



Unité 1 : Débuter la programmation en Python

Compétence 3 : Les fonctions en Python

Dans cette troisième leçon de l'unité 1, vous allez utiliser l'éditeur de programme (script) afin de créer des fonctions, puis exécuter celui-ci afin d'observer les résultats dans la console

Objectifs :

- Découvrir la notion de fonction en Python
- Créer une fonction

Vers la notion de fonction en Python.

Mettre en œuvre l'algorithme suivant :

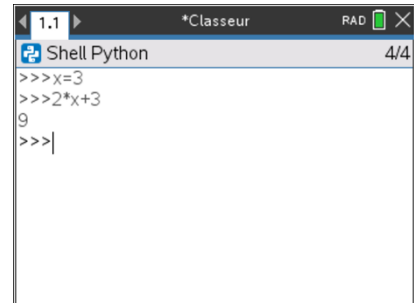
$$x \leftarrow 3$$

$$y \leftarrow 2 \times x + 3$$

Dans une console, ceci se réalise très simplement.

Mais si l'on souhaite répéter ce type de calcul pour une autre valeur de x il faut écrire de nouveau l'ensemble de la séquence. Ce qui, sur un exemple moins trivial, peut s'avérer vite fastidieux.

On est donc conduit à créer une fonction qui nous permettra de dupliquer aisément le traitement de l'algorithme.



En algorithmique, une fonction peut être considérée comme une séquence d'instructions, réalisant une certaine tâche, en utilisant un ou plusieurs **arguments** (ou aucun dans certains cas).

Cette fonction reçoit un nom.

- La programmation d'une fonction commence toujours par **def** suivi du nom de la fonction, suivi des arguments de celle-ci. Cette ligne se termine par le symbole **:**
- Les deux points indiquent le début du bloc d'instructions définissant la fonction : toutes ces instructions sont **indentées**, c'est-à-dire décalées vers la droite par rapport à la première ligne. On ajoute en tête de chaque ligne, le même nombre d'espaces.

La fonction renvoie un seul résultat par l'intermédiaire de la commande **return**. Le résultat peut être constitué d'une liste de résultats, une chaîne de caractères etc.

def nom_fonction(liste des arguments) :

- .. bloc d'instructions
- .. **return** (résultats)

L'indentation, obtenue avec la touche de tabulation ou avec des espaces, est **primordiale** : tout ce qui est indenté après le **def()** sera exécuté comme un bloc. Il ne faut pas que l'indentation varie (nombre d'espaces, passer de la tabulation à des espaces. . .) en cours de bloc.

Conseil à l'enseignant : Une fonction permet de découper le problème étudié en sous-problèmes et d'éviter ainsi la répétition d'instructions. Une fois définie, elle peut être « appelée » tout au long de l'exécution du programme autant de fois que nécessaire.

Une fonction peut n'avoir aucun argument. Elle peut également être appelée dans un autre programme : il suffit pour cela de l'insérer dans une instruction en saisissant son nom et les valeurs des arguments.





10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

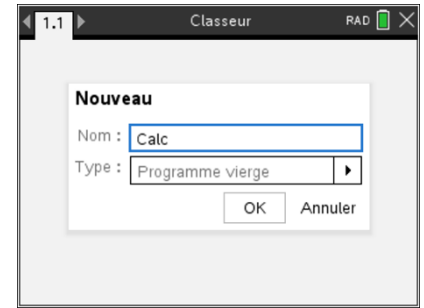
Mise en œuvre d'un premier exemple.

- Créer un nouveau script

Appuyer sur la touche et choisir le menu **A Ajouter Python**, puis choisir : **1 Nouveau**. Dans la fenêtre qui s'ouvre, donner un nom à votre script.

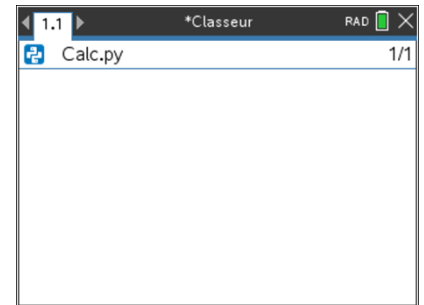
Valider le nom de votre script en cliquant sur OK, ou en appuyant sur la touche **enter**.

