

Les structures conditionnelles



Jean Yves Aristide YAO

Table des matières



I - Objectifs	3
II - Introduction	4
III - Instructions conditionnelles	5
1. Forme minimale "if"	5
2. Forme normale "if" et "else"	5
3. Forme complète "if", "elif" et "else"	6
4. Exercice	6
5. Exercice	7
6. Exercice	7
IV - Condition : prédicat ou prédicats	8
1. Opérateurs de comparaison	8
2. Prédicat et booléens	8
3. Imbrication de prédicat ou de condition	9
4. prédicat(s)	9
5. Exercice	10
6. Exercice	10
7. Exercice	11
8. Exercice	11
9. Exercice	11
V - Conclusion	12
VI - Solutions des exercices	13



Objectifs

- Écrire une structure conditionnelle
- Utiliser les conditions pour réaliser des programmes avancés

Introduction



Python exécute normalement les instructions de la première à la dernière ligne, sauf lorsqu'il rencontre une *instruction conditionnelle*. Une telle instruction va permettre de faire une action précise si, par exemple, une variable est positive, une autre action si cette variable est négative, ou une troisième action si la variable est nulle.

Pour permettre une bonne manipulation des structures conditionnelles, cette leçon présente les différentes formes de structure conditionnelles et les conditions

Instructions conditionnelles



Objectifs

- Décrire la structure d'une instruction conditionnelle
- Identifier une indentation

1. Forme minimale "if"

Blocs d'instructions

On entend par bloc d'instructions une série d'instructions qui s'exécutent dans un cas précis.

if pour dire "si"

L'instruction *if* permet d'exécuter un bloc d'instruction si une condition est vraie. Si la condition n'est pas vraie alors le bloc d'instruction n'est pas exécuté.



Syntaxe : python

1. `>>>if condition :`
2. `[indentation] bloc d'instruction en python`

La ligne 1 représente un test conditionnel. Il se compose de :

1. Le mot clé « *if* », en anglais qui signifie « *si* » en français
2. De la « condition », qui est un test généralement réalisé sur des variables.
3. Du signe deux points, « `:` », qui termine la condition et est indispensable ; son omission entraîne l'affichage d'un message d'erreur.

La ligne 2, représente le bloc d'instruction qui sera exécuté si et seulement si la condition de la ligne 1 est vraie. *[indentation]* signifie que toutes les instructions contenues dans le bloc d'instruction doivent être décalées vers la droite (soit par des espaces ou par une tabulation).



Attention : indentation

En Python l'*indentation* est très importante car elle permet à l'interpréteur de savoir où se trouve le début et la fin d'un bloc d'instruction.

2. Forme normale "if" et "else"

else pour dire "sinon"

L'instruction *else* permet d'exécuter un autre bloc d'instruction au cas où la condition qui suit l'instruction *if* n'est pas vraie (la condition est fausse).

Syntaxe : python

1. >>> if condition :
2. [indentation] bloc d'instruction 1
3. else :
4. [indentation] bloc d'instruction 2

Si la condition de la ligne 1 est fausse alors le bloc d'instruction 1 n'est pas exécuté mais le bloc d'instruction 2 est exécuté.

Remarque

L'instruction if et l'instruction else sont au même niveau d'indentation.

3. Forme complète "if", "elif" et "else"

elif pour dire "sinon si"

La forme normale permet de tester seulement une condition et son contraire (c'est à dire deux cas). Cependant il existe des situations où il faudrait traiter plus que deux cas. Par exemple, pour vérifier la mention qu'un étudiant a obtenu à un examen, il existe quatre mentions (passable, bien, très bien, honorable) ; dans ce cas il faudrait traiter quatre cas. L'instruction *elif* se place toujours entre le *if* et le *else*, elle permet de tester des conditions intermédiaire et elle peut se répéter autant de fois que possible dans un code.

