

Informatique Pour Tous : Fiche bilan sur les Itérables utiles

Itérable	<code>list</code>	<code>numpy.array</code>
Format	<code>[item1, item2, ..., item]</code>	<code>[item1, item2, ..., item]</code>
Typage	<i>Dynamique</i>	<i>Fixe</i>
Longueur	<code>len(list)</code>	<code>array.shape</code>
Accès	<code>list[i]</code>	<code>array[i]</code>
Modification	<code>list[i] = item</code>	<code>array[i] = item</code>
Création	<code>list.append(item)</code> ajoute à la fin <code>list.insert(index, item)</code> ajoute item en position index	
Destruction	<code>del list[i]</code> <code>list[i].pop(index)</code>	
Création de La variable	<code>A = []</code> <code>A = [1, 2, 3]</code>	<code>A = np.array([item, ...])</code> <code>A = np.linspace(min, max, nbr)</code> <code>A = np.zeros(nombre)</code> <code>A = np.zeros_like(array)</code> <code>A = np.ones(nombre)</code> <code>A = np.ones_like(array)</code>
Parcours	<code>for element in A</code> ou <code>for i in range(...)</code> <code>A[i] = ...</code>	idem <code>list</code>

Itérable	<code>tuple</code>	<code>dict</code>
Format	<code>(item1, item2, item3)</code>	<code>{key1 : item1, key2 : item2, ...}</code>
Typage	<i>Dynamique</i>	<i>Dynamique</i>
Longueur	<code>len(tuple)</code>	<code>len(dict)</code>
Accès	<code>tuple[i]</code>	<code>dict[key]</code> accès à la valeur associée <code>dict.items()</code> couples (clé, valeur) <code>dict.values()</code> liste des valeurs <code>dict.keys()</code> liste des clés
Modification		<code>dict[key] = value</code>
Création		<code>dict[key] = value</code>
Destruction		<code>del dict[key]</code> supprime la paire (clé, valeur)
Création de La variable	<code>A = (item1, item2, item3)</code>	<code>A = {}</code> <code>A = {clé : valeur}</code>
Parcours	idem <code>list</code>	<code>for cles in A</code> ou <code>for cles in A.keys() :</code> <code>for valeurs in A.values() :</code> <code>for cles, valeurs in A.items() :</code>

Itérable	string	range(from, to, pas)
Format	"Quantum"	(from, from+pas, ..., from+n*pas < to)
Typage	<i>Fixe (modulo les encodages)</i>	<i>Entier</i>
Longueur	len(string)	
Accès	string[i]	
Modification		
Création	string = string + 'a' ajoute 'a' à la fin, concaténation	
Destruction	string = string[:-1]	
Création de La variable	A = "" A = "Quantum"	L = range(from, to, pas) for i in range(from=0, to, pas=1):
Méthodes	list(A) = ["Q", "u", "a", "n", "t", "u", "m"] A.replace("Q", "q") retourne "quantum" A.split("t") retourne ['Quan', 'um'] A.strip("m") retourne "Quantu" équivalent de A[:-1]	

Itérable	zip(liste1, liste2, ...)	enumerate(iterable)
Format	[(i1_l1, i1_l2), (i2_l1, i2_l2), ... (in_l1, in_l2)]	[(0, it0), (1, it1), (2, it2), ...]
Typage	<i>Fixe (liste de tuple)</i>	<i>Fixe (liste de tuple (index, élément))</i>
Longueur	len(liste1) = len(liste2) = len(liste3) = ...	len(iterable)
Accès		
Modification		
Création		
Destruction		
Création de La variable	Pour liste1 = [i1_l1, i2_l1, i3_l1, ...] Liste2 = [i1_l2, i2_l2, i3_l2, ...] ... L = zip(liste1, liste2, liste3, ...) for em1, em2, em3 in zip(liste1, liste2, lsite3): em1, em2, em3 sont les ième éléments des listes respectivement liste1, liste2, liste3, ...	liste = ['a', 'b', 'c', ...] for i, letter in enumerate(liste): ici i est le compteur et letter l'élément i de la liste.