

## Contexte

La calculatrice permet de développer en Python.

On va voir comment mettre en œuvre ce langage de programmation avec la construction d'une première fonction **CM** qui transforme un pourcentage passé en paramètre, en son équivalent sous forme de coefficient multiplicateur. Si l'on souhaite augmenter un prix de 30%, la fonction renverra le coefficient multiplicateur 1,3.

## 1. Créer un nouveau script

On accède à l'éditeur Python à l'aide de la touche **prgm** puis en exécutant l'application Python.

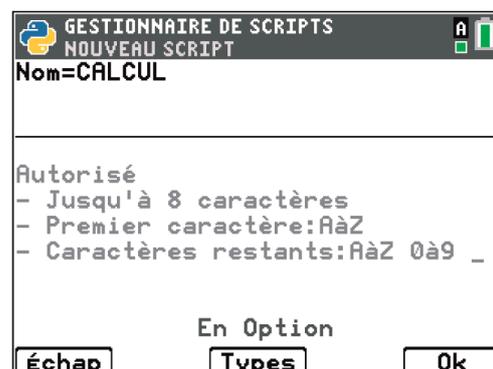
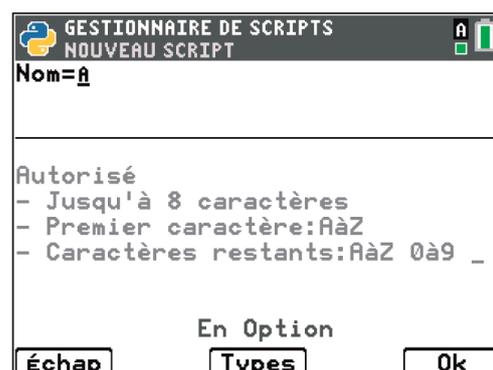
Pour créer un nouveau script, on sélectionne l'onglet **Nouv** à l'aide de la touche **zoom** qui est située sous l'onglet visé.

On obtient alors la fenêtre ci-contre. Le curseur nous indique que l'on peut saisir du texte en continu. Si on veut saisir un chiffre, je dois alors désactiver la saisie de caractères en appuyant sur la combinaison de touches **2nde** + **alpha**

Le curseur change d'aspect et la lettre A disparaît 

On nomme notre script **CALCUL**. On pourrait se servir de l'onglet **Types** pour définir le type de contenu à venir dans notre script. Cela facilite notamment la saisie de certains imports de librairie (**random**, **math** par exemple) mais n'est pas obligatoire. On aura la possibilité de les importer par la suite.

On valide la création du script à l'aide de l'onglet **Ok** avec la touche **graphe**



## 2. Saisir une nouvelle fonction

On va créer une première fonction **CM** qui prend en paramètre **p** un taux d'évolution et qui renvoie le coefficient multiplicateur correspondant.

La calculatrice offre une interface de saisie efficace que nous allons découvrir.

Commençons par aller dans l'onglet **Fns...** à l'aide de la touche **f(x)**

Cela nous permet d'accéder à des menus contenant la plupart des fonctions dont nous aurons besoin. On peut ainsi les repérer en naviguant dans les onglets à l'aide des flèches de la calculatrice et les sélectionner à l'aide de la touche **entrer**

Une fois validées, les instructions, structures, fonctions etc. sont importées dans notre script sans que nous ayons besoin de les saisir. C'est un gain de temps appréciable.

On valide donc l'instruction **def fonction() :**



Notre script est implémenté par ces premiers éléments.

Il faut activer le mode « écriture continue » (« n'écrire que des lettres ») à l'aide des touches   pour écrire en minuscule  

ou    pour écrire en majuscule  

Une fois activé, pour passer d'un mode à l'autre, il suffit d'appuyer seulement sur 

On complète notre première ligne de script de la manière ci-contre.

Les points gris sont des marqueurs d'indentation. Nous aurons l'occasion d'en reparler.

Si on les efface par mégarde, il est possible de corriger une ligne à l'aide de l'onglet  via la touche .

On y trouvera les commandes pour augmenter ou diminuer l'indentation.

On complète la deuxième ligne du script en allant chercher l'instruction return dans l'onglet 

On quitte le mode « écriture continue » pour saisir nos différentes valeurs et opérateurs. Ponctuellement on active la saisie d'un caractère en appuyant une seule fois sur la touche 

Notre saisie terminée, on va pouvoir tester notre première fonction dans la console Python en se rendant dans l'onglet  à l'aide de la touche 

Un message sur l'exécution du script apparaît.

On peut saisir en toutes lettres notre instruction ou appuyer sur la touche  pour sélectionner le nom de la fonction et compléter la valeur passée en paramètre.

