

# MODIFICANDO EL DOM

Para modificar el DOM debemos primero capturar el elemento sobre el cual vamos a trabajar.

Luego podremos modificar su apariencia (estilo), cambiar su contenido, agregar o quitar clases.



# MODIFICANDO ESTILOS

## Propiedad `style`

La propiedad **`style`** permite leer y sobrescribir el estilo de un elemento.

Javascript no conoce de la existencia de archivos CSS por lo tanto todo el estilo aplicado usando la propiedad **`style`** se verá dentro del html como estilo en línea.

```
let titulo = document.querySelector(".title");  
  
{} titulo.style.color = "cyan";  
   titulo.style.fontSize = "12px";  
   titulo.style.backgroundColor = "#dddddd";
```



Las reglas CSS que llevan guiones (como font-size) en Javascript se escriben ahora en camelCase.



# MODIFICANDO CONTENIDO

## innerText

Retorna el texto dentro de un elemento html y permite modificarlo.

***innerText*** no reconoce estructuras html internas del elemento.

## innerHTML

Retorna la estructura html de un elemento y permite modificarla.

***innerHTML*** reconoce estructuras html internas del elemento.

# MODIFICANDO EL DOM - CONTENIDO

Dada la siguiente estructura HTML

TUESDAY, MAY 18 @ 2:30 PM GMT-3

## GitHub Automation with Octokit

GitHub has powerful APIs, and with Octokit we can do just about anything on GitHub through code. Gregor Martynus will teach us how!

```
▼<div class="episode-preview-details">  
  <p class="gradient-subheading">...</p>  
  <h3>...</h3>  
  <p class="episode-description">...</p>  
  <div class="episode-links">...</div> flex  
</div>
```

**innerText** retorna

"TUESDAY, MAY 18 @ 2:30 PM GMT-3\n\nGitHub Automation with Octokit\n\nGitHub has powerful APIs, and with Octokit we can do just about anything on GitHub through code. Gregor Martynus will teach us how!"

**innerHTML** retorna

"<p class=\"gradient-subheading\">Tuesday, May 18 @ 2:30 PM GMT-3</p><h3><a href=\"/github-automation-with-octokit\">GitHub Automation with Octokit</a></h3><p class=\"episode-description\">GitHub has powerful APIs, and with Octokit we can do just about ..."

## innerText

Reemplazamos el texto de un elemento.



```
<h3>Este es el título original del artículo</h3>  
document.querySelector("h3").innerText = "Nuevo texto del título";
```

## innerHTML

Reemplazamos la estructura interna de un elemento aplicando html.



```
<p>Hola mundo</p>  
document.querySelector("p").innerHTML = "<p>Hola <strong>mundo</strong></p>";
```



Para agregar contenido manteniendo el original recordá que podés usar el operador de asignación "+="





# MANIPULANDO CLASES

## classList

La propiedad **classList** de un elemento retorna la cantidad de clases que posee y permite: agregar, eliminar o alternarlas.

- elemento.classList.add()
- elemento.classList.remove()
- elemento.classList.toggle()

`.classList.add()`

`element.classList.add()` permite agregar clases a un elemento html.

Las clases a agregar deben pasarse como parámetros dentro de los paréntesis.

```
// html
```

```
<ul> ... </ul>
```

```
// javascript
```

```
let lista = document.querySelector('ul');
```

```
lista.classList.add('nueva-clase');
```

```
// html
```

```
<ul class="nueva-clase"> ... </ul>
```

## .classList.remove()

**element.classList.remove()** permite eliminar clases de un elemento html.

La clase a eliminar debe pasarse como parámetros dentro de los paréntesis.

```
// html
```

```
<ul class="clase-base nueva-clase"> ... </ul>
```

```
// javascript
```

```
let lista = document.querySelector('ul');
```

```
lista.classList.remove('nueva-clase');
```

```
// html
```

```
<ul class="clase-base"> ... </ul>
```

## .classList.toggle()

`element.classList.toggle()` permite alternar clases de un elemento html.

Si la clase **no está** `toggle()` la **agrega** y si la clase **está presente** `toggle()` la **elimina**.

```
// html
```

```
<ul class="clase-base nueva-clase"> ... </ul>
```

```
// javascript
```

```
let lista = document.querySelector('ul');
```

```
lista.classList.toggle('nueva-clase');
```

```
// html
```

```
<ul class="clase-base"> ... </ul>
```

# GRACIAS