

* Tapez l'exemple suivant dans l'éditeur Edupython. Attention : l'indentation d'un programme Python (c'est-à-dire le décalage des lignes par rapport au bord gauche de l'écran) est fondamentale. Par exemple la 3^{ème} ligne du programme doit être légèrement décalée vers la droite. Ce décalage peut se faire soit en utilisant la touche « Tabulations » (la touche juste au-dessus de la touche « Verrouillage majuscule »), soit en tapant 4 fois sur la barre d'espace.



```
nombre = int(input("Veuillez entrer un nombre entier :"))
if nombre > 10:
    print("Le nombre est supérieur à 10")
print("Ce message est affiché tout le temps")
```

* Lancez le programme en cliquant sur la flèche verte en haut de la fenêtre Edupython. Le programme demande un nombre entier. Entrez 5. Que se passe-t-il ? Relancez le programme puis entrez 24. Que se passe-t-il ?

Le programme réalise un test : il regarde si la variable « nombre » est plus grande que 10, et agit en conséquence. Ce programme serait équivalent au « programme » Scratch suivant :

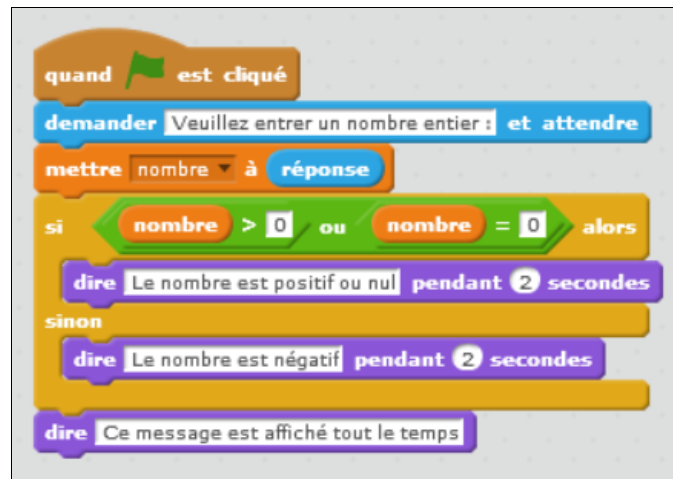


* Modifier le programme pour qu'il affiche un message lorsque le nombre est inférieur à 5.

* Tapez ensuite le programme suivant :

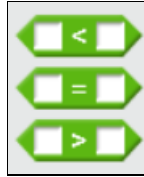
```
nombre = int(input("Veuillez entrer un nombre entier :"))
if nombre >= 0:
    print("Le nombre est positif ou nul")
else:
    print("Le nombre est négatif")
print("Ce message est affiché tout le temps")
```

Le programme réalise un test : il regarde si la variable « nombre » est plus grande ou égale à 0, et agit en conséquence. Ce programme serait équivalent au « programme » Scratch suivant :



Remarques :

- Alors qu'en Scratch il n'y a que trois façons de comparer des nombres (<, > et =), il y en a 6 en Python (<, >, =, ≤, ≥ et ≠).



- en langage Python, dans un test, ≥ s'écrit >= , ≤ s'écrit <= , ≠ s'écrit != , et = s'écrit ==. Voir par exemple le code suivant : (à tester avec les valeurs 5, 12 et 18 par exemple).

```
nombre = int(input("Veuillez entrer un nombre entier :"))
if nombre == 12:
    print("Le nombre est égal à 12")
if nombre != 5:
    print("Le nombre est différent de 5")
print("Ce message est affiché tout le temps")
```

De la même façon que Scratch, on peut combiner plusieurs conditions à l'aide des opérateurs **OU** (or) et **ET** (et).



