

Les comparaisons

il est possible d'effectuer une série de comparaisons entre le contenu de deux variables.

Opérateurs de comparaison	
<code>x == y</code>	# x est égal à y
<code>x != y</code>	# x est différent de y
<code>x > y</code>	# x est plus grand que y
<code>x < y</code>	# x est plus petit que y
<code>x >= y</code>	# x est plus grand que, ou égal à y
<code>x <= y</code>	# x est plus petit que, ou égal à y

Python va renvoyer `True` si la comparaison est correct, et `False` si elle est fausse.

Attention à ne pas confondre l'opérateur d'affectation `=` et l'opérateur de comparaison `==`.

Il est également possible de faire des comparaisons sur des chaînes de caractères. Cependant, seuls les tests `!=` et `==` ont un sens. Les autres opérateurs comparent l'ordre alphabétique des caractères.

Les tests

Dans cette partie, on va apprendre à utiliser des structures conditionnelles qui vont nous permettre d'exécuter ou non une ligne de code selon certaine condition. Il va permettre à l'ordinateur de prendre des décisions. Par exemple soustraire 1 à un nombre si celui ci est supérieur à 1.

La structure pour écrire une condition est de la forme:

```
if _condition_ :  
    je_fais_cette_action  
else:  
    je_fais_une_autre_action
```

Noter ici la position des lignes (leurs indentations). Toutes les instructions qui sont relatives à une condition doivent se situer une tabulation en arrière de la ligne conditionnelle (appelé bloc d'instruction).

Par exemple :

```
nb=10  
if nb>1:      # Si nb est supérieur à 1  
    nb=nb-1   # On retire 1 à nb  
else :  
    nb=nb/2
```

Exercice

Créer une variable avec votre age. Créer une condition qui teste si l'age saisi correspond à une personne majeure ou non, et afficher `Vous êtes majeur.` ou `Vous êtes mineur.` en fonction des cas.

In []:

Comme vous pouvez le voir, jusqu'à présent on ne peut gérer uniquement des cas du type `si...alors...sinon.. alors` . Or il peut être intéressant d'utiliser des structures conditionnelles plus complexe du type `si...alors... sinon si... alors` .
Par exemple, on peut afficher l'état de l'eau en fonction de sa température (temperature solide, temperature >100 -> gazeux, sinon liquide):

```
if temperature<0:
    print('solide')
elif (temperature>100):
    print('gazeux')
else:
    print('liquide')
```

Exercice

Créer une variable avec votre age. Créer unprogramme qui teste si l'age saisie correspond à une personne enfant (<12ans), adulte(>18ans) ou adolescent(entre 12 et 18 ans).

In []:

Les Tests multiples

Les tests multiples permettent de tester plusieurs conditions en même temps en utilisant des opérateurs **ET** (and), **OU**(or) (ce sont des opérateurs booléens).
Voici les tableaux de rappel de fonctionnement de l'opérateur **ET** :

Condition 1	Opérateur	Condition 2	Résultat
Vrai	ET	Vrai	Vrai
Vrai	ET	Faux	Faux
Faux	ET	Vrai	Faux
Faux	ET	Faux	Faux

Et de l'opérateur **OU**:

Condition 1	Opérateur	Condition 2	Résultat
Vrai	OU	Vrai	Vrai
Vrai	OU	Faux	Vrai
Faux	OU	Vrai	Vrai
Faux	OU	Faux	Faux

Par exemple on peut se demander si les deux variables a et b sont positives.

```
a=2
b=4
if a>0 and b>0:
    print('Les deux variables sont positives')
```

Remarquez que l'on peut obtenir le même résultat en utilisant deux instructions if imbriquées:

```
if a>0:
    if b>0:
        print('Les deux variables sont positives')
```

Exercice

Compléter l'exemple des tests multiples, et afficher les autres cas possible ('Les deux variables sont négatives', 'Une variable est positive et l'autre est négative').

In []: