

# THÈME 8: BOUCLES FOR

# Notions du thème:

- Boucles For
  - Sur les structures de données
  - Avec range

# Itération

- Boucle while : la condition détermine le nombre de fois que la boucle est exécutée  
**boucle conditionnelle**
- Si on connaît ce nombre à l'avance, on peut utiliser le for...  
**boucle inconditionnelle**

# La boucle for

- Permet de parcourir des structures :  
Listes de nombres, d'objets, lettres d'un mots

```
for e in [1, 4, 5, 0, 9, 1] :
```

```
    print(e)
```

```
for e in ["a", "e", "i", "o", "u", "y"] :
```

```
    print(e)
```

```
for e in "python":
```

```
    print(e)
```

**e** prend successivement  
les valeurs de la liste parcourue

```
for i in range(1,6) :  
    print (i,end=",")
```



```
1, 2, 3, 4, 5,
```

- **range** (*deb*, *fin*, *pas*)
  - Fonction qui prend des arguments entiers
  - génère une séquence d'entiers entre [*deb*, *fin*[ avec le *pas* choisi.
- Les paramètres *deb* et *pas* sont **optionnels**
  - range** (**a**) : séquence des entiers dans [0, a[ , c'est-à-dire dans [0, a-1]
  - range** (**b**, **c**) : séquence des valeurs [b, c[, c'est-à-dire dans [b, c-1]
  - range** (**e**, **f**, **g**) : séquence des valeurs [e, f[ avec un pas de g
- **for var in range** (*deb*, *fin*, *pas*) :  
*instructions*

# A noter

- En cas d'**incohérence**, la boucle est **ignorée** et l'on passe aux instructions suivantes :

```

for k in range(200, 210, -2) :
    print(k)
for k in range(110, 100, -2) :
    print(k)
  
```

} *ignorée*

110  
108  
106  
104  
102

- Quoi qu'il arrive** dans le corps de la boucle, la variable du compteur prend la **valeur suivante** du *range* ou de la liste à chaque nouvelle étape de la boucle

```

for i in range(1, 5) :
    print(i)
    i = i*2
  
```

1  
2  
3  
4

# Exercice

- Pour les cas suivants, indiquer les valeurs successives affichées sur la console

```
for i in range(4) :  
    print(i)  
  
for j in range(2,5):  
    print(j)  
  
for k in range(3,12,3):  
    print(k)  
  
for l in range(12, 3):  
    print(l)  
  
for m in range(12,3,-2):  
    print(m)
```

```
# la liste en arg. doit ne contenir que des nb  
def somme_des_positifs(liste) :  
    s=0  
    for e in liste:  
        if e>0:  
            s=s+e  
    return s  
  
# prog. principal  
ma_liste = [2,-4,6,0,-5,1]  
for e in ma_liste:  
    print(e+1)  
t=somme_des_positifs(ma_liste)  
print(t)
```

- Ces slides ont été réalisés par:
  - Amir Charif
  - Lydie Du Bousquet
  - Aurélie Lagoutte
  - Julie Peyre
- Leur contenu est placé sous les termes de la licence **Creative Commons CC BY-NC-SA**

