

JavaScript en front-end

**Carrera
Programador
full-stack**

JavaScript: Origen

- Se popularizó con DHTML (páginas dinámicas).
- La web dejó de ser un conjunto de markup documents, para tener comportamiento de acuerdo al usuario.
- $\text{DHTML} = \text{HTML} + \text{CSS} + \text{JavaScript} + \text{DOM}$



JavaScript: Presente

- Es uno de los lenguajes de programación más utilizados
- Hay un intérprete en cada máquina (Navegadores)
- Principal uso, del lado del cliente
- Por su popularidad se ha extendido a otras aplicaciones y entornos
- Existe también del lado del servidor (Node.js)

JavaScript: ¿Para que?

Algunos ejemplos:

- Validar formularios (lo que no permita HTML5)
- Reaccionar a lo que haga el usuario (click, teclear, etc)
- Cambiar algo al pasar el mouse (si no lo permite CSS)
- Partes de páginas que se muestran/ocultan
- Hacer cálculos complejos

JavaScript y TypeScript

- El navegador entiende solo JavaScript
- Transpilar es el proceso de interpretar ciertos lenguajes de programación y traducirlos a un idioma específico. En este contexto, el transpiling tomará los idiomas de compilación a JS y los traducirá al idioma de destino de JavaScript
- TypeScript se transpila a JavaScript
- Son similares, pero JavaScript es SIN tipos estáticos.
- Vamos a usar JS en dos ejemplos, para entender mejor cómo funciona el navegador.

JavaScript y TypeScript

```
let num1 = 5;  
let num2 = 6;  
let nombre = "Jhon Doe";  
let variableBooleana = true;  
  
function sumarDosNumeros(numero1, numero2) {  
    console.log(numero1 + numero2);  
}  
  
sumarDosNumeros(num1, num2);
```

```
let num1: number = 5;  
let num2: number = 6;  
let nombre: string = "Jhon Doe";  
let variableBooleana: boolean = true;  
  
function sumarDosNumeros(numero1: number, numero2: number): void {  
    console.log(numero1 + numero2);  
}  
  
sumarDosNumeros(num1, num2);
```

Ejemplo 1

Carrera Programador full-stack

Mensaje al usuario

Hacer una página que salude al usuario al entrar

Mostrar un mensaje saludando al usuario al entrar a la página


¿Qué vamos a aprender?

- Incluir un archivo JavaScript y ejecutarlo
- Mostrar un mensaje en la consola del navegador

Como incluir un JavaScript

- Conviene incluir un archivo JavaScript separado
- **Se ejecuta su código en la línea donde se incluye**

```
<!DOCTYPE html>
<html>
...
<body>
    ....
    <script type="text/javascript" src="js/main.js"></script>
</body>
</html>
```



- Incluirlo al final del body, luego de que ya se cargo el html con **todos** sus elementos.
- **Se pueden agregar varios archivos .js**

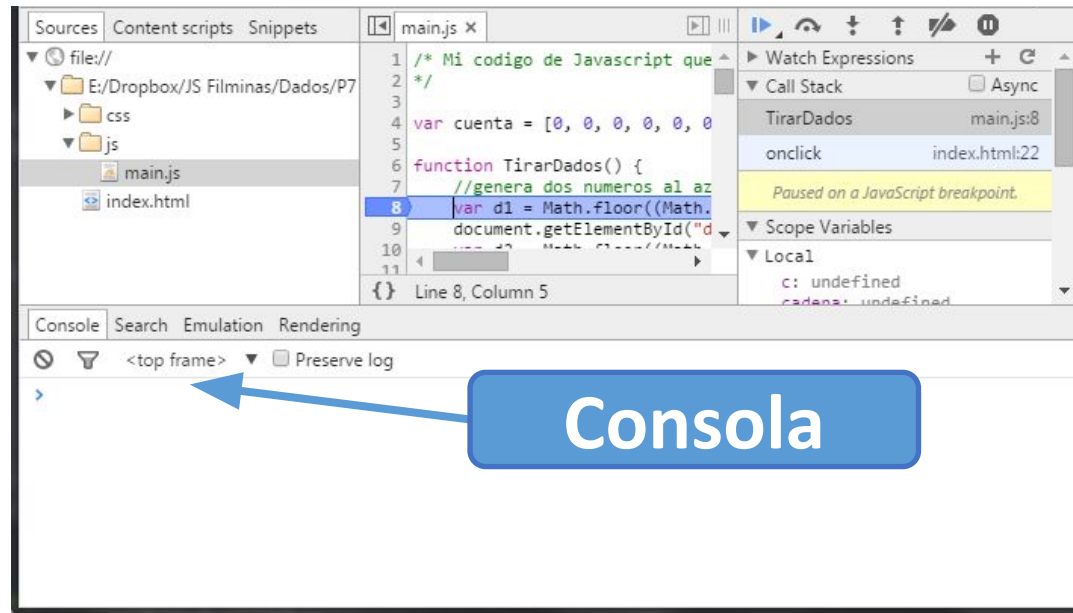


Ver la consola

Menú Chrome > Más herramientas > Herramientas Desarrollador

Atajos de teclado y mouse:

