

String Operations

The Designer supports string manipulation operations. Internally, strings are represented as UTF-8 strings.

The following operations are offered by the **String** class:

- [capture\(\) Operation](#)
- [castValue\(\) Operation](#)
- [concat\(\) Operation](#)
- [convertBase64ToBlob\(\) Operation](#)
- [convertDurationToDateTIme\(\) Operation](#)
- [convertHexToBlob\(\) Operation](#)
- [convert.ToBoolean Operation\(\)](#)
- [convertToDateTIme\(\) Operation](#)
- [convertToFloat\(\) Operation for Strings](#)
- [convertToInteger\(\) Operation for Strings](#)
- [endsWith\(\) Operation](#)
- [escapeURI\(\) Operation](#)
- [extendedJSONToClass\(\) Operation](#)
- [findPattern\(\) Operation](#)
- [findPatterns\(\) Operation](#)
- [findString\(\) Operation](#)
- [jsonToClass\(\) Operation](#)
- [match\(\) Operation](#)
- [normalizeSpaces\(\) Operation](#)
- [padLeft\(\) Operation](#)
- [padRight\(\) Operation](#)
- [parseDateTimeExpression\(\) Operation](#)
- [parseFloatExpression\(\) Operation](#)
- [parseIntegerExpression\(\) Operation](#)
- [parseLocalDateTimeExpression\(\) Operation](#)
- [removeAccents\(\) Operation](#)
- [replace\(\) Operation](#)
- [split\(\) Operation](#)
- [startsWith\(\) Operation](#)
- [stringLength\(\) Operation](#)
- [substring\(\) Operation](#)
- [substringAfter\(\) Operation](#)
- [substringBefore\(\) Operation](#)
- [toASCII\(\) Operation](#)
- [toLower\(\) Operation](#)
- [toUpper\(\) Operation](#)
- [transcodeToBlob\(\) Operation](#)
- [transliterate\(\) Operation](#)
- [unescapeURI\(\) Operation](#)
- [xmlToClass\(\) Operation for Strings](#)

On this Page:

- [Escaping Characters](#)

Related Pages:

- [text\(\) Macro](#)
- [Available Base Types](#)
- [Overview on All Type Conversion Operations](#)

Escaping Characters

String literals are created by putting character sequences into single ('...') or double quotes ("..."). If the string literal contains double quotes, they can be used without escaping if the literal is enclosed by single quotes and vice versa:

- "myString'StringWithinQuotes'end"
- 'myString"StringWithinDoubleQuotes"end'

An alternative to handle quotes within a string is escaping the quotes with a backslash (\" or \'):

- "myString\"StringWithinDoubleQuotes\"end"
- 'myString\'StringWithinDoubleQuotes\'end'

Another backslash character is escaping the backslash character itself (\\).



If you want to use white spaces like tabs, new lines, or carriage returns in a string, use the [text\(\)](#) macro.