

# Suite de Fibonacci : quel est le meilleur algorithme ?

il suffit de comparer les temps avec timeit...

```
<sxh python; title : fibonacci09_fonctions_comparaison.py> #! /usr/bin/env python # -*- coding: utf-8
-*- """ Calculs des premiers éléments de la suite de Fibonacci. Référence :
```

[http://fr.wikipedia.org/wiki/Suite\\_de\\_Fibonacci](http://fr.wikipedia.org/wiki/Suite_de_Fibonacci) Comparaison de différentes fonctions avec Timeit

<http://docs.python.org/2/library/timeit.html>

[http://www.diveintopython.net/performance\\_tuning/timeit.html](http://www.diveintopython.net/performance_tuning/timeit.html) ; traduit en français ici :

[http://python.developpez.com/cours/DiveIntoPython/php/frdiveintopython/performance\\_tuning/timeit.php](http://python.developpez.com/cours/DiveIntoPython/php/frdiveintopython/performance_tuning/timeit.php) """ import timeit from fibonacci05\_fonction import fibonacci\_item from fibonacci06\_fonctions

import fibonacci\_item\_from\_list from fibonacci07\_fonction\_recursive import fibonacci\_item\_recursive

from fibonacci08\_fonction\_algo\_log import fibonacci\_item\_logarithmic

```
from fibonacci08_fonction_algo_log import fibonacci_item_logarithmic
```

```
if name == 'main':
```

```
    print "Calculs exemples préliminaires..."
    i=input("Suite de Fibonacci. Donnez l'indice de l'élément souhaité ? ")
    print ("Élément de la suite : "),
    if i <= 10: print fibonacci_item_recursive(i)
    print fibonacci_item_from_list(i)
    print fibonacci_item(i)
    print fibonacci_item_logarithmic(i)
    print "Temps d'exécution"
    #using timeit :
    t1 = timeit.Timer("fibonacci_item_recursive(" + str(i) + ")", "from
fibonacci07_fonction_recursive import fibonacci_item_recursive")
    t2 = timeit.Timer("fibonacci_item_from_list(" + str(i) + ")", "from
fibonacci06_fonctions import fibonacci_item_from_list")
    t3 = timeit.Timer("fibonacci_item(" + str(i) + ")", "from
fibonacci05_fonction import fibonacci_item")
    t4 = timeit.Timer("fibonacci_item_logarithmic(" + str(i) + ")", "from
fibonacci08_fonction_algo_log import fibonacci_item_logarithmic")
    if i <= 10:
        print "fibonacci_item_recursive"
        print "1000 exécutions : ",t1.timeit(1000)
        print "Cinq ensembles de 10000 exécutions : ",t1.repeat(5, 1000)
    print "fibonacci_item_from_list"
    print "1000 exécutions : ",t2.timeit(1000)
    print "Cinq ensembles de 10000 exécutions : ",t2.repeat(5, 1000)
    print "fibonacci_item_"
    print "1000 exécutions : ",t3.timeit(1000)
    print "Cinq ensembles de 10000 exécutions : ",t3.repeat(5, 1000)
    print "fibonacci_item_logarithmic"
    print "1000 exécutions : ",t4.timeit(1000)
    print "Cinq ensembles de 10000 exécutions : ",t4.repeat(5, 1000)
```

Last update: 2013/11/08 08:55 teaching:progappchim:suite\_de\_fibonacci-5 [https://dvillers.umons.ac.be/wiki/teaching:progappchim:suite\\_de\\_fibonacci-5?rev=1383897321](https://dvillers.umons.ac.be/wiki/teaching:progappchim:suite_de_fibonacci-5?rev=1383897321)

---

</sxh>

From: <https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link: [https://dvillers.umons.ac.be/wiki/teaching:progappchim:suite\\_de\\_fibonacci-5?rev=1383897321](https://dvillers.umons.ac.be/wiki/teaching:progappchim:suite_de_fibonacci-5?rev=1383897321)

Last update: **2013/11/08 08:55**

