02-cours_python_conditions_indent

January 11, 2020

1 Programmation conditionnelle et indentation

1.1 Condition

Une condition est le résultat d'une **évaluation** booléenne.

Voici les comparateurs :

- Egalité : ==
- Non égalité (différent) : !=
- Supérieur (ou égal) : >, >=
- Inférieur: (ou égal) : <, <=

[1]:
```
nombre = 7
# Notez le comparateur avec un *double* signe '=' (qui n'est donc pas ici une␣
 ↪affectation !).
print(nombre == 7)
print(nombre != 7)
```

```
True
False
```

1.2 Conditions *if* / *elif* / *else*

[6]:
```
# Exemple simple avec l'instruction if

nombre = 0
# Suite du programme
# ...
if nombre == 0:  # nombre == 0 est une expression booléenne
    print("Bonjour, c'est votre jour de chance !")
```

```
Bonjour, c'est votre jour de chance !
```

[1]:
```
# Exemple avec if / elif / else

valeur1 = 12
valeur2 = 14
```

```python
if valeur1 > valeur2:
    print("valeur1 est plus grande que valeur2")
elif valeur1 == valeur2:
    print("valeur1 et valeur2 sont égales")
else:
    print("valeur1 est plus petite que valeur2")
```

```
valeur1 est plus petite que valeur2
```

1.3 Le rôle fondamental de l'indentation

1.3.1 Définition

- L'indentation définit un bloc d'instructions.
- L'indentation est obtenue en appuyant sur la touche "TAB" du clavier.

Par exemple ici les trois instructions *print* ne sont exécutées que si a vaut 2 :

```python
# Notez la présence du caractère ":"
# Il marque le début d'un bloc
if a == 2:
    print(f"a vaut {a}")
    print("Ce programme est fascinant")
    print("Il ne peut être compris que par des êtres intelligents")
```

Remarques :

- Il ne faut jamais mixer les espaces et les tabulations, sinon les ennuis sont garantis.
- On peut régler son éditeur pour que lorsqu'on appuie sur la touche *tab* il génère 4 espaces.
- Le faire dans Spyder "Préférences… Editeur…".

1.4 Exercices

1.4.1 Devinette

Sans taper le code, indiquer quelle est la différence entre ces deux extraits de code :

```python
if a == 2:
    print("Je me surpasse en programmation")
    print("Je suis un Dieu en Python")
```

et :

```python
if a == 2:
    print("Je me surpasse en programmation")
print("Je suis un Dieu en Python")
```

1.4.2 Exercice : VRAI / FAUX

Demander à l'utilisateur d'entrer un nombre décimal. À l'aide de Python, vérifier si le nombre rentré par l'utilisateur est égal à 11.5. Si oui, afficher le message "VRAI", sinon afficher "FAUX".

[3]:
```
reponse = input("Entrez une valeur décimale : ")
valeur_d = float(reponse)
if valeur_d == 11.5:
    print("VRAI")
else:
    print("FAUX")
```

```
Entrez une valeur décimale : 11.5
VRAI
```

1.4.3 Exercice : nombre pair ou impair

Demander à l'utilisateur de saisir un entier. Le programme affichera si le nombre est pair ou impair.

[5]:
```
reponse = input("Saisissez un nombre entier : ")
nb = int(reponse)
if nb%2 == 0:
    print("Ce nombre est pair")
else:
    print("Ce nombre est impair")
```

```
Saisissez un nombre entier : 11
Ce nombre est impair
```

1.4.4 Exercice : comparaisons

Afin de mettre en place un système de caisses automatiques, le directeur du cinéma ''Les Studios" vous demande de coder un programme demandant au client son âge, s'il est abonné et lui annonçant en retour le tarif appliqué :

- Moins de 18 ans : 4 €
- Abonné : 5,50 €
- Sinon : 8 €

[6]:
```
reponse = input("Quel est votre âge ? ")
age = int(reponse)
if age < 18:
    print("Tarif enfants : 4 €")
else:
    abonne = input("Etes-vous abonné ? (o/n) ")
    if abonne == 'o':
        print("Tarif réduit : 5.50 €")
    else:
        print("Plein tarif : 8 €")
```

```
Quel est votre âge ? 19
Etes-vous abonné ? (o/n) o
Tarif réduit : 5.50 €
```

1.4.5 Exercice : le plus grand des nombres

Soit trois nombres, n1, n2 et n3. Ecrire un programme indiquant lequel de ces trois nombres est le plus grand.

On peut utiliser le **module** *random* de Python pour "tirer" des nombres au hasard. Il s'utilise de la façon suivante :

[2]:
```
import random

n1 = random.randint(0, 100)
print(n1)
```

```
41
```

[10]:
```
import random

NB_MIN = 0  # Par convention, en majuscules, car c'est une constante
NB_MAX = 100

n1 = random.randint(NB_MIN, NB_MAX)
n2 = random.randint(NB_MIN, NB_MAX)
n3 = random.randint(NB_MIN, NB_MAX)

# On recherche le plus grand des 3
if n1 >= n2 and n1 >= n3 :
    plus_grand = n1
elif n2 >= n1 and n2 >= n3 :
    plus_grand = n2
else :
    plus_grand = n3

print(f"Nombres tirés : {n1}, {n2}, {n3}")
print(f"Nombre le plus grand : {plus_grand}")
```

```
Nombre tirés : 74, 14, 66
Nombre le plus grand 74
```

1.5 Le type booléen

Nous avons vu jusqu'à présent essentiellement trois types : les nombres entiers (integer en anglais, ***int*** en Python), les nombres décimaux (***float*** en Python) et les chaines de caractères (string en anglais, ***str*** en python). Nous avons pu manipuler un autre type: le type booléen (boolean en anglais, ***bool*** en Python). Ce type accepte deux valeurs : ***True*** et ***False*** (ce sont des mots-clés Python et ils commencent bien par une majuscule).

1.5.1 Exercice : sans exécuter le programme, indiquer ce qu'affichent les instructions *print* ci-dessous

```
a = 1
b = 2
c = 2
print(a == b)
print(b == c)
d = a == b
print(d)
print(type(d))
e = False  # Attention à bien mettre la majuscule (idem pour True)
print(e)
f = True
if not d:  # Équivalent à if d == False
    print("Pas bon!")
elif f:
    print("Toujours pas")
if f:  # Équivalent à if f == True
    print("C'est tout bon !")
```

Notez que le choix du nom des variables est tout sauf explicite. Comme il s'agit d'un exercice de gymnastique mentale, cela est autorisé…

[5]:
```
a = 1
b = 2
c = 2
print(a == b)
print(b == c)
d = a == b
print(d)
print(type(d))
e = False  # Attention à bien mettre la majuscule (idem pour True)
print(e)
f = True
if not d:  # Équivalent à if d == False
    print("Pas bon!")
elif f:
    print("Toujours pas")
if f:  # Équivalent à if f == True
    print("C'est tout bon !")
```

```
False
True
False
<class 'bool'>
False
Pas bon!
C'est tout bon !
```