



Les structures de contrôle

Enseignante: Mme Lamia MANSOURI

1. Les structures conditionnelles



1.1 l'instruction if

L'instruction if qui permet d'exécuter certaines instructions uniquement si une condition est vérifiée.
La forme la plus simple est :

```
if condition:  
    instructions
```

La forme complète :

```
if condition:  
    instructions  
elif condition:  
    instructions  
elif condition:  
    instructions  
...  
else:  
    instructions
```

```
max = 0  
note = float(input())  
if note > max :  
    max = note  
    print("Nous avons un nouveau record")  
print("Maximum note :", max)
```

Remarques :

- Une instruction composée se compose d'une ligne d'en-tête terminée par deux points :
- Importance de l'indentation : d'un bloc d'instructions indenté par rapport à la ligne d'en-tête. On utilise habituellement 4 espaces par indentation

1. Les structures conditionnelles



Exemple

Selon le calendrier grégorien, l'année sera bissextile :

- si l'année est divisible par 4 et non divisible par 100, ou
- si l'année est divisible par 400.

Sinon, l'année n'est pas bissextile.

correction

```
annee = int(input())
if annee % 400 == 0:
    print('bissextile')
elif annee % 100 == 0:
    print('non bissextile')
elif annee % 4 == 0:
    print('bissextile')
else:
    print('non bissextile')
```

Activité

On veut calculer la ou les éventuelles racines d'une équation du second degré.

Rappel : ayant l'équation du second degré $ax^2 + bx + c = 0$ où a , b et c sont des valeurs réelles et $a \neq 0$, pour savoir si cette équation a deux, une ou aucune racines réelles, il faut calculer le discriminant $\Delta = b^2 - 4ac$

Si la valeur de Δ est strictement positive, l'équation a deux racines réelles :

- $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$
- $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$

Si la valeur de Δ est égale à 0, l'équation a une racine : $x = -b / 2a$

Si la valeur de Δ est strictement négative, l'équation n'a pas de racine réelle.

1. Les structures conditionnelles



Correction

```
from math import sqrt
a= float(input('Donnez la valeur de a : '))
b= float(input('Donnez la valeur de b : '))
c= float(input('Donnez la valeur de c : '))
delta = b**2 - 4*a*c
if delta < 0:
    print(" pas de racines réelles")
elif delta==0:
    print("une racine : ")
    print("x = ",-b/2*a)
else:
    racined=sqrt(delta)
    print("deux racines : ")
    print("x1 = ", (-b + racined)/(2*a))
    print("x2 = ", (-b - racined)/(2*a))
```

2. Les structures itératives



2.1 l'instruction while

```
while cond :  
    instruction 1  
    ...  
    instruction n
```

Exemples:

```
n = 0  
while n < 3:  
    print("à l'intérieur ", n)  
    n += 1  
print("à l'extérieur ", n)
```

```
n = int(input("Entrez un entier [1 .. 10] : "))  
while not(1<=n<= 10):  
    n = int(input("Entrez un entier [1 .. 10], S.V.P. : "))
```

2. Les structures itératives



2.1 l'instruction while

```
while cond :  
    instruction 1  
    ...  
    instruction n
```

Exemples:

```
n = 0  
while n < 3:  
    print("à l'intérieur ", n)  
    n += 1  
print("à l'extérieur ", n)
```

```
n = int(input("Entrez un entier [1 .. 10] : "))  
while not(1<=n<= 10):  
    n = int(input("Entrez un entier [1 .. 10], S.V.P. : "))
```

2. Les structures itératives



Activité

Un épargnant place une somme d'argent en DT sur un compte rémunérant à un taux annuel de 3% à intérêts composés. Tous les ans, le jour de la perception des intérêts, il dépose 1000 DT sur ce compte.

Cet épargnant ne retire jamais d'argent de ce compte.

On veut calculer la valeur acquise au bout de 10 ans de placement après y avoir déposé une dernière fois les 1000 DT.

Correction

```
S=float(input('donner votre épargne initiale en DT:'))
n=1
while n<=10:
    S=1.03*S+1000
    n=n+1
print("Le capital acquis au bout de 10 ans s'élève ",S,"DT")
```


2. Les structures itératives



2.2 l'instruction for

```
for x in ensemble:  
    instruction 1  
    ...  
    instruction n
```

Où :

- x est un élément de l'ensemble ensemble.
- Les instructions 1 à n sont exécutées pour chaque élément x de l'ensemble ensemble. Cet ensemble peut être une chaîne de caractères, un tuple, une liste, un dictionnaire, un set ou tout autre type.

Instruction for utilisée avec range

```
for x in range (val_min,val_max,nbre_pas)  
    instruction 1  
    ...  
    instruction n
```

- Où x varie entre val_min et val_max -1 avec un pas égale à nbre_pas

| Boucle | Affichage |
|--|---|
| <pre>for lettre in "hello": print(lettre)</pre> | <pre>H E L L O</pre> |
| <pre>for x in [2, 'a', 3.14]: print(x)</pre> | <pre>2 a 3.14</pre> |
| <pre>for i in range(1,4): print(i)</pre> | <pre>1 2 3</pre> |
| <pre>for i in range(1,4,2): print(i)</pre> | <pre>1 3</pre> |
| <pre>for i in range(4): print(i)</pre> | <pre>0 1 2 3</pre> |
| <pre>nb_v = 0 for lettre in "Python est un langage interprété": if lettre in "aeiouy": nb_v = nb_v + 1 print("le nombre de voyelle dans la chaine est : ", nb_v)</pre> | <pre>le nombre de voyelle dans la chaine est : 10</pre> |

3. Ruptures des séquences



3.1 Interrompre une boucle

L'instruction `break` permet de sortir immédiatement de la boucle `for` ou `while` en cours contenant l'instruction :

```
for x in range(1, 11):  
    if x == 5:  
        break  
    print("x= ",x)
```

La boucle sera interrompue à la valeur de `x = 5` et on aura l'affichage de ces valeurs seulement :

```
x =1  
x=2  
x=3  
x=4
```

3.2 l'instruction continue

L'instruction `continue` passe immédiatement à l'itération suivante de la boucle `for` ou `while` en cours contenant l'instruction et reprend à la ligne de l'en-tête de la boucle

```
for x in range(1, 7):  
    if x == 5:  
        continue  
    print("x= ",x)
```

On aura cet affichage :

```
x=1  
x=2  
x=3  
x=4  
x=6 # la boucle a sauté la valeur 5
```

4. Utilisation avancée des boucles

→ while – else/ for - else

Les boucles **while** et **for** peuvent posséder une clause **else** qui ne s'exécute que si la boucle se termine normalement, c'est-à-dire sans interruption :

Exemple :

```
y = int(input("Entrez un entier positif : "))
while not(y > 0) :
    y = int(input('Entrez un entier positif, S.V.P. : '))

x = y // 2
while x > 1:
    if y % x == 0:
        print(x, "a pour facteur", y)
        break # voici l'interruption !
    x -= 1
else :
    print(y, "est premier.")
```