

Tableau récapitulatif des instructions en Python

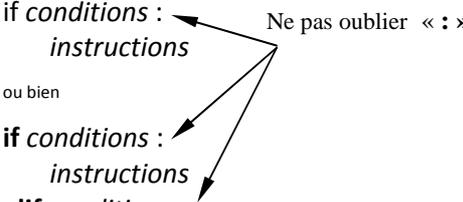
En Python, pour faire appel à des modules supplémentaires avec des commandes particulières on doit l'indiquer avec des instructions **en début du programme** du type :

from math import* : pour des fonctions mathématiques telles que « pi » ou « cosinus »,...

from random import* : pour faire un calcul aléatoire

from tkinter import * : pour créer des interfaces graphiques

Si vous programmez sous EduPython, vous pouvez également appeler la bibliothèque lycée par l'appel **from lycee import ***, qui comprend toutes les bibliothèques citées plus haut.

Type d'instruction	Ecriture en Python	Exemples
Pour afficher un message à l'écran	<code>Print("message")</code>	<code>Print("Bonjour !")</code>
Entrer une valeur au clavier pour une variable X, mais précédée par un message * si X est une chaîne de caractères : * si X est un entier : * si X est un réel :	<code>X=input("message")</code> <code>X=int(input("message"))</code> <code>X=float(input("message"))</code>	<code>X=int(input("Entrer un nombre entier : "))</code>
Afficher le résultat d'une variable X à l'écran	<code>print (X)</code>	<code>print (X)</code>
Afficher le résultat d'une variable X à l'écran, mais précédée ou suivi de messages	<code>print ("message n°1" , X , "message n°2")</code>	<code>print ("la longueur est de " , X , " cm")</code>
Affecter dans l'algorithme une valeur à une variable A	<i>A=valeur</i> <i>A=formule</i>	<code>A=5</code> le programmeur affecte 5 à A <code>A=X+2</code> le programmeur affecte à A la valeur de X à laquelle on ajoute 2 <code>A=A+1</code> le programmeur affecte à A la précédente valeur de A à laquelle on ajoute 1
Nombres au hasard : un réel $\in [0 ; 1[$ un réel $\in [a ; b[$ un entier $\in [a ; b]$	<code>random()</code> <i>from random import*</i> <code>a+(b-a)*random()</code> <i>from random import*</i> <code>randint(a,b)</code> <i>from random import*</i>	<code>random()</code> <code>3+5*random()</code> donne un réel $\in [3 ; 8[$ <code>randint(5,13)</code> donne un entier $\in [5 ; 13]$
Opérations particulières : Division euclidienne de A par B Reste de la division euclidienne de A par B A^n \sqrt{A} partie entière de A	<code>A//B</code> <code>A%B</code> <code>A**n</code> <code>sqrt(A)</code> <i>from math import*</i> <code>floor(A)</code> <i>from math import*</i>	<code>X//3</code> <code>X%3</code> <code>X**2</code> calcule le carré de X <code>sqrt(X)</code> calcule la racine carré de X <code>floor(pi)</code> donne 3 : partie entière de Pi
Tests pour les conditions des boucles : A=B ? A≠B ? A≤B ? A≥B ? A<B<C ?	<code>A==B</code> <code>A!=B</code> <code>A<=B</code> <code>A>=B</code> <code>A<B<C</code> ou bien <code>A<B and B<C</code>	Effectue des instructions si : <code>A==2</code> A est égal à 2 <code>A!=2</code> A est différent de 2 <code>A<=2</code> A est inférieur ou égal à 2 <code>A>=2</code> A est supérieur ou égal à 2 <code>2<A<5</code> A est compris entre 2 et 5 exclus
Instruction conditionnelle Si <i>condition</i> alors <i>instructions</i> ou bien Si <i>condition</i> alors <i>instructions</i> Sinon <i>instructions</i>	<code>if conditions :</code> <i>instructions</i> <code>if conditions :</code> <i>instructions</i> <code>elif conditions :</code> <i>instructions</i> <code>else :</code> <i>instructions</i> 	Exemple : <code>if X>0:</code> <code>print "le nombre X est positif"</code> <code>elif X<0 :</code> <code>print "le nombre X est négatif"</code> <code>else :</code> <code>print "le nombre 'X' est nul"</code> <i>ce programme permet de savoir si un nombre est positif , négatif ou nul.</i>

