

# Strings

	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	.	.	4 <sup>th</sup> to Last	3 <sup>rd</sup> to Last	2 <sup>nd</sup> to Last	Last	END	str.substring( 0, 3 ) → “abc” str.substring( 1, 3 ) → “bc” str.substring( str.length()-2 ) → “yz” str.substring( str.length()-4, str.length()-1 ) → “wxy”
<b>String str</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	.	.	<b>w</b>	<b>x</b>	<b>y</b>	<b>z</b>		
<b>Index (position)</b>	0	1	2	3	.	.	str.length()-4	str.length()-3	str.length()-2	str.length()-1	str.length()	

<b>EVEN</b>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	<b>5<sup>th</sup></b>	6 <sup>th</sup>	7 <sup>th</sup>	8 <sup>th</sup>	END	<b>int mid = str.length() / 2; // 8/2 = 4</b> For even-length strings, there is <b><u>NO MIDDLE CHARACTER</u></b> . The character at index <b>mid</b> will be the first character in the right-half of the string.  str.substring( mid-1, mid+1 ) gives the middle 2 characters.
<b>String str</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>		
<b>Index (position)</b>	0	1	2	3 mid-1	4 mid	5 mid+1	6	7	str.length()	

<b>ODD</b>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	<b>5<sup>th</sup></b>	6 <sup>th</sup>	7 <sup>th</sup>	8 <sup>th</sup>	9 <sup>th</sup>	END	<b>int mid = str.length() / 2; // 9/2 = 4</b> For odd-length strings, mid gives the value of the index of the middle character.  str.substring( mid-1, mid+2 ) gives the middle 3 characters.
<b>String str</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>		
<b>Index (position)</b>	0	1	2	3 mid-1	4 mid	5 mid+1	6 mid+2	7	8	str.length()	

# Strings

String Methods: These methods are called using **DOT** syntax using a String object, e.g.:

String + **DOT** + Method:     word.length()            “abcde”.substring(0,1)

Method	Input Parameters	Description	Return Type
<b>length()</b>	No Parameters	Returns the number of characters in the String.	<b>int</b> (NUMBER)
<b>substring</b> (int <i>startIdx</i> , int <i>stopIdx</i> )	<i>startIdx</i> : Index of the 1 <sup>st</sup> character in the new string <i>endIdx</i> : Stopping index; the character at this index is NOT included; that is, the last character in the new string is at index <i>endIdx-1</i> .	Returns a new string containing ( <i>stopIdx-startIdx</i> ) # of characters. The 1 <sup>st</sup> character of the new String is the character at position <i>stopIdx</i> . Characters are added up to, but not including the character at index <i>stopIdx</i> .	<b>String</b> (TEXT)
<b>substring</b> ( <i>startIdx</i> )	<i>startIdx</i> : Index of the 1 <sup>st</sup> character in the new string	Returns a new string containing ( <i>stopIdx-length()+1</i> ) # of characters. The new String includes all characters in the original string starting with the character at <i>startIdx</i> and continuing to the end.	<b>String</b> (TEXT)
<b>equals</b> (String <i>strTest</i> )	<i>strTest</i> : The String used to compare the String object to.	Returns true if <b>strTest</b> matches the object String “abc” matches “abc” “abc” does <b>NOT</b> match “Abc”	<b>Boolean</b> (true/false)
<b>equalsIgnoreCase</b> (String <i>strTest</i> )	<i>strTest</i> : The String used to compare the String object to. Matches strings regardless of uppercase or lowercase letters	Returns true if <b>strTest</b> matches the object String regardless of uppercase or lowercase, i.e. treats uppercase and lowercase letters as if they are the same. “abc” matches “abc” “abc” matches “ABC”	<b>Boolean</b> (true/false)
<b>endsWith</b> (String <i>suffix</i> )	<i>suffix</i> : The String you want to match at the end of the String object.	Returns true if the String object ends with <b>suffix</b> .	<b>Boolean</b> (true/false)
<b>startsWith</b> (String <i>prefix</i> )	<i>prefix</i> : Does the object String start with <b>prefix</b> ?	Returns true if the String object begins with <b>prefix</b> .	<b>Boolean</b> (true/false)
<b>startsWith</b> (String <i>prefix</i> , int <i>offsetIdx</i> )	<i>prefix</i> : Imagine that this creates a substring starting at position <i>offsetIdx</i> . Does that substring start with <b>prefix</b> ?	Returns true if the imagined substring begins with <b>prefix</b> .	<b>Boolean</b> (true/false)
<b>isEmpty</b> ()	No Parameters	Returns true if the String object has no characters. This is the same thing as <b>return str.length() == 0;</b>	<b>Boolean</b> (true/false)