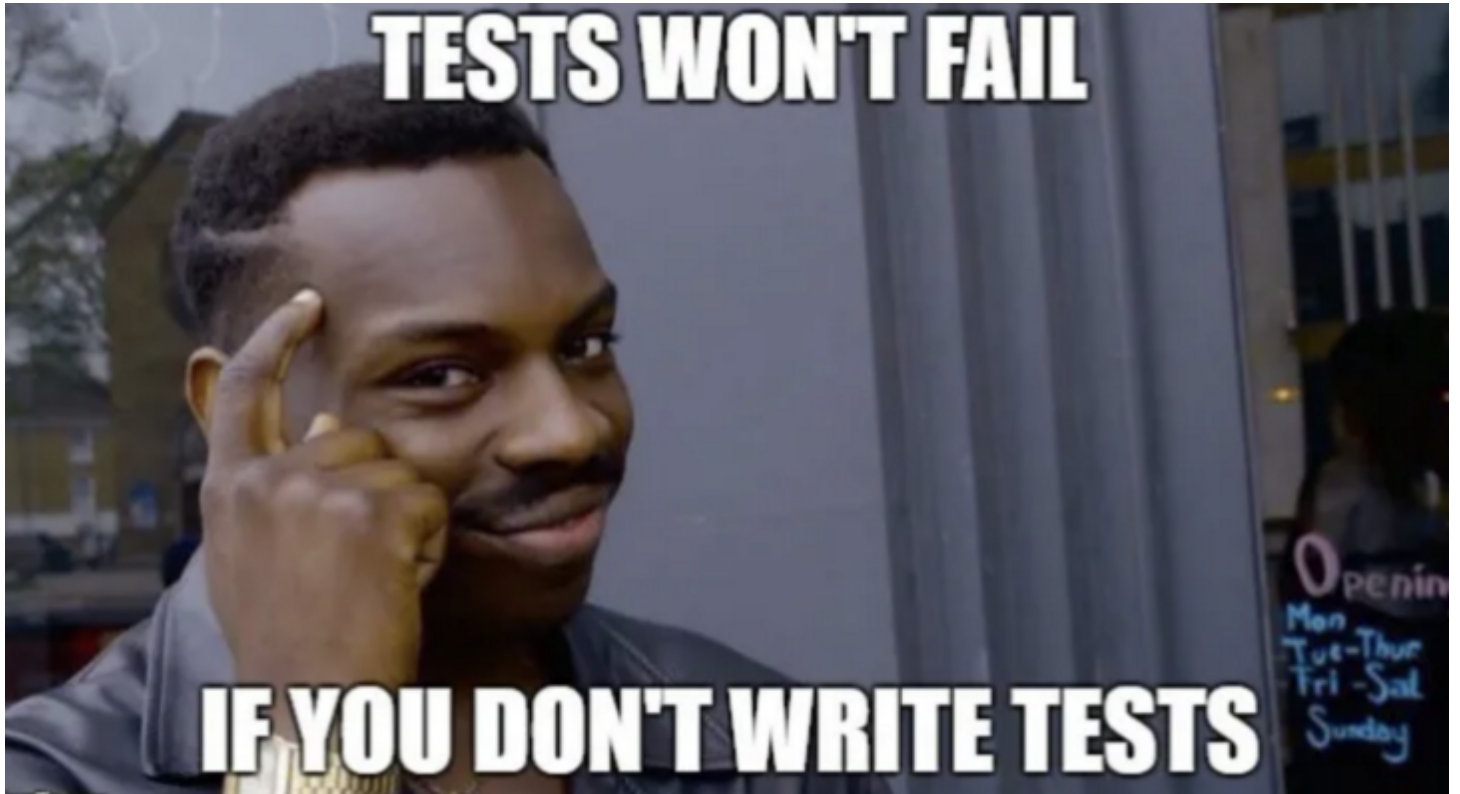


## T6.3 Tester une fonction



Normalement, la première chose qu'on fait après avoir écrit une fonction, c'est la tester. Pour cela, on passe des arguments à la fonction pour lesquels on connaît le résultat attendu.

Par exemple, reprenons la fonction `maximum` écrite au chapitre 6.1.5:

```
Script Python

1  def maximum(n1, n2):
2      if n1 > n2:
3          return n1
4      else:
5          return n2
```

Ensuite, en console, on la teste en passant des arguments simples:

```
Script Python

>>> maximum(2, 5)
5
>>> maximum(10, 3)
10
```

Dans la vraie vie des programmeurs, c'est le contraire. On écrit d'abord les tests. Puis la fonction. Et enfin on effectue les tests.

## 1. 6.3.1 Assert

Ces tests reposent sur le mot-clé `assert`, qui va lever une erreur lorsqu'il est suivi d'une expression évaluée à `False` :

### Script Python

```
>>> assert 2 == 3
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
AssertionError
>>> assert 2 <= 3
>>>
```

Avant d'écrire cette fonction `maximum`, on écrirait alors:

### Script Python

```
1  assert maximum(2, 5) == 2
2  assert maximum(10, 3) == 10
3  assert maximum(4, 4) == 4
```

Ce *jeu de tests* peut se regrouper dans une fonction pour vérifier qu'elle a bien le comportement désiré.

### Script Python

```
1  def test_maximum():
2      assert maximum(2, 5) == 2
3      assert maximum(10, 3) == 10
4      assert maximum(4, 4) == 4
5      print("tests ok") # cette ligne ne s'exécutera que si aucune erreur n'est levée par assert
```

### Un autre exemple

Toujours au chapitre 6.1.5, je vous ai demandé d'écrire une fonction `est_bissextile` permettant de déterminer si une année est bissextile ou non (en renvoyant un booléen).

Comme on sait (?) que 2000 était bissextile, que 2020 également, mais ni 1900 ni 2021 ne le sont, on aurait pu commencer par écrire notre fonction de jeu de tests ainsi:

### Script Python

```
1  def test_est_bissextile():
2      assert est_bissextile(2000) == True
3      assert est_bissextile(2020) == True
4      assert est_bissextile(1900) == False
5      assert est_bissextile(2021) == False
6      print("tests ok")
```

### ⚠ Jeu de tests couvrant ?

Il faut vérifier que les tests couvrent toutes les situations possibles, mais ce n'est pas toujours facile !



## 2. 6.3.2 Exercices

### ☰ Exercice 1

#### Énoncé

Écrire un jeu de tests pour la fonction `compte_voyelles` de l'exercice 2 du chapitre 6.1.5.

#### Correction

## Exercice 2

### Énoncé

1. Écrire un jeu de tests pour une fonction `fizzbuzz` qui prend un entier en paramètre et renvoie ... vous savez quoi (cf. ex 6 de 6.1.4).
2. Écrire la fonction `fizzbuzz` et la tester.

### Correction