PROGRAMMATION PYTHON

Chapitre 3: Structures conditionnelles

Sommaire

Introduction

Les structures conditionnelles vont nous permettre de faire des tests afin d'adapter les instructions que notre ordinateur doit exécuter dans certains cas.

Les structures conditionnelles permettent de vérifier une ou plusieurs conditions appelées **prédicats.**

On rencontre les structures conditionnelles dans la quasi-totalité des algorithmes de nos jours.

Dans les structures conditionnelles on rencontre:

- Le test sans alternative: exécution d'une instruction quand un test est vrai.
- Le test avec alternative: exécution d'une instruction quand le test est vrai et exécution d'une autre instruction quand le test faux.
- Le test composé: test avec plusieurs conditions associées par les opérateurs logiques **Ou (or), Et (and), non (not)**
- Le test imbriqué: test qui contient un autre test, c'est une suite de tests.

Commentaires

Pour donner des indications sur un code qu'on écrit, on utilisera les commentaires qui sont des messages qui non exécutés par l'interpréteur lors de l'exécution. On distingue deux types de commentaires en Python:

Les commentaires sur une ligne: commence par le caractère dièse #.

```
# Affichage du contenu de la variable
a = 1
print("La variable a contient:", a)
```

Les commentaires multilignes: commentaires sur plusieurs lignes encadrées par 3 guillemets doubles ou simples.

```
"""
    Ce code affiche le message de bienvenue
"""

print("Hello World")
```

```
'''
    Ce code affiche le message de bienvenue
'''

print("Hello World")
```

Condition if

En Python pour écrire un test sans alternative on utilise le mot clé **if** avec le format suivant:

```
if condition:
    instructions
```

Les deux points à la fin de la condition sont obligatoires ainsi que l'indentation avant écriture des instructions.

```
"""
   Ce programme affiche le message Bienvenue si et seulement si le nom d'utilisateur est Christophe.
"""

username = "Christophe"

if username == "Christophe":
    print("Bienvenue !")
```

Programme 1

```
"""
   Ce programme affiche le message Bienvenue si et seulement si le nom d'utilisateur est Christophe.
"""

username = "John"

if username == "Christophe":
    print("Bienvenue !")
```

Programme 2

Le programme 1 affichera bien « Bienvenue » lors de l'exécution, contrairement au programme 2 qui ne fera rien lors de son exécution.

Conditions if/else

En Python pour écrire un test avec alternative on utilise les mots clés **if/else** avec le format suivant:

```
if condition:
    instructions
else:
    instructions alternatives
```

Avec if/else, le code aura un comportement différent selon que la condition soit vraie ou fausse.

```
"""
    Ce programme affichera 'Vous êtes Christophe' si le nom d'utilisateur est Christophe.
    Sinon il affichera 'Vous n'êtes pas Christophe'.
"""

username = "Christophe"

if username == "Christophe":
    print("Vous êtes Christophe")
else:
    print("Vous n'êtes pas Christophe")
```

Le programme précédent exécutera une action peut importe le résultat du test sur la variable **username.**

En Python la simple égalité permet d'assigner une valeur à une variable alors que la double égalité == permet d'exprimer la condition d'égalité de deux valeurs.

Conditions if/elif/else

Parfois on a besoin de faire des tests imbriqués ou suite de tests. Avec Python, on utilisera les mots clés **if/elif/else** avec le format suivant:

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

Suite de tests

```
if condition1:
    if condition2:
        instructions1
    else:
        instructions2
else:
    intructions
```

Tests imbriqués

Généralement une suite de tests peut être convertie en tests imbriqués et vice versa.

```
username = "Christophe"


if username == "Christophe":
    print("Bienvenue")
elif username == "John":
    print("Bienvenue")
else:
    print("Utilisateur inconnu")
```

Est équivalent à

```
if username == "Christophe":
    print("Bienvenue")
else:
    if username == "John":
        print("Bienvenue")
    else:
        print("Utilisateur inconnu")
```

Opérateurs de comparaison

Pour faire les tests en Python on dispose des operateurs de comparaison suivants:

Opérateurs	Signification littérale
<	Strictement inférieur à
>	Strictement supérieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
==	Egal à
!=	Différent de

Ces operateurs de comparaison ont pour résultat **True/False** selon la condition à vérifier.

```
>>> a = 1
>>> a == 1
True
>>> a > 1
False
>>> a >= 1
True
>>> a < 1
False
>>> a <= 1
True
>>> a != 1
False
>>>
```

Opérateurs logiques

On peut faire une association de conditions avec les opérateurs logiques suivants:

Opérateurs	Signification littérale
and	Et
or	ou
is not	non

Ces operateurs de comparaison ont pour résultat **True/False** selon la condition à vérifier.

```
>>> a = 1
>>> b = 2
>>> a == 1 and b == 2
True
>>> a == 1 and b == 3
False
>>> a == 1 or b == 2
True
>>> a == 1 and b is not 2
False
>>> b is not 3
True
```

Ne pas oublier de mettre des parenthèses en cas d'importance d'ordre de priorité des conditions et pour éviter toute ambiguïté.

Programme interactif

En Python, pour afficher un message ou le contenu d'une variable, on utilise la fonction **print.** Pour demander à l'utilisateur de saisir une valeur on utilisera la fonction **input** avec le format suivant:

nomVariable = input(message_a_afficher)

```
saisie = input("Entrer votre age:")

# La saisie de l'utilisateur est une chaine de caractère de type string
print(type(saisie))

# Conversion en entier car un age est en entier
age = int(saisie)

# Condition en fonction de l'âge
if age >= 18:
    print("Vous êtes majeur")
else:
    print("Vous êtes mineur")
```

Exécution

```
Entrer votre age:12
<class 'str'>
Vous êtes mineur
```

```
Entrer votre age:18
<class 'str'>
Vous êtes majeur
```

Le retour de la fonction **input** est une chaine de caractère de type String. Il faut faire une conversion de la saisie de l'utilisateur si on veut faire des opérations numériques…

On peut utiliser le **cast** qui consiste à convertir un type de données en un autre type de données avec le format suivant:

type_de_destination(variable_à_convertir)

Exercices

Exercice 1

Réécrire le programme suivant en utilisant les opérateurs logiques.

```
username = "Christophe"


if username == "Christophe":
    print("Bienvenue")
elif username == "John":
    print("Bienvenue")
else:
    print("Utilisateur inconnu")
```

Exercice 2

Ecrire un programme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif. On prendra en compte le cas où le nombre est nul.

Exercice 3

Ecrire un programme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif ou nul.

Exercices

Exercice 4

Ecrire un programme qui demande l'âge d'une personne à l'utilisateur. Ensuite, il l'informe de sa catégorie: Poussin de 0 à 7 ans, Pupille de 8 à 9 ans, Minime de 10 à 11 ans, Cadet de 12 à 17 ans, Junior de 18 à 49 ans et Senior après 50 ans.

Exercice 5

Soit un utilisateur dont le nom d'utilisateur est « christophe » et son mot de passe « chris1234 ».
Ecrire un programme qui demande à l'utilisateur son nom d'utilisateur et son mot de passe, le programme doit afficher « Nom d'utilisateur et/ou mot de passe incorrect(s) » si l'une des deux valeurs est incorrecte sinon afficher « Bienvenue Christophe ! » .

Exercice 6

Reprendre le programme de l'exercice 5 mais cette fois-ci donner une indication sur l'erreur de l'utilisateur comme « Nom d'utilisateur incorrect ! » ou « Mot de passe incorrect ! » ou « Nom d'utilisateur et mot de passe incorrects ! ».

FIN CHAPITRE 3