

Some Methods in the Class String

Method	Return Type	Example for	Description
charAt (<i>index</i>)	char	String s = "Java"; c = s.charAt(2); // c='v'	Returns the character at <i>index</i> in the string. Index numbers begin at 0.
concat (<i>a_string</i>)	String	s2 = s.concat("rocks"); // s2 = "Javarocks"	Returns a new string with this string concatenated with <i>a_string</i> . You can use the + operator instead.
equals (<i>a_string</i>)	boolean	b = s.equals("Java"); // b = true	Returns true if this string and <i>a_string</i> are equal. Otherwise returns false.
equals IgnoreCase (<i>a_string</i>)	boolean	b = s.equalsIgnoreCase("Java"); // b = true	Returns true if this string and <i>a_string</i> are equal, considering upper and lower case versions of a letter to be the same. Otherwise returns false.
indexOf (<i>a_string</i>)	int	i = s.indexOf("va"); // i = 2	Returns the index of the first occurrence of the substring <i>a_string</i> within this string or -1 if <i>a_string</i> is not found. Index numbers begin at 0.
lastIndexOf (<i>a_string</i>)	int	i = s.lastIndexOf("a"); // i = 3	Returns the index of the last occurrence of the substring <i>a_string</i> within this string or -1 if <i>a_string</i> is not found. Index numbers begin at 0.
length()	int	i = s.length(); // i = 4	Returns the length of this string.
toLowerCase()	String	s2 = s.toLowerCase(); // s = "java"	Returns a new string having the same characters as this string, but with any uppercase letters converted to lowercase. This string is unchanged.
toUpperCase()	String	s2 = s.toUpperCase(); // s2 = "JAVA"	Returns a new string having the same characters as this string, but with any lowercase letters converted to uppercase. This string is unchanged.
replace (<i>oldchar</i> , <i>newchar</i>)	String	s2 = s.replace('a', 'o'); // s2 = "Jovo";	Returns a new string having the same characters as this string, but with each occurrence of <i>oldchar</i> replaced by <i>newchar</i> .
substring (<i>start</i>)	String	s2 = s.substring(2); // s2 = "va";	Returns a new string having the same characters as the substring that begins at index <i>start</i> through to the end of the string. Index numbers begin at 0.
substring (<i>start</i> , <i>end</i>)	String	s2 = s.substring(1, 3); // s2 = "av";	Returns a new string having the same characters as the substring that begins at index <i>start</i> through to but not including the character at index <i>end</i> . Index numbers begin at 0.