

☛ Mesurer le temps en OCAML

La fonction `time` du module `Sys` donne le temps en seconde écoulé lors de l'exécution du programme.

```
1 | Sys.time : unit -> float
```

Attention, le temps retourné est le temps cumulé au cours des différents appels de fonctions, donc la valeur retournée par cette fonction est à utiliser comme un instant plutôt que comme une durée. On obtient alors un temps de calcul par différence entre l'instant de début et l'instant de fin.

```
1 | let rec fibo_bete (n : int) : int =
2 |   match n with
3 |   | 0 -> 0
4 |   | 1 -> 1
5 |   | _ -> (fibo_bete (n-1)) + (fibo_bete (n-2))
6 |
7 | let fibo_affiche_tps (n : int) : unit =
8 |   let debut = Sys.time() in
9 |   let res = fibo_bete n in
10 |  let fin = Sys.time() in
11 |  print_string (string_of_float (fin-.debut))
```

Dans l'exemple ci-dessus on a défini une fonction qu'on sait être longue à exécuter sur des grandes valeurs. On a ensuite encapsulé un appel à cette fonction dans une deuxième fonction qui prend soin de noter l'instant avant l'appel, l'instant après l'appel, et qui affiche ensuite la différence entre les deux (on note l'usage de `-.` et non de `-`) soit la durée d'exécution de cet appel.