# Chapter 4 - Strings

Strings are used to store and manipulate text.
String can be created using the following syntax:

let name = "Harry" → Creates a string
name.length
     → This property prints length of the string

Strings can also be created using single quotes

let name = 'Harry'

## Template Literals
Template literals use backtics instead of quotes to define
a string

let name = `Harry`

With template literals, it is possible to use both
single as well as double quotes inside a string

                                             ↗ Single quote
let Sentence = ` The name "is" Harry's
            backtic                double quote

We can insert variable directly in template literal. This
is called string interpolation

let a = ` This is ${ name } ` → Prints `This is a Harry`
                        name is a
                        variable

Escape Sequence Characters

If you try to print the following string, JavaScript will "misunderstand" it

```
let name = 'Adam D'Angelo'
```

We can use single quote escape sequence to solve the problem

```
let name = 'Adam D\'Angelo'
```

Similarly we can use \" inside a string with double quotes

Other escape sequence characters are as follows

```
\n    → Newline
\t    → Tab
\r    → Carriage Return
```

String properties and Methods

1) 
```
let name = "Harry"
name.length    → prints 5
```

2) 
```
let name = "Harry"
name.toUpperCase ()    → prints HARRY
```

3) 
```
let name = "Harry"
name.toLowerCase()    → prints harry
```

4> let name = "Harry"

    name.slice (2, 4) ⟶ prints rr

                        (from 2 to 4, 4 not included)

5> let name = "Harry"

    name.slice (2) ⟶ prints rry

                        (from 2 to end)

6> let name = "Harry Bhai"

   let newName = name.replace ("Bhai", "Bhau")

7> let name1 = "Harry"

   let name2 = "Naman"

   let name3 = name1.concat (name2, "Yes")

               ↳ We can even use + operator

8> let name = " Harry "

   let newName = name.trim()

               ↳ Removes whitespaces

Strings are immutable. Inorder to access the character at an index we use the following syntax

    let name = "Harry"

    name [0] ⟶ Prints H

    name [1] ⟶ Prints a