

Les structures conditionnelles



Plan

1. Structures conditionnelles simples
2. Structures conditionnelles généralisées

Structures conditionnelles simples

3

► Syntaxe

if condition:

 bloc_instruction1

else :

 bloc_instruction2

Structures conditionnelles simples

4

Indentation de bloc :

- ▶ Un bloc est défini par une indentation obtenue en décalant le début des instructions vers la droite grâce à des espaces en début de ligne (habituellement 4 espaces mais ce n'est pas obligatoire).
- ▶ Toutes les instructions d'un même bloc doivent être indentées exactement au même niveau (c'est-à-dire décalées à droite d'un même nombre d'espaces). L'irrespect de l'indentation génère une erreur.

Structures conditionnelles simples

5

Exercice

- ▶ Ecrire un programme qui demande d'introduire la moyenne de l'utilisateur et affiche s'il a de moyenne ou s'il est sous la moyenne

Structures conditionnelles simples

6

Solution

```
chaine = input("Note sur 20 : ")
note = float(chaine)
if note >= 10.0:
    print("J'ai la moyenne")
else:
    print("C'est en dessous de la moyenne")
print("Fin du programme")
>>>
Note sur 20 : 15
J'ai la moyenne
Fin du programme
```

Structures conditionnelles simples

7

▶ Deux niveaux de test :

▶ Syntaxe :

if condition1:

if condition1_1:

 bloc_instruction1

else:

 bloc_instruction2

Structures conditionnelles simples

8

Exercice

- ▶ Ecrire un programme qui demande d'introduire la moyenne de l'utilisateur et affiche la mention
- ▶ Bien si supérieur à 14
- ▶ Passable si entre 10 et 14
- ▶ Sous la moyenne si inférieur à 10

Structures conditionnelles simples

9

Exemple 1

```
chaîne = input("Note sur 20 : ")
note = float(chaîne)
if note >= 10.0:
    if note >= 13:
        print("mention Bien")
    else:
        print("mention Passable")
else:
    print("C'est en dessous de la moyenne")
print("Fin du programme")
>>>
Note sur 20 : 15
J'ai la moyenne
Fin du programme
```

Structures conditionnelles généralisée

10

► Syntaxe en utilisant else if

```
if condition_1:
    bloc_instruction_1_1
else :
    if condition_2:
        bloc_instruction_1_2
    else :
        bloc_instruction_2_2
else:
    bloc_instruction _1_2
```

► Syntaxe en utilisant elif

```
if expression 1:
    bloc d'instructions 1
elif expression 2:
    bloc d'instructions 2
elif expression 3:
    bloc d'instructions 3
else:
    bloc d'instructions 4
# suite du programme
```

Exemple :

```
chaîne = input("Note sur 20 : ")
note = float(chaîne)
if note>20.0 or note<0.0:
    print("Note invalide !")
else:
    if note>=10.0:
        print("J'ai la moyenne")
        if note==20.0:
            print("C'est même excellent !")
    else:
        print("C'est en dessous de la moyenne")
        if note==0.0:
            print("lamentable !")
print("Fin du programme")
>>>
Note sur 20 : 20
J'ai la moyenne
C'est même excellent !
Fin du programme
>>>
Note sur 20 : 3
C'est en dessous de la moyenne
Fin du programme
```

Exemple :

```
note = float(input("Note sur 20 : "))
if note==0.0:
    print("C'est en dessous de la moyenne")
    print("... lamentable !")
elif note==20.0:
    print("J'ai la moyenne")
    print("C'est même excellent !")
elif note<10.0 and note>0.0:
    # ou bien : elif 0.0 < note < 10.0:
    print("C'est en dessous de la moyenne")
elif note>=10.0 and note<20.0:
    # ou bien : elif 10.0 <= note < 20.0:
    print("J'ai la moyenne")
else:
    print("Note invalide !")
    print("Fin du programme")
```

>>>

Note sur 20 : 20

J'ai la moyenne

C'est même excellent

!Fin du programme

>>>

Note sur 20 : 3

C'est en dessous de
la moyenne

Fin du programme

>>>

Note sur 20 : 77

Note invalide !

Fin du programme

Exercice :

13

- Une année est bissextile (contient 366 jours) si elle est multiple de 4, sauf les années de début de siècle (qui se terminent par 00) qui ne sont bissextiles que si elles sont divisibles par 400.