- Ctrl + Shift + I
- F12
- click derecho > inspeccionar

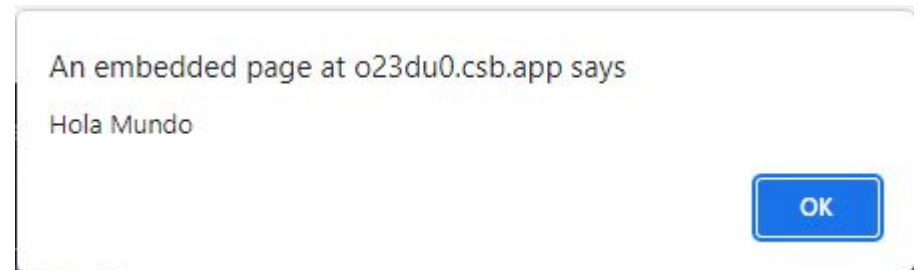
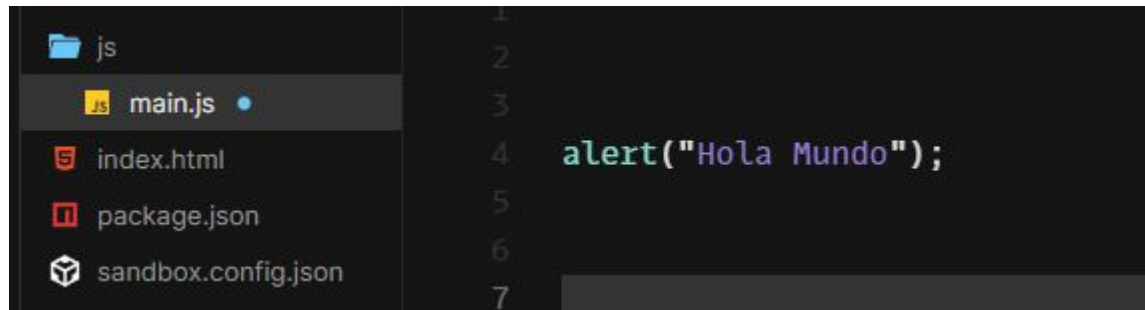


Funcion Alert

- La función alert nos muestra una alerta en nuestro navegador
- La forma de usarla es:

```
alert([mensaje])
```

- No se suele usar en páginas reales, ya que no se integra visualmente con el resto del sitio



Botón para saludar

**Carrera
Programador
full-stack**

Resolver el problema

¿Qué vamos a aprender?

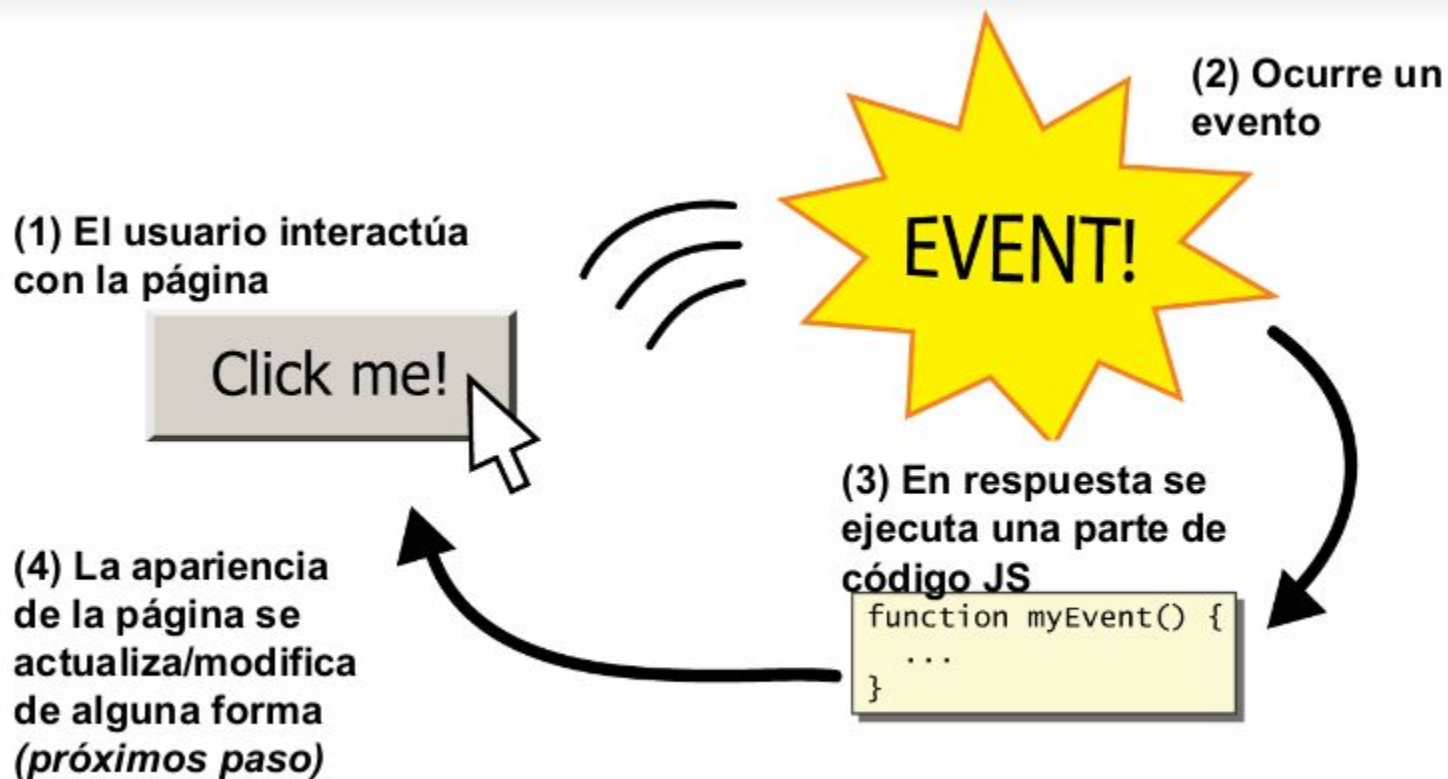
- Ejecutar un código al hacer click en un botón