Ayudantía I: JavaScript

Juan Fernández
Manuel Muñoz

Contenidos

01 DOM y Javascript

02 Eventos del navegador

03 localStorage

04 Ejercicio

¿Qué es el DOM?
(Document Object Model)

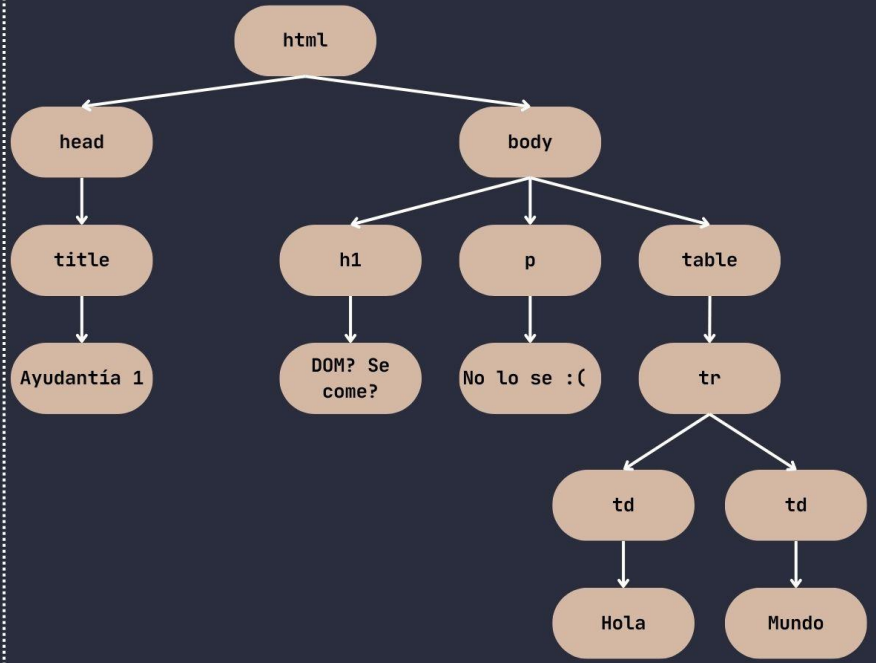
Corresponde a una representación jerárquica estructurada en forma de árbol de un documento HTML[1].

1. También representa documentos XML y XHTML.

HTML

```
<html>
    <head>
        <title>Ayudantía 1</title>
    </head>
    <body>
        <h1>DOM? Se come?</h1>
        <p>No lo se :(</p>
        <table>
            <tr>
                <td>Hola</td>
                <td>Mundo</td>
            </tr>
        </table>
    </body>
</html>
```

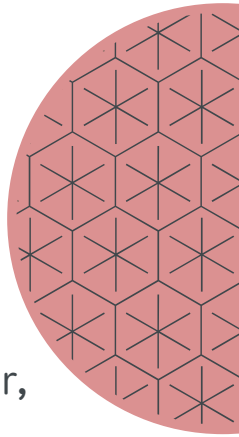
Diagrama del DOM



HTML DOM desde JavaScript

El HTML DOM es un **modelo de objetos estandarizado** y una **interfaz de programación** para HTML, la cual define:

- Los elementos HTML como objetos
- Las propiedades de todos los elementos HTML
- Los métodos para acceder a todos los elementos HTML
- Los eventos para todos los elementos HTML



De esta forma HTML DOM establece un estándar sobre cómo obtener, cambiar, agregar o eliminar elementos HTML.

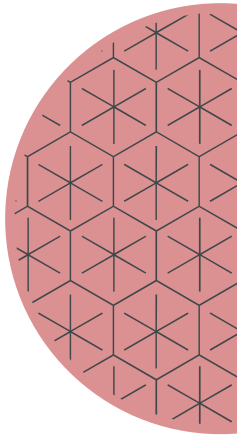
- JavaScript puede hacer consultas y cambios sobre el DOM

Nodos vs Elementos

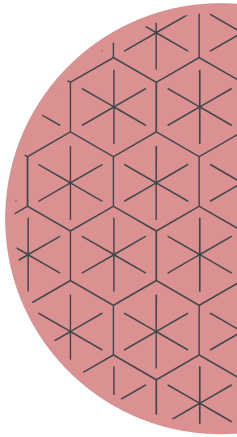
Un nodo es cualquier objeto del DOM, pudiendo ser el documento mismo (document), tag HTML, texto e incluso los comentarios.

Por otro lado, un elemento en el DOM es específicamente una etiqueta HTML.

Veremos un ejemplo más adelante 😉



¿Cómo accedemos a los nodos?

