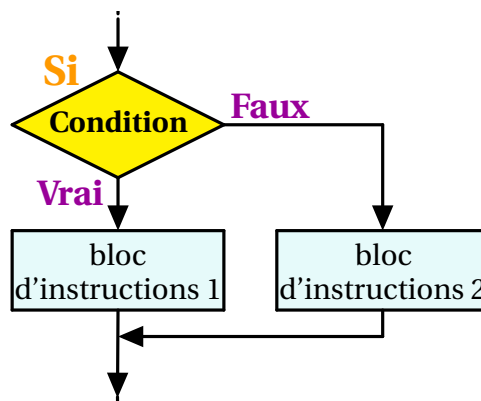


# T.P. chapitre 3 : l'instruction conditionnelle if



## 1 Expressions booléennes



### Définition :

- Une **expression booléenne** est une expression qui peut être **Vrai** ou **Faux**.
- Cela permet au programme, selon qu'une condition est vraie ou fausse, d'effectuer une action ou une autre. On parle d'**exécution conditionnelle**.

### Exercice 1 : expressions booléennes

Pour chacune des instructions ou séquences d'instructions du tableau suivant, prévoir le résultat qu'affichera la console, puis vérifier en saisissant ces instructions dans la console.

>>> 3 + 1 == 4	>>> 3 + 1 == 5	>>> 3 + 1 = 4
>>> 5 <= 3	>>> 3.1 < 17.3	>>> -8 > -7
>>> "bonjour" == "bonjour"	>>> "bonjour" == "Bonjour"	>>> "bonjour" != "Bonjour"
>>> x = 5 >>> y = 3 >>> x == y	>>> x = 3 >>> x = x + 1 >>> x = 8 * x >>> x == 36	

### Exercice 2 :

Prévoir le résultat des instructions suivantes et vérifier dans la console :

```
1 >>> age = 15
2 >>> age <= 18
3 ?
4 >>> age_papa = 45
5 >>> age_papa <= 18
6 ?
```

## 2 Instruction conditionnelle if

### Exercice 3 : instruction conditionnelle

Nous aurons souvent besoin de savoir programmer des actions différentes selon qu'une condition est vraie ou fausse. C'est ce que permet de faire l'instruction conditionnelle `if`.

a. Que renvoie la séquence d'instructions suivante? Vérifier dans la console.

```
1 >>> age = 16
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
```

b. Que renvoie la séquence d'instructions suivante? Vérifier dans la console.

```
1 >>> age = 25
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
```

c. Que renvoie la séquence d'instructions suivante? Vérifier dans la console.

```
1 >>> age = 25
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
4 ... else :
5 ...     print("Vous êtes majeur")
```

d. Que renvoie la séquence d'instructions suivante? Vérifier dans la console.

```
1 >>> age = 15
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
4 ... else :
5 ...     print("Vous êtes majeur")
```

Nous pouvons ainsi schématiser le fonctionnement de l'instruction conditionnelle `if` :