Testons les retours de `CM(30)` et `CM(-30)`

La fonction nous renvoie bien les coefficients multiplicateurs 1,3 et 0,7

Augmentons 15 de 30% en saisissant `15*CM(30)`

On obtient bien 19,5



```
ÉDITEUR : CALCUL
LIGNE DU SCRIPT 0001
def ():
..
```



```
ÉDITEUR : CALCUL
LIGNE DU SCRIPT 0002
def CM(p):
..
```



```
ÉDITEUR : CALCUL
Fonc Ctl Ops List Type E/S Modul
1: def fonction():
2: return
```

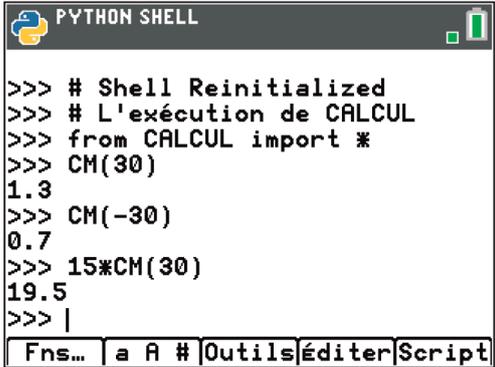


```
ÉDITEUR : CALCUL
LIGNE DU SCRIPT 0002
def CM(p):
..return 1+p/100
```

Fns... a A # Outils Exéc Script



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de CALCUL
>>> from CALCUL import *
>>> |
```



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de CALCUL
>>> from CALCUL import *
>>> CM(30)
1.3
>>> CM(-30)
0.7
>>> 15*CM(30)
19.5
>>> |
```

Fns... a A # Outils Éditer Script

## Contexte

On poursuit notre travail d'appropriation en créant un nouveau script à partir du précédent.

L'objectif est la construction d'une fonction **Capital** qui calcule, pour un placement initial, le montant récupéré au bout de  $t$  années à un certain pourcentage. Toutes ces informations devront être fournies en paramètres à la fonction. Si on place 1000 euros pendant 5 ans à 1%, l'instruction **Capital(1000,1,5)** renverra 1010.

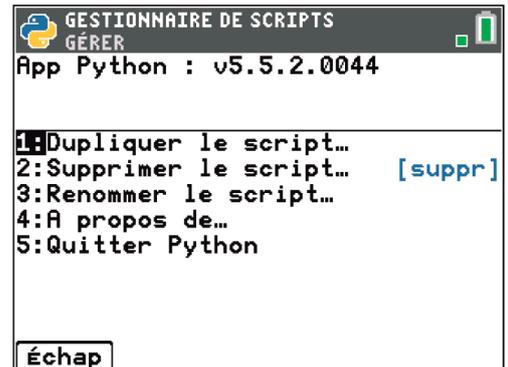
## 1. Dupliquer un script

Lorsqu'on lance l'environnement Python, on arrive en premier dans le gestionnaire de script. En positionnant le curseur devant le nom d'un script existant, il est possible de l'éditer à l'aide de l'onglet **Édit** mais également de le dupliquer pour en recopier le contenu dans un nouveau.

Pour cela, il faut se rendre dans l'onglet **Gérer** à l'aide de la touche **graphe**

On accède à plusieurs options pour gérer nos scripts.

On a choisi de dupliquer le script **CALCUL** précédent pour appeler le nouveau **EPARGNE**



## 2. Ajouter une nouvelle fonction

Nous allons créer une nouvelle fonction **AppliqPourcent** dans notre script.

Elle prend en paramètre un montant  $m$  et un pourcentage  $p$ . Elle renvoie un nouveau montant auquel on aura appliqué l'évolution  $p$ .

Lorsque nous positionnons le curseur en fin de ligne de la fonction **CM** et validons, l'éditeur Python choisit de maintenir l'indentation précédente.

Il faut la supprimer pour pouvoir créer une nouvelle fonction, à l'aide de la touche **suppr**

On utilise la fonction **CM** pour convertir le pourcentage en coefficient multiplicateur.

Il est important de se souvenir que le langage Python est également sensible à la « casse ». C'est-à-dire qu'il faut respecter les majuscules et les minuscules lors de la saisie de nos fonctions.

