

## Opérateurs de base

---

Opérateur	opération
+	addition
-	soustraction
/	division
*	multiplication
%	modulo (reste de la division)
//	division entière
**	puissance

## Opérateurs avancés

---

La librairie math permet d'importer des fonctions pour réaliser des **opérations mathématiques** plus complexes.

```
import math

racine = math.sqrt(16) # racine carrée
print(racine) # 4

arrondi = math.floor(3.7) # arrondi à l'inférieur
print(arrondi) # 3
```

 Fonctions proposées par la librairie 'math'

## L'opérateur d'assignation

---

On l'a vu depuis le début de ce cours, l'opérateur d'assignation d'une valeur à une variable est l'opérateur `=`.

L'opérateur d'assignation peut être couplé à un opérateur mathématique.

```
x = 3
x += 1 # x = x + 1 --> 4
x *= 3 # x = x * 3 --> 12
x /= 2 # x = x / 2 --> 6
print(x)
```

⚠ Contrairement à beaucoup d'autres langages de programmation, python n'autorise pas l'incrémentation d'une variable `x` en utilisant la synthaxe `x++`

## Les opérateurs de comparaison

---

Utilisés dans les [structures conditionnelles](#).

opérateur	opération
<code>&gt;</code>	supérieur à
<code>&lt;</code>	inférieur à
<code>&gt;=</code>	supérieur ou égal à
<code>&lt;=</code>	inférieur ou égal à
<code>==</code>	égalité
<code>!=</code>	inégalité
<code>is</code>	même adresse en mémoire

⚠ Ne pas confondre l'opérateur d'assignation `=` avec l'opérateur d'égalité `==`

### Opérateur d'égalité

Il existe certains langages comme le Javascript qui utilise aussi le `==` pour vérifier plus de condition que le `==` qui est plus léger.

## L'opérateur d'appartenance `in`

---

L'opérateur `in` va nous permettre de déterminer si oui ou non, un élément est présent dans une collection d'objet.

```
phrase = "Le python c'est trop cool"

print('x' in phrase)      # False
print("trop" in phrase) # True

ma_liste = [1, 2, 3, 4]

print(1 not in ma_liste)      # False
print("test" not in ma_liste) # True
```