

## Tableau récapitulatif des instructions en Python 3 (niveau 2<sup>nde</sup>)

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 3	Exemples
<b>Afficher</b> un message à l'écran	<code>print("message")</code>	<code>print("bonjour")</code>
<b>Saisir</b> la valeur de X au clavier Accompagné de l'affichage d'un message	<code>X=input()</code> <code>X=input("message")</code>	<code>X=input()</code> <code>X=input("saisir votre âge")</code>
<b>Afficher</b> le résultat d'une variable X Accompagné de l'affichage de messages	<code>print(X)</code> <code>print("message n°1",X, "message n°2")</code>	<code>print(X)</code> <code>print("la longueur est de ",X," cm")</code>
<b>Affecter</b> une valeur à une variable A  Echanger les valeurs de deux variables A et B  Connaitre le type d'une variable (entier, réel, chaîne de caractères)	<code>A=valeur</code> <code>A=formule</code>  <code>A,B=B,A</code>  <code>type(variable)</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 <code>A=5</code> <code>B=3</code> <code>A,B=B,A</code> <code>print(A,B)</code> donne 3 pour A et 5 pour B donne <b>int</b> pour un entier donne <b>float</b> pour un réel donne <b>str</b> pour une chaîne de caractères
<b>Nombres :</b> le réel 3,245 $\pi$ e Nombre de type entier Nombre de type réel Arrondi à n chiffres après la virgule	3.245 <code>pi</code> <code>from math</code> <code>import*</code> e <code>from math</code> <code>import*</code> <code>int(5/3)</code> <code>float(variable)</code> <code>round(x,n)</code>	<code>int(5/3)</code> donne la valeur 1 car $5/3 \approx 1,666\dots$ <code>x=float(input("saisir x"))</code> indique que la valeur saisie pour x sera considérée comme étant un nombre réel <code>round(5/3,0)</code> donne 2
<b>Nombres au hasard :</b> un réel $\in [0 ; 1[$ un réel $\in [a ; b]$ un entier $\in [a ; b]$	<code>random()</code> <code>from random</code> <code>import*</code> <code>uniform(a,b)</code> <code>from random</code> <code>import*</code> <code>randint(a,b)</code> <code>from random import*</code>	<code>random()</code> donne un réel entre 0 et 1 <code>uniforme(3,8)</code> donne un réel $\in [3 ; 8]$ <code>randint(5,13)</code> donne un entier $\in [5 ; 13]$
<b>Opérations :</b> A+B A-B B A×B A÷B reste de la division euclidienne de A par B A <sup>n</sup>	A+B A-B A*B A/B A%B  A**n <code>sqrt(A)</code> <code>from math</code> <code>import*</code> <code>floor(A)</code> <code>from math import*</code>	X+5 additionne X et 5 X-3 soustrait 3 à X 2*X multiplie X par 2 X/3 divise X par 3 14%3 donne 2 car $14=4 \times 3 + 2$  X**2 calcule le carré de X <code>sqrt(X)</code> calcule la racine carrée de X <code>floor(pi)</code> donne 3

$\sqrt{A}$ partie entière de A		
<b>Fonction :</b> $f(x) = 3x + 1$ <i>(au sens mathématique)</i>	def f(x): return 3*x +1	L'image de 5 par f se note alors f(5)
<b>Chaîne de caractères</b> Longueur d'une chaîne de caractères Addition de chaînes de caractères	" <i>chaîne de caractères</i> " len( <i>chaîne de caractères</i> ) <i>chaîne n°1 + chaîne n°2</i>	"ceci est une chaîne" len("ceci est une chaîne")    donne 19 "par"+"tir"    donne la chaîne "partir"
<b>Tests pour les conditions :</b> A=B ? A≠B ? A≤B ? A≥B ? A<B<C ? condition n°1 <b>et</b> condition n°2 condition n°1 <b>ou</b> condition n°2	A==B A!=B A<=B A>=B A<B<C            ou bien    A<B and B<C <i>condition n°1 and condition n°2</i> <i>condition n°1 or condition n°2</i>	effectue les 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 A<10 and A!=0        A est inférieur à 10 <b>et</b> A différent de 0 A>10 or A==0        A est supérieur à 10 <b>ou</b> A est égal à 0

Type d'instruction	Ecriture en Python 3	Exemples
<b>Instructions conditionnelles</b> <b>Si condition alors instructions</b>  ou bien <b>Si condition alors instructions</b> <b>Sinon instructions</b>  ou bien <b>Si condition alors instructions</b> <b>Sinon Si condition</b> <b>alors instructions</b> <b>Sinon Si condition</b> <b>alors instructions</b> ... <b>Sinon instructions</b>	if conditions: instructions  if conditions: instructions else: instructions  if conditions: instructions elif conditions: instructions elif conditions: instructions ... else: instructions	age=int(input("âge ?")) if age>=18: print("la personne est majeure") if age<18: print("la personne est mineure") ce script permet de savoir si une personne est majeure ou non.  age=int(input("âge ?")) if age>=18: print("la personne est majeure") else: print("la personne est mineure")  age=int(input("âge ?")) if 2<=age<=6: print("école maternelle") elif age<=11: print("école primaire") elif age<=15: print("collège") else: print("lycée") ce script donne l'établissement en fonction de l'âge de l'élève non majeur.
<b>Boucle bornée</b> <b>Pour i un entier variant de 1 à n,</b> <b>instructions</b>	for i in range(1,n+1): instructions <div style="margin-left: 200px;"> ← Pour s'arrêter à n on doit saisir n+1 </div>	S=0 for i in range(1,11): S=S+i print("la somme S est égale à ",S) ce script permet de calculer la somme des 10 premiers entiers
<b>Boucle non bornée</b> <b>Tant que condition, instructions</b>	while conditions: instructions	X=-1 while X<0: X=float(input("saisir un positif")) Ce script permet d'obliger à saisir un réel positif pour la valeur de X
<b>Les fonctions</b> (au sens informatique) <b>Définir</b> une fonction : cas général  sans variable pour être exécutée sans résultat en sortie  <b>Récupérer</b> les résultats d'une fonction dans des variables A, B, ... avec un seul résultat dans la variable A  <b>Exécuter</b> une fonction sans transmettre de résultat dans des variables	def nom de la fonction (variable n°1, variable n°2, ...): instructions return résultat n°1,résultat n°2,...  def nom de la fonction (): supprimer la dernière ligne avec l'instruction "return"  A,B,...=nom de la fonction(variable n°1,variable n°2, ...) A=nom de la fonction(variable n°1, variable n°2, ...)  nom de la fonction(variable n°1, variable n°2, ...)	def Somme(a,b): S=0 for i in range(a,b+1): S=S+i return S  début=int(input("entier du début ? ")) fin=int(input("entier de la fin ? ")) total=Somme(début,fin) print("total =",total)  Ce script est composé de la fonction Somme permettant de calculer la somme des entiers d'un entier a à un entier b, inclus. La suite du script montre comment exécuter cette fonction.