

Sequency things in Python

	Lists	Strings	Tuples	Sets	Dictionaries
Declaration of constants	Use square brackets: <code>a = [1, 2, 3, 4, 5]</code> For an empty list: <code>a = []</code>	Use quotes: <code>a = "Kflortnic"</code> For an empty string: <code>a = ""</code>	Use parentheses (parens optional): <code>a = (1, 2, 3, 4, 5)</code> For an empty tuple: <code>a = ()</code>	Use curly brackets <code>a = {1, 2, 3, 4}</code> For an empty set: <code>a = set()</code>	Use curly brackets & colons <code>a = {1: "one", 5: "five"}</code> For an empty dictionary: <code>a = {}</code>
Order matters	Yes	Yes	Yes	No	Yes (colon-wise) No (comma-wise)
Length function	<code>len(a)</code> returns number of elements in the list	<code>len(a)</code> returns number of characters in the string	<code>len(a)</code> returns number of items in the tuple	<code>len(a)</code> returns number of elements in the set	<code>len(a)</code> returns number of pairs in the dictionary
Random access through indexing	<code>a[2]</code> returns the third element of the list	<code>a[2]</code> returns the third character of the string	<code>a[2]</code> returns the third item of the tuple	Since order does not matter, <code>a[2]</code> generates an error	<code>a[2]</code> will generate an error, but <code>a[5]</code> will return the string "five" in this case.
Creating subsequences through slicers	<code>a[1:4]</code> returns the list <code>[2, 3, 4]</code>	<code>a[1:4]</code> returns the string <code>flor</code>	<code>a[1:4]</code> returns the tuple <code>(2, 3, 4)</code>	Slicers cannot be used	Slicers cannot be used
Assignment through indexing	<code>a[2] = 9</code> will set the third element to 9	Generates an error	Generates an error	Generates an error	<code>a[2] = "nine"</code> will add a new dictionary item, while <code>a[5] = "nine"</code> reassigns key 5 to the value "nine"
Adding elements	<code>a.append(9)</code> will add a 9 to the end of the list; <code>a.extend([8, 9])</code> will add two elements to the end of the list	<code>a = a + "!!"</code> will add to the string	<code>a = a + (6, 7)</code> will add items to the tuple	<code>a.add(9)</code> will add a 9 to the set. (note: if 9 is already in the set, this statement does nothing); <code>a = a.union(b)</code> will add set b items to set a	<code>a[2] = "two"</code> will add an element to the dictionary if no key of 2 exists yet
Deleting elements	<code>del a[2]</code> will delete the third element	<code>a = a[:2]+a[3:]</code> will effectively delete the third character	<code>a = a[:2]+a[3:]</code> will effectively delete the third item	<code>a.discard(2)</code> will remove the item 2 (not second item) from the set	<code>del a[5]</code> will delete the item with key 5 from the dictionary
Inserting elements	<code>a.insert(2, 2.5)</code> will insert the value 2.5 after the second element	Use slicers: <code>a=a[:4]+"u"+a[4:]</code> inserts a "u" to get "Kflourtnic"	Use slicers: <code>a=a[:2]+(2.5)+a[2:]</code> inserts 2.5 after the second item	Since order does not matter, this is not different from adding	Since order does not matter, this is not different from adding
Membership test	<code>(3 in a)</code> will be true if the list contains the element 3	<code>("k" in a)</code> will be true if the string contains letter "k"	<code>(3 in a)</code> will be true if the tuple contains the item 3	<code>(3 in a)</code> will be true if the set contains the element 3	<code>(3 in a)</code> will be true if the dictionary contains a pair with a key of 3
Other neat things	sort, index, count	find, format, zfill, isdigit	index, count	issubset, intersection	already really cool