Syntaxe

1. >>> if condition 0 :
2. [indentation] bloc d'instruction 0
3. elif condition 1 :
4. [indentation] bloc d'instruction 1
5. elif condition 2 :
6. [indentation] bloc d'instruction 2
7. else :
8. [indentation] bloc d'instruction 3

Si aucune *condition n* (n appartient à $[0, 2]$) n'est vraie alors le bloc d'instruction 3 sera exécuté. Sinon si une *condition n* est vraie alors seulement le bloc d'instruction n correspondant sera exécuté.

Remarque

l'instruction else ne peut figurer qu'une fois, clôturant le bloc de la condition. Deux instructions else dans une même condition ne sont pas envisageables.

4. Exercice

[Solution n°1 p 13]

Quelle partie d'une instruction conditionnelle if doit être indentée ?

- ☐ La première ligne
- ☐ Toutes les lignes
- ☐ Le bloc d'instruction qui est à l'intérieur du if

5. Exercice

[Solution n°2 p 13]

Agencez dans l'ordre les mots suivants pour que le code fonctionne correctement

1. :
2. condition:
3. bloc d'instruction pour condition vrai
4. if
5. [indentation]
6. [indentation] bloc d'instruction pour condition fausse
7. else

Réponse : _ _ _ _ _

6. Exercice

[Solution n°3 p 13]

1. elif 1
2. if
3. elif 2
4. else

Réponse : _ _ _ _ _

Condition : prédicat ou prédicats



Objectifs

- Identifier les opérateurs de comparaison en Python
- Construire un programme avancé en utilisant les conditions

1. Opérateurs de comparaison

Opérateur	Signification
<	Strictement inférieur à
>	Strictement supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
==	Égal à
!=	Différent de

Les conditions introduisent ces opérateurs appelés *opérateurs de comparaison*

Attention

Il existe une différence entre l'opérateur d'affectation "=" et l'opérateur d'égalité "=="

2. Prédicat et booléens

Prédicat

Un *prédicat* est la condition qui est située après le "if" et les ":" lorsqu'on écrit une instruction conditionnelle.

Rapport entre prédicat et booléen

Un prédicat fournit toujours en sortie une *valeur booléenne* ; c'est-à-dire soit *True* pour vrai soit *False* pour faux. En fonction de la valeur retournée par un prédicat, un bloc d'instruction peut être exécuté ou non.

Exemple : Python 2 ou Python 3

```
>>>age = 25
>>>age > 12 -> cette instruction représente un prédicat
```


True -> valeur retournée par le prédicat

```
>>> age < 25
```

False

```
>>> age == 25
```

True

❖ Rappel : Les variables booléennes

True et *False* sont des valeurs ayant leur première lettre en *majuscule*. Écrire *False* sans un '*F*' majuscule ou *True* sans un '*T*' majuscule va entraîner une erreur lors de l'interprétation de l'instruction Python.

3. Imbrication de prédicat ou de condition

Imbrication

Une imbrication consiste à écrire une condition à l'intérieur d'une autre condition. Dans ce cas, l'indentation permet de comprendre clairement le schéma d'exécution du programme

☞ Exemple : Python 2 ou Python 3

On voudrait vérifier si une variable *age* est comprise dans un intervalle afin de décider si un individu est adolescent ou pas. On sait qu'un individu est adolescent si son âge est compris entre 14 et 21 ans

```
>>> age = 20
>>> if age >= 14:
    if age <= 21:
        print("l'individu est adolescent")
    else:
        print("l'individu n'est pas adolescent")
else:
    print("l'individu n'est pas adolescent")

l'individu est adolescent
```

Exécution de conditions imbriquées

le prédicat 1 renvoie True, dans le bloc d'instruction qui est exécuté, on y voit le prédicat 2 qui renvoie aussi True. Le bloc d'instruction indenté est alors affiché à l'écran et le programme s'arrête. On dit que l'instruction conditionnelle *if age <= 21* est imbriquée à l'intérieur de l'instruction *if age >= 14*.

