

Base des bases

<code>a = 3</code>	Mets la valeur 3 dans la variable <code>a</code> (type <code>int</code>)
<code>b = 3</code>	Pareil
<code>c = a + b</code>	Dans <code>c</code> il y a la somme de <code>a</code> et <code>b</code> ce qui donne 6 (type <code>int</code>)
<code>type(a)</code>	Donne le type de la variable (<code>str</code> , <code>int</code> , <code>list</code> , <code>tuple</code>)
<code>print(a)</code>	Affiche la valeur de <code>a</code> ici 3
<code>#commentaire</code>	Le <code>"#"</code> permet d'écrire qqch qui ne sera pas interpréter comme du code.
<code>d = " Toto o"</code>	Mets la chaîne de caractère "Toto" dans <code>d</code> (type <code>str</code>)
<code>print(d)</code>	Affiche la chaîne de caractère "Toto" présente dans la variable <code>d</code>
<code>e = []</code>	Initialise la variable <code>e</code> d'un tableau vide (type <code>list</code>)
<code>f = (0, 1, 2)</code>	Initialise la variable <code>f</code> d'un tuple très pratique pour des coordonnées (type <code>tuple</code>)

Le plus simple à connaître et comprendre. Généralement, on donne des noms plus parlant et pas d'une lettre au variable. Généralement, les tuples ne sont pas utilisés en Mathématique au niveau de Maths Spécialités (Année 2023-2024)

Boucle et conditions

<code>while <condition> :</code>	Boucle qui continue tant que la condition n'est pas respecter.
<code>for i in range(a) :</code>	Boucle qui initialise un index <code>i</code> a 0 pour aller jusqu'à <code>a</code> de 1 en 1.
<code>if <condition> :</code>	Vérifie si la condition est respecter si oui alors executé le code dans le <code>if</code> .
<code>elif <condition> :</code>	Vérifie si la condition est respecter si oui alors executé le code dans le <code>elif</code> contraction des mots clés <code>else</code> et <code>if</code> .
<code>else :</code>	Si les conditions des <code>if</code> ou <code>elif</code> alors le code dans le <code>else</code> sera executé. Ne prends pas de conditions

Niveau supérieur mais abordable facilement.

Mots clés

<code>break</code>	Casse la boucle
<code>pass</code>	Ignore ce qu'il y a d'écrit et casse la boucle
<code>continue</code>	Passe au code en casse la boucle
<code>e.append(a)</code>	Ici nous ajoutons à la fin du tableau <code>e</code> créer plutôt, la variable <code>a</code> contenant 3 (type <code>list</code>)
<code>print(e)</code>	Affichage de la variable <code>e</code> qui montre désormais : <code>[3]</code>

Des mots clés parfois pratique mais qui ont tous de petites subtilités. A utiliser a vos risques et péril (sauf pour `break` mais pas a trop utilisé non plus).

Fonctions

<code>def <nom> (<variable>) :</code>	On peut penser a définir <code>def</code> ce qui nous permet de dire a notre programme, ce que l'on veut qu'il fasse si on appelle la fonction.
--	---



Fonctions (cont)

`return <variable_quelconque>` A mettre a la fin de la fonction pour qu'elle renvoie le contenu de la variable.

Basique aussi, une fonction peut prendre plusieurs variable en compte, on appelle ça les arguments de la fonction.

Quand on l'appelle dans la ligne de commande python (la ligne ou il y a écrit >>>) on doit juste écrire `<no m>(<va ria ble >)` avec autant d'argument que la fonction peut prendre en compte.



By **JustWireless**
(JustWireless)

cheatography.com/justwireless/

Published 1st November, 2023.
Last updated 1st November, 2023.
Page 3 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

