

Modificando el DOM

Índice

1. [.innerHTML](#)
2. [.innerText](#)
3. [Propiedad style](#)

“

Para poder hacer modificaciones al **DOM** siempre tenemos que tener seleccionado el objeto que queremos modificar. Esto lo podemos hacer usando selectores.



”

1 | .innerHTML

.innerHTML

Si queremos **leer o modificar** el contenido de una etiqueta HTML, vamos a utilizar esta propiedad.

```
{ } document.querySelector('div.nombre').innerHTML ;
```

En este caso, la utilizamos para leer el contenido, pero ¿qué pasa si queremos modificarlo?



Si queremos **guardar** el valor, debemos asignar esa línea de código a una variable. De otra manera, cuando la ejecución continúe, se perderá el valor que hayamos buscado.

.innerHTML

Para **modificar** el contenido de una etiqueta HTML, vamos a utilizar esta propiedad de la siguiente manera.

```
{} document.querySelector('div.nombre').innerHTML = 'Darío';
```

Si utilizamos la propiedad de esta forma, todo el contenido que teníamos en el **div** con clase **nombre** se va a cambiar por el string “Darío”.



Sin embargo, también podríamos modificar el contenido sin perder lo que teníamos anteriormente...

.innerHTML

Para **modificar** el contenido de una etiqueta HTML, sin perder el contenido que ya estaba, vamos a utilizar esta propiedad de la siguiente manera:

```
{ } document.querySelector('div.nombre').innerHTML += 'Papitas';
```

De esta forma, estamos agregando al **div** con clase **compras** la palabra "Papitas". De tal manera que si lo leyéramos, diría:

```
{ } <div class="compras"> “Jamón, Queso, Pan” Papitas</div>
```

Contenido que ya estaba.

Contenido que agregamos.

2 | .innerText

.innerText

Si queremos **leer o modificar** el texto de una etiqueta HTML, vamos a utilizar esta propiedad.

```
{} document.querySelector('div.nombre').innerText ;
```

En este caso, si en mi **div** con clase **nombre** estuviera escrito “Leo”, la propiedad me retornaría el **string** “Leo”.



Si queremos **guardar** el valor, debemos asignar esa línea de código a una variable. De otra manera, cuando la ejecución continúe, se perderá el valor que hayamos buscado.

.innerText

Si queremos **modificar** el texto de una etiqueta HTML, vamos a utilizar esta propiedad de la siguiente manera:

```
{} document.querySelector('div.nombre').innerText = 'Maria';
```

Si utilizamos la propiedad de esta forma, todo el texto que teníamos en el **div** con clase **nombre** se va a cambiar por el string “Maria”.



Sin embargo, también podríamos modificar el contenido sin perder lo que teníamos anteriormente...

.innerText

Si queremos **agregar** contenido al texto de una etiqueta HTML, vamos a utilizar esta propiedad de la siguiente manera:

```
{} document.querySelector('div.nombre').innerText += 'Messi';
```

En este caso, lo que sucedería es similar a lo que sucede con el otro selector, pero el texto se incluiría dentro de la etiqueta div, quedando:

```
{} <div class="nombre">Leo Messi</div>
```



Contenido que ya estaba.



Contenido que agregamos.

3 | Propiedad style

Propiedad **Style**

Nos permite leer y sobreescribir las reglas CSS que se aplican sobre un elemento que hayamos seleccionado.

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



Nótese que las reglas CSS que llevaban guiones (como **font-size**), en JavaScript se escriben en camelCase (es decir: **fontSize**).

DigitalHouse>
Coding School