Exemples :

- 1980 et 1996 sont bissextiles car elles sont divisibles par 4
- 2000 est une année bissextile car elle est divisible par 400
- 2100 et 3000 ne sont pas bissextiles car elles ne sont pas divisibles par 400.

Ecrire un script python qui permet de déterminer si un entier positif donné correspond à une année bissextile ou non.

Correction :

```
annee= int (input ("saisir une annee"));
```

```
if annee % 4 != 0:
```

```
    print("année non bisseccstile")
```

```
elif annee % 100 == 0 :
```

```
    if annee % 400 == 0 :
```

```
        print("année bisseccstile")
```

```
    else :
```

```
        print("année non bisseccstile")
```

```
else :
```

```
    print("année bisseccstile")
```

Les structures itératives



Plan

1. La boucle « for »
2. La boucle « while »
3. Utilisation avancée des boucles

Structures itératives : for

17

► Syntaxe

for élément in itérable :
instructions

Structures itératives : for

18

- ▶ Grâce à la fonction **range([début], fin, [pas])** la boucle **for** peut parcourir un intervalle de début à fin-1 ou en définissant la borne initiale.
- ▶ Elle peut également parcourir d'autres objets altérables (listes, chaînes, tuples, ensembles, dictionnaires, tableaux...).

Exemple 1 :

```
for i in range(5):  
    print(i)  
print("Fin de la boucle")
```

```
>>>  
0  
1  
2  
3  
4  
Fin de la boucle
```

Structures itératives : for

19

Exemple 2 :

```
for compteur in range(1,11):
```

```
    print(compteur, '* 9 =', compteur*9)
```

```
print("Fin de la boucle")
```

```
>>>
```

```
(1, '* 9 =', 9)
```

```
(2, '* 9 =', 18)
```

```
(3, '* 9 =', 27)
```

```
(4, '* 9 =', 36)
```

```
(5, '* 9 =', 45)
```

```
(6, '* 9 =', 54)
```

```
(7, '* 9 =', 63)
```

```
(8, '* 9 =', 72)
```

```
(9, '* 9 =', 81)
```

```
(10, '* 9 =', 90)
```

```
Fin de la boucle
```

Structures itératives : for

20

Exemple 3 :

```
for i in range(1,8,2):  
    print(i**2)  
print("Fin de la boucle")
```

```
>>>
```

```
1
```

```
9
```

```
25
```

```
49
```

```
Fin de la boucle
```

Structures itératives : for

21

Importance des indentations : C'est quoi la valeur de x et y en fin de deux programmes ?

#1^{er} programme

```
>>> x = y = 1
>>> for i in range(5):
    x += 1
    y += 1
>>> print (x,y)
```

#2^{ème} programme

```
>>> x=y=1
>>> for i in range(5)
    x+=1
>>> y+=1
>>> print (x,y)
```

Résultats :

<http://prepa-info.blogspot.com>

Programme 1 : 6,6

Programme2 : 6,2

Structures itératives : while

22

► Syntaxe

while expression :
 bloc instructions

Structures itératives : while

23

Exemple 1 :

```
Compteur = 1
```

```
while compteur<5:
```

```
    print(compteur, compteur<5)
```

```
    compteur += 1
```

```
print(compteur<5)
```

```
print("Fin de la boucle")
```

```
>>>
```

```
1 True
```

```
2 True
```

```
3 True
```

```
4 True
```

```
False
```

```
Fin de la boucle
```

Utilisation avancée des boucles

24

L'instruction break

- ▶ L'instruction break provoque une sortie immédiate d'une boucle while ou d'une boucle for.

- ▶ **Exemple : Que fait ce programme ?**

```
somme=0
while True:
    n=int(input("Entrez un nombre 0 pour arrêter") )
    if n==0 :
        break
    somme=somme+n
print("la somme des nombres est",somme)
```

- ▶ Dans cet exemple, l'expression True est toujours vraie : on a une boucle sans fin.
- ▶ L'instruction break est donc le seul moyen de sortir de la boucle.

