

## TP de familiarisation avec Python

Pierre Pompidor

### Ecrivez un script Python qui affiche "Bonjour !"

La première étape est d'écrire un "Hello world".  
Saisissez pour cela le code suivant (je suis tellement bon) sous votre éditeur préféré (fichier `bonjour.py`) :

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

print("Bonjour !")
```

La ligne `# -*- coding: utf-8 -*-` n'est pas obligatoire mais elle permet que certains caractères accentués soient affichés correctement.

**A partir du terminal** n'oubliez pas tout d'abord de lui donner le droit d'exécution :

`chmod +x bonjour.py`

puis pour l'exécuter :

- si vous avez rajouté le répertoire courant (.) dans la variable système `PATH` :  
ligne `export PATH=$PATH:.` à placer en fin du fichier `~/.bashrc` vous pouvez directement écrire dans le terminal `bonjour.py`
- sinon (mais quel déshonneur) : `./bonjour.py`

### Ecrivez un script qui vérifie qu'un paramètre lui a bien été passé

Pour contrôler la présence de paramètres, testez le nombre d'éléments d'une liste (mais laquelle (\*) ?) par la fonction `len()`.

(\*) vous pouvez vous faire spoiler en lisant ce qui apparaît en début de page suivante.  
Si aucun paramètre n'a été passé, affichez un message d'erreur.

Ce paramètre est stocké dans un tableau de paramètres appelé `argv` (accessible à condition que le module `sys` soit inclus), et dont le premier élément (c'est à dire le premier vrai paramètre passé au script) est appelé `sys.argv[1]` (`sys.argv[0]` étant le nom du script).

### Affichez le premier paramètre du script

Maintenant, passez un paramètre au script `bonjour.py` (par exemple votre prénom) pour qu'il soit affiché.

### Affichez tous les paramètres du script :

Pour faire afficher une liste quelconque de paramètres, mettez en place une boucle automatique par :

```
for parametre in sys.argv :  
    # instructions à exécuter ...
```

(la variable `parametre` étant instanciée par chaque élément de la liste `sys.argv`).

### Affichez tous les paramètres du script sauf le premier :

Rappelez-vous que l'on peut saucissonner une liste !

### Ecrivez un script qui calcule en itératif la factorielle de n

Quelques consignes :

- `n` doit être donné en paramètre ;
- `n` doit être transtypé en entier (car les paramètres sont des chaînes de caractères) ;
- il est important de se rappeler que  $0!$  vaut 1.

### Ecrivez un script qui affiche les nombres premiers contenus dans les n premiers entiers positifs

Quelques consignes :

- `n` doit être donné en paramètre ;
- il est important de se souvenir que le premier nombre premier est 2 ;
- les nombres premiers que vous trouvez doivent être stockés dans une liste et **réutilisés lors des vérifications suivantes** ;
- c'est le contenu de cette liste qui doit être affiché.

### Modifiez le script précédent pour afficher les n premiers nombres premiers

C'est en fait ce que j'avais marqué dans le support de cours...