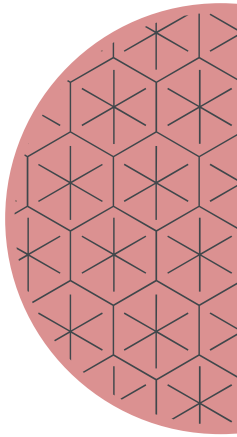


HTML DOM Document

El HTML DOM Document corresponde a la representación completa de una página web, en donde se encuentran contenidos todos los objetos de la web.

A partir de este elemento se puede acceder a los nodos del DOM 😱, y se seguirá la siguiente estructura para utilizar los métodos del DOM:

document.<método>

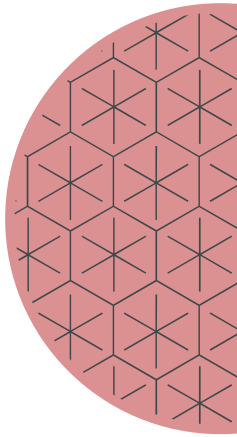


Según su id

Para acceder a un objeto del DOM según su id hacemos uso del método →**getElementById**←

Ingresemos a la página de Ingeniería y probemos esto 🤩

https://www.ing.uc.cl/



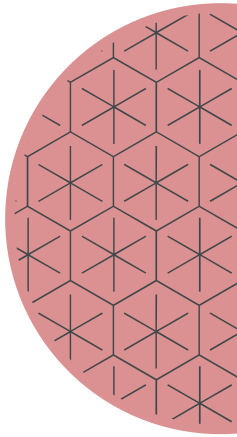
Según su tag HTML

Para acceder a un objeto del DOM según el tag (p, h, etc) hacemos uso del método

→**getElementsByTagName**←

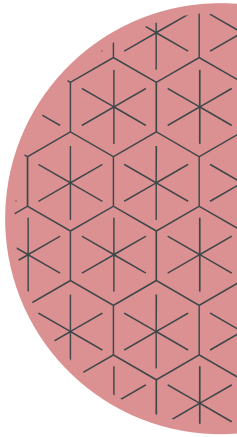
Ingresemos a la página de Ingeniería y probemos esto 🤩

https://www.ing.uc.cl/



Según su clase CSS

Para acceder a un objeto del DOM según su clase hacemos uso del método
→**getElementsByClassName**←



Ingresemos a la página de Ingeniería y probemos esto 🤩

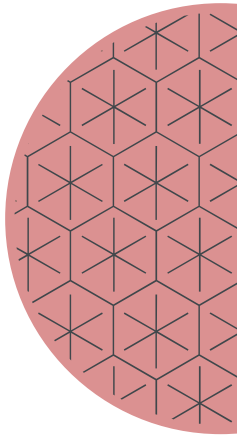
https://www.ing.uc.cl/

Según su nombre

El acceso a objetos del DOM según su nombre se utiliza cuando se trabaja con forms para así distinguir los input. El método asociado es →**getElementsByName**←

Ingresemos a la página de Ingeniería y probemos esto 🤩

https://www.ing.uc.cl/

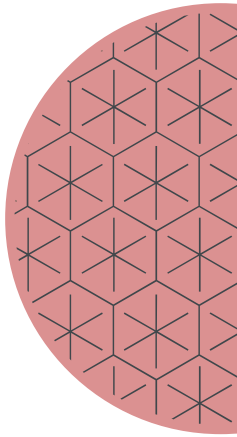


Algunas propiedades de los nodos/elementos

👀 Ojo, las propiedades se aplican sobre objetos del DOM (nodos o elementos).

Identidad

- nodeName: Indica el tipo de elemento del nodo

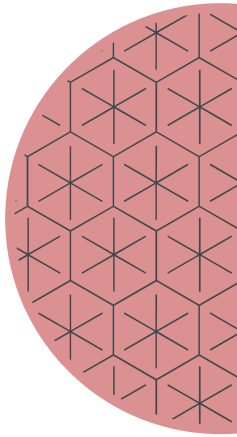


Algunas propiedades de los nodos/elementos

Jerarquía estructural:

- parentNode
- childNodes
- firstChild, lastChild
- firstElementChild, lastElementChild

Pueden encontrar más propiedades en el siguiente link.



