

Cours python_100

Types	Opérateurs de base
Variables, affectation et expression	Types structurés
<p>Les noms de variables ne peuvent contenir que des signes alphanumériques (a-z, A-Z, et 0-9) et des underscore (blancs soulignés). Le nom ne peut pas commencer par un chiffre.</p> <p>L'affectation attribue une valeur à une variable à l'aide du symbole = Syntaxe : nom variable=expression</p>	
Instructions	Fonctions
<p>Un conteneur est une variable d'un type structuré ou un intervalle défini par la fonction range Syntaxe : range(4) # conteneur d'entiers de 0 à 3 range(d, n, p) #conteneur d'entiers de d inclus à n exclu par pas de p. range(0,4,1) est identique à range(4) range(d,n) est identique à range(d,n,1)</p> <p>Une boucle bornée (for) exécute un bloc un nombre prédéfini de fois en changeant la valeur d'une variable. Syntaxe : for var in conteneur : #attention aux : Bloc #attention à l'indentation</p>	<p>Une fonction permet de donner un nom à un bloc de code. Syntaxe : def nom_fonction(para1,para2,...): #att aux : corps #att indentation</p> <p>L'appel d'une fonction à la forme suivante : Syntaxe : nom_fonction(exp1,exp2,...)</p> <p>para1, ... sont les paramètres : des noms de variables qui peuvent être utiliser dans le bloc corps, et qui sont initialisés par les expressions exp1, ... passées en arguments lorsque la fonction est appelée.</p>
Bibliothèques	
<p>Une bibliothèque contient des fonctions qui ne sont pas directement accessibles. Il faut importer celle-ci en début de programme pour y avoir accès. Syntaxe : from turtle import* #ici, nous importons toutes les fonctions contenues dans la bibliothèque turtle.</p> <p>Pour avoir accès à la liste de ces fonctions, il suffit de taper dans le shell : import turtle help(turtle)</p>	
math	random
turtle	nsi_ui
forward() ou fd() #avance de ... pixels backward() ou back() #recule de ... pixels left() #tourne à gauche de ... ° right() #tourne à droite de ... ° clear() #efface penup() #lève le crayon pendown() #descend le crayon setx() #déplace le crayon horizontalement de ... pixels sety() #déplace le crayon verticalement de ... pixels	
matplotlib.pyplot	

