

La suite de Fibonacci (suite)

Complément - niveau intermédiaire

Nous reprenons le cas de la fonction `fibonacci` que nous avons vue déjà, mais cette fois nous voulons que l'utilisateur puisse entrer l'entier en entrée de l'algorithme, non plus en répondant une question, mais sur la ligne de commande, c'est-à-dire en tapant

```
$ python fibonacci.py 12
```

Avertissement Attention cette version-ci **ne fonctionne pas dans ce notebook** sous IPython, justement car on n'a pas de moyen dans un notebook d'invoquer un programme en lui passant des arguments de cette façon. Ce notebook est rédigé pour vous permettre de vous entraîner avec la fonction de téléchargement au format python, qu'on a vu dans la vidéo, et de faire tourner ce programme sur votre propre ordinateur.

Cette fois nous importons le module qui va nous permettre d'interpréter les arguments passés à la ligne de commande

```
from argparse import ArgumentParser
```

Puis nous répétons la fonction `fibonacci`

```
def fibonacci(n):
    "retourne le nombre de fibonacci pour l'entier n"
    # pour les petites valeurs de n il n'y a rien à calculer
    if n <= 1:
        return 1
    # sinon on initialise f1 pour n-1 et f2 pour n-2
    f2, f1 = 1, 1
    # et on itérera n-1 fois pour additionner
    for i in range(2, n + 1):
        f2, f1 = f1, f1 + f2
    # print i, f2, f1
    # le résultat est dans f1
    return f1
```

Remarque Certains d'entre vous auront évidemment remarqué qu'on aurait pu éviter de copier-coller la fonction `fibonacci` comme cela; c'est à ça que servent les modules mais nous n'en sommes pas là.

À présent nous utilisons le module `argparse` pour lui dire qu'on attend exactement un argument sur la ligne de commande, qui doit être un entier. Ici encore ne vous inquiétez pas si vous ne comprenez pas tout le code, l'objectif est de vous donner un morceau de code

utilisable tout de suite pour jouer avec les notebooks.

```
[-] parser = ArgumentParser()
    parser.add_argument(dest="entier", type=int,
                        help="entier d'entree")
    input_args = parser.parse_args()
    entier = input_args.entier
```

Nous pouvons à présent afficher le résultat

```
[-] print "fibonacci({}) = {}".format(entier, fibonacci(entier))
```

Vous pouvez donc à présent

- télécharger ce code sur votre disque comme un fichier `fibonacci.py` en utilisant le menu *"File -> Download as -> python"*
- l'exécuter avec simplement python comme ceci

```
$ python fibonacci_prompt.py 56
```