FICHE-OUTILS : LANGAGE PYTHON EN SECONDE

IMPORTER UN MODULE OU BIBLIOTHEQUE

Importer le module "math" :	<code>from math import*</code>
Importer le module "random" :	<code>from random import*</code>

TYPES DE VARIABLES, CONVERSION

Type	Nature	Syntaxe	Opération
<code>int (integer)</code>	Nombre entier	<code>int (C)</code>	Valeur de C convertie en entier, si C est un nombre decimal c'est sa troncature qui est renvoyée
<code>float (floating)</code>	Flottant : Nombre décimal / réel	<code>float (C)</code>	Valeur de C convertie en flottant
<code>string</code>	Chaîne de caractère (texte)	<code>str (x)</code>	Valeur de C convertie en chaîne de caractère (texte)

EFFECTUER DES CALCULS

Syntaxe	Opération	Syntaxe	Opération
<code>x=2</code> ou <code>x=y</code>		<code>a/b</code>	
<code>a+b</code> ou <code>a-b</code>		<code>a//b</code>	
<code>a*b</code>		<code>a%b</code>	
<code>a**b</code>		<code>min (a , b)</code>	
<code>a**2</code>		<code>max (a , b)</code>	
<code>round (a)</code>		<code>round (a , b)</code>	

Bibliothèque *math* : il faut avoir importé la bibliothèque *math* avant d'utiliser ces fonctions

Syntaxe	Opération	Syntaxe	Opération
<code>pi</code>		<code>sin (a)</code>	Sinus de a
<code>sqrt (a)</code>		<code>cos (a)</code>	Cosinus de a

Bibliothèque *random* : il faut avoir importé la bibliothèque *random* avant d'utiliser ces fonctions

Syntaxe	Opération	Syntaxe	Opération
<code>random ()</code>		<code>randint (a , b)</code>	Nombre entier aléatoire compris entre deux entiers a et b inclus
<code>uniform (a , b)</code>	Nombre décimal aléatoire compris entre deux nombres décimaux a et b		

Nom :

Prénom :

Classe :

E.H. - 13/10/2019

DEMANDE ET AFFICHAGE

<code>a=input("Message")</code>	
<code>print(m) ou print("Message") ou print("Message",m)</code>	

DEFINITION D'UNE FONCTION

<pre>def nom-fonction(arg1, arg2, ...): instruction1 instruction 2... return resultat</pre>	<p>Une fonction informatique est définie par un nom, peut comporter un, aucun ou plusieurs arguments (arg1, arg2...) et renvoie un resultat.</p>
---	--

INSTRUCTIONS CONDITIONNELLES, TESTS ET CONNECTEURS LOGIQUES

<code>if condition : instructions</code>		<code>a==b</code>	
<code>if condition : instructions1 else : instructions2</code>		<code>a!=b</code>	
<pre>if condition1 : instructions1 elif condition2 : instructions2 else : instructions3</pre>	<p>Si la condition 1 est vérifiée, exécute les instructions 1, sinon, si la condition 2 est vérifiée, exécute les instructions 2, sinon exécute les instructions 3</p>	<code>a<b, a>b a<=b, a>=b</code>	
		<code>Condition1 and condition2</code>	
		<code>Condition1 or condition2</code>	

BOUCLES

<code>for variable in range(n) : instructions</code>	Boucle Pour : Exécute n fois les instructions en faisant varier la valeur de la <i>variable</i> de 0 à $n - 1$ inclus
<code>for variable in range(n,m) : instructions</code>	Boucle Pour : Exécute $m - n$ fois les instructions en faisant varier la valeur de la <i>variable</i> de n à $m - 1$
<code>for variable in range(n,m,p) : instructions</code>	Boucle Pour : Exécute les instructions en faisant varier la valeur de la <i>variable</i> de n à $m - 1$ avec un pas de p
<code>while condition : instructions</code>	Exécute les instructions tant que la condition est vérifiée

Nom :

Prénom :

Classe :

E.H. - 13/10/2019