

Strings have an index

s = "computer"

Position	0	1	2	3	4	5	6	7
String characters	c	o	m	p	u	t	e	r
Right	-8	-7	-6	-5	-4	-3	-2	-1

Function	Purpose	Example
len(string)	Returns the length of the string or list	<pre>word = "Hello" my_string = len(word) &gt;&gt; 5</pre> <p>Can also be used with a list:</p> <pre>words = ["cat", "dog", "pig"] my_list = len(words) &gt;&gt;3</pre>
The simplest syntax for a slice of a string is:  s[startIndex: endIndex]	This refers to the substring of s starting at index startIndex and stopping just before index endIndex	<p>First 4 characters:</p> <pre>s = "computer" print(s[0:4]) &gt;&gt; comp</pre> <p>Third character from the end:</p> <pre>print(s[-3]) &gt;&gt;t</pre> <p>Get the last 5 characters of a string</p> <pre>print(s[-5:]) &gt;&gt; ute</pre>
stringname.subString(startingPosition, numberOfCharacters)  subject.substring(3,5) subject.left(4)	This gets a substring but the string will start at the 0 <sup>th</sup> character. subject.right(3) will return "nce"	<pre>subject.substring(3,5) will return "uter" subject.left(4) will return "Comp"</pre>

## Working with Strings & String Methods

<pre>subject.right(3)</pre>		
<pre>str.split()</pre>	Splits a word or sentence into a list where there is a space	<pre>sentence = "This is a sentence of words that will be split where there is a space" my_list = sentence.split()</pre>
<pre>str.split(",")</pre>	Splits a word or sentence into a list where there is a comma	<pre>x = 'blue,red,green' my_list = x.split(',')</pre>
<pre>str.upper() str.lower()</pre>	Performing the .upper() method on a string converts all of the characters to uppercase, whereas the .lower() method converts all of the characters to lowercase.	<pre>s = "Whereof one cannot speak, thereof one must be silent." print(s.upper()) print(s.lower())</pre>
<pre>ASC(character) CHR(asciinumber) ASC(A) CHR(97) &gt;&gt;&gt; ord("b") 98 &gt;&gt;&gt; chr(13) \r</pre>	<p>Uppercase letters and lowercase letters have different ASCII values as does numbers represented in a string.</p> <p>The ord function gives the integer value of a character.</p> <p>The chr function returns an integer into ascii.</p>	<p>This converts to and from ASCII.</p> <p>ASC(A) will return 65 (numerical)</p> <p>CHR(97) will return "a" (char)</p>
<pre>" ".join(list)</pre>	Join is roughly the reverse of split. It joins together a list into a string and adds a space between each word	<pre>words= ["Self", "contained", "underwater", "breathing", "apparatus"] my_String = " ".join(words)</pre>
<pre>",".join(list)</pre>	Join is roughly the reverse of split. It joins together a list into a string and adds a comma between each word	<pre>my_String = ", ".join(words)</pre>

## Working with Strings & String Methods

count(str)	The .count() method adds up the number of times a character or sequence of characters appears in a string.	s = "That that is is that that is not is not is that it it is" print(s.count("t")) >>>13
str.strip()	Sometimes you need to remove unwanted spaces from a string. You can use the strip() function	sentence = ' hello apple' print(sentence.strip()) >>> 'hello apple'
str.replace(" ", "")  str.replace("find", "replace")	Sometimes you need to remove characters or unwanted spaces from a string. You can use the replace()	sentence = ' hello apple' print(sentence.replace(" ", "")) >>> 'helloapple'  s = "I intend to live forever, or die trying." print(s.replace("to", "three")) >>>'I intend three live forever, or die trying.'
find()	We search for a specific character or characters in a string with the .find() method	s = "On the other hand, you have different fingers."  s. find("hand") >>>13  The results tell us that "hand" begins at the 13th position in the sequence.