Utilisation avancée des boucles

25

L'instruction continue

- ▶ Permet de passer immédiatement à l'itération suivante de la boucle for ou while;

- ▶ **Exemple :**

```
>>> for x in range(1, 11):
```

```
    if x == 5:
```

```
        continue
```

```
    print(x, end=" ")
```

```
1 2 3 4 6 7 8 9 10
```

```
>>> # la boucle a sauté la valeur 5
```

while – else et **for- else**

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

Utilisation avancée des boucles

27

Exemple : Ce programme python permet de vérifier si un entier positif saisi au clavier est premier, sinon il affiche ses diviseurs

```
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.")
```

Exercice 1

28

Question :

Ecrire un script python qui permet de déterminer si un entier donné est totalement divisible par 2

Exemple :

8 est totalement divisible par 2 car :

$$8 \text{ div } 2 = 4,$$

$$4 \text{ div } 2 = 2,$$

$$2 \text{ div } 2 = \underline{1}$$

24 n'est pas totalement divisible par 2 car :

$$24 \text{ div } 2 = 12,$$

$$12 \text{ div } 2 = 6,$$

$$6 \text{ div } 2 = \underline{3}$$

Solution : Exercice 1

```
n1 = int(input("Entrez un entier "))
```

```
n = n1
```

```
while n % 2 == 0 and n != 1 :
```

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

```
    print(n)
```

```
else :
```

```
    print(n1, " totalement divisible par 2")
```

Exercice 2

30

Question :

Ecrire un script python qui demande à l'utilisateur un nombre entier, puis affiche une liste de tous les diviseurs de ce nombre.

Solution : Exercice 2

31

```
n = int(input("Entrez un entier "))  
for nb in range (1,n+1):  
    if n % nb == 0:  
        print(nb)
```

Exercice 3

32

Questions :

- ▶ Générer un nombre aléatoire entre 1 et 9 (y compris 1 et 9). `random.randint(x,y)`
- ▶ Demandez à l'utilisateur de deviner le nombre, puis leur dire qu'ils devinaient trop faible, trop élevé, ou tout à fait exact.

Extras:

- ▶ Garder le jeu en cours d'exécution jusqu'à ce que l'utilisateur tape "exit"
- ▶ Quand le jeu se termine, afficher le nombre d'essais faites pour arriver au bon choix.

Solution 3

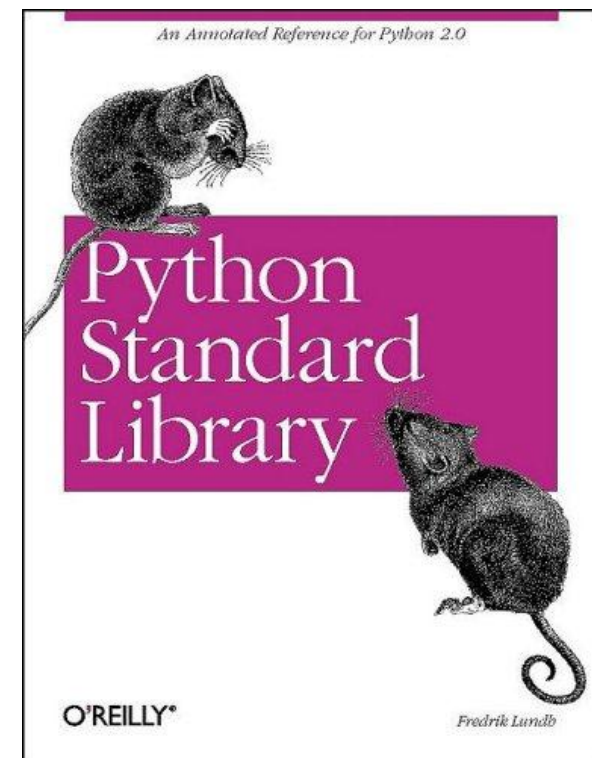
33

```
import random
nombre = random.randint(1,9)
choix = 0
compteur= 0
while choix != nombre and choix != "exit":
    choix = input("Votre choix?")
    if choix == "exit":
        Break
    choix = int(choix)
    compteur += 1
    if choix < nombre:
        print("Très faible!")
    elif choix > nombre:
        print("Très élevé!")
    else:
        print("Tout à fait exact!")
        print("Vous avez fait",compteur," essais!")
```

Et vraiment beaucoup de choses à savoir



Livres



Liens utiles

- ▶ www.python.org
- ▶ www.python.org
- ▶ www.docs.python.org
- ▶ www.codecademy.com/tracks/python
- ▶ <http://effbot.org/media/downloads/librarybook-core-modules.pdf>

ESSAYEZ-LE, VOUS
FINIREZ PAR
L'AIMER 😊