

Memento Python

Manipulation de listes			Dictionnaires	
<p>list1 = [] : créé une liste vide <i>list1</i></p> <p>list2 =[1,'a',[1,2]] : créé <i>list2</i> une liste à 3 éléments</p> <p>list3=[1]*10 : créé <i>list3</i> qui contient 10 éléments égaux à 1</p> <p>list4 = [i for i in range(5)] équivalent à list4=[0,1,2,3,4]</p> <p>list1[i] : renvoie l'élément de <i>list1</i> à l'index i (i>=0). Si i<0, on part de la fin de la liste (list1[-1] pour le dernier, list1[-2] pour l'avant dernier...)</p> <p>len(list1) : renvoie le nombre d'élément de <i>list1</i></p> <p style="text-align: center;">Méthodes sur les listes</p> <p>list1.append(x) : ajoute l'élément x à la fin de la liste <i>list1</i></p> <p>list1.append(list2) : ajoute tous les éléments de <i>list2</i> à la fin de <i>list1</i></p> <p>list1.pop([i]) : enlève de <i>list1</i> l'élément situé à la position indiquée et le renvoie en valeur de retour.</p> <p>list1.insert(i, x) : insère un élément à la position indiquée (i est la position de l'élément avant lequel l'insertion doit s'effectuer).</p> <p>list1.remove(x) : supprime de <i>list1</i> le premier élément dont la valeur est égale à x.</p> <p>list1.clear() : supprime tous les éléments de <i>list1</i>.</p> <p>list1.index(x[, start[, end]]) : renvoie la position du premier élément de <i>list1</i> dont la valeur égale x (en commençant par zéro).</p> <p>list1.count(x) : renvoie le nombre d'éléments ayant la valeur x dans <i>list1</i>.</p> <p>list1.extend(list2) : concatène <i>list2</i> à la suite de <i>list1</i>.</p> <p>list1.copy() : renvoie une copie superficielle de <i>list1</i></p> <p>list(list1) : renvoie une copie profonde de <i>list1</i> (clone)</p>			<p>dico1 = { } : Créé un dictionnaire vide</p> <p>dico2= {'a':1,2:'oui','c':[1,2]} : créé un dictionnaire à 3 clés 'a', 2 et 'c'. Ces clés permettant d'accéder aux valeurs du dictionnaire : 1, 'oui' et [1,2].</p> <p>dico2['a'] : renvoie la valeur correspondante (ici 1) ou une erreur si la clé 'a' n'est pas dans le dictionnaire.</p> <p>del dico2['a'] : supprime une association dans <i>dico2</i> (erreur si la clé n'existe pas)</p> <p>'a' in dico2 : vérifie si la clé 'a' se trouve (True) ou non (False) dans <i>dico2</i></p> <p>list(dico2) : renvoie la liste des clés de <i>dico2</i>, ici ['a',2,'c']</p>	
			Types	
			<p>type() : pour connaître le type d'une variable</p> <p>int() : permet la transformation en un entier</p> <p>float() : permet la transformation en flottant</p> <p>str() : permet la transformation en chaîne</p>	
			Procédure - Fonction	
			<p>Fonction : renvoie un résultat par l'intstruction return</p> <p>def nom_fonction(paramètres) <i>Instructions</i> return résultat</p> <p>Procédure : ne renvoie pas de résultat (pas de return)</p> <p>def nom_fonction(paramètres) <i>Instructions</i></p>	
Valeurs particulières	Instructions conditionnelles		Itération for	
<p>True : vrai</p> <p>False : faux</p> <p>float('inf') : +∞</p>	<p>if condition : <i>instructions</i></p> <p>if condition : <i>instructions</i> else <i>instructions</i></p>	<p>if condition : <i>instructions</i> elif condition2 : <i>instructions</i> else : <i>instructions</i></p>	<p>for i in list_chaine_ou_dico : <i>instructions</i></p> <p>Parcours des couples d'un dictionnaire (items)</p> <p>for (i,j) in dic.items(): print(i,j)</p> <p>for var in range (deb, fin, pas) : <i>instructions</i></p>	
Boucle Tant que		Range		
<p>While condition : <i>instructions</i></p>		<p>range(a) : séquence des valeurs [0, a[</p> <p>range (b,c) : séquence des valeurs [b, c[(pas de 1)</p> <p>range (b, c, g) : idem avec un pas de g</p> <p>range(b,c,-1) : valeurs de b(incl.) à c (excl.) , pas -1</p>		
Opérateurs				
Arithmétiques		Logiques	Comparaison	Texte
<p>+ : addition - : soustraction</p> <p>* : multiplication ** : puissance</p> <p>/ : division</p> <p>// : division euclidienne(quotient),</p> <p>% : reste de la division euclienne</p>		<p>and : et</p> <p>or : ou</p> <p>not : négation</p>	<p>== égalité != différence</p> <p>< inférieur <= inférieur ou égal</p> <p>> supérieur >= supérieur ou égal</p> <p style="text-align: center;">Ensemblistes</p> <p>in : appartient</p> <p>not in : n'appartient pas</p>	<p>len(text) : renvoie la longueur de la chaîne <i>text</i></p> <p>text1 + text2 : concatène les chaînes <i>text1</i> et <i>text2</i></p> <p>text* n : répétition de n fois de la chaîne <i>text</i></p>