

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

Working with Strings & String Methods

<code>subject.right(3)</code>		
<code>str.split()</code>	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>
<code>str.split(",")</code>	Splits a word or sentence into a list where there is a comma	<pre>x = 'blue,red,green' my_list = x.split(",")</pre>
<code>str.upper()</code> <code>str.lower()</code>	Performing the <code>.upper()</code> method on a string converts all of the characters to uppercase, whereas the <code>.lower()</code> 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>
<code>ASC(character)</code> <code>CHR(ascii number)</code> <code>ASC(A)</code> <code>CHR(97)</code> <code>>>> ord("b")</code> <code>98</code> <code>>>> chr(13)</code> <code>\r</code>	<p>Uppercase letters and lowercase letters have different ASCII values as does numbers represented in a string.</p> <p>The <code>ord</code> function gives the integer value of a character.</p> <p>The <code>chr</code> function returns an integer into <code>ascii</code>.</p>	<p>This converts to and from ASCII. <code>ASC(A)</code> will return 65 (numerical) <code>CHR(97)</code> will return "a" (char)</p>
<code>" ".join(list)</code>	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","apparatu s"] my_String = " ".join(words))</pre>
<code>",".join(list)</code>	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>

<code>count(str)</code>	The <code>.count()</code> method adds up the number of times a character or sequence of characters appears in a string.	<pre>s = "That that is is that that is not is not is that it it is" print(s.count("t")) >>>13</pre>
<code>str.strip()</code>	Sometimes you need to remove unwanted spaces from a string. You can use the <code>strip()</code> function	<pre>sentence = ' hello apple' print(sentence.strip()) >>> 'hello apple'</pre>
<code>str.replace(" ")</code> <code>str.replace("find", "replace")</code>	Sometimes you need to remove characters or unwanted spaces from a string. You can use the <code>replace()</code>	<pre>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.'</pre>
<code>find()</code>	We search for a specific character or characters in a string with the <code>.find()</code> method	<pre>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.</pre>