

### Guía Teórica N°03

#### ¡Atributos Globales, Estilos y más!

Ya sabiendo cómo seleccionar los diferentes nodos que conforman al Modelo de Objetos del Documento o simplemente conocido como DOM, es turno de manipular aquellos atributos y métodos de los diferentes elementos que vayamos a utilizar en el ciclo de desarrollo de nuestra página web.

Esta guía englobará temas de diferentes clases para simplificar la cantidad de guías teóricas a ofrecer. Si hay algún tema que no llegues a comprender, no te preocupes, en próximas clases tus dudas pueden llegar a ser resueltas y si en dado caso siguen persistiendo, recuerda que puedes contactar directamente con nosotros al correo electrónico [educa2ucv@gmail.com](mailto:educa2ucv@gmail.com) o escribiarnos a través de algunos de los medios de comunicación que te hemos facilitado (Facebook, Twitter, Instagram, WhatsApp, Telegram o Discord).

#### Métodos para definir, obtener y eliminar valores de atributos:

A continuación, vamos a presentar algunos métodos que nos permitirán manipular los atributos definidos en las etiquetas HTML de nuestro documento. Algunos ejemplos de estos atributos pueden ser como **class**, **id**, **name**, **placeholder**, **required**, **style**, entre otros.

- `setAttribute()`: Nos permite indicarle un nuevo valor a un atributo ya definido o que no se encuentra aún definido a un elemento HTML.
- `getAttribute()`: Obtenemos el valor del atributo indicado como parámetro de entrada al invocar el método.

- `removeAttribute()`: Nos da la posibilidad de eliminar un atributo definido con anterioridad a un elemento HTML.

## Atributos Globales:

Hay atributos que son comunes entre todos los elementos pertenecientes al DOM y JavaScript nos permite acceder a estos atributos como si fuese un objeto perteneciente directamente a una clase (recordemos del curso de Programación en JavaScript, que la definición de **objeto** no existe como tal, sino que nos basamos de prototipos que van siendo desarrollados a partir de la herencia). Veamos algunos de estos atributos que podemos manipular directamente:

- `contenteditable`: Este atributo puede tener solamente dos valores, **true** o **false**, y nos permite dar la posibilidad de si un elemento HTML, por ejemplo, una etiqueta **p**, el contenido de esta pueda ser modificada directamente por parte del usuario al ingresar a la página.
- `dir`: Recibe los valores **ltr** o **rtl**, los cuales indican la dirección de escritura.
- `hidden`: Al igual que el atributo **contenteditable**, recibe únicamente valores booleanos y nos permite indicar si un elemento estará visible o no en nuestra página.
- `tabindex`: Recibe un valor numérico entero positivo y este se verá observado cuando el usuario haga **tabulación** y se seleccionen diferentes elementos HTML.

## Atributos de elementos inputs:

Los elementos inputs que utilizamos para diseñar nuestros formularios también comparten varios atributos en común que pueden llegar a ser de mucha utilidad si conocemos los más importantes (o más utilizados al desarrollar). Veamos qué atributos podemos manipular de estos elementos:

- `className`: Nos permite definir a qué clase pertenecerá un elemento input.

- **value:** Nos da la posibilidad de definir un valor u obtener el valor de un correspondiente input de entrada. Muy utilizado para cuando el usuario activa el evento ***submit***.
- **type:** Este atributo nos permite indicar el tipo de input que utilizaremos para el formulario a diseñar, por ejemplo, ***text, color, date, password***, entre otros.
- **minlength:** Define la cantidad mínima de caracteres que debe tener un input al momento de realizar el envío del formulario.
- **required:** Indica si un elemento de entrada es obligatorio o no de rellenar.
- **placeholder:** A diferencia del atributo ***value***, este nos permite definir un pequeño mensaje informativo al usuario para indicar que valor debe ingresar dentro de un elemento input.

## Atributo Style:

A través de JavaScript podemos definirle algunos estilos a diferentes elementos según alguna acción que active el usuario mientras se encuentra navegando a través de nuestra página web. Cabe mencionar que este atributo es utilizado para ocasiones específicas y debemos recordar que a través de los pseudoelementos y pseudoclasas podemos definir atributos especiales según alguna acción generada.

La sintaxis para hacer uso de este atributo es la siguiente:

```
element.style.propiedadCss = 'valor de propiedad CSS';
```

## Clases, classList y sus Métodos:

Sabemos que las clases definidas para un elemento HTML son un atributo y podemos acceder a este atributo con algunos métodos mencionados anteriormente, pero JavaScript nos ofrece una manera más adecuada cuando deseamos manipular las clases pertenecientes de un elemento. Esto a través de la ***classList*** y sus métodos.

- `add()`: Nos permite agregarle una nueva clase a un elemento HTML y el valor que tendrá directamente esta clase añadida.
- `remove()`: Caso contrario al método **`add()`**, este nos permite eliminar la clase que le indiquemos como parámetro de entrada al elemento que le definamos.
- `item()`: Las `classList` la podemos ver como un arreglo normal de JavaScript donde cada clase está posicionada en un índice definido, a través de este método el valor numérico que indiquemos (entero positivo), accederemos a esa clase en particular.
- `contains()`: Nos permite saber si un elemento HTML pertenece o no a una clase indicada.
- `replace()`: Nos da la posibilidad de reemplazar una clase ya definida con una nueva clase, es decir, cambiar el nombre de una clase por otra.
- `toggle()`: Si la clase indicada se encuentra definida en el elemento HTML, esta será eliminada, en caso contrario, la definirá.

## ¡Nos vemos en la próxima!

Wow... Que gran cantidad de teoría, ¿no crees?

No te preocupes, en los vídeos explicativos hacemos una excelente cantidad de ejemplos que nos ayudarán a entender muy bien muchas de las cosas explicadas en esta guía, así que no entres en colapso, esto te será de ayuda cuando no recuerdes exactamente qué hacía un atributo o método.