

Semaine 4

Initiation à l'algorithmique et programmation

Revekka Kyriakoglou

Plan du cours

- 1 Iteration
 - Boucle while

Iteration



Itération

Une **itération** permet de répéter plusieurs fois une même série d'instructions, permettant de faire des récurrences ou de traiter de gros volumes de données.



Boucle

Une boucle dans un programme d'ordinateur est une instruction qui se répète jusqu'à ce qu'une condition spécifiée soit atteinte.

L'instruction while



Dans la boucle **while**, le programme répète un bloc d'instructions tant qu'une certaine condition est vraie.

```
while ( condition ):  
    instruction
```

- Tout d'abord, l'expression est évaluée.
- Si l'expression est **True**, l'instruction s'exécute et l'expression est évaluée de nouveau. Ensuite, la procédure recommence (c'est-à-dire que l'expression est évaluée)
- Si l'expression est **False**, la procédure se termine et nous sortons de la boucle while.



La condition s'appelle une **condition d'arrêt**.



Exercise 1

Essayez de décrire ce qui se passe avec le code suivant :

```
expression 1  
while ( expression 2 ):  
    instruction  
    expression 3  
expression 4
```

- Quelle est l'expression de comparaison ?
- Ce qui arrivera si nous supprimons l'expression 3 ?

Exemple :

```
chaine = "Python"  
chaine_inverse = ""  
  
index = len(chaine) - 1  
  
while (index >= 0):  
    chaine_inverse += chaine[index]  
    index -= 1  
  
print("original_str:", chaine)  
print("inversed_str:", chaine_inverse)
```

Exemple :

```
chaine = "Bonjour, bienvenue a Python!"
lettre_rech = "o"
compteur = 0
index = 0

while index < len(chaine):
    if chaine[index] == lettre_rech:
        compteur += 1
    index += 1

print(f"The '{lettre_rech}' appears {compteur} times.")
```

Exemple (puissance)

Exemple

Calculer la puissance d'un nombre entier.

```
i = 0 //initialisation
result = 1 //initialisation a 1 (produit)
while (i < k ):
    result = result * n
    i = i + 1 //progression

print(result)
```