



# OPÉRATIONS SUR LES VARIABLES EN PYTHON

[fabricemvah@gmail.com](mailto:fabricemvah@gmail.com)



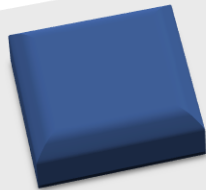
# SOMMAIRE



**Variables**



**Opérateurs et expressions**



**Structures de contrôle**

# DÉFINITION D'UNE VARIABLE



***Une variable*** est un espace mémoire permettant de contenir une donnée ou une valeur.

**mavariabale = valeur**

***Exemple:***

**phone\_number = '0065466785545'**

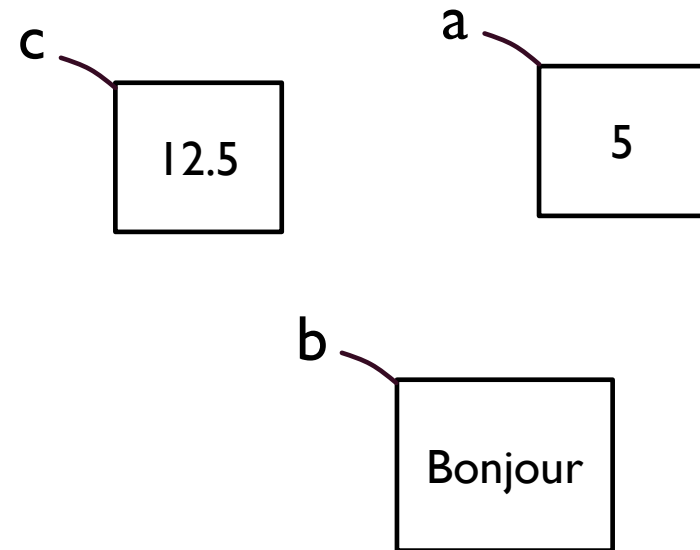
# DÉFINITION D'UNE VARIABLE

a = 5

b = 'Bonjour'

c = 12.5

## Ordinateur



# EXERCICES

## Exercice 1

```
a = 5  
b = 'Bonjour'  
c = 12.5
```

```
a = 10  
b = c  
c = a
```

Donnez les contenus de a, b et c à la fin d'exécution des instructions ci-dessus.

## Exercice 2

```
a = 5  
b = 10  
c = 0
```

```
c = a  
a = b  
b = c
```

- 1- Donnez les contenus de a et b à la fin d'exécution des instructions ci-dessus;
- 2- Que fait ce pseudo-code ?

# TYPES DE VARIABLE

## Types primitifs

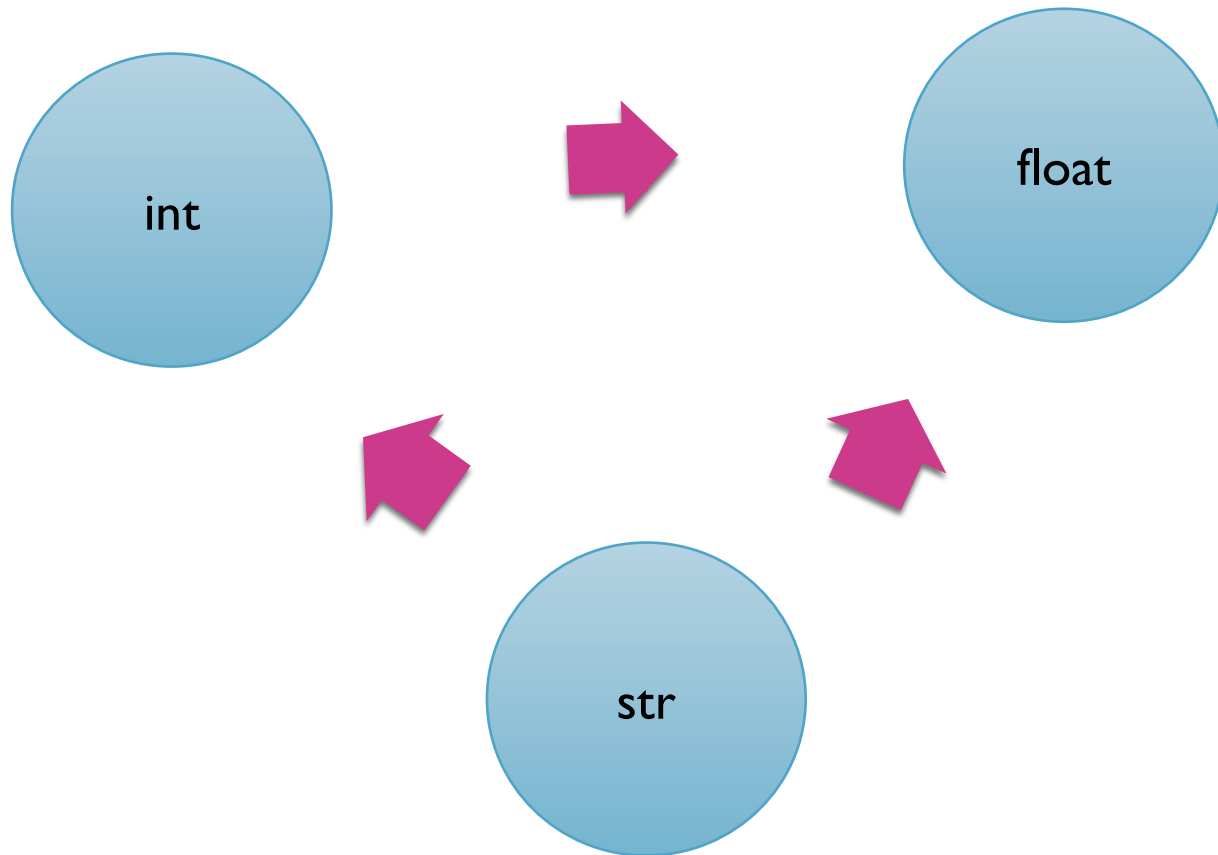
- Entier (int en python)
- Réel (float en python)
- Chaîne de caractère (str en python)
- Booléen (bool en python)

