

LCLC séance 1

guilers

Opérateurs

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')
```

If then else

1. Majeur / Mineur

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

```
age = input('Quel est votre âge ?')
if int(age) < 18:
    print('Vous êtes mineur.')
else:
    print('Vous êtes majeur !')
```

2. Nombres pairs

```
n = 3  # nombre choisi
```

```
if n % 2 == 0:
    print("Le nombre est pair !")
else:
    print("Le nombre est impair...")
```

Boucles for

1. somme des entiers

1) "En dur" :

```
x = 1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20
```

```
print(x)
```

2) avec une boucle :

```
y = 0
```

```
for i in range(21):
```

```
    y += i
```

```
print(y)
```

2. Tables de multiplication

1) "en dur":

```
print('3 * 0 =', 3)
print('3 * 1 =', 6)
print('3 * 2 =', 9)
...
print('3 * 10 =', 30)
```

2) avec boucle for :

```
for i in range(11):
    print(f'3 * {i} =', 3 * i)
```

Boucles while

1. Quotient d'une division

a = 42 *# nbre de départ (dividende)*

b = 3 *# nbre par lequel on divise (diviseur)*

q = 0

```
while a >= b:
```

```
    a = a - b
```

```
    q = q + 1
```

```
print(q)
```

```
print(a // b == q)
```

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 ?

```
h = 11000
```

```
t = 0
```

```
while h >= 2000:
```

```
    h = h - 300
```

```
    t = t + 1
```

```
print(t)
```


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.

```
n = 42  # nombre de départ
```

```
while n != 1:
```

```
    if n % 2 == 0 :
```

```
        n = n // 2
```

```
    else:
```

```
        n = n * 3 + 1
```

```
    print(n)
```

```
print(n)
```

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 ?

Solution :

```
code = 2242
N = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
for c1 in N:
    for c2 in N:
        for c3 in N:
            for c4 in N:
                tentative = c1 * 1000 + c2 * 100 + \
                            c3 * 10 + c4
                if tentative == code:
                    print("Trouvé !", tentative)
                    quit()
                else:
                    print('Dommage...', tentative)
```

Autre solution :

```
code = 2242
for essai in range(10000):
    if essai == code:
        print('Trouvé !', essai)
```