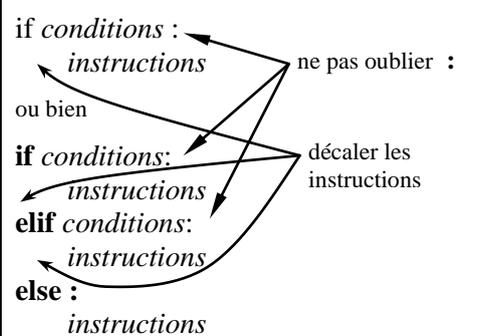
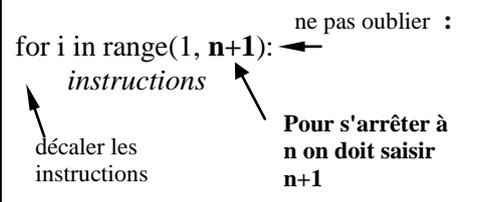


Tableau récapitulatif des instructions en Python

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

Type d'instruction	Ecriture en Python	Exemples
Pour afficher un message à l'écran	Print("message")	Print("bonjour")
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 :	X=input("message") X=int(input("message")) X=float(input("message"))	X=input("saisir la valeur de X")
Afficher le résultat d'une variable X à l'écran	print (X)	print (X)
Afficher le résultat d'une variable X à l'écran, mais précédée ou suivi de messages	print ("message n°1" , X , "message n°2")	print ("la longueur est de " , X , "cm")
Affecter dans l'algorithme une valeur à une variable A	A=valeur A=formule	A=5 le programmeur affecte 5 à A A=X+2 le programmeur affecte à A la valeur de X à laquelle on ajoute 2 A=A+1 le programmeur affecte à A la précédente valeur de A à laquelle on ajoute 1
Nombres au hasard : un réel ∈ [0 ; 1[un réel ∈ [a ; b[un entier ∈ [a ; b]	random() from random import* a+(b-a)*random() from random import* randint(a,b) from random import*	random() 3+5*random() donne un réel ∈ [3 ; 8[randint(5,13) donne un entier ∈ [5 ; 13]
Opérations : Reste de la division euclidienne de A par B A^n \sqrt{A} partie entière de A	A%B pow(A,n) from math import* sqrt(A) from math import* floor(A) from math import*	X%3 pow(X,2) calcule le carré de X sqrt(X) calcule la racine carrée de X floor(pi) donne 3 : partie entière de Pi
Tests pour les conditions des boucles : A=B ? A≠B ? A≤B ? A≥B ? A<B<C ?	A==B A!=B A<=B A>=B A<B<C ou bien A<B and B<C	effectue des instructions si : A==2 A est égal à 2 A!=2 A est différent de 2 A<=2 A est inférieur ou égal à 2 A>=2 A est supérieur ou égal à 2 2<A<5 A est compris entre 2 et 5 exclus
Boucle conditionnelle Si condition alors instructions ou bien Si condition alors instructions Sinon instructions		if X>0: print "le nombre X est positif" elif X<0 : print "le nombre X est négatif" else : print "le nombre 'X' est nul" ce programme permet de savoir si un nombre est positif , négatif ou nul.
Boucle itérative Pour i un entier variant de 1 à n, Instructions		S=0 for i in range(1;11): S=S+i print "la somme S est égale à ",S ce programme permet de calculer la somme des 10 premiers entiers

<p>Boucle conditionnelle</p> <p>Tant que <i>condition, instructions</i></p>	<pre>while condition: ← ne pas oublier : instructions ← décaler les instructions</pre>	<pre>X= -1 while X<0: X=input("saisir un nbre positif")</pre> <p>Ce programme permet d'obliger à saisir un nombre positif</p>
<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() attention chaque élément de la liste doit être de même type ! liste.reverse()</pre>	<p>A tester pour vous ré-entraîner aux listes</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>
<p>Le type String</p> <p>Une donnée de type string peut se définir comme une <u>suite</u> quelconque <u>de caractères</u>. On peut la délimiter par des simples quotes '...' ou par des doubles quotes « ... ».</p> <p>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.</p> <ul style="list-style-type: none"> * Les fonctions len ou + (pour concaténer) sont les mêmes que pour les listes. * Les chaînes de caractères sont non modifiables (l'instruction <i>chaîne[i] = 'a'</i> est impossible) * On peut comparer des chaînes de caractères. <p>Fonctions qui peuvent également vous servir : (notons ch notre chaîne de caractère)</p> <p>ch.find(sch) : cherche la position d'une sous-chaîne sch dans la chaîne</p> <p>ch.count(sch) : compte le nombre de sous-chaînes sch dans la chaîne</p> <p>ch.lower() : convertit une chaîne en minuscules</p> <p>ch.upper() : convertit une chaîne en majuscules</p> <p>float(ch) convertit la chaîne ch en un nombre réel (<i>float</i>) (pour fonctionner il faut que la chaîne représente bien un tel nombre)</p> <p>int(ch) convertit la chaîne ch en un nombre entier (<i>idem</i>)</p> <p>ord(ch[caractere]) : permet de savoir la valeur ASCII d'un caractère de la chaîne.</p>		