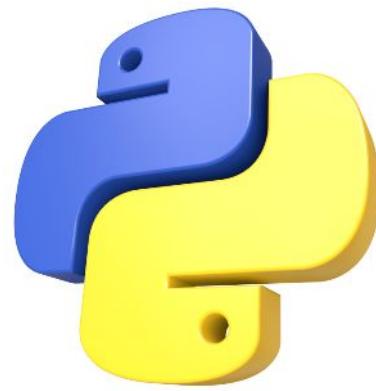


Python

Entrées et sorties texte

Affectation



Structure d'un programme

Un programme comporte en général trois grandes parties dans cet ordre :

- L'importation des modules :
.....

- La déclaration des fonctions :
.....

- Le programme principal :
.....

Il faut toujours ranger les instructions dans la bonne partie du programme. On les identifie en écrivant leur titre avec au moins un symbole **#** en début de ligne.

1. Ecrire et exécuter le programme ci dessous. Il sera appelé **structure.py**.

```
1 ## Importation des modules
2
3
4 ## Déclaration des fonctions
5
6
7 ## Programme principal
```

2. Ce programme ne fait rien. Pourquoi ?

.....
.....

Sortie texte dans la console, la fonction print

La fonction **print** permet d'afficher du texte dans la console. On dit que le texte « sort » de la machine puisqu'il est produit par l'ordinateur à destination de l'utilisateur. L'écran est un périphérique de sortie.

C'est une fonction car elle nécessite, comme en Mathématiques, des parenthèses pour être utilisée et elle produit un résultat.

1. Coder dans le programme principal d'un programme appelé **sortie_console.py** l'instruction **print('Hello world !')**. L'exécuter.

2. A quoi servent les apostrophes avant et après le texte à afficher. Que peut-on utiliser d'autre ?
Comment s'appelle un texte en programmation ?

.....
.....
.....

3. Ajouter la seconde instruction **print("J'ai envie d'apprendre à programmer.")**.

4. Pourquoi a-t-on utilisé les guillemets cette fois-ci ?

.....

5. Essayer **print('Hello world !', "J'ai envie d'apprendre à programmer.")**

6. Quelle est la différence d'affichage avec les deux instructions précédentes ? En déduire des propriétés de la fonction **print** et à quoi sert la virgule en Python.

.....
.....
.....

7. Ajouter une fonction **print** à parenthèses vides entre deux des autres. Qu'est-ce qu'il s'affiche ?
Pourquoi ?

.....
.....

Variables et opérateurs

Une variable est un espace mémoire de l'ordinateur, « case », dans laquelle on stocke une donnée. Elle est associée à une étiquette dont le nom est choisi par l'utilisateur. En **python** une variable se définit par le nom choisi par le programmeur, suivi du symbole **=** puis de la donnée à stocker, dans cet ordre, de gauche à droite.

Le symbole **=** est un opérateur car il permet de faire une opération avec une donnée. Il est appelé opérateur d'affectation et a une signification différente de celle des Mathématiques.

8. Dans un programme **variables_operateurs.py**, écrire les instructions **a = 7** et **print(a)** l'une en dessous de l'autre.

9. Expliquer en détail ce que va faire python et la machine en exécutant ces deux instructions.

.....
.....
.....
.....

10. Il y a beaucoup d'opérateurs qui combinent des données pour en produire de nouvelles. Essayer et interpréter :

```
1 a = 5 + 8
2 b = 92 - 32
3 c = b / a
4
5 print('Les valeurs des trois variables sont', a, b, c)
```

Les opérateurs peuvent s'appliquer à différents types de données.

11. Essayer et interpréter **d** = 'Sarah' + 'Massey'.

12. Comment s'appelle l'opération + entre deux nombres ou entre deux chaînes de caractères ?

Le type d'une donnée s'obtient avec la fonction **type**.

13. Afficher le type des variables **a**, **b**, **c** et **d**. Recopier l'instruction et expliquer les résultats.

14. Enfin les fonctions **int**, **float** et **str** permettent de convertir les types entre eux. Afficher les résultats des conversions en entier de '**57**' et de **5.89**, puis les conversions en réels de '**5.89**' et de **6**, et enfin les conversion en chaînes de caractères de **12.47** et de **89**. Par exemple pour le premier on écrira et expliquera ce qui se passe avec le programme ci-dessous.

```
1 a = '57'  
2 b = int(a)  
3  
4 print('Le type du contenu de la variable a est', type(a))  
5 print('Le type du contenu de la variable b est', type(b))  
6  
7 print('a vaut :', a)  
8 print('b vaut :', b)  
9  
10 print('a + a =', a + a)  
11 print('b + b =', b + b)
```

Entrée texte dans la console, la fonction input

La fonction **input** permet à l'utilisateur d'entrer un texte au clavier. On dit que le texte « entre » dans la machine puisqu'il est produit par l'utilisateur à destination de l'ordinateur. Le clavier est désigné comme un périphérique d'entrée.

15. Dans un programme **entree_console.py**, essayer et expliquer le programme ci-dessous.

```
1 prenom = input('Entrer votre prénom : ')
2
3 print('Bonjour', prenom)
```

.....
.....
.....

16. Le compléter afin qu'il demande aussi votre nom et votre âge. Il affichera les résultats exactement comme ci-après.

```
1 prenom = input('Entrer votre prénom : ')
2
3
5
6 print('Bonjour', prenom)
7
```

.....
.....
.....

Bonjour Sam Agace !
Je sais que vous avez 15 ans.

17. Tester le programme ci-dessous. Le résultat est-il correct ? Expliquer.

```
1 x = input('Entrer un nombre : ')
2
3 print('Multiplié par 3, le résultat vaut', x * 3)
```

.....
.....

La variable **x** contenant une chaîne de caractère, **python** effectue une concaténation, plusieurs fois. L'opération s'appelle alors une répétition. Il faut donc la convertir en nombre entier ou réel avant de faire la multiplication.

18. Proposer et tester une modification permettant d'obtenir un résultat correct. La noter ci-dessous.

.....

19. Ecrire un programme qui demande d'entrer deux nombres et affiche le résultat de leur multiplication. Le recopier ci-dessous.

.....
.....
.....
.....