant vers un autre écran du mode **Python**), le contenu dans la zone du tas du SHELL jusqu'à ce point est effacé. L'effacement du contenu dans la zone du tas du SHELL est appelé « initialisation du SHELL ».

- Lorsque vous redémarrez le SHELL dans le mode **Python**, le message « \* SHELL Initialized \* » apparaîtra sur la ligne au-dessus de la ligne du bas (ligne d'invite) sur l'écran SHELL.
- Ce message s'affichera uniquement si vous réaffichez l'écran SHELL, mais il n'apparaîtra pas la première fois que vous affichez l'écran SHELL après être entré dans le mode **Python**.



```
>>>from example1 impo
a=45
a>5
>>>a
45
* SHELL Initialized *
>>>|
```

- Si le SHELL est redémarré en exécutant un script py depuis l'écran de la liste des fichiers ou l'écran de l'éditeur de script, le SHELL sera initialisé avant l'exécution du script py. C'est pour cette raison que l'écran SHELL s'affichera comme illustré dans la capture d'écran ci-dessous.

```

>>>f=60
>>>d+f*2
160
* SHELL Initialized *
>>>from example3 impo
6
>>>|
  
```

## 5. Modification d'un fichier py

### ■ Affichage et modification d'un fichier py

Vous pouvez utiliser la procédure ci-dessous pour ouvrir un fichier py enregistré et afficher son contenu sur l'écran de l'éditeur de script, où vous pouvez le modifier, si vous le souhaitez.

#### • Pour ouvrir un fichier py et afficher l'écran de l'éditeur de script

1. À partir du menu principal, entrez dans le mode **Python**.
2. Sur l'écran de la liste des fichiers qui s'affiche, utilisez ▲ et ▼ pour mettre en surbrillance le fichier py que vous souhaitez ouvrir, puis appuyez sur **F2** (OPEN).
  - Cela ouvre le fichier py sélectionné et affiche l'écran de l'éditeur de script.
  - Veillez à ne pas appuyer sur **EXE** par erreur alors que l'écran de la liste des fichiers s'affiche. Cela lancerait l'exécution du fichier py et afficherait l'écran SHELL.

#### • Pour sauter à la première ligne ou à la dernière ligne sur l'écran de l'éditeur de script

- Pour sauter à la première ligne de l'écran de l'éditeur de script, appuyez sur **F6** (▷) **F3** (JUMP) **F1** (TOP).
- Pour sauter à la dernière ligne de l'écran de l'éditeur de script, appuyez sur **F6** (▷) **F3** (JUMP) **F2** (BOTTOM).

#### • Pour sauter à un numéro de ligne spécifique de l'écran de l'éditeur de script

1. Depuis l'écran de l'éditeur de script, appuyez sur **F6** (▷) **F3** (JUMP) **F3** (LINE).
2. Dans la boîte de dialogue qui apparaît, saisissez le numéro de ligne à laquelle vous souhaitez sauter, puis appuyez sur **EXE**.

---

### • Pour rechercher du texte sur l'écran de l'éditeur de script

1. Depuis l'écran de l'éditeur de script, appuyez sur **F6** (▷) **F4** (SEARCH).
2. Sur l'écran qui apparaît, saisissez la chaîne de caractères à rechercher puis appuyez sur **EXE**.
  - Cela commence la recherche depuis le haut du script py, et déplace le curseur à gauche du premier caractère de la première chaîne de caractères trouvée. Si aucune chaîne de caractères n'est trouvée, le message « Non trouvé » s'affiche. Si cela se produit, appuyez sur **EXIT**.
  - Pour reprendre la recherche en utilisant la même chaîne de caractères, appuyez sur **F1** (SEARCH).
  - Vous pourrez reprendre une opération de recherche seulement lorsque « SEARCH » est affiché comme touche de menu de fonctions **F1**, ce qui veut dire qu'il y a au moins une autre chaîne de caractères correspondante dans le script. Pour annuler une opération de recherche en cours, appuyez sur **EXIT**. L'opération de recherche se terminera automatiquement s'il n'y a plus de texte correspondant à la chaîne de texte spécifiée.

---

### • Pour copier ou couper une chaîne de texte à partir de l'écran de l'éditeur de script et la stocker dans le presse-papiers

1. Sur l'écran de l'éditeur de script, déplacez le curseur au début du texte que vous voulez copier ou couper puis appuyez sur **SHIFT** **8** (CLIP).
2. Déplacez le curseur à la fin du texte que vous voulez copier ou couper.
  - Cela met en surbrillance le texte sélectionné.
  - Sélectionner le texte du début à la fin ou de la fin au début produit le même résultat.
3. Appuyez sur **F1** (COPY) ou **F2** (CUT).

---

### • Pour coller une chaîne de caractères qui est dans le presse-papiers

1. Sur l'écran de l'éditeur de script, déplacez le curseur à l'emplacement où vous souhaitez coller la chaîne de texte.
2. Appuyez sur **SHIFT** **9** (PASTE).

---

## ■ Débogage d'un script py

Si un fichier py ne fonctionne pas comme prévu, cela peut être dû à un bogue (erreur) dans le script py.

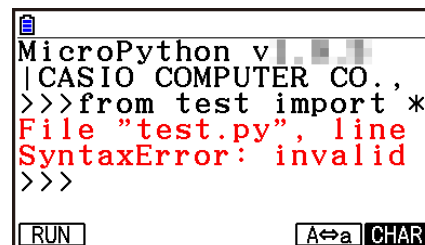
Les symptômes ci-dessous indiquent qu'un fichier py doit être débogué.

- Lors de son exécution, un fichier de script py génère un message d'erreur.
- Lors de son exécution, un fichier py ne produit pas les opérations ou les résultats escomptés.




---

## • Utilisation des messages d'erreur pour effectuer le débogage

Si un message d'erreur en texte rouge apparaît sur l'écran SHELL lorsque vous exécutez un fichier py, effectuez les étapes ci-dessous.



```
MicroPython v...
| CASIO COMPUTER CO.,
>>>from test import *
File "test.py", line
SyntaxError: invalid
>>>
[RUN] [A↔a] [CHAR]
```

1. Utilisez  pour mettre la ligne de message d'erreur en surbrillance, et ensuite utilisez  et  pour vérifier les détails du message d'erreur.
2. Appuyez sur **EXIT**.
  - Cela permet de retourner à l'écran à partir duquel le fichier py a été exécuté (écran de l'éditeur de script ou écran de la liste des fichiers). Ouvrez le fichier py où l'erreur est survenue et vérifiez le contenu de la ligne qui a généré le message d'erreur. Apportez les corrections nécessaires.
  - Notez qu'un message d'erreur peut ne pas identifier le véritable problème.
  - Notez qu'un message d'erreur s'affiche également lorsqu'il y a un problème de saisie dans le SHELL, il peut donc sembler qu'il y a une erreur dans le fichier py. Un exemple serait des données saisies ne correspondant pas au type de données spécifié par *input*, etc. Si vous ne parvenez pas à trouver un problème avec la ligne pour laquelle un message d'erreur s'affiche, vérifiez si la saisie dans le SHELL est correcte.

Le menu de fonctions pour la fonction {JUMP} est utile lorsque vous avez besoin de sauter à une ligne particulière sur l'écran de l'éditeur de script. Voir « Pour sauter à un numéro de ligne spécifique de l'écran de l'éditeur de script » (page 17-18).

---

## • Débogage basé sur les résultats d'exécution du fichier py

Si l'exécution d'un fichier py produit un résultat inattendu, vérifiez le contenu de tout le fichier py et apportez les corrections nécessaires.