




<b>Créer</b> un programme	- On va à la ligne après chaque instruction. - On peut séparer plusieurs instructions sur la même ligne en les séparant par « ; »	
<b>Saisir</b> une variable	- <b>A=input("A=")</b> si A est une chaîne de caractère ( c'est le type par défaut) <b>type str</b> : Chaîne de caractères	
Il existe aussi d'autres types numériques :  <b>type long</b> : Entier compris entre - inf et 2 147 483 647 ou entre 2 147 483 648 et + inf  <b>type complex</b> : Nombre complexe <a href="https://fr.wikiversity.org/wiki/Python/Les_types_de_base">Pour connaître tous les types ...</a> <a href="https://fr.wikiversity.org/wiki/Python/Les_types_de_base">https://fr.wikiversity.org/wiki/Python/Les_types_de_base</a>	- <b>A=float(input("A="))</b> si A est un flottant <b>type float</b> : Valeur spécifiée avec un point dans le programme (exemple : a=2.0) permettant une approximation de nombre réel  - <b>A=int(input("A="))</b> si A est un entier <b>type int</b> : Entier compris entre -2 147 483 648 et 2 147 483 647 (codage sur 32 bits soit 4 octets)	
<b>Afficher</b>	- <b>print(A)</b> affiche la valeur de la variable A - <b>print("Vive les maths")</b> affiche le texte Vive les maths - On peut aussi mélanger texte et variable : <b>print("la valeur de A est ",A)</b>	
<b>Affecter</b>	<b>B=A</b> affecte la valeur A ou le contenu de la variable A à la variable B	
<b>Écrire un commentaire</b>	Les commentaires s'écrivent après le signe #	
<b>Opérations élémentaires</b>	addition + soustraction - multiplication * puissance **	division / reste de division entière % ( 9%2 donne 1 ) quotient de division entière // ( 9//2 donne 4 )
<b>Tester ...</b>	<b>A==B</b> (égal) <b>A!=B</b> (différent) <b>A&gt;B</b> (supérieur) <b>A&lt;B</b> (inférieur) <b>A&gt;=B</b> (supérieur ou égal) <b>A&lt;=B</b> (inférieur ou égal)	
<b>Et / Ou</b>	A <b>and</b> B / A <b>or</b> B	
<b>Si ... Sinon Si ... Sinon</b>	<b>if condition C1 :</b> <i>instruction A1</i> <b>elif condition C2 :</b> <i>instruction A2</i> <b>else :</b> <i>instruction A3</i>	C'est le <b>décalage</b> vers la droite qui indique les instructions faisant partie de la structure conditionnelle.  Il n'y a pas d'instruction de fin.  Il en est de même pour <b>for</b> , <b>while</b> et <b>def</b> .
<b>Boucle Pour</b>	<b>for i in range(1,n+1) :</b> <i>instruction A</i>  la variable i parcourt tous les entiers de 1 à n	- <b>for i in range(n)</b> : la variable i parcourt tous les entiers de 0 à n-1 - <b>for i in range(m,n)</b> : la variable i parcourt tous les entiers de m à n-1 - <b>for i in range(m,n,p)</b> : la variable i parcourt tous les entiers de m à n-1 avec un <b>pas</b> de p.
<b>Boucle Tant que</b>	<b>while condition :</b> <i>instruction A</i>	
<b>Fonctions</b>	<b>Def exemple(a,b ...):</b> <i>instruction ... y=...</i> <b>return(y)</b>	a,b,... sont les <b>arguments</b> de la fonction exemple  On peut aussi retourner plusieurs valeurs : <b>return(x,y,z,...)</b>
Insérer un <b>module</b>	Un module est une bibliothèque comportant un ensemble de fonctions. Je présente ci-dessous les modules utilisés au lycée.	
<b>Opérations mathématiques : math</b> <a href="https://www.afpy.org/doc/python/3.5/library/math.html">Toutes les fonctions du module math</a> <a href="https://www.afpy.org/doc/python/3.5/library/math.html">https://www.afpy.org/doc/python/3.5/library/math.html</a>		<b>from math import *</b> On peut aussi importer uniquement la fonction souhaitée : <b>from math import sqrt</b> Le module math, contient les définitions de nombreuses fonctions mathématiques telles que <b>sin, cos, tan, sqrt, pi ...</b>
<b>Nombres aléatoires : random</b> <a href="https://docs.python.org/fr/3.5/library/random.html">Toutes les fonctions du module random</a> <a href="https://docs.python.org/fr/3.5/library/random.html">https://docs.python.org/fr/3.5/library/random.html</a>		<b>from random import *</b> Le module random contient les définitions de nombreuses fonctions faisant référence au hasard telles que : - <b>uniform(a,b)</b> qui retourne un nombre aléatoire compris entre a et b - <b>randint(a,b)</b> qui retourne un entier aléatoire compris entre a et b
<b>Graphiques : pylab</b> <a href="http://matplotlib.free.fr/bases.html">Bases du module pylab</a> <a href="http://matplotlib.free.fr/bases.html">http://matplotlib.free.fr/bases.html</a>		<b>from pylab import *</b> Le module pylab contient de nombreuses fonctions graphiques, ce qui en fait un outil très puissant pour créer des graphiques scientifiques. Ce module possède aussi les fonctions usuelles du module math , il n'est donc pas utile d'importer aussi celle-ci lorsqu'on utilise pylab. On peut aussi utiliser une version plus légère : <b>matplotlib.pyplot</b> . Mais celui-ci ne possède pas les fonctions du module math.
<b>Listes et chaînes de caractères</b>	<b>A=[]</b> permet de définir la liste vide A <b>A.append(x)</b> ajoute la valeur x à la liste (Si la liste était définie jusqu'au 10 ème terme, x sera le 11ème terme)	
<b>Longueur</b>	<b>len(A)</b> renvoie la longueur de la liste ou de la chaîne de caractères A	
<b>Extraire</b>	<b>A[k]</b> renvoie le k+1 ème élément de la liste ou de la chaîne de caractères A. Attention A[0] est le premier terme de la liste.	
<b>Concaténer</b>	"mathé "+"matiques" donne la chaîne de caractères "mathématiques" [1,2,3,4]+[5,6,7,8] donne la liste [1,2,3,4,5,6,7,8]	

