

Semaine 5

Initiation à l'algorithmique et programmation

Revekka Kyriakoglou

Plan du cours

- Exemple : while
- Break
- Boucle for

Exemple

Exemple

Écrivez un programme qui génère un nombre aléatoire entre 1 et 100 en utilisant :

```
random.randint(debut, fin)
```

Ensuite, demandez à l'utilisateur de deviner le nombre.

- Si la supposition de l'utilisateur est incorrecte, le programme doit lui donner un indice pour l'aider à deviner si le nombre secret est plus grand ou plus petit que la supposition précédente.
- Le programme doit continuer de demander à l'utilisateur de deviner jusqu'à ce qu'il ait deviné correctement.

Bonus : Calculez le nombre de fois qu'il a essayé jusqu'à ce qu'il fonde l'entreprise.

Exemple (Solution)

```
import random
nombre_secret = random.randint(1, 100)
print("Devinez le nombre secret entre 1 et 100.")
supposition = int(input("Votre supposition:"))
essais = 1

while (supposition != nombre_secret)
    supposition = int(input("Votre supposition:"))
    essais += 1
    if supposition < nombre_secret:
        print("Le nombre secret est plus grand.")
    else:
        print("Le nombre secret est plus petit.")

print(f"Fini en {essais} essais.")
```

Break

- Quitte immédiatement la boucle dans laquelle il se trouve.
- Saute l'expression restante dans le bloc de code.
- Ne sort que de la boucle la plus proche.

Exemple (Solution 2)

```
import random
nombre_secret = random.randint(1, 100)
essais = 0
print("Devinez le nombre secret entre 1 et 100.")

while True:
    supposition = int(input("Votre supposition: "))
    essais += 1
    if supposition == nombre_secret:
        print(f"Fini en {essais} essais.")
        break
    elif supposition < nombre_secret:
        print("Le nombre secret est plus grand.")
    else:
        print("Le nombre secret est plus petit.")
```

L'instruction for

L'instruction for (= pour)

```
for element in sequence:  
    instruction
```

- element est une variable qui prend la valeur de chaque élément de la séquence à chaque itération
- sequence est l'objet itérable sur lequel vous itérez, comme une liste, un tuple, ou une chaîne de caractères

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



range(start, stop, step)



boocle jusqu'à stop-1

Example

```
s = 0
for i in range(5, 10):
    s += i
print('The sum is ' + s)
```

```
# We can use a step
s = 0
for i in range(5, 10, 2):
    s += i
print('The sum is ' + s)
```


for VS while LOOPS

for loops

- **know** number of iterations
- can **end early** via `break`
- uses a **counter**
- **can rewrite** a `for` loop using a `while` loop

while loops

- **unbounded** number of iterations
- can **end early** via `break`
- can use a **counter but must initialize** before loop and increment it inside loop
- **may not be able to rewrite** a `while` loop using a `for` loop

Figure – While VS for (Source : Introduction To Computer Science And Programming In Python, MIT)