<p>Boucle itérative</p> <p>Exécute en boucle la (ou les) instruction(s) n fois</p> <p>Exécute en boucle la (ou les) instruction(s) pour une variable allant de n à $m - 1$</p> <p>Exécute en boucle la (ou les) instruction(s) pour une variable allant de n à $m - 1$ avec un pas de k.</p> <p>Exécute en boucle la (ou les) instruction(s) pour chaque caractère de la chaîne de caractères ch</p>	<pre>for i in range(n) : instruction(s) for i in range(n, m) : instruction(s) for i in range(n, m, k) : instruction(s) for caractere in ch : instruction(s)</pre> <p style="text-align: center;">Ne pas oublier « : »</p>	<p>Exemple :</p> <pre>S=0 for i in range(1;11) : S=S+i print "la somme S est égale à ",S</pre> <p><i>ce programme permet de calculer la somme des 10 premiers entiers</i></p>
<p>Boucle conditionnelle</p> <p>Tant que <i>condition</i> instructions</p>	<pre>while condition : instructions</pre> <p style="text-align: center;">Ne pas oublier « : »</p>	<pre>X= -1 while X<0 : X=input("saisir un nb positif")</pre> <p><i>Ce programme permet d'obliger à saisir un nombre positif</i></p>
<p>Fonction</p> <p>Une fonction peut avoir aucun, un ou plusieurs paramètres. Une fonction peut renvoyer un résultat (avec l'instruction return). Celui-ci est alors réutilisable dans un programme ou une autre fonction.</p>	<pre>def fonction(param1,param2, etc.) : instruction(s) return resultat</pre>	
<p>Le type Liste</p> <p>Affecter une liste vide à une variable</p> <p><i>Rappel : une liste est un tableau à une ligne et n colonnes n étant la longueur de la liste.</i></p> <p><i>Chaque élément est indexé de 0 à n-1</i></p> <p>Récupérer la longueur de la liste</p> <p>Supprimer un élément de la liste</p> <p>Concaténer deux listes</p> <p>Ajouter un élément à la fin de la liste</p> <p>Insérer au rang indiqué un élément</p> <p>Supprimer un élément de la liste</p> <p>Supprimer un élément à un rang donné</p> <p>Récupérer le rang (=l'index) de l'élément souhaité</p> <p>Trier une liste par ordre croissant ou alphabétique</p> <p>Inverser l'ordre des éléments d'une liste</p>	<pre>liste=[] l = len(liste) del(liste[élément]) liste1 + liste2 liste.append(élément) liste.insert(rang ,élément) liste.remove(élément) liste.pop(rang) liste.index(élément) liste.sort() <i>attention chaque élément de la liste doit être de même type !</i> liste.reverse()</pre>	<p><i>A tester pour vous ré-entraîner aux listes</i></p> <pre>s=['lundi' , 'mardi' , 230 , 'dimanche'] print(len(s)) del(s[2]) s s.append('samedi') s s.insert(3,29) s s.remove('lundi') s s.pop(3) s s.index('mardi') s s.index('jeudi') s s.reverse() s s.sort()</pre>

Le type String

Une donnée de type **string** peut se définir comme une suite quelconque de caractères. On peut la délimiter par des simples quotes '...' ou par des doubles quotes « ... ».

Une chaîne de caractères peut alors être considérée comme une suite de caractères indexés de 0 à $n-1$, n étant la longueur de la chaîne.

- * Les fonctions **len** ou **+** (pour concaténer) sont les mêmes que pour les listes.
- * Les chaînes de caractères sont non modifiables (l'instruction `chaine[j] = 'a'` est impossible)
- * On peut comparer des chaînes de caractères.

Fonctions qui peuvent également vous servir : (notons ch notre chaîne de caractère)

`ch.find(sch)` : cherche la position d'une sous-chaîne sch dans la chaîne

`ch.count(sch)` : compte le nombre de sous-chaînes sch dans la chaîne

`ch.lower()` : convertit une chaîne en minuscules

`ch.upper()` : convertit une chaîne en majuscules

`float(ch)` convertit la chaîne ch en un nombre réel (*float*) (pour fonctionner il faut que la chaîne représente bien un tel nombre)

`int(ch)` convertit la chaîne ch en un nombre entier (idem)

`ord(ch[caractere])` : permet de savoir la valeur ASCII d'un caractère de la chaîne.

Représentation graphique d'une fonction

Pour tracer la courbe représentative d'une fonction, on fera l'appel de 2 bibliothèques :

- La bibliothèque **numpy** que l'on peut renommer avec l'alias np – qui permet de faire du calcul scientifique
- La bibliothèque **matplotlib.pyplot** que l'on peut renommer avec l'alias plt – qui permet de tracer les courbes

Si vous programmez **avec Edu Python**, la bibliothèque numpy est intégrée au module lycée proposé donc vous pouvez remplacer l'appel classique « `import numpy as np` » par « `from lycée import *` ».

Toutefois ce n'est pas le cas du module matplotlib. Vous devrez en faire l'appel ! (voir exemple ci-dessous)

		Exemple :
Définir les dimensions du repère : Axe des abscisses de val1 à val2 Axe des ordonnées de val3 à val4	<code>plt.axis([val1,val2,val3,val4])</code>	<code>from lycée import *</code> <code>import matplotlib.pyplot as plt</code> <code>x=np.linspace(-2,3,100)</code> <code>y=x**2</code> <code>plt.axis([-3,3,-1,10])</code> <code>plt.plot(x,y)</code> <code>plt.grid()</code> <code>plt.show()</code>
Affiche un quadrillage	<code>plt.grid()</code>	
Définit l'intervalle [val1 ; val2] dans lequel x varie et le nombre de points calculés par tracer la courbe	<code>np.linspace(val1, val2, nb_points)</code>	
Trace les points de coordonnées x et y	<code>plt.plot(x,y)</code>	
Permet l'affichage de la courbe	<code>plt.show()</code>	