

Fonctions spécifications et tests - Cours

– décembre 2022

1 Les fonctions

Définition : Les fonctions

Les **fonctions** sont des séries de blocs d'instructions que l'on souhaite isoler et réutiliser. Elles permettent de **factoriser** des instructions et, quand elles sont bien nommées, donner des informations sur l'intention du programmeur.

En Python, les fonctions ont la forme suivante

```
1 def nom_de_la_fonction(argument1, argument2, ...):
2     instruction 1
3     instruction 2
4     instruction 3
5     ...
6     return valeur
```

Le mot clé **return** permet de renseigner la valeur que la fonction retournera après son exécution. L'exécution de la fonction s'arrêtera au premier **return** rencontré. Si aucun **return** la fonction ne renverra aucune valeur, mais toutes les instructions s'exécuteront.

Si une variable est définie dans une fonction, on ne pourra pas accéder à sa valeur en dehors sauf si elle est retournée.

Exemples

- Fonction qui calcule les coordonnées d'un vecteur

```
1 def coord_vecteur(xA, xB, yA, yB):
2     x = xB - xA
3     y = yB - yA
4     return x, y
5
6 print(coord_vecteur(1, 2, 3, 4))
7 xAB, yAB = coord_vecteur(12, 34, 23, 15)
8
```

- Fonction qui teste si un mot de passe est valide

```
1 mdp = "NSI"
2 def pass_est_valide(password):
3     if password == mdp:
4         return True
5
6     print("Mot de passe invalide")
7     return False
8
9 acces = pass_est_valide("plop")
10 acces2 = pass_est_valide("NSI")
11 print(acces)
12 print(accesé)
```

- Fonction qui décore du texte

```
1 def decor_pyramide(texte, symbole):
2     chapeau = ""
3     for i in range(len(texte)):
4         chapeau = chapeau + symbole*(i+1) + "\n"
5
6     bas = ""
7     for i in range(len(texte)):
8         chapeau = chapeau + symbole*(len(texte)-i) + "\n"
9
10    return chapeau + "\n" + texte + "\n" + bas
11
12 decoration = decor_pyramide("NSI", "#")
13 print(decoration)
14 print(chapeau)
```