

# Javascript Cadenas

---

## Declaración

Hay tres maneras de declarar cadenas en Javascript:

- Comilla Simple *Simple Quote*
- Comilla Doble *Double Quote*
- Comilla Invertida *Backtick* (a partir de ES6)

Declaración simple de una cadena:

```
const cad = "Esto es una cadena de caracteres"  
const cad = 'Esto también es una cadena de caracteres'
```

Declaración si la cadena tiene en su contenido comillas simple o dobles:

```
const cad = 'Esto es una "cadena" de caracteres' //Cadena con comillas dobles dentro, declarar con comillas simples  
const cad = 'M\'hija me dijo: "P\'aque haces eso papá"' //Podemos "escapar" las comillas internas con \
```

Declaración con backticks de una cadena conteniendo comillas simples y dobles:

```
const cad = `M'hija me dijo: "P'aque haces eso papá"``
```

# Javascript Cadenas

---

## Interpolación

Anteriormente para concatenar variables y cadenas, se tenía que hacer con el operador `+`

```
const nombre = "Pepito";
const edad = 8;
const cadena = "Mi nombre es " + nombre + " y tengo " + edad + " años";
```

Pero ahora podemos también *inyectar* variables en una cadena con backticks usando la sintaxis: `${nombreVariable}`

```
const nombre = `Pepito`;
const edad = 8;
const cadena = `Mi nombre es ${nombre} y tengo ${edad} años`;
```

De esta manera incluso podemos hacer operaciones matemáticas en las variables:

```
const nombre = `Pepito`;
const edad = 8;
const cadena = `Mi nombre es ${nombre} y tengo ${edad} años, el próximo año tendré ${edad+1} años`;
```

# Javascript Cadenas

---

## Metodos de las cadenas

En Javascript las cadenas se comportan como objetos, es decir, tienen métodos:

```
let cadenaEjemplo = "El que madruga Dios lo ayuda";

console.log(cadenaEjemplo.toUpperCase());
console.log(cadenaEjemplo.substring(15,cadenaEjemplo.length));
console.log(cadenaEjemplo.replace("Dios","Diosito"));
```