

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 tout début du programme du type **from math import*** ou **from random import***.

Type d'instruction	Ecriture en Python	Exemples
Pour afficher un message à l'écran	<code>print"message"</code>	<code>print"bonjour"</code>
Pour faire entrer une valeur au clavier pour une variable X	<code>X=input()</code>	<code>X=input()</code>
Pour faire entrer une valeur au clavier pour une variable X, mais précédée par un message	<code>X=input("message")</code>	<code>X=input("saisir la valeur de X")</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	<code>A=valeur</code> <code>A=formule</code>	<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 : l'entier 3 le réel 3 π Nombres au hasard : un réel $\in [0 ; 1[$ un réel $\in [a ; b[$ un entier $\in [a ; b]$	3 3. pi <code>from math import*</code> <code>random()</code> <code>from math import*</code> <code>a+(b-a)*random()</code> <code>from math import*</code> <code>randint(a,b)</code> <code>from math import*</code>	<code>3/2</code> <code>3/2</code> donne 1 la partie entière du résultat <code>3./2</code> <code>3./2</code> donne 1.5 comme <code>3/2.</code> et <code>3./2.</code> pi <code>from math import*</code> <code>random()</code> <code>from math import*</code> <code>3+5*random()</code> donne un réel $\in [3 ; 8[$ <code>randint(5,13)</code> donne un entier $\in [5 ; 13]$
Opérations : <code>A+B</code> <code>A-B</code> <code>A*B</code> <code>A/B</code> <code>Aⁿ</code> \sqrt{A} partie entière de A	<code>A+B</code> <code>A-B</code> <code>A*B</code> <code>A/B</code> <code>pow(A,n)</code> <code>from math import*</code> <code>sqrt(A)</code> <code>from math import*</code> <code>floor(A)</code> <code>from math import*</code>	<code>X+5</code> additionne X et 5 <code>X-3</code> soustrait 3 à X <code>2*X</code> multiplie X par 2 <code>X/3</code> divise X par 3 <code>pow(X,2)</code> calcule le carré de X <code>sqrt(X)</code> calcule la racine carré de X <code>floor(pi)</code> donne 3
Tests pour les conditions des boucles : <code>A=B ?</code> <code>A≠B ?</code> <code>A<B ?</code> <code>A>B ?</code> <code>A<B<C ?</code>	<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
Boucle conditionnelle Si condition alors instructions ou bien Si condition alors instructions Sinon instructions		<code>if X>=0:</code> <code>print "le nombre ",X," est positif</code> <code>else:</code> <code>print "le nombre ",X," est négatif</code> ce programme permet de savoir si un nombre est positif ou négatif.
Boucle itérative Pour i un entier variant de 1 à n, instructions	<code>for i in range(1, n+1):</code> ← ne pas oublier : <code>instructions</code> ← décaler les instructions ← Pour s'arrêter à n on doit saisir n+1	<code>S=0</code> <code>for i in range(1;11):</code> <code>S=S+i</code> <code>print "la somme S est égale à ",S</code> ce programme permet de calculer la somme des 10 premiers entiers
Boucle conditionnelle Tant que condition, instructions	<code>while condition:</code> ← ne pas oublier : <code>instructions</code> ← décaler les instructions	<code>X=-1</code> <code>while X<0:</code> <code>X=input("saisir un nbre positif")</code> Ce programme permet d'obliger à saisir un nombre positif