
ITC MPSI - TP 1 : TYPES DE BASE EN PYTHON

7 SEPTEMBRE 2026

Exercice 1 (Tableau des opérations possibles)

Faites un tableau double entrée où chacune des lignes et colonnes correspond à l'un des types vu en cours : entiers, flottants, booléens, chaînes de caractères, listes. Pour chaque croisement entre deux types, vous testerez s'il est possible ou pas de faire les opérations suivantes : +, *, -, == et `or`. Dans chaque case du tableau, marquez en trois couleurs différentes les opérations qui :

- sont possibles
- sont impossibles
- sont possibles mais ont un comportement surprenant

La fonction `print`

En python, vous pouvez utiliser `print` pour afficher des messages dans la console. `print` peut prendre plusieurs arguments, entre parenthèses et séparés par des virgules. Par exemple, l'instruction `print("Bonjour!")` affiche `Bonjour!` dans la console, tandis que l'instruction

```
print("L'addition de", 3, "et", 2, "fait", 3+2)
```

affiche dans la console `L'addition de 3 et 2 fait 5`.

La fonction `input`

En Python, `input` permet de demander à l'utilisateur·ice du programme de rentrer une valeur. La fonction `input` prend en argument un message à afficher à l'utilisateur·e. L'appel à `input` s'évalue vers ce que l'utilisateur·e a rentré, en tant que chaîne de caractères.

Par exemple, le code

```
prenom = input("Quel est votre prénom ?")  
print("Bonjour " + prenom + "!")
```

a pour effet de demander son prénom à l'utilisateur·ice, et d'ensuite lui dire bonjour, en utilisant le bon prénom. Notez que la variable `prenom` est de type chaîne de caractères.

Les instructions conditionnelles

En Python, vous pouvez utiliser la syntaxe

```
if expression :  
    code1  
else :  
    code2
```

Si *expression* s'évalue vers `True`, alors *code1* est exécuté. Sinon, c'est *code2* qui est exécuté.

Par exemple, le code suivant :

```
nombre = int( input("Entrez un nombre") )  
if nombre % 2 == 0:  
    print( "Le nombre est pair" )  
else:  
    print( "Le nombre est impair" )
```

demande un nombre à l'utilisateur·ice, puis affiche un message qui indique si le nombre est pair ou impair.

Exercice 2 (Années bissextiles)

Une année est bissextile si elle est multiple de 4 sans être multiple de 400. Écrivez un code python qui demande de rentrer un année, et qui affiche un message qui indique si elle est bissextile ou pas.

Exercice 3 (Conversion températures)

Écrire un programme qui demande une température (soit en degrés Celsius soit en degrés Fahrenheit) et qui affiche la température en l'autre unité (respectivement Fahrenheit ou Celsius).

Exercice 4 (Bonus : jour de la semaine à une date donnée)

Écrivez un programme qui demande une date, puis affiche de quel jour de la semaine il s'agit (lundi, mardi, etc...).

Vous pourrez utiliser [la méthode des jours pivots](#).