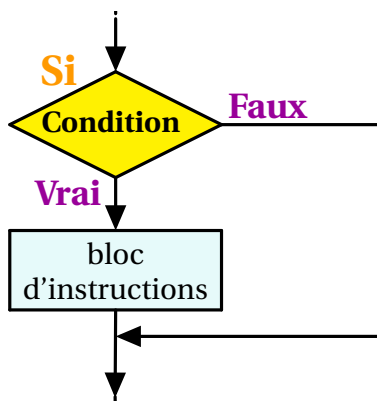


Figure 1 : `if`

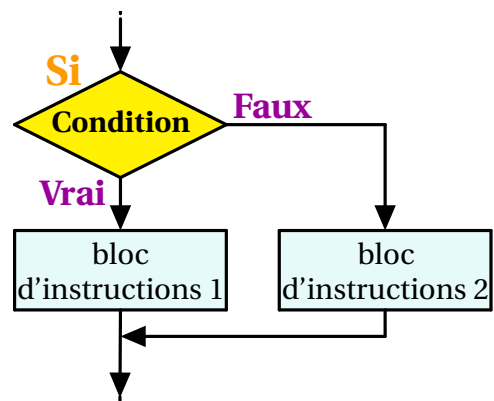


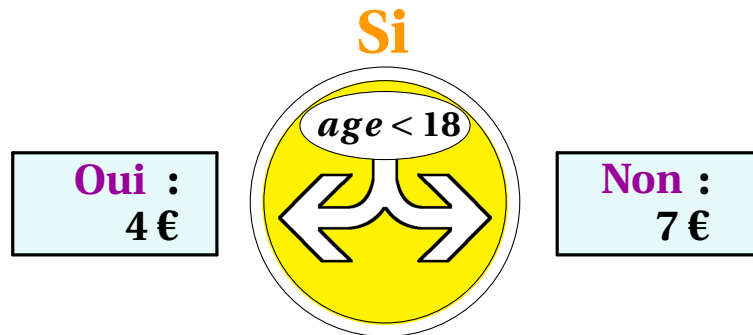
Figure 2 : `if ... else ...`

### 3 Un logiciel de caisse automatique au cinéma

#### Exercice 4 :

Nous programmerons ici une version très simplifiée d'un tel logiciel. Il devra pouvoir :

- recevoir en entrée l'âge du client,
- choisir de renvoyer en retour le juste tarif à appliquer :



a. Écrire une fonction `tarif(age)` :

- prenant en argument l'âge du client,
- et renvoyant, à l'aide de l'instruction `return`, le tarif appliqué :  
4 € si moins de 18 ans, 7 € sinon.

```
1 def tarif(age) :  
2     if ..... :  
3         prix =  
4     else :  
5         .....  
6     return .....
```

- b. • Cliquer sur le bouton lecture pour placer le programme en mémoire de la console.  
• Se rendre dans la console et tester votre fonction avec différentes valeurs de l'âge :

```
1 >>> tarif(15)  
2 ?  
3 >>> tarif(47)  
4 ?  
5 >>> tarif(0)  
6 ?  
7 >>> tarif(103)  
8 ?
```

### Exercice 5 : impressions de photos en ligne

Un site internet propose deux tarifs d'impressions de photos :

- moins de 50 photos : 0,5 € par photo imprimée,
- à partir de 50 photos : 0,35 € par photo imprimée.

a. Écrire une fonction `tarif(photo)` :

- prenant en argument le nombre de photo,
- et renvoyant, à l'aide de l'instruction `return`, le tarif appliqué.

b. Utiliser votre fonction pour déterminer :

```
1 >>> tarif(27)
2 ?
3 >>> tarif(253)
4 ?
5 >>> tarif(50)
6 ?
```

### Exercice 6 : instruction `randint` de la bibliothèque `random`

Nous avons parfois besoin de charger des fonctionnalités supplémentaires de Python. Pour cela, nous pouvons importer des bibliothèque supplémentaires. Nous pouvons aussi choisir de n'importer que certaines fonctions de ces bibliothèques.

Ainsi, pour importer la fonction `randint` depuis la bibliothèque `random`, nous saisisons en première ligne des programmes ou dans la console :

```
1 >>> from random import randint
```

a. Saisir dans la console l'instruction :

```
1 >>> from random import randint
```

b. Tester 10 fois de suite l'instruction :

```
1 >>> randint(1, 3)
```

c. Quel est le rôle de cette instruction?

d. Tirer des nombres aléatoires entre 0 et 10.

e. Tirer des nombres aléatoires entre 1 et 100.

### Exercice 7 : chifoumi

Pour programmer un jeu de chifoumi, nous souhaiterions programmer une fonction **chifoumi()** :

- qui ne prend pas d'argument en entrée,
- qui commence par tirer un nombre au hasard entre 1 et 3,
- qui renvoie :
  - 'pierre' si le nombre tiré au sort est 1,
  - 'feuille' si le nombre tiré au sort est 2,
  - 'ciseaux' si le nombre tiré au sort est 3.

#### Exemple :

```
1 >>> chifoumi()
2 'feuille'
3 >>> chifoumi()
4 'feuille'
5 >>> chifoumi()
6 'pierre'
7 >>> chifoumi()
8 'feuille'
9 >>> chifoumi()
10 'ciseaux'
```

Notre programme aura la structure suivante :

```
1 from random import randint
2
3 def chifoumi() :
4     nombre = .....
5     if ..... :
6
7     elif ..... :
8
9     else :
10
11     return .....
```

- Écrire et compléter ce programme.
- Le tester en l'appelant plusieurs fois depuis la console.