Cours Introduction à Python
structures de données : Les types non mutables

Introduction

Ce sont des données ou objets composites destinés à contenir d'autres objets (une variable dans laquelle on peut mettre plusieurs variables).
On distingue des types modifiables ou mutables et des types non mutables.
Non modifiable signifie qu'une donnée, une fois créée en mémoire, ne pourra plus être changée, toute transformation résultera en la création d'une nouvelle valeur distincte avec une autre adresse mémoire.

1. Les tuples:

Un tuple est une séquence ordonnée et non modifiable d'éléments éventuellement hétérogènes séparés par des virgules et entourés de parenthèses (Les parenthèses ne sont pas obligatoires).

Fonctions	**Syntaxe**	**Exemples**	**Résultats**
Créer un tuple vide	nom_tuple = ( )	T = ( )	.....
Créer un tuple non vide	nom_tuple = (…)	T = (1, "ok", "non")	.....
Créer un tuple à l'aide de la fonction **tuple**() qui prend en argument un objet séquentiel et renvoie le tuple correspondant	nom_tuple = tuple(…)	T1 = tuple(x*x for x in range(1,9));	.....
Créer un tuple sans les parenthèses ( ). Les parenthèses ne sont pas obligatoires mais facilite la lisibilité du code (rappelons que la force de python est sa simplicité de lecture)		T2 = 1, 2, 3 type(T2)	.....
Créez un tuple avec une seule valeur, n'oubliez pas d'y ajouter une virgule, sinon ce n'est pas un tuple.		T3 = ("ok") type(T3) T4 = ("ok",) ;Type(T4)	.....
Ajouter et supprimer une valeur	nom_tuple.append(elt) nom_tuple.remove(1)	Ces méthodes n'existe pas ! Les tuples n'ont pas de méthodes, sauf les suivantes	
Trouver l'index d'une valeur	nom_tuple.index(entree)	T.index("b")	.....
Nombre d'occurrences d'une entrée	nom_tuple.count(entree)	tup = ("a","a","a","b","c","c") tup.count("a")	.....
Extraire tous les éléments du tuple		tup[:]	.....
Accès à un élément :	Nom_tuple [index] Avec : index ∈ [0,1,2,…,-2,-1]	tup = ("a","d","m") tup[0] tup[-1]	.....
Les n premiers éléments	tup[:n]	tup[:2]	.....
Les n dernières éléments	Tup[-n :]	tup[-1 :] ; tup[-3 :]	.....
Les éléments entre deux index n et m	Tup[n:m+1]	tup[1 : 3]	.....
Nombre des éléments	len(nom_tuple)	len(tup)	.....
Vider la tuple	nom_tuple = ( )	tup = ()	.....
Boucler sur un tuple	for element in tup : **ou** for pos in range(len(tup)): print (element) print (tup[pos])		.....

Les valeurs retournées par la boucle sont des **tuples**	for element in enumerate (tup) : print (element)		…… …… ……
Tester si un item existe dans un tuple	item **in** nom_tuple item **not in** nom_tuple	3 in tup 11 not in tup	…… …… ……
Agrandir un tuple par un autre tuple	tuple1 = tuple1 + tuple2 tuple1 * nombre_entier	t1 = t1 + t2 t1 * 3 (0,) * 5 #utile pour l'initialisation	…… …… …… ……
Afficher le maximum, le minimum et la somme des éléments d'un tuple		max(tup) ; min(tup) ; sum(tup)	…… …… ……
Comparer le contenu de deux tuples		t1= = t2 t1 != t2	…… ……

A quoi servent donc les tuples ?

Votre code est plus sûr si vous «protégez en écriture» les données qui n'ont pas besoin d'être modifiées.
Utiliser un tuple à la place d'une liste revient à avoir une assertion implicite que les données sont constantes

2. Les chaines de caractères :

Une chaîne de caractères est une suite ou séquence **ordonnée** de caractères Unicode (lettres et symboles) entre simple ou double côtes non modifiable. Elle appartient au type **str**.
On peut dire que les chaînes sont des **listes non modifiables**. Elles supportent donc le test d'appartenance, la concaténation, la répétition, l'accès par indice, par tranche, la taille,... mais pas de changement de ses éléments. On peut convertir une chaîne en une liste et la modifier.