## Utilisation

- **a = 3**     *# a est de type entier*
- **b = 2.5**   *# b est une variable de type réel*
- **c = 'Hello Worlds!'**   *# c est de type chaîne de caractère*
- **d = False**   *# d est de type booléen*

# CONVERSION DE TYPE

La conversion de type consiste à aller d'un type de variable vers un autre type



❑ `int()`

❑ `float()`

❑ `str()`

# EXERCICES

Donnez le résultat qu'affiche chacun des programmes python ci-dessous

programme 1

```
a = 44.5  
b = int(a)  
print(b)
```

programme 2

```
a = '44.5'  
b = float(a)  
print(b)
```

programme 3

```
a = 44  
b = float(a)  
print(b)
```

programme 4

```
a = 44  
b = str(a)  
print(b)
```

programme 5

```
a = '44'  
b = int(a)  
print(b)
```

programme 6

```
a = 10  
b = 5  
c = a + b  
print(c)
```

programme 7

```
a = 10  
b = 5  
c = str(a) + str(b)  
print(c)
```

programme 8

```
a = '10'  
b = '5'  
c = a - b  
print(c)
```



# AUTRES TYPES DE VARIABLE

## Les listes

maliste = []                      maliste1 = ['bonjour', 'me', 'voici']                      maliste2 = [1.3, 2.4, 6.0]

**Accéder à l'élément situé à la position zéro: *maliste1[0]* renvoie *'bonjour'***

**Modifier un élément de la liste : *maliste1[0] = 'Salut'***

# AUTRES TYPES DE VARIABLE

## Les tuples

Un tuple sert à stocker les données dont la structure est importante et ne change pas.

Exemple : `paris(23.456, 4.678)`      `point(3, 4)`      `coord('A', 'B', 'C')`

On peut utiliser la fonction ***tuple()*** pour convertir une liste en tuple : `tuple(['1', '2', '3', '4']) = ('1', '2', '3', '4')`

De même, on peut convertir n'importe quel type itérable en tuple:

`tuple('sam') = ('s', 'a', 'm')`      `tuple(str(50)) = ('5', '0')`

Conversion en liste: `list(('a', 'b', 'c')) = ['a', 'b', 'c']`      `list('abc') = ['a', 'b', 'c']`

# AUTRES TYPES DE VARIABLE

## Dictionnaire

C'est un ensemble de données structurées suivant le modèle ***clé : valeur***

**Exemple:** dico1 = {}    dico2 = {'je' : 'i', 'tu' : 'you', 'me' : 'moi'}

On accède à une valeur à partir de sa clé: dico2['me'] retourne 'moi'

La valeur peut aussi être un entier, un réel, un booléen ou une chaîne de caractère.

