

Les fonctions

1. Les fonctions

Pour définir une fonction nous utiliserons l'instruction **def**. Comme tout, en Python, l'indentation sera importante pour définir ce qui appartient à la fonction définie.

```
def ma_fonction(param1, param2,...) :  
    print ("je suis dans ma fonction ")
```

Exemple :

```
def bonjour5() :  
    for _ in range(5) :  
        print (« Bonjour »)  
  
>>> bonjour5()
```

1.1. le mot clé return des fonctions

return est utilisé dans les fonctions pour renvoyer une valeur suite à l'exécution du code. Cette dernière se met en fin de programme et agira sur la variable ou instruction à renvoyer.

Le retour de la fonction pourra être placé dans une variable afin d'être exploité plus tard dans le programme.

Exercice : 1-*renvoie_str.py*

Tâches à réaliser :

Créer une fonction qui prendra en paramètre une chaîne de caractères
nommée : **return_str(chaîne)**.
→ cette fonction renvoie la chaîne mise en paramètre.

Exercice : 2-*renvoie_int.py*

Tâches à réaliser :

Créer une fonction qui prendra en paramètre un chiffre ou un nombre
nommée : **return_double(nombre)**
→ cette fonction renvoie le double du nombre mis en paramètre

Exercice : 3-*renvoie_bool.py*

Tâches à réaliser :

Créer une fonction qui prendra en paramètre un chiffre ou un nombre
nommée : **return_bool(nombre)**
→ cette fonction teste si la variable est un nombre
si oui return True
sinon return False

lien utile :

<https://docs.python.org/3/library/functions.html#isinstance>

<https://www.programiz.com/python-programming/methods/built-in/isinstance>

2. les saisies utilisateur

L'instruction **input()** permet d'effectuer cette saisie au clavier.

```
>>> var = input('entrer quelque chose au clavier\n')
entrer quelque chose au clavier
12
>>> type(var)
<type 'str'>
>>> var = input('entrer quelque chose au clavier\n')
entrer quelque chose au clavier
bonjour
>>> type(var)
<type 'str'>
>>> var = input('entrer quelque chose au clavier\n')
entrer quelque chose au clavier
3.9
>>> type(var)
<type 'str'>
```

Exercice :

4.saisie_user.py

Tâches à réaliser :

Demander à l'utilisateur d'entrer un nombre ou des caractères.

Tester si la valeur entrée :

si chiffre → *afficher* : C'est un chiffre

si caractère :

→ compte la taille

→ si 1 caractère *afficher* : c'est une lettre

→ si plus *afficher* : c'est une chaîne de caractères

sinon → *afficher* : c'est autre chose

5.saisie_user_evolution.py

Tâches à réaliser :

Demander à l'utilisateur d'entrer un nombre ou des caractères.

Le programme doit compter le nombre de caractères, de chiffres et signes de ponctuation ;

Affichage sera du type :

entrer un nombre ou des caractères :

-> dsc6&'4('".3.té""(r6(4é""3-"4é3""-é""-45-é

il y a 5 lettres, 10 chiffres, 22 signes de ponctuation

[6.check_password.py](#)

Tâches à réaliser :

Demander à l'utilisateur de saisir la lettre C ou Q afin de vérifier le mot de passe ou de quitter le programme.

Le mot de passe devra respecter les consignes suivantes :

12 caractères minimums

Etre en alphanumérique et comportée des caractères spéciaux

Astuce :

Pour terminer le programme, la fonction `exit()` de la lib `sys` peut être judicieuse.

Affichage sera du type :

[C]heck

[Q]uitter

Votre choix ->