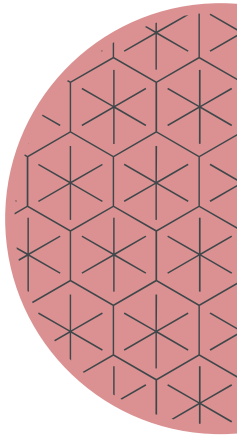
Algunos métodos de los nodos/elementos

Relacionados con el contenido:

- textContent: Entrega un string con el contenido de todos los elementos.
- innerText: Entrega un string con el contenido “leíble”
- innerHtml: Retorna el HTML que está dentro de un elemento
- getAttribute(<atributo>)/setAttribute(<atributo>)

Ingresemos a la página de Ingeniería y probemos esto 🤩

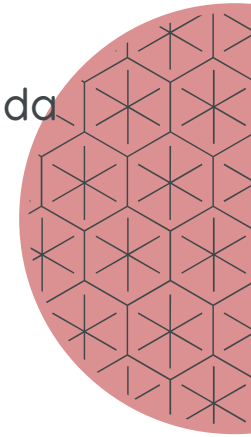
https://www.ing.uc.cl/



Algunos métodos de los nodos/elementos

Relacionados con el estilo:

- classList.add(<nombre_clase>): Agrega una clase con el nombre indicado en el parámetro del método.
- style.<propiedad_css>: Permite modificar la propiedad especificada para el objeto sobre el que se ejecuta el método.
- querySelector(): Retorna el primer elemento que coincide con el selector.



Ingresemos a la página de Ingeniería y probemos esto 🤩

https://www.ing.uc.cl/

¿Qué son los Eventos en el navegador?🤔

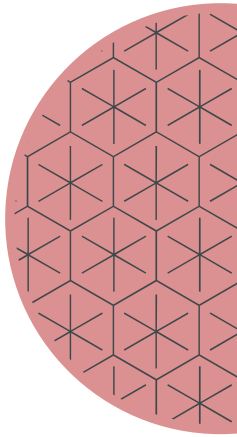
Corresponden a acciones que ocurren en el navegador web, como click, pulsaciones de teclas, entre otros.

Estos eventos permiten **dinamizar** la aplicación web.

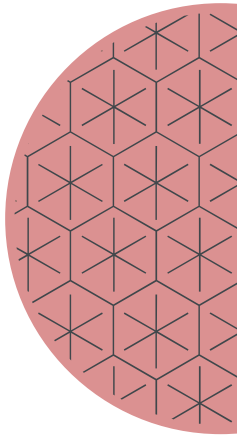
Tipos de eventos

Los principales eventos son los siguientes, sin embargo, existen muchos más.

- **Asociados al mouse:** selección, clicks, movimientos, hover.
- **Asociados al teclado:** pulsación de teclas.
- **Asociados a tiempo.**
- **Cambios en un input, submit de un form.**



¿Cómo manejamos los eventos?



Event listener

Corresponde a la forma más recomendada de capturar eventos. Para esto se hace uso del método **addEventListener**.

Este método adjunta un controlador de eventos al **elemento** sobre el cual se aplica el método.

La estructura de uso es:

```
elemento.addEventListener(evento, función)
```

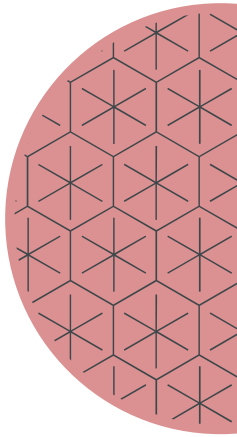
Ingresemos al siguiente link 💻 y probemos esto

HTML Event

De esta forma se puede asignar directamente las funciones de manejo de eventos como un atributo HTML, el cual corresponde al evento HTML que se quiere atrapar.

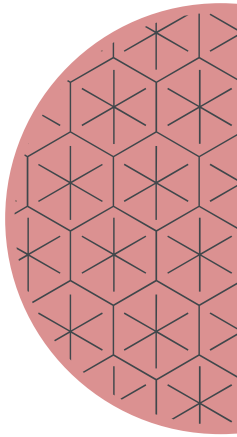
Para utilizarlo, solo se debe agregar un atributo con el nombre del evento al HTML, junto con la función que se quiere ejecutar cuando esto ocurra.

Ingresemos al siguiente link 💻 y probemos esto.



Almacenamiento de datos: localStorage

localStorage es un objeto de almacenamiento que nos permite guardar pares clave-valor, lo cual nos da la posibilidad de almacenar información en el navegador incluso después de su cierre.



```
localStorage.valorImportante = 'Un valor';  // Manera 1
localStorage.setItem('valorImportante', 'Un valor')   // Manera 2


localStorage.getItem("valorImportante")
```

Output: ("Un valor");

Actividad de práctica

Pongamos a prueba lo que hemos aprendido con una pequeña actividad. Deberás agregar el comportamiento a una página para que se puedan agregar actividades enumeradas a una lista y cambiar el color de fondo mediante botones y eventos.