Fonctions	**Syntaxe**	**Exemples**	**Résultats**
Créer une chaine de caractères	nom_chaine = "catactères"	chaine= "Bonjour "	…… …… ……
Longueur de chaine	len(nom_chaine)	len (chaine)	…… ……
Afficher toute la chaine		chaine [:]	……
Accès à un caractère de la chaine	nom_chaine [index] Avec : index ∈ [0,1,2,…,-2,-1]	ch = "01234321" ch [0] ch [-1]	…… …… ……
Les n premiers caractères	nom_chaine [:n]	ch [:2]	…… ……
Les n dernières caractères	nom_chaine [-n :]	ch [-1 :] ; ch [-3 :]	……
Les caractères entre deux index n et m	nom_chaine [n:m+1]	ch [1 : 3]	…… ……
Boucler sur une chaine	for carc in ch : **ou** for pos in range(len(ch)) : print (carc) print (ch[pos])		…… ……
Les valeurs retournées par la boucle sont des **tuples**	for carc in enumerate (ch): print (carc)		…… …… ……

Le caractère : ' **"ab'c " ou ' ab\'c '** **' ab"c ' ou "ab\"c "**	'Aujourd\'hui' ou "Aujourd'hui"	ch = 'Aujourd'hui' ch ch1 = 'Aujourd\'hui' ch1	…… …… …… ……
saut ligne	\n : retour à la ligne """ : pour encadrer une chaîne définie sur plusieurs lignes	ch2 = 'Première ligne\nDeuxième ligne' print(ch2) ch3 = """Première ligne Deuxième ligne""" print(ch3)	…… …… …… …… …… ……
Concaténation	+ : concaténation de deux chaines	ch4 = "Nom" + "Prenom" ch4	…… ……
répétition	* : opérateur de répétition de chaine	ch5 = "Nom" *3 ch5	…… ……
Convertir une chaine en une liste	list (chaine)	ch6 = list ("Bonjour ") ch6 ; type (ch6)	…… ……
Renvoyer l'index de la première occurrence de la valeur et renvoie erreur si non trouvée	nom_chaine.index(valeur)	ch2.index("n") ch2.index("a")	…… …… …… ……
Tester le début et la fin		chaine.startswith('B') chaine.endswith('f')	…… ……
Éliminer les espaces au début et à la fin	nom_chaine.strip( )	chaine= " B o n j o u r " chaine.strip()	…… ……
Rechercher la première occurrence d'une sous-chaîne; renvoie -1 si non trouvé.	chaine1.find(chaine2) Renvoie l'index de première occurrence de ch2 dans ch1	chaine1= "Bonjour tout le monde " chaine2= "tout" chaine1.find(chaine2)	…… …… …… ……
compte le nombre de sous-chaînes dans la chaîne		chaine1.count(chaine2)	…… …… ……
Remplace une sous-chaîne par une autre		chaine1.replace(" ","*") chaine1	…… ……
Changer la casse		chaine1.upper() ; chaine1.lower() ;	…… ……
Met majuscule en entête		chaine1.capitalize()	……
Vérifier si la chaîne ne contient respectivement que des caractères alphanumériques, alphabétiques, numériques ou des espaces		chaine1.isalpha() chaine1.isdigit() chaine1.isalnum() chaine1.isspace()	…… …… …… …… …… …… ……
Retourner True si seule la première lettre de chaque mot de la chaine est en majuscule		chaine1.istitle()	…… ……
On peut convertir un entier en chaîne et vice versa	str(255); int('0xA0',16) ; bin(255) ; int('0b110',2) float('160.3') ;		…… …… ……
ord('c') : retourne le code ASCII de 'c' chr (code ASCII) : retourne le caractère correspondant		ord ('m') chr(109)	…… …… ……

Exercices [Les chaines de caractères]

1. Ecrire un script qui détermine si une chaîne contient ou non le caractère « e »
2. Ecrire un script qui compte le nombre d'occurrences du caractère « e » dans une chaine
3. Ecrivez un script qui recopie une chaîne (dans un nouvelle variable), en insérant des astérisques entre les caractères. Ainsi par exemple, « Docteur » devra devenir « d*o*c*t*e*u*r »
4. Ecrivez un script qui recopie une chaîne (dans une nouvelle variable) en l'inversant. Ainsi par exemple, « python » deviendra « nohtyp ».
5. En partant de l'exercice précédent, écrivez un script qui détermine si une chaine de caractères est un palindrome (c'est-à-dire une chaîne qui peut se lire indifféremment dans les deux sens), comme par exemple « radar » ou « s.o.s »