

# Programath

## Contents

<b>1</b>	<b>= vs ==</b>	<b>2</b>
<b>2</b>	<b>Syntaxe de base</b>	<b>2</b>
<b>3</b>	<b>Ensembles, intervalles</b>	<b>2</b>
3.1	$\square$ vs $\{ \}$ . . . . .	2

## 1 = vs ==

En maths, on note indifféremment le = de la **déclaration** et le = de l'**hypothèse**.

- Ce que j'appelle le = *de la déclaration*, c'est celui qu'on utilise pour poser une variable: "Soit  $a = 45$ ".
- Ce que j'appelle le = *de l'hypothèse*, c'est le = dont on n'est pas sûr, celui que l'on veut prouver ou réfuter: "Supposons  $b = c + d$ ".

En programmation, ce que vous dites est interprété par une machine, qui ne peut pas déduire cette différence cruciale toute seule. On est donc obligé de noter les "deux =" différemment:

Python	Math
<code>a = 45</code>	Soit $a = 45$
<code>a == 45</code>	$a = 45$
<code>P = a == 45</code>	Soit $P$ la propriété " $a = 45$ "

Comme en programmation on pose (ie on déclare) plus que l'on ne teste, le "=" tout simple sert de "= de la déclaration".

## 2 Syntaxe de base

+, × et tout ça.

<code>(1 + 2) * 3 - 4**6</code>	$(1 + 2) \cdot 3 - 4^6$
<code>a//b + a/b</code>	$\left\lfloor \frac{a}{b} \right\rfloor + \frac{a}{b}$
<code>r = a%b</code>	Posons $r$ tel que $a = \left\lfloor \frac{a}{b} \right\rfloor b + r$
<code>from math import sqrt</code> <code>sqrt(5)</code>	$\sqrt{5}$

## 3 Ensembles, intervalles

<code>range(a, b)</code>	$\llbracket a, b \llbracket$
<code>{ 2*a for a in A }</code>	$\{2a, a \in A\}$
<code>{ a for a in A if a**2 == a/2 }</code>	$\{a \in A, a^2 = \frac{a}{2}\}$
Une version qui mélange les deux	
<code>{ f(a) for a in A if P(a) }</code>	$\{f(a), a \in A, P(a)\}$ (pas vraiment légal mais on l'a utilisé une fois)

### 3.1 [] vs {}