

String Object

The string object represents a text value and includes functions to test, manipulate and convert to other types.

Name and Parameters	Description	Return Type	Usage Example	
Append (string value)	Returns a new string with the specified string added to the end	String	<code>\$Utils.GetString("Build Config").Append(" v1.2.3")\$</code>	"Build Config v1.2.3"
Contains (string value [, bool ignoreCase])	Returns a value indicating whether the specified string occurs within the current text. Use ignoreCase parameter is false by default	Boolean	<code>\$Utils.GetString("Build Config v1.2.3").Contains("v1")\$</code>	True
Insert (int startIndex, string value)	Returns a new string in which a specified string is inserted at a specified index position in the current text	String	<code>\$Utils.GetString("Build Config v1.2.3").Insert(12, "uration")\$</code>	"Build Configuration v1.2.3"
Length	Returns the number of characters in the text	Integer	<code>\$Utils.GetString("Build Config v1.2.3").Length\$</code>	19
PadLeft (int totalWidth [, char paddingChar])	Returns a new string that right-aligns the characters in the current text by padding them on the left with a specified character, for a specified total length	String	<code>\$Utils.GetString("Build Config v1.2.3").PadLeft(3, "**")\$</code>	*****Build Config v1.2.3"
PadRight (int totalWidth [, char paddingChar])	Returns a new string that left-aligns the characters in the current text by padding them on the right with a specified character, for a specified total length	String	<code>\$Utils.GetString("Build Config v1.2.3").PadRight(3, "0")\$</code>	"Build Config v1.2.300000"
Remove (int startIndex [, int count])	Returns a new string in which a specified number of characters in the current text beginning at a specified position have been deleted	String	<code>\$Utils.GetString("Build Config v1.2.3").Remove(12)\$</code> <code>\$Utils.GetString("Build Config v1.2.3").Remove(5, 7)\$</code>	"Build Config" "Build v1.2.3"
Replace (string oldValue, string newValue)	Replaces all occurrences of the first specified string in the current string with the second specified string	String	<code>\$Configuration.Name.Replace("v.", "version")\$</code> <code>\$Utils.GetString("Build Config v1.2.3").Replace("Config", "Project")\$</code>	"Test Config version 10" (given "Test Config v. 10") "Build Project v1.2.3"
Replace (string pattern, string replacement, bool ignoreCase)	Returns a new string in which all matches of the specified regular expression pattern in the current text are replaced with a specified replacement string	String	<code>\$Utils.GetString("Build Config v1.2.3").Replace("\d", "9", true)\$</code>	"Build Config v9.9.9"
ReplaceNewlines ([string replacement])	Returns a new string in which all newlines in the current text are replaced with an optional specified string	String	<code>\$Utils.GetString("a\nb\nc").ReplaceNewlines()\$</code> <code>\$Utils.GetString("a\nb\nc").ReplaceNewlines(" ")\$</code>	"abc" "a b c"
Split (string separators, [bool trimWhitespace,] [bool removeEmptyEntries])	Splits a string into substrings separated by any of the characters in separator parameter	String Collection	<code>\$Utils.GetString("a, b, c").Split(",").Last()\$</code> <code>\$Utils.GetString("a, b, c").Split(",", True).Last()\$</code>	" c" "c"
SplitWithQuotes (string separators [, string quotetype] [, bool trimQuotes] [, bool trimWhitespace] [, bool removeEmptyEntries])	Splits a string into substrings separated by any of the characters in separator parameter. Separators within quotes are ignored. The quoteType parameter can be "single" or "double" and defaults to "double"	String Collection	<code>\$Utils.GetString(""" stage 1 ' , ' stage 2 ' , ' stage 3 """).SplitWithQuotes("'", "single", false, true).First()\$</code>	"" stage 1 ""
Substring (int startIndex [, int length])	Returns a substring of the current text starting at a specified character position with an optional specified length	String	<code>\$Utils.GetString("Build Config v1.2.3").Substring(5)\$</code> <code>\$Utils.GetString("Build Config v1.2.3").Substring(6, 6)\$</code>	"Config v1.2.3" "Config"
ToBoolean ()	Parses text to a boolean, allowing boolean operations and functions	Boolean	<code>\$Utils.GetString("True").ToBoolean().Not()\$</code>	False
ToLower ()	Converts a string to lowercase	String	<code>\$Utils.GetString("Build Config v1.2.3").ToLower()\$</code>	"build config v1.2.3"

ToNumber()	Parses text to a number, allowing numeric operations and functions	Number	\$Utils.GetString("999.999").ToNumber().Add(0.001)\$	1000.000
ToUpper()	Converts a string to uppercase	String	\$Utils.GetString("Build Config v1.2.3").ToUpper()\$	"BUILD CONFIG V1.2.3"
TrimStart ([string trimChars])	Returns a new string with all leading occurrences of white space or a specified set of characters removed	String	\$Utils.GetString(" Build Config v1.2.3 ").TrimStart()\$ \$Utils.GetString("****Build Config v1.2.3****").TrimStart("")\$	"Build Config v1.2.3 " "Build Config v1.2.3****"
Trim ([string trimChars])	Returns a new string with all leading and trailing occurrences of white space or a specified set of characters removed	String	\$Utils.GetString(" Build Config v1.2.3 ").Trim()\$ \$Utils.GetString("****Build Config v1.2.3****").Trim("")\$	"Build Config v1.2.3" "Build Config v1.2.3"
TrimEnd ([string trimChars])	Returns a new string with all trailing occurrences of white space or a specified set of characters removed	String	\$Utils.GetString(" Build Config v1.2.3 ").TrimEnd()\$ \$Utils.GetString("****Build Config v1.2.3****").TrimEnd("")\$	" Build Config v1.2.3" "****Build Config v1.2.3"