Working With the String Type

In this lesson, we get introduced to the string type in JavaScript.
Let's begin!

WE'LL COVER THE FOLLOWING

When you intend to work with text or characters, you should use the `String` type. This data type represents a sequence of **16-bit Unicode** characters, including the empty string with zero length.

JavaScript allows you to define strings enclosed between two single quotes (`'`) or two double quotes (`"`).

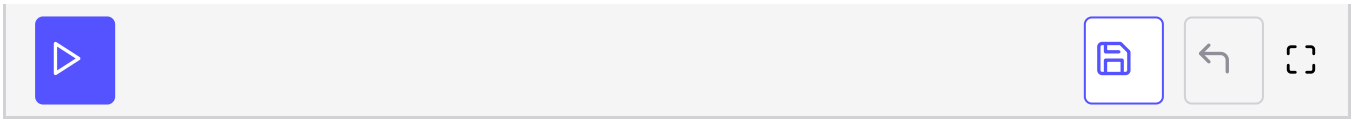
So, the following string declarations are all valid:

```
var name1 = 'Steve';
var name2 = "Jane";
```

Quotes must **match**, so these are invalid declarations.

> Hence, the following code will not compile ✕

```
var name1 = "Steve';
var name2 = 'Jane";
```



When you need to specify characters that are not available on the keyboard, or are non-printable, you can use character escape sequences that begin with a backslash (`\`) character.

The table below summarizes the available escape sequences:

Character escape sequences

Sequence	Description
`\n`	New line character (U+000A)
`\t`	Horizontal tab (U+0009)
`\v`	Vertical tab (U+000B)
`\b`	Backspace (U+0008)
`\r`	Carriage return (U+000D)
`\f`	Form feed (U+000A)
`\\`	Backslash (\)
`\'`	Single quote (')
`\"`	Double quote (")
`\0`	Null character (U+0000)
`\0ooo`	A character represented by octal code *ooo*, where *o* is an octal digit (0-7).
`\unnnn`	A Unicode character represented by the hexadecimal code *nnnn*, where n is a hexadecimal digit (0-F).
`\xnn`	A character represented by hexadecimal code *nn*, where *n* is a hexadecimal digit (0-F).

Examples

Let's see a few examples:

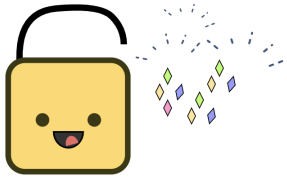
```
// I'm here.
var str1 = 'I\'m here.';
// Greek alpha: α
var str2 = "Greek alpha: \u03b1";
// 42
var str3 = "4\0o102";
// C:\Program Files\
var str4 = "C:\\Program Files\\";
```

If the string is very long, you can break it into multiple lines with a backslash character directly followed by a line terminator character:

```
// This is a long text broken into short lines
var str5 = "This is \
a long text \
broken into \
short lines";
```

Strings are **immutable**, so once they are created, their value cannot be changed.

Achievement unlocked!



Congratulations! You've learned how to deal with the string type in JavaScript.

Great work! Give yourself a round of applause! :)

In the *next lesson*, let's see how to convert other JavaScript types to string.