

Fonctions spécifications et tests - Cours

– décembre 2022

2 Documenter

Programmer c'est écrire du code pour que la machine le comprenne mais aussi et surtout pour que d'autres personnes le comprennent et soit en mesure de l'utiliser ou le modifier. C'est pour cela qu'il est primordial de **documenter** les fonctions.

Définition : Les docstrings

Une **docstring** (chaîne de documentation) est la manière choisie pour documenter les fonctions en Python. Elles se placent en dessous de la signature de la fonction (ligne avec **def**) et sont entourées de `""" ... """`.

```
1 def nom_de_la_fonction(argument1, argument2, ...):
2     """ Explication du fonctionnement de la fonction """
3     instruction 1
4     instruction 2
5     instruction 3
6     ...
7     return valeur
```

Pour accéder à la docstring, on utilise l'attribut `__doc__` ou la fonction `help` en mode interactif.

```
1 >>> from math import sqrt
2 >>> print(sqrt.__doc__)
3 Return the square root of x.
```

3 Tester

Écrire une fonction qui marche c'est bien mais écrire une fonction qui fait ce qu'on attend d'elle c'est mieux. Pour s'assurer qu'une fonction a un comportement attendu, il faut écrire des **tests** en utilisant le mot clé **assert**.

```
1 assert condition
```

Si la condition est vérifiée rien ne se passera. Par contre, si elle est fautive, une erreur sera remontée.

Exemple

```
1 def mention(note):
2     """ Renvoie la mention qui correspond à la note """
3     if note >= 16:
4         return "TB"
5     elif note >= 14:
6         return "B"
7     elif note >= 12:
8         return "AB"
9     elif note >= 9:
10        return "A"
11    else:
12        return "R"
13
14 assert mention(13) == "AB"
15 assert mention(18) == "TB"
16 assert mention(9.5) == "R"
```

Les deux premiers tests passent. Par contre, le dernier rate car nous avons fait une erreur dans la programmation de la fonction.

Remarques : Écrire les tests avant de coder la fonction est considéré comme une bonne pratique.