

# LCLC séance 1

guilers

## Opérateurs

Entrer chaque ligne dans la console Python. Regarder le résultat.

```
# opérations de base
print(2 + 3)
print(6 - 2)
print(2 * 6)
# variables et affectation
n = 2
m = 42
print(n, m)
print(n * m)
# divisions, reste
print(m / n)
print(m // n)
print(m % n)
# égalités, comparaisons, différences
print(n == m)
print(n != m)
print(n == 42)
print(n == '42')
s = '42'
print(m == s)
print(n > 400)
print(n <= 430)
y = n + n
print(y)
z = s + s
print(s)
# entrée
i = input("Comment vous appelez-vous ?")
print(i)
# listes
L = [1, '2', 3]
print(L)
```

```
print(L[1])
print(L[0])
```

## If then else

### 1. Majeur / Mineur

On demande à l'utilisateur son âge, et on dit s'il est majeur ou mineur.

### 2. Nombres pairs

Écrire un script Python qui, pour un nombre `n` choisi, dit s'il est pair ou impair.

## Boucles for

### 1. somme des entiers

Écrire un script Python qui donne la somme des nombres entiers de 1 à 20.

Existe-t-il un moyen plus simple de programmer cette somme ?

### 2. Tables de multiplication

Écrire un script Python qui détaille la table de 3.

Comment le programmer plus simplement ?

## Boucles while

### 1. Quotient d'une division

Écrire un script Python qui, à l'aide d'une boucle `while`, calcule pour deux nombres `a` et `b` le quotient de la division de `a` par `b`.

### 2. Avion

Un avion, initialement à 11 000 mètres d'altitude, descend de 300 mètres chaque minute. Au bout de combien de temps passera-t-il sous les 2000 m d'altitude ?

Écrire un script permettant d'avoir la réponse.

## BOSS FINAL : conjecture de Syracuse

Les règles : - on choisit un nombre entier supérieur à 0 ; - s'il est pair : on le divise par 2 ; - s'il est impair : on le multiplie par 3 et on ajoute 1 ; - on recommence jusqu'à obtenir 1.

=> Programmer cet algorithme sur Python.

## LE BOSS FINAL MAIS LE VRAI CETTE FOIS QU'ON AVAIT PAS VU VENIR JUSTE APRÈS LE PREMIER BOSS FINAL :

**Le problème:** On dispose d'un code de 4 chiffres. Écrire un script Python permettant de le trouver (en essayant toutes les combinaisons possibles jusqu'à trouver la bonne).

***Pour les matheux :*** combien de combinaisons différentes vont être testées dans le pire des cas ?