

Camino al Certamen 2: Resumen de Contenidos

	Lista	Diccionario	Tupla	Conjunto
Características especiales	Puede ser ordenada Acceso a datos mediante su posición	Acceso más rápido a los datos	No es modificable una vez creada No tiene orden	No permite elementos repetidos No tiene orden
Creación	<code>[]</code> , <code>[elem1, elem2, elem3, ...]</code>	<code>dict()</code> , <code>{llave1:valor1, llave2:valor2, ...}</code>	<code>(elem1, elem2, elem3, ...)</code>	<code>set(elem_iterable)</code> , <code>set()</code> , <code>{obj1, obj2, ...}</code>
Acceso a Datos	<code>lista[i]</code>	<code>dicc[llave]</code>	<code>obj1, obj2, ... = tupla</code> <code>tupla[i]</code>	
Asignación	<code>lista[i]=obj</code> <code>lista.append(obj)</code>	<code>dicc[llave]=valor</code>		<code>conj.add(obj)</code>
Contar elementos	<code>lista.count(obj)</code>		<code>tupla.count(obj)</code>	
Cantidad de elementos	<code>len(lista)</code>	<code>len(dicc)</code>	<code>len(tupla)</code>	<code>len(conj)</code>
Posición de un elemento	<code>lista.index(obj)</code>		<code>tupla.index(obj)</code>	
Eliminar un elemento	<code>lista.remove(obj)</code> <code>del lista[i]</code>	<code>del dicc[llave]</code>		<code>conj.remove(obj)</code>
Existencia de un elemento	<code>obj (not) in lista</code>	<code>llave (not) in dicc</code> <code>valor (not) in dicc.values()</code>	<code>obj (not) in tupla</code>	<code>obj (not) in conj</code>
Suma de todos los elementos	<code>sum(lista)</code>	<code>sum(dicc)</code> <code>sum(dicc.values())</code>	<code>sum(tupla)</code>	<code>sum(conj)</code>
Funciones específicas	Unir dos listas <code>lista1 + lista2</code>	Lista con las llaves de un diccionario <code>list(dicc)</code>	Comparación entre tuplas <code>tupla1 == tupla2</code> <code>tupla1 < tupla2</code> <code>tupla1 > tupla2</code>	Intersección entre <i>conj1</i> y <i>conj2</i> <code>conj1 & conj2</code>
	Repetir <i>n</i> veces una lista <code>lista * n</code>	Lista con los valores de un diccionario <code>list(dicc.values())</code>		Unión entre <i>conj1</i> y <i>conj2</i> <code>conj1 conj2</code>
	Cortar una lista entre las pos. <i>i</i> y <i>j</i> <code>lista[i:j]</code>	Iterar sobre las llaves <code>for llave in diccionario</code>		Diferencia entre 2 conjuntos <code>conj1 - conj2</code>
	Invertir una lista <code>lista.reverse()</code>	Iterar sobre los valores <code>for valor in diccionario.values()</code>		Diferencia Simétrica entre 2 conjuntos <code>conj1 ^ conj2</code>
	Ordenar una lista <code>lista.sort()</code>	Iterar sobre las llaves y los valores <code>for llave, valor in diccionario.items()</code>		<i>conj1</i> subconjunto de <i>conj2</i> ? <code>conj1 < conj2</code>
				<i>conj1</i> subconjunto de (o igual a) <i>conj2</i> ? <code>conj1 <= conj2</code>

Creado por Camilo Saldías

<https://github.com/csaldias/python-usm>