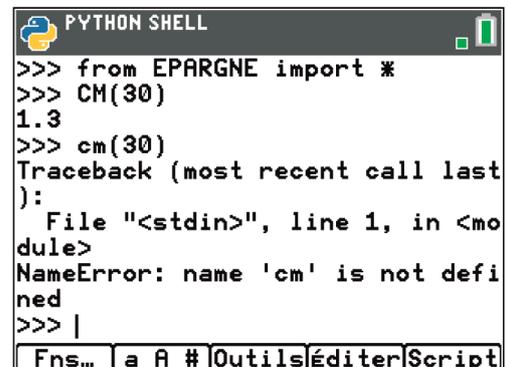
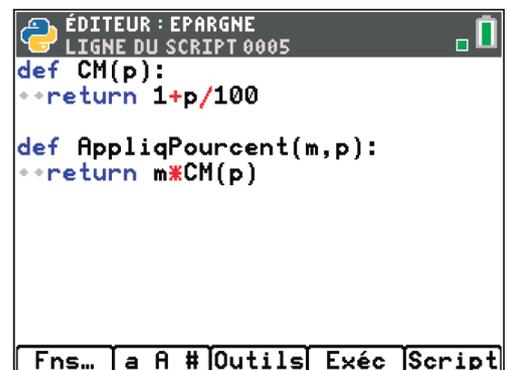
Dans l'exemple ci-contre, la saisie dans la console de l'instruction **CM(30)** fonctionne tandis que **cm(30)** renvoie une erreur.

On peut vérifier en console que si on applique une augmentation de 30% à 15 euros, on obtient bien 19,50 euros

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de EPARGNE
>>> from EPARGNE import *
>>> AppliqPourcent(15,30)
19.5

```



### 3. Connaître des raccourcis

Pour accélérer encore davantage la saisie de script Python, il est intéressant de connaître certaines manipulations que nous allons découvrir lors de la saisie de notre fonction **Capital**

Par exemple, la calculatrice offre plusieurs raccourcis clavier.

Le signe = est accessible directement en appuyant sur la touche 

Comme pour la définition de fonction, il est possible d'importer des structures de contrôle que l'éditeur de la calculatrice se chargera d'indenter.

Il ne reste alors, ici, plus qu'à compléter l'instruction `range()` avec la variable `t`.

Nous souhaitons saisir l'instruction :

```
c=AppliqPourcent(m,p)
```

L'écriture peut être fastidieuse mais là encore, la calculatrice dispose de fonctionnalité pour accélérer la saisie comme un copier/coller disponible dans l'onglet **Outils**

Il ne reste plus que quelques modifications à faire manuellement.

Pour conclure, voici le script complet que vous devez obtenir.

Vous pourrez tester en console que 1000 euros placés pendant 5 année à 1% d'intérêts par an rapporteront 10 euros puisque le capital sera alors de 1010 euros.

```
ÉDITEUR : EPARGNE
LIGNE DU SCRIPT 0009
def CM(p):
  **return 1+p/100

def AppliqPourcent(m,p):
  **return m*CM(p)

def Capital(m,p,t):
  **c=0
  **
```

Fns... a A # Outils Exéc Script

```
ÉDITEUR : EPARGNE
Fonc Ctrl Ops List Type E/S Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(taille):
5:for i in range(début,fin):
6:for i in range(début,fin,pas):
7:for i in liste:
8:while condition:
9:elif :
0:else:
```

```
ÉDITEUR : EPARGNE
Outils
1:Indent ▶
2:Indent ◀
3:Annuler Effacer
4:Insérer Ligne ▲
5:Couper Ligne
6:Copier Ligne
7:Coller Ligne ▼
8:Aller à la Ligne du Script...
9:Réinitialiser le Shell ▶
0:Retour au Shell
```

Échapp

```
ÉDITEUR : EPARGNE
LIGNE DU SCRIPT 0009
def CM(p):
  **return 1+p/100

def AppliqPourcent(m,p):
  **return m*CM(p)

def Capital(m,p,t):
  **c=0
  **for i in range():
  **
```

Fns... a A # Outils Exéc Script

```
ÉDITEUR : EPARGNE
LIGNE DU SCRIPT 0010
def CM(p):
  **return 1+p/100

def AppliqPourcent(m,p):
  **return m*CM(p)

def Capital(m,p,t):
  **c=0
  **for i in range(t):
  **def AppliqPourcent(m,p):
  **
```

Fns... a A # Outils Exéc Script

```
ÉDITEUR : EPARGNE
LIGNE DU SCRIPT 0011
def CM(p):
  **return 1+p/100

def AppliqPourcent(m,p):
  **return m*CM(p)

def Capital(m,p,t):
  **c=0
  **for i in range(t):
  **c=AppliqPourcent(m,p)
  **return c
```

Fns... a A # Outils Exéc Script

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de EPARGNE
>>> from EPARGNE import *
>>> Capital(1000,1,5)
1010.0
```