

Fiche 3.0 : scripts et modules

bUT Informatique
IUT de Vélizy



Ce symbole indique un texte à lire attentivement avant de se lancer dans les exercices qui suivent.

bla

1 Notion de script

2 Next



Ce symbole précise les compétences SACoche liées aux exercices qui suivent.

TYPE

- ☐ 0. Dans le mode console : définissez deux variables a et b valant respectivement 10 et 3, calculez ce que valent $a + b$, $a * b$ ou a/b . Changez la valeur de b en 3.0 et recommencez. Quelle est la différence ?
- ☐ 1. Toujours dans le mode console, utilisez la fonction `type` pour demandez le type des objets. Créez des variables entières, des réels, des chaînes de caractères et affichez leur type.
- ☐ 2. Affichez maintenant une phrase mélangeant une chaîne de caractère et la valeur d'une variable.



À partir de maintenant les exercices seront à traiter en écrivant des scripts (ou programmes) python. N'hésitez pas à revoir les vidéos correspondantes si nécessaire. N'oubliez pas d'ajouter la ligne

`#coding: utf-8`

au début du programme si vous n'utilisez pas que des caractères ascii (accents, etc.)

- ☐ 3. Écrivez un programme `age.py` qui demande son âge à l'utilisateur et répond en utilisant sa réponse au milieu d'une phrase, comme par exemple en lui donnant l'âge qu'il aura en 2027 (ou sinon, soyez imaginatifs, c'est vous les jeunes :))
- ☐ 4. Écrivez `matieres.py` : demandez maintenant deux informations à l'utilisateur, sous forme de chaînes de caractères, et faites une phrase avec les deux informations. Par exemple, on lui demande le nom de ses deux matières préférées, et on répond "moi aussi j'adore (...) et (...) ." (Réponse : programmation et algorithmique). *Attention à bien différencier les fonctions natives `input` qui renvoie une valeur et `raw_input` qui renvoie une chaîne de caractères.*
- ☐ 5. Dans `quetype.py`, demandez à l'utilisateur d'entrer une valeur avec `input`, stockez la dans une variable et affichez son type. Rappel : `type(a)` renvoie le type d'une variable a . Vous pouvez l'afficher avec `print`. Testez avec différentes valeurs : 35, 284736537, -35, 2.4, -10.6, "bonjour", "b", [2,4,3], ["bonjour", "au revoir"], ... regardez bien le type à chaque fois.
- ☐ 6. Dans `affichage.py`, affichez (sans utiliser de boucle) :
 - la chaîne "aaaaa..." (250 "a")
 - la chaîne "bbbb....a" (250 "b" suivis d'un "a")

- la chaîne "bbbb....aaabbb..." (250 "b" suivis d'un "a" suivis de 100 b)
- La liste [0,1,...1000] (au complet)
- La liste [1000,1001,...1507] (au complet)
- 50 fois "bonjour" en allant à la ligne à chaque fois (utilisez "\n")

3 Les boucles



La boucle `while` en python
La boucle `for` en python



Partie 4 du tutoriel :
<https://docs.python.org/fr/2.7/tutorial/controlflow.html>



BOUCLES1, BOUCLES2, BOUCLES ?

- ☐ 7. Écrire un programme qui affiche 50 fois "bonjour", cette fois-ci en utilisant une boucle `while`.
- ☐ 8. Même exercice avec une boucle `for`.
- ☐ 9. Écrire un programme qui demande un mot et un entier, et affiche autant de fois le mot entré. Attention à distinguer `raw_input` et `input`.
- ☐ 10. Écrire un programme qui affiche un rectangle d'étoiles de p lignes et q colonnes. Par exemple si on entre 3 pour p et 12 pour q, la sortie sera :

```
*****  
*****  
*****
```

- ☐ 11. Écrire un programme qui affiche un triangle d'étoiles de taille donnée, par exemple si on entre 6 le programme affiche :

```
*  
**  
***  
****  
*****  
*****
```

- ☐ 12. Même exercice mais cette fois la sortie est

```
          *  
        * *  
      * * *  
    * * * *  
  * * * * *  
* * * * *  
* * * * *
```