

```
#!/coding: utf-8
from __future__ import print_function, division

#####
# Base UNE : "=" #
#####

# "=" permet de changer la valeur d'une variable
# print permet de voir à l'écran la valeur d'une variable

a = 5
print(a) # en python 2, les parenthèses n'étaient pas nécessaires
a = 6
print(a)
# on peut faire des maths
b = a + 1
c = a + b * 2 # priorité des opérations
d = (a + b) * 2 # parenthèses si nécessaire !
a = a + 1 # a est augmenté de 1
print(a)
print(b)
print(c)
print(d)

# pour afficher du texte, il faut le mettre entre guillemets
print("fin")

#####
# Base DEUX : "if" #
#####

a = 7

if a < 10:
    print("Coucou")
    print("Hello")

# ici le programme va afficher "Coucou", puis "Hello" seulement si a est plus petit que 10
# sinon, il saute le bloc et donc, ne fait rien

# un autre exemple, on veut donner 50 points de vie à un personnage, sans dépasser 100...
vie = 75
vie = vie + 50
if vie > 100:
    vie = 100

# pour afficher plusieurs choses on écrit une virgule
print("Votre vie est", vie) # en python 2, les parenthèses n'étaient pas obligatoires

# on peut faire un else, le code ira dans le else si la condition est fausse
if vie == 100:
    print("You are full !")
else:
    print("You can drink potions.")

a = 8
b = 2
# les opérateurs de comparaison sont "<", ">", "<=", ">=", "==", "!=" (différent)
# attention, pour comparer deux valeurs il faut utiliser "=="
# par exemple, ce programme ci

if a == 5:
    print("Yo")
else:
    print("Da")

# affiche "Yo" si a est égal à 5 et "Da" sinon
# essaie de réécrire ce code en utilisant l'opérateur "!=" (différent)

# Dans un if, on peut mettre n'importe quel code
```

```

# Comme un "=", un print, ou... un autre if !

if a == 5:
    a = 2
    print("Yo")
    if b == 5:
        print("Hello")
    else:
        print("Tada")
else:
    print("Hum")

print(a)

# Essaie ce programme avec a=5 b=5, a=5 b=2, a=2 b=2 et voit ce qu'il se passe

# Fais maintenant l'exercice 0 (trier_deux_nombres)
# regarde la correction sur pythontutor
(http://robertvandeneuynde.be/parascolaire/pythontutor.html)

# Ensuite tu peux faire les exercices 1 à 4 sans lire la suite
# Cependant la suite PEUT être utile pour les exercices suivants

# On peut écrire des conditions combinées avec "and" et "or", par exemple :

if a == 1 and b == 2:
    print("Yo")
else:
    print("Da")

# est un programme qui affiche "Yo" si a est égal à 1 et b est égal à 2, sinon Da
# les Deux conditions doivent être vraies
# Celui ci

if a == 1 or b == 2:
    print("Yo")
else:
    print("Da")

# affiche Yo si a est égal à 1 ou b est égal à 2
# ou moins une des deux conditions doit être vraie

# attention, si tu mélanges des and et des or,
# utilise des parenthèses pour bien préciser l'ordre des opérations

if a == 2 or b == 2 and c == 2: # qui du "or" ou du "and" a la priorité ?
    print("Yo")

# équivalent au précédent : or est "comme" un +, and est "comme" un *
if a == 2 or (b == 2 and c == 2):
    print("Yo")

if (a == 2 or b == 2) and c == 2:
    print("Yo")

#####
# Pour en savoir plus #
#####

## =

# des raccourcis pour incrémenter/décrémenter !

a = a + 1
a += 1 # raccourci pour a = a + 1
a -= 1 # raccourci pour a = a - 1
a *= 2 # a = a * 2
# etc.

# floats
a = 2.5

# division entière // et modulo %
# si on divise 14 par 4 on a 3 avec un reste de 2 (comme en primaire !)

```

```
a = 14 / 4 # 3 (attention python 3 ! 14 / 4 = 3.5)
```

```
a = 14 / 4.0 # 3.5
```

```
a = 14 / float(4) # 3.5
```

```
a = 14 // 4 # 3
```

```
a = 14 // 4.0 # 3.0
```

```
div = 14 // 4 # 3, la partie entière
```

```
mod = 14 % 4 # 2, le reste
```

```
# exposant
```

```
a = 10 ** 5 # 100000
```

