

---

## TP (2/7) initiation au langage Python : les structures conditionnelles

### Remarque préliminaire

#### 1- if ou if...else

##### a) Instruction if

Exécutez la *cellules Code* suivante (sélectionner la "cellule Code", puis cliquer sur *Exécuter* ou utiliser le raccourci clavier *CTRL + Entrée*).

```
x = 11
if x > 10:
    print("Vous avez la moyenne")
print("Fin")
```

On constate ici que lorsque  $x$  contient l'entier 11, la condition booléenne  $x > 10$  est vraie donc la ligne 3 est exécutée. Une fois que cette ligne 3 est exécutée, le script se poursuit, on "sort du if" et le script se poursuit à la ligne 4.

Attention :

- le : (deux-points) à la fin de la ligne 2 est obligatoire pour respecter la syntaxe Python.
- L'indentation (décalage vers le droite d'une tabulation) marque les lignes qui composent le bloc du if. Ici ce bloc du if est composé de la seule ligne 3.
- La ligne 4 n'est plus indentée, ce qui signale qu'elle ne fait plus partie du bloc du if.

Exécutez la *cellules Code* suivante :

```
x = 9
if x > 10:
    print("Vous avez la moyenne")
print("Fin")
```

Ici pour la valeur de 9, la condition booléenne  $x > 10$  de la ligne 2 est fausse donc le bloc du if n'est pas exécuté. Autrement dit on passe directement à la ligne 4 sans exécuter la ligne 3.

##### b) Instructions if...else

Exécutez le script suivant :

---

```
x = 11
if x > 10:
    print("Vous avez la moyenne")
else:
    print("Vous n'avez pas la moyenne")
print("Fin")
```

Lorsque  $x$  contient l'entier 11, la condition booléenne  $x > 10$  est vraie donc la ligne 3 est exécutée. Une fois cette ligne 3 est exécutée, le script se poursuit, on "sort du if" et le else n'est pas exécuté, le script se poursuit à la ligne 6.

Dans la ligne 4, le else est suivi d'un : (deux-points) sans condition booléenne.

Exécutez le script suivant :

```
x = 9
if x > 10:
    print("Vous avez la moyenne")
else:
    print("Vous n'avez pas la moyenne")
print("Fin")
```

Lorsque  $x$  contient l'entier 9, la condition booléenne  $x > 10$  est fausse donc la ligne 3 n'est pas exécutée. On passe alors au else, la ligne 5 est exécutée, et le script se termine par la ligne 6.

## 2- instructions if...elif...else

Exécuter les *cellules Code* suivantes :

```
x = -5
if x < 0:
    print("Le nombre est strictement négatif")
elif x == 0:
    print("Le nombre est nul")
else:
    print("Le nombre est strictement positif")
print("Fin")

x = 0
if x < 0:
    print("Le nombre est strictement négatif")
elif x == 0:
    print("Le nombre est nul")
else:
```

---

```
    print("Le nombre est strictement positif")
print("Fin")

x = 110
if x < 0:
    print("Le nombre est strictement négatif")
elif x == 0:
    print("Le nombre est nul")
else:
    print("Le nombre est strictement positif")
print("Fin")
```

A noter que :

- elif est obligatoirement suivi d'une condition booléenne et se termine par un : (deux-points).
- On n'est forcément dans un des trois cas :  $x < 10$  (cas du if) ou  $x == 0$  (cas du elif) ou  $x > 10$  (cas du else). Lorsqu'un de ces trois booléens est vrai, on passe directement à la ligne 8 puisque les autres situations ne peuvent pas arriver.

Le script suivant semble donner le même résultat, pourtant il est moins performant que le précédent car chacune des lignes 2, 4 et 6 est toujours exécutée.

```
x = 110
if x < 0:
    print("Le nombre est strictement négatif")
if x == 0:
    print("Le nombre est nul")
if x > 0:
    print("Le nombre est strictement positif")
print("Fin")
```

Il est possible d'ajouter autant de elif qu'on souhaite comme le montre l'exemple suivant :

```
def tarif(age, prix):
    if age <= 5:
        tar = 0
    elif age < 10:
        tar = prix * 0.5
    elif age < 15:
        tar = prix * 0.75
    elif age < 65:
        tar = prix
    else:
        tar = prix * 0.9
    return tar
```

---

```
tarif(25, 20)
```

Lorsque *age* prend la valeur de 25, les booléens des lignes 2, 4 et 6 sont testés et sont à False. Donc les lignes qui suivent ne sont pas exécutées. Puis, le booléen de la ligne 8 est testé. Comme *age* < 65 est True, on exécute la ligne 9, et on passe directement à la ligne 12 du return tarif.

### 3- Exercices

#### Exercice 1 :

Soit *x* un réel. On rappelle la définition de la **valeur absolue** de *x*.

$$|x| = -x \text{ si } x \leq 0.$$

$$|x| = x \text{ si } x > 0.$$

Compléter le script suivant pour qu'il renvoie la valeur absolue de -5 et de 7.

```
def absolue(x):  
    if ...:  
        return -x  
    else:  
        return ...
```

```
absolue(-5)
```

```
absolue(7)
```

#### Exercice 2 :

Compléter le script ci-après pour que :

- Si la température *T* est inférieure ou égale à 0°C, il affiche « C'est de la glace ».
- Si elle est strictement supérieure à 0°C et strictement inférieure à 100°C, il affiche « C'est de l'eau liquide ».
- Si elle est supérieure ou égale à 100°C, il affiche « C'est de la vapeur ».

```
def etat(T):  
    if ...:  
        ...  
    elif ...:  
        ...  
    else:
```

---

...

```
etat(-10)
etat(50)
etat(200)
```

### Exercice 3 :



On souhaite modéliser le jeu de Pierre-feuille-ciseaux contre un ordinateur.

On suppose que l'ordinateur choisit un entier *ord* aléatoirement :

- Si *ord*=1 alors l'ordinateur joue feuille
- Si *ord*=2 alors l'ordinateur joue ciseaux
- Si *ord*=3 alors l'ordinateur joue pierre

L'entier *choix* représentera le choix de l'utilisateur. De la même manière :

- Si *choix*=1 alors l'utilisateur joue feuille
- Si *choix*=2 alors l'utilisateur joue ciseaux
- Si *choix*=3 alors l'utilisateur joue pierre

De façon générale, la pierre bat les ciseaux (en les émoussant), les ciseaux battent la feuille (en la coupant), la feuille bat la pierre (en l'enveloppant).

Compléter le programme suivant pour qu'il modélise le jeu contre l'ordinateur.

```
from random import *
```

```
def jeu(choix):
    ord = randint(1, 3) # randint(1,3) renvoie un entier aléatoire parmi 1,
    ↪ 2 ou 3.
```

---

...

jeu(1)