* Tapez les deux programmes suivants et testez leur comportement en tapant plusieurs valeurs en entrée :

```
# test de l'opérateur "or"
nombre = int(input("Veuillez entrer un nombre entier :"))
if nombre < 0 OR nombre > 10:
    print("Le nombre est plus petit que 0 ou plus grand que 10")
else:
    print("Le nombre est compris entre 0 et 10 inclus")
print("Ce message est affiché tout le temps")
```

```
# test de l'opérateur "and"
nombre = int(input("Veuillez entrer un nombre entier :"))
if nombre >= 0 and nombre <= 10:
    print("Le nombre est compris entre 0 et 10 inclus")
else:
    print("Le nombre est plus petit que 0 ou plus grand que 10")
print("Ce message est affiché tout le temps")
```

Un autre opérateur très utile est **NON** (not) :



```
# test de l'opérateur "not"
nombre = int(input("Veuillez entrer un nombre entier :"))
if not(nombre == 6):
    print("Le nombre n'est pas égal à 6")
print("Ce message est affiché tout le temps")
```

Remarque : dans les trois programmes précédents, les lignes commençant par « # » sont des commentaires : ce sont des lignes qui ne seront pas interprétées par l'ordinateur, et qui sert au programmeur pour commenter les différentes parties de son code.

Lorsqu'on veut réaliser plusieurs tests successifs, on peut utiliser la structure if ... elif ... else.

```
nombre = int(input("Veuillez entrer un nombre entier entre 1 et 5 :"))
if nombre == 1:
    print("Vous avez tapé 1")
elif nombre == 2:
    print("Vous avez tapé 2")
elif nombre == 3:
    print("Vous avez tapé 3")
elif nombre == 4:
    print("Vous avez tapé 4")
elif nombre == 5:
    print("Vous avez tapé 5")
else:
    print("Vous ne savez pas lire ou quoi ?")
print("Le programme est terminé")
```

Mettons maintenant en pratique ce que nous venons de voir. Pour cela vous allez devoir réaliser les programmes suivants. Pensez à les enregistrer, soit sur votre lecteur réseau, soit sur votre clef USB.

1./ Réalisez un programme qui demande l'année de naissance de l'utilisateur, et qui affiche un message d'erreur si cette année de naissance est antérieure à 1900, ou postérieure à l'année courante. Si la valeur entrée se trouve dans la fourchette autorisée, le programme renvoie l'âge de l'utilisateur (on fera un calcul simplifié qui ne tient pas compte de la date de naissance, et on calculera juste la différence entre l'année en cours et l'année de naissance).

2./ Réalisez un programme qui demande deux **nombres flottants a et b**. Le programme demandera ensuite à l'utilisateur de taper « 1 » pour une addition, ou « 2 » pour une soustraction. Si l'utilisateur entre « 1 », le programme devra afficher le résultat de l'addition **a+b**. Si l'utilisateur entre « 2 », le programme devra afficher le résultat de la soustraction **a-b**. Si l'utilisateur entre une autre valeur, le programme devra le signaler par un message d'erreur.

3./ Réalisez une mini calculatrice en Python. Pour ce faire, modifiez le programme précédent pour qu'il puisse réaliser les quatre opérations mathématiques de base (+, -, * et /).

Exemple de fonctionnement :

```
Entrez un nombre a : 4.0
Entrez un nombre b : 5.0
Tapez 1 pour +, 2 pour -, 3 pour * ou 4 pour / : 4
Le résultat de l'opération est : 0.8
```

4./ Modifiez le programme précédent afin qu'il affiche l'opération effectuée en clair.

Avec l'exemple précédent :

```
Entrez un nombre a : 4.0
Entrez un nombre b : 5.0
Tapez 1 pour +, 2 pour -, 3 pour * ou 4 pour / : 4
On a : 4.0 / 5.0 = 0.8
```

5./ Notre calculatrice a toujours un gros défaut : le programme « plante » quand on essaie de diviser par zéro. Modifiez le programme précédent afin qu'il affiche un message avertissant l'utilisateur sur l'impossibilité de diviser par zéro quand cette situation se produit.