# EXERCICES

Donnez le résultat qu'affiche chacun des programmes python ci-dessous

## programme 1

```
liste = ['a','b','c']  
print(liste)  
b = tuple(liste)  
print(b)  
Print('affiche')  
print(liste[0])  
print(liste[1])  
print(liste[2])  
print('fin')
```

## programme 2

```
a = 345.56  
b = int(a)  
liste = list(str(b))  
print(liste)  
Print(tuple(liste))
```

## programme 3

```
liste = ['l3','b','c']  
a = liste[0]  
print(a)  
b = list(a)  
print(b)  
Print(tuple(b))
```

## programme 4

```
dico = {'l' : '234'}  
a = dico['l']  
print(a)  
b = list(a)  
print(b)  
print(tuple(b))
```

# OPÉRATEURS ET EXPRESSIONS

## Opérateurs arithmétiques

Égale ==

Différent !=

Inférieur <

Supérieur >

Inférieur ou égale <=

Supérieur ou égale >=

Addition +

Soustraction -

Multiplication \*

Division /

Reste de la division entre deux nombres %

Division entière //

Puissance \*\*

# OPÉRATEURS ET EXPRESSIONS

## Opérateurs sur les chaînes de caractère

Concaténation    +                      Exemple : 'Bonjour' + 'Monsieur' retourne 'Bonjour Monsieur'

Répétition        \*                      4\*'moi' retourne 'moimoimoimoi'

Opérateurs logiques : **not, and, or**

# OPÉRATEURS ET EXPRESSIONS

## Opérateurs d'affectation simple et composé

Opérateur	Exemple	Equivalent à	Description
=	<code>x = 1</code>	<code>x = 1</code>	Affecte 1 à la variable x
+=	<code>x += 1</code>	<code>x = x + 1</code>	Ajoute 1 à la dernière valeur connue de x et affecte la nouvelle valeur (l'ancienne + 1) à x
-=	<code>x -= 1</code>	<code>x = x - 1</code>	Enlève 1 à la dernière valeur connue de x et affecte la nouvelle valeur à x
*=	<code>x *= 2</code>	<code>x = x * 2</code>	Mutliplie par 2 la dernière valeur connue de x et affecte la nouvelle valeur à x
/=	<code>x /= 2</code>	<code>x = x / 2</code>	Divise par 2 la dernière valeur connue de x et affecte la nouvelle valeur à x
%=	<code>x %= 2</code>	<code>x = x % 2</code>	Calcule le reste de la division entière de x par 2 et affecte ce reste à x
//=	<code>x //= 2</code>	<code>x = x // 2</code>	Calcule le résultat entier de la division de x par 2 et affecte ce résultat à x
**=	<code>x **= 4</code>	<code>x = x ** 4</code>	Elève x à la puissance 4 et affecte la nouvelle valeur dans x

# EXERCICES

Donnez le résultat qu'affiche chacun des programmes python ci-dessous

programme 1

```
a = 1+2*3  
Print(a)
```

programme 2

```
a = (1+2)*3  
Print(a)
```

programme 3

```
a = 4  
a *=3  
Print(a)
```

programme 4

```
a = 5 > 3  
Print(a)
```

programme 5

```
a = (3 > 5) and (2<5)  
print(a)
```

programme 6

```
a = (3 > 5) or (2<5)  
print(a)
```

programme 7

```
a = 10  
b = 5  
c = str(a) + str(b)  
print(c)
```

programme 8

```
a = '10'  
print(4*a)
```



# STRUCTURES DE CONTRÔLE

## La structure : if....else

### If

```
if 2==2:  
    print('vrai')
```

### If...else

```
if 2==2:  
    print('vrai')  
else:  
    print('faux')
```

### If....elif

```
if 2==0:  
    print('vrai')  
elif 2 > 1 :  
    print('faux')  
else:  
    Print('Non déterminé')
```

# STRUCTURES DE CONTRÔLE

## Structure répétitive: boucle for et la boucle while

### syntaxe

```
for item in list:  
    # do something
```

### application

```
maliste = ['hey', 'toi']  
  
for i in maliste:  
    print(maliste[i])
```

### syntaxe

```
while condition:  
    # do something
```

### application

```
maliste = ['hey', 'toi']  
i = 0  
while i < 3:  
    print(maliste[i])  
    i = i + 1
```



**Merci pour votre attention !**