4. prédicat(s)

Une condition peut contenir plusieurs prédicats. Dans ce cas ils sont associés par des *connecteurs*. Dans certains cas, il est possible de transformer des conditions imbriquées en une seule condition où les prédicats sont reliés par un ou des connecteurs. Le tableau suivant montre les différents connecteurs

Connecteur	Signification	Valeur de retour
and	et	True and True = True True and False = False
or	ou	True or False = True
not	non	not True = False not False = True

☞ *Exemple : reprise de l'exemple précédent*

```
>>> age = 25
>>> if age >= 14 and age <= 21:
    print("l'individu est adolescent")
else:
    print("l'individu n'est pas adolescent")

l'individu n'est pas adolescent
```

prédicats reliés par un connecteur

On a le prédicat 1 qui renvoie la valeur True et le prédicat 2 qui renvoie la valeur False. Or *True and False = False* donc le bloc d'instruction exécuté est celui qui est contenu dans l'instruction else.

✂ *Conseil*

En reliant les prédicats par le *connecteur and*, le code devient beaucoup plus court et surtout il est lisible par rapport au code imbriqué qui est un peu moins lisible.

5. Exercice

[Solution n°4 p 13]

Les variables Python de type booléen peuvent prendre deux valeurs :

- ☐ true and false
- ☐ True and false
- ☐ True and False
- ☐ Vrai et Faux

6. Exercice

[Solution n°5 p 13]

Quelle est la valeur retournée par ce prédicat ?

```
>>> 7 != 8
```

- ☐ False
- ☐ True
- ☐ Ce prédicat entraînera une erreur

7. Exercice

[Solution n°6 p 13]

Donnez le résultat d'exécution de ce code ?

```
oeuf = 7
if oeuf > 5 :
[indentation] print("cinq")
if oeuf > 8 :
[indentation] print("huit")
```



8. Exercice

[Solution n°7 p 14]

Donnez le résultat d'exécution de ce code ?

```
numero = 7
if numero > 3 :
[indentation] print("3")
[indentation] if numero < 5 :
[indentation][indentation] print("5")
[indentation][indentation] if numero == 7 :
[indentation][indentation][indentation] print("7")
```



9. Exercice

[Solution n°8 p 14]

Quel est le résultat d'exécution de ce code ?

```
age = 25
if age >= 25 or age/2 > 30 :
[indentation] print("Vrai")
else :
[indentation] print("Faux")
```



Conclusion



Cette leçon a présenté la notion de structure conditionnelle qui permet à l'apprenant de réaliser des programmes un peu plus complexe que ceux vu lors des deux première leçon. Il pourra par exemple écrire un programme qui résout une équation de second degré.

La prochaine leçon s'intéressera aux structures répétitives.



Solutions des exercices

> Solution n°1

Exercice p. 6

- ☐ La première ligne
- ☐ Toutes les lignes
- ☒ Le bloc d'instruction qui est à l'intérieur du if

> Solution n° 2

Exercice p. 7

1. if
2. condition:
3. [indentation]
4. bloc d'instruction pour condition vrai
5. else
6. :
7. [indentation] bloc d'instruction pour condition fausse

> Solution n° 3

Exercice p. 7

1. if
2. elif 1
3. elif 2
4. else

> Solution n°4

Exercice p. 10

- ☐ true and false
- ☐ True and false
- ☒ True and False
- ☐ Vrai et Faux

> Solution n°5

Exercice p. 10

- ☐ False
- ☒ True
- ☐ Ce prédicat entraînera une erreur

> **Solution n° 6***Exercice p. 11*

cinq

> **Solution n° 7***Exercice p. 11*

3

toutes les instructions en dessous de l'instruction *if numero < 5* constitue un bloc d'instruction. Donc elle s'exécute si et seulement l'instruction *if numero < 5* vaut True.

> **Solution n° 8***Exercice p. 11*

Vrai