

# Méthodologie de la programmation

## TP 1 : Premiers pas avec Python

Dans ce TP :

- Écrire et exécuter du code Python.  
Documentation de Python : <https://docs.python.org/3/>.

### Exercice 0.

Jouer avec Python.

1. Pour exécuter un programme écrit en Python, on fait appel à l'*interpréteur* Python. L'interpréteur Python est la commande `python3` (parce qu'une vieille version 2 du langage pas totalement compatible avec la 3 s'appelle encore `python`).  
Pour l'utiliser il suffit de lui passer en argument le fichier Python à exécuter.

(a) → Recopier le code suivant dans un fichier `hello.py` :

```
1 print("Hello, world!\n")
```

(b) → Lancer le programme Python `hello.py`.

2. Écrire un programme qui compte de 0 à 100, mais remplace les nombres divisibles par 3 par "Fizz", ceux divisibles par 5 par "Buzz", et ceux qui le sont à la fois par 3 et par 5 par "FizzBuzz". Ne recopiez pas celui du cours, essayez de le refaire par vous même.

(a) → Écrire le script Python `fizzbuzz.py`.

(b) → Tester le code pour vous assurer qu'il fonctionne comme attendu.

### Exercice 1.

Un petit jeu.

1. Écrire une fonction qui prend un entier en argument, choisi un nombre aléatoire entre 0 et cet argument, et le fait deviner au joueur ou à la joueuse en lui donnant un indice après chaque tentative.

Pour générer un nombre aléatoire entre  $a$  et  $b$ , vous avez besoin de la fonction `random.randint(a, b)`. Pour utiliser cette fonction, vous devez importer le module `random` de Python dans votre programme. Vous pouvez faire cela avec la directive `import random` en début de fichier.

Pour lire une entrée clavier, vous pouvez utiliser la fonction `input()`, qui prend en argument une chaîne de caractère à afficher, et qui retournera ce que la personne a tapé. Par exemple `nom = input("Nom : ")` affichera "Nom : " et stockera dans la variable `nom` le texte entré au clavier.

Pour lire un entier, il faut convertir l'entrée clavier avec la fonction `int()`, cela peut être fait directement en sorti de `input()`, c'est à dire qu'on peut faire `int(input("Âge : "))`.

Pour au contraire manipuler un nombre comme une chaîne de caractères (par exemple pour le concaténer avec l'opérateur `+` a une autre chaîne de caractères), on peut utiliser la fonction `str()`.

→ Créer un programme Python `guessing_game.py` avec dedans la fonction `guess_the_number`.

2. Quand le nombre est correctement deviné par le joueur ou la joueuse, on voudrait lui afficher le nombre de coups que ce-tte derniere a dû utiliser.

→ Modifier la fonction `guess_the_number` en conséquence, et lui faire retourner cette valeur.

3. On voudrait proposer au joueur ou à la joueuse de faire plusieurs parties.

En dehors de la fonction, on veut que le programme demande le nombre maximum à faire deviner, si celui ci est 0 alors il s'arrête, sinon il lance le jeu (c'est à dire appelle la fonction `guess_the_number`) avec ce nombre comme maximum.

→ Modifier le programme `guessing_game.py` pour implémenter ce comportement.

4. Maintenant, on voudrait afficher un résumé des parties quand le jeu est quitté : pour chaque partie jouée, on veut afficher le nombre maximum à deviner et combien de coups il a fallu pour trouver le bon nombre.

→ En utilisant une liste (un élément par partie) de dictionnaire (avec une entrée pour le maximum et une pour le nombre de coups), implémenter le comportement souhaité dans le programme `guessing_game.py`.