Tester avec `doctest`

Objectifs

- utiliser un outil de test : `doctest`
- établir une méthodologie pour les TP

Motivation

Vous savez documenter les fonctions à l'aide d'une «chaîne de documentation» (ou «`docstring`»), c'est-à-dire une chaîne de caractères placée immédiatement après l'en-tête de la fonction. Voici un tel exemple de documentation

```
def fact(n):
    """
        paramètre n : (int) un entier
        valeur renvoyée : (int) la factorielle de n.

    CU : n >= 0

    Exemples :

    >>> fact(3)
    6
    >>> fact(5)
    120
    """
    res = 1
    for i in range(2, n + 1):
        res = res * i
    return res
```

Cette documentation peut être exploitée avec la fonction `help` :

```
>>> help(fact)
Help on function fact in module __main__:

fact(n)
        paramètre n : (int) un entier
        valeur renvoyée : (int) la factorielle de n.

    CU : n >= 0

    Exemples :

    >>> fact(3)
    6
    >>> fact(5)
    120
```

À faire

Utilisez `Spyder` pour

1. recopier la fonction `fact` avec sa docstring dans un fichier que vous nommerez `exples_doctest.py`,
2. et utiliser la fonction `help` au niveau de l'interpréteur.

Réaliser une telle chaîne de documentation permet

- à l'utilisateur de la fonction de savoir
  - à quoi peut servir la fonction ;
  - comment il peut l'utiliser ;
  - et quelles conditions il doit respecter pour l'utiliser (CU).
- et au programmeur de la fonction de préciser
  - le nombre et la nature de ses paramètres ;
  - la relation entre la valeur renvoyée et celle du ou des paramètres ;
  - ses idées avec quelques exemples.

  (Tout cela bien entendu à condition que cette documentation soit rédigée avant la réalisation du programme et non le contraire.)

Mais vous allez découvrir que cela permet davantage encore !

Utiliser le module **`doctest`**

Les exemples donnés dans une chaîne de documentation peuvent être testés à l'aide d'un module de Python nommé **`doctest`**.

À faire

Depuis l'interpréteur (shell), dans lequel la fonction `fact` ci-dessus est supposée chargée, tapez les deux lignes

```
>>> import doctest
```

pour importer le module, et

```
>>> doctest.testmod()
```

Vous devez obtenir

```
>>> doctest.testmod()
TestResults(failed=0, attempted=2)
```

La fonction `testmod` du module **`doctest`** est allée chercher dans les docstring des fonctions du module actuellement chargé, c'est-à-dire **`exples_doctest`**, tous les exemples (reconnaissables à la présence des triples chevrons >>>), et a vérifié que la fonction documentée satisfait bien ces exemples. Dans le cas présent, une seule fonction dont la documentation contient deux exemples (`attempted=2`) a été testée, et il n'y a eu aucun échec (`failed=0`).

Et si un exemple et la fonction ne sont pas d'accord ?

À faire

Modifiez le deuxième exemple, en mettant `121` à la place de `120` dans le second exemple. Chargez le fichier dans l'interpréteur (touche F5) et retapez les deux lignes

```
>>> import doctest
>>> doctest.testmod()
```

Vous devez obtenir

```
>>> doctest.testmod()
**********************************************************************
File "/home/eric/AP1/exples_doctest.py", line 24, in __main__.fact
Failed example:
   fact(5)
Expected:
   121
Got:
   120
**********************************************************************
1 items had failures:
  1 of   2 in __main__.fact
***Test Failed*** 1 failures.
```

Qu'est ce que tout cela révèle ?

- Tout d'abord que les tests ont échoué et qu'il y a eu 1 échec (cf dernière ligne) et que cet échec est dû à la fonction fact (cf avant dernière ligne).
- Ensuite que le test incriminé est celui concernant fact(5) pour lequel le test a obtenu (Got) 120 en exécutant la fonction fact, alors qu'il attendait (Expected) 121 selon l'exemple donné par la documentation.

Lorsqu'il y a de tels échecs, cela invite le programmeur à vérifier son programme, … ou bien les exemples de sa documentation, comme c'est le cas ici.

Rendre automatique les tests

Il est très facile de rendre automatique les tests et ainsi de ne plus avoir à faire appel explicitement (et manuellement) à la fonction testmod.
Il suffit pour cela d'inclure en fin de fichier les trois lignes :

```
if __name__ == '__main__':
   import doctest
   doctest.testmod()
```