UNITE 1: COMPETENCE 3 NOTES DU PROFESSEUR

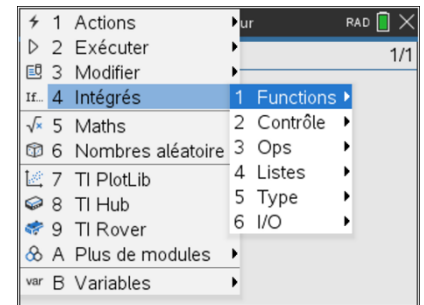


Conseil à l'enseignant : Par défaut l'éditeur de programmes s'ouvre sur un script vierge (n'intégrant aucune librairie). Cependant, l'édition peut être facilitée, lorsque l'on connaît par avance, les librairies qui seront nécessaires à la création d'un nouveau projet.

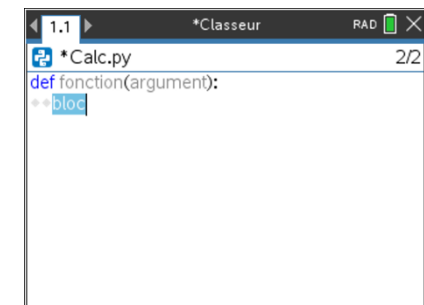
- Le nom de votre script apparaît ensuite dans le bandeau supérieur, avec l'extension **.py**.



- Appuyer ensuite sur la touche
 - Les instructions du langage Python sont intégrées dans le menu **4 Intégrés** et contiennent toutes les instructions nécessaires à l'écriture d'un script.
 1. Définitions de fonctions
 2. Boucles et instructions conditionnelles
 3. Tests
 4. Listes
 5. Type
 6. Entrées sorties ...



- Observer l'action réalisée par la calculatrice. La structure de la fonction est directement affichée.
- Les parties grisées peuvent être facilement complétées en utilisant la touche de la calculatrice.
- Lorsque l'éditeur est sur une ligne pouvant être complétée, l'instruction qui s'y trouve est colorée en bleu clair.



Ce document est mis à disposition sous licence Creative Commons
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>





- Vous devriez obtenir l'écran ci-contre. Vous remarquerez l'indentation automatique du curseur.
- Continuez ensuite par les instructions de traitement de l'algorithme. Souvenez-vous que l'affectation d'une variable s'effectue en appuyant sur la touche = .
- Appuyer ensuite sur la touche puis **4 Intégrés, 1 Fonctions** et enfin **2 : return** puis compléter l'instruction.

L'instruction peut également être tapée manuellement, la coloration syntaxique sera mise en œuvre dès reconnaissance de celle-ci.

```
1.1 *Classeur RAD 3/3
*Calc.py
def f(x):
  y=2*x+3
  return y
```

A présent vous êtes prêts pour exécuter votre script.

- Appuyer sur .
- La console s'affiche et un message vous informe du chargement du script.
- Compléter l'invite de commande par le nom de la fonction avec les arguments prévus (un seul dans cet exemple), puis valider.
- Les touches de déplacement ▲ et **enter** permettent de rappeler une instruction.

```
1.1 1.2 *Classeur RAD 7/7
Shell Python
>>>#Running Calc.py
>>>from Calc import *
>>>f(3)
9
>>>f(5)
13
>>>|
```

Conseil à l'enseignant : L'édition et l'exécution d'un script Python peuvent être facilités avec la TI-Nspire en créant une seconde page (puis **5 : Format de page**. Choisir le format qui vous convient et, dans la seconde page, insérer une application correspondant au **Shell**. Ainsi, vous obtiendrez une fenêtre correspondant à celle d'un éditeur complet.

Par la suite, la commande passera directement dans la console afin d'exécuter le script, directement sous la fenêtre dans laquelle il a été édité.

```
1.1 1.2 *Classeur RAD 1/1
Calc.py
Shell Python
>>>|
```

