

## TP 1 : Python, une super calculatrice ?

### Quelques sites à consulter

- Installer un interpréteur : le site officiel de Python : (<https://www.python.org>) ; ou utiliser la distribution Anaconda (<https://www.anaconda.com>), ou Edupython (<http://edupython.tuxfamily.org>) ou encore utiliser un interpréteur en ligne (<https://trinket.io>).
- Quelques environnements de développement intégré (EDI ou IDE en anglais) possibles : Pyzo (<https://pyzo.org/>), Spyder (<https://www.spyder-ide.org>), Pycharm (<https://www.jetbrains.com/fr-fr/pycharm/download/#section=windows>)
- Le site <http://pythontutor.com> permet de tester un code ligne par ligne.
- La documentation officielle se trouve sur la page <https://www.python.org>
- Le livre "apprendre à coder avec Python 3" de Gérard Swinnen et disponible sur <https://inforef.be/swi/python.htm> pourra vous servir si vous voulez en savoir plus.

### Python comme calculatrice

#### Exercice 1 :

Dans la console, saisir successivement `3 + 4 * 5`, `(3 + 4) * 5`, `8 / 4`, `8 // 4`, `8 \% 4`, `8 // 3`, `8 % 3`, `3 ** 3`, `3 ** 3 ** 3`, `3 ** (3 ** 3)`, `(3 ** 3) ** 3`, `0.00001`, `exp(2)`, `1j ** 2`, `(1 + 3j) ** 2`, `j ** 2`, `1==1`, `1==2`, `1!=2`, `2>1`.

Commenter vos observations et appliquer la fonction `type` aux différentes expressions précédentes.

#### Exercice 2 :

Saisir les commandes et commenter : `0.1 + 0.1`, `0.1 + 0.1 + 0.1`, `3*0.1 == 0.3`, `4*0.1 == 0.4`, `1e-324`, `1e-323`, `1e308`, `1e309`

#### Exercice 3 (Python comme traitement de texte) :

Dans la console, saisir successivement `"Bonjour"`, `'Bonjour'`.

Écrivez `"Bonjour, c'est moi"`.

`"Vive " + "la BCPST"`

`10 * "toto"`, `"la" * 12`

`len("anticonstitutionnellement")`

`Classe="BCPST"`, `Classe[0]`, `Classe[1]`, `Classe[4]`, `Classe[5]`, `Classe[1:4]`, `Classe[-2]`

### Gérer les variables

#### Exercice 4 :

Dans la console, saisir : `x = 2`; `x`; `y = 3 * 4`; `y`; `z = x + 5`; `z`; `ma_classe = "BCPST 1"`; `x,y = 2, "toto"`; `x`; `y`; `x+=1`; `x`.

#### Exercice 5 :

Sans trop vous fatiguer (mais sans copier-coller), affichez successivement

Veuillez agréer, madame, l'expression de mes salutations distinguées

Veillez agréer, monsieur, l'expression de mes salutations distinguées  
Veillez agréer, cher confrère, l'expression de mes salutations distinguées  
Veillez agréer, chère amie, l'expression de mes salutations distinguées

**Exercice 6 :**

Écrire, cette-fois-ci dans l'éditeur, un script Python (c'est-à-dire quelques lignes d'instructions) qui demandent à l'utilisateur de saisir un réel positif  $r$  et retourne la circonférence d'un cercle de rayon  $r$ , la surface d'un disque de rayon  $r$  et le volume d'une sphère de rayon  $r$ .

On prendra comme valeur de  $\pi$  la valeur 3.14.

On pourra utiliser les commandes `input`, `print`, et ajouter des commentaires avec `#` et enfin documenter le script avec `""" ... """`

**Exercice 7 :**

Écrire un script qui demande à l'utilisateur d'entrer deux entiers et retourne leur moyenne.

**Exercice 8 :**

Écrire un script qui demande à l'utilisateur d'entrer une durée en secondes et indique la durée correspondante en heures, minutes, secondes.

## Les bibliothèques ou modules

Ils s'appellent en utilisant `import`

Par exemple : `import math` ou `import matplotlib.pyplot as plt`

**Exercice 9 :**

Reprendre l'exercice 6 avec une valeur de  $\pi$  connue par Python.

**Exercice 10 :**

Tracer la courbe représentative de la fonction cosinus sur un intervalle de votre choix.

## Les fonctions

Voici un exemple de fonction.

```
def double(n):  
    """Renvoie le double de n"""  
    return 2*n
```

On visualise le résultat en faisant `print(double(10))`

Et en attendant le TP suivant, que font ces deux fonctions mystérieuses ?

```
def mystere_1(n):  
    if n == 0 :  
        return 1  
    else :  
        return n*mystere_1(n-1)  
  
def mystere_2(n):  
    resultat = 1  
    for k in range(2,n+1):  
        resultat = resultat * k  
    return resultat
```