

# La ligne *shebang*

---

```
❏ #!/usr/bin/env python
```

## Complément - niveau avancé

Ce complément est uniquement valable pour MacOS et linux

### Le besoin

Nous avons vu dans la vidéo que pour lancer un programme python on fait essentiellement depuis le terminal

```
❏ $ python mon_module.py
```

Lorsqu'il s'agit d'un programme qu'on utilise fréquemment, on n'est pas forcément dans le même répertoire que là où se trouve le programme python, aussi dans ce cas on peut utiliser un chemin "absolu", c'est-à-dire à partir de la racine des noms de fichiers, comme par exemple

```
❏ $ python /le/chemin/jusqu/a/mon_module.py
```

Sauf que si effectivement on utilise cela souvent, c'est très malcommode de devoir s'y prendre de cette façon

### La solution

Sur linux et MacOS, il existe une astuce utile pour simplifier cela. Voyons comment s'y prendre, avec par exemple le programme `fibonacci.py` que vous pouvez télécharger ici ([data/fibonacci.py](#)) (nous verrons ce code en détail dans les deux prochains compléments). Commencez par sauver ce code sur votre ordinateur dans un fichier qui s'appelle, bien entendu, `fibonacci.py`.

On commence par éditer le tout début du fichier pour lui ajouter une **première ligne**; il faut faire attention que la ligne contenant le **coding**: soit bien en deuxième position:

```
❏ #!/usr/bin/env python
  # coding: utf-8

  ## La suite de Fibonacci (Suite)
  ...etc...
```

Cette première ligne s'appelle un Shebang ([http://en.wikipedia.org/wiki/Shebang\\_%28Unix%29](http://en.wikipedia.org/wiki/Shebang_%28Unix%29)) dans le jargon Unix. Unix stipule que le Shebang doit être en **première position** dans le fichier. C'est d'ailleurs pourquoi, on l'a vu précédemment au sujet des caractères accentués, la ligne décrivant l'encodage avec `coding`: peut être mise dans un fichier python en première **ou deuxième** position dans le fichier.

Ensuite on rajoute au fichier, depuis le terminal, le caractère exécutable comme ceci:

```
$ pwd
/le/chemin/jusqu/a/

$ chmod +x fibonacci.py
```

À partir de là vous pouvez utiliser le fichier `fibonacci.py` comme une commande, sans avoir à mentionner `python`, qui sera invoqué au travers du shebang.

```
$ /le/chemin/jusqu/a/fibonacci.py 20
fibonacci(20) = 10946
```

Et donc vous pouvez aussi le déplacer dans un répertoire qui est dans votre variable `PATH`, et le rendre ainsi accessible depuis n'importe quel répertoire en faisant simplement

```
$ cd /tmp

$ fibonacci.py 20
fibonacci(20) = 10946
```

**Remarque** tout ceci fonctionne très bien tant que votre point d'entrée – ici `fibonacci.py` – n'utilise que des modules standard. Dans le cas où le point d'entrée vient avec au moins un module, il est également nécessaire d'installer ces modules quelque part, et d'indiquer au point d'entrée comment les trouver, nous y reviendrons avec la vidéo sur les modules.