

# Premières instructions

## A Faire des calculs « directs »

In [1]: `4 * (3 + 2 / 5)` Permet de calculer  $4 \left( 3 + \frac{2}{5} \right)$

Out[1]: 13.6

## B Variable

### Instruction d'affectation

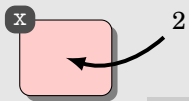
In [2]: `x = 2`

Correspond à l'instruction d'affectation  
**x prend la valeur 2**

In [3]: `x`

On constate bien voir son contenu !

Out[3]: 2



On vient de créer une boîte, nommée **x** qui contient 2.  
Cette instruction se note en langage « naturel » :  $x \leftarrow 2$

In [4]: `x, y = 2, 3` correspond à  $x \leftarrow 2$  et  $y \leftarrow 3$

## C Type d'une variable

### Définitions

Le **type** d'une variable **x** dépend de l'élément qu'elle contient :

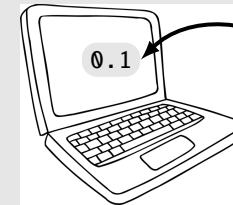
En Python	Type de x	Commentaires
<code>x = 'baba7:)'</code>	<b>string</b>	x contient une chaîne de caractères <i>C'est une succession de caractères mis entre des guillemets ' ' ou " ".</i>
<code>x = 64</code>	<b>int</b>	x contient l'entier 64
<code>x = 3.5</code>	<b>float</b>	x contient le nombre réel 3,5 <i>3,5 est représenté en machine par son nombre flottant 3.5</i>

## D Opérations de base

Maths	+	-	×	/	=	≠	>	<	≤	≥	$a^n$
Python	+	-	*	/	==	!=	>	<	<=	>=	<code>a ** n</code>

### Attention

### Mauvaise représentation des réels en machine



Le nombre décimal 0,1 est représenté en machine par son **nombre flottant** 0.1

In [5]: `3 * 0.1`

Out[5]: 0.30000000000000004

In [6]: `3 * 0.1 == 0.3`

Out[6]: False

### Morale :

0.1 ≠ 0,1  
Nombre flottante      Nombre décimal

**Attention** : des nombres comme 0,1 sont **mal représentés** en machine !

### Arrondir et racine carrée

Instruction Python	Rôle
<code>round(x, n)</code>	Arrondit x avec n chiffres après la virgule
<code>round(x)</code>	Arrondit x à l'unité près
<code>int(x)</code>	Renvoie l'entier associé à x
<code>from math import sqrt</code>	On importe la fonction <b>sqrt</b> du module <b>math</b>
<code>sqrt(x)</code>	Calcule $\sqrt{x}$
<code>x ** 0.5</code>	

### Division euclidienne

La **division euclidienne** des entiers (*non nuls*) **a** et **b** consiste à déterminer les uniques entiers **q** et **r** vérifiant :

$$a = bq + r \quad \text{et} \quad 0 \leq r < b$$

Dividende

a

Diviseur

b

Reste :

`a % b`  
type int

Quotient :

`a // b`  
type int