

## Commentaires au sein d'un programme: #

En python, tout ce qui suit le symbole # n'est pas exécuté : ce sont des **commentaires**.

## Afficher : `print`

La fonction **`print`** permet d'afficher des informations dans la console python.

Afficher dans la console	Exemples	Remarques
...du texte	<code>print ("Hello world")</code>	<i>avec des guillemets</i>
...la valeur d'une variable	<code>print (a)</code>	<i>sans guillemets</i>
...du texte et des variables	<code>print ("La valeur de a est ", a)</code>	<i>séparation à l'aide de virgules</i>

## Affecter une valeur à une variable : =

Les **types** les plus courants de variables sont :

- **`str`** : chaîne de caractères
- **`float`** : réel
- **`int`** : entier
- **`boolean`** : booléen (*True* et *False*)

Affecter une valeur une variable	Exemples	Remarques
type <b>entier</b> ( <i>int</i> )	<code>a = 4</code>	La variable <i>a</i> est <b>entière</b> et prend la valeur 4
type <b>réel</b> ( <i>float</i> )	<code>x = 4.0</code>	La variable <i>x</i> est de type <b>réel</b> et prend la valeur 4 Attention : Le séparateur décimal est le point : « . »
type <b>texte</b> ( <i>str</i> )	<code>mot = "python"</code>	<b>Chaîne de caractères</b> valant "python"

## Lire (Demander à l'utilisateur de saisir une valeur) : `input`

La fonction **`input`** renvoie une **chaîne de caractères** saisie par l'utilisateur.

S'il s'agit d'une valeur numérique, il faut la convertir en entier (fonction **`int`**) ou en décimal (fonction **`float`**).

Saisie dans la console	Exemples
...de type entier	<code>n = int(input("Donner une valeur entière :"))</code>
... de type réel	<code>x = float(input("Donner une valeur réelle :"))</code>
...de type texte	<code>texte = input("Saisir du texte :")</code>

## Instructions conditionnelles : "Si", "Sinon si", "Sinon" ... : `if / elif / else`

<b><code>if condition :</code></b> → <i>instruction 1</i> → <i>instruction 2</i>	<b><code>if condition :</code></b> → <i>instruction 1</i> → <i>instruction 2</i> <b><code>else :</code></b> → <i>instruction 3</i> → <i>instruction 4</i>	<b><code>if condition :</code></b> → <i>instruction 1</i> <b><code>elif condition :</code></b> → <i>instruction 2</i> <b><code>else :</code></b> → <i>instruction 3</i>
--	--	--



**Attention à l'indentation** → (elle détermine les blocks d'instructions à exécuter)

Tests	Opérateurs de comparaison	Exemples			
Égalité : =	==	if a == 1 :			
Ordre : < > ≤ ≥	< > ≤ ≥	if a < 0 :	if a > 0 :	if a ≤ 0 :	if a ≥ 0 :
Non égalité : ≠	!=	if a != 1 :			

Exemples		
<pre>if a == 1 : → print("a vaut 1 !")</pre>	<pre>if a == 1 : → print("a vaut 1 !") else : → print("a ne vaut pas 1 !")</pre>	<pre>if a &lt; 0 : → print("a est négatif.") elif a &gt; 0 : → print("a est positif.") else : → print("a est nul.")</pre>

## Instructions itératives : Boucles "Pour" et "Tant Que" : for et while

Range : liste de nombres.

range	Listes d'entiers	Exemples
range(n)	Liste des entiers naturels strictement inférieurs à n	range(5) [0,1,2,3,4]
range(a,n)	Liste des entiers supérieurs ou égaux à a et strictement inférieurs à n	range(2,10) [2,3,4,5,6,7,8,9]
range(a,n,pas)	Liste identique à la précédente mais avec un écart défini entre deux entiers consécutifs	range(1,10,2) [1,3,5,7,9]

Structure de la boucle « Pour »	Exemple 1	Exemple 2
<pre>for variable in listeDeValeurs : → instruction 1 → instruction 2</pre>	<pre>for i in range(4) : → print (i)</pre>	<pre>voyelles = ['a', 'e', 'i', 'o', 'u'] for lettre in voyelles : → print(lettre)</pre>

Structure de la boucle « Tant que »	Exemple
<pre>while condition : → instruction 1 → instruction 2</pre>	<pre>i = 0 while i &lt; 10 : → print (i) → i = i + 1</pre>

## Les Fonctions : def

Structure d'une fonction	Exemples		
<pre>def fonction(arguments) : → instruction 1 → instruction 2 → return resultat</pre>	<pre>def trinome(x) : → y = x*x - 2*x + 3 → return y</pre>	<pre>def suite (n) : → u = 110 → for i in range(n) : → → u = 1.1*u - 5 → return u</pre>	<pre>def max (a,b) : → if a &lt; b : → → return b → else : → → return a</pre>
	print ( trinome(1) )	print ( u(10) )	print ( max(7,5) )