```
# modulo négatifs (python is cool !)
```

```
n = -1 % 5 # 4
```

```
n = 0 % 5 # 0
```

```
n = 1 % 5 # 1
```

```
n = 2 % 5 # 2
```

```
n = 3 % 5 # 3
```

```
n = 4 % 5 # 4
```

```
n = 5 % 5 # 0
```

```
n = 6 % 5 # 1
```

```
## plusieurs lignes
```

```
# si on ouvre une parenthèse, on peut passer à la ligne autant que l'ont veut
```

```
x = (5 + 2 * 3  
     + 7 * 2  
     + 1  
     - 2)
```

```
## multiple comparaisons
```

```
if 2 <= a <= 5: # 2 <= a and a <= 5  
    print("a est entre 2 et 5")
```

```
if a == b == 0: # a == b and b == 0  
    print("a et b valent 0")
```

```
## not "inverser" une condition
```

```
if a == 5: # si a == 5...  
    pass # ne rien faire  
else:  
    print("a n'est pas égal à 5")
```

```
# équivalent à
```

```
if not(a == 5):  
    print("a n'est pas égal à 5")  
else:  
    pass
```

```
# équivalent à
```

```
if not(a == 5):  
    print("a n'est pas égal à 5")
```

```
# au choix du programmeur, le not peut être simplifié
```

```
# not(a == b)   ↔ a != b
```

```
# not(a < b)   ↔ a >= b (ATTENTION : plus grand ou ÉGAL)
```

```
# not(X and Y) ↔ (not X) or (not Y) (ATTENTION : OR)
```

```
# not(X or Y)  ↔ (not X) and (not Y)
```

```
if not(a == 5 and b < 7):  
    print("not(a == 5 and b < 7)")
```

```
if a != 5 or b >= 7: # équivalent au précédent  
    print("a != 5 or b >= 7")
```

```
# quand on a un if / else quelconque
```

```
# on peut, au choix, inverser la condition en mettant un "not"
```

```
# conseil: quand vous avez un long if et un court "else", c'est souvent plus clair d'inverser
```

```

if a != 5 and a > 0:
    print("yoyo")
    print("tada")
    print("truc")

    if a == 4:
        print("da")
    else:
        print("yo")

else: # ce "else" est court et oublié :(
    print("coucou")

# on peut (au choix) l'inverser en "not(X)" sa condition :
if not(a != 5 and a > 0):
    print("coucou") # aaah, l'ancien else est mis en avant :)
else:
    print("yoyo")
    print("tada")
    print("truc")
    if a == 4:
        print("da")
    else:
        print("yo")

## elif : parfois, on a un "else" qui ne contient qu'une seule instruction, qui est un "if"

if a < 5:
    print("petit")
else:
    if a < 10:
        print("moyen")
    else:
        if a < 15:
            print("grand")
        else:
            print("graaaand")

# raccourci : elif (else if)

if a < 5: # si a < 5
    print("petit")
elif a < 10: # sinon... si a < 10
    print("moyen")
elif a < 15:
    print("grand")
else:
    print("graaaand")

## bool: les conditions peuvent être mises dans des variables
# un (booléen) vaut Vrai ou Faux (True / False)

condition = (a < 5)
if condition == True:
    print("Plus petit !")
else:
    print("Plus grand ou égal")

# le "if" attend un bool, on peut donc enlever "== True"

condition = a < 5 # parenthèse non nécessaires
if condition: # == True enlevé
    print("Plus petit")
else:
    print("Plus grand ou égal")

# on peut donc faire des "opérations" sur les bool

x = True
y = False
z = x or y # z = True or False = True
n = not x # n = not True = False
g = a < 5 and z

```

```
## if fonctionnel : parfois, on a un "if/else" qui ne fait qu'assigner une variable (et rien d'autre !)
```

```
if a == 5:  
    b = 8  
else:  
    b = 3
```

```
# raccourci: le "if/else" "en une ligne"  
b = (8 if a == 5 else 3) # même code qu'au dessus
```

```
b = 8 if a == 5 else 3 # parenthèses non nécessaires
```

```
b = (8 if a == 5 else  
    3) # deux lignes c'est plus clair !
```

```
c = (8 if a == 5 else  
    4 if a == 2 else  
    1 if a < 0 else  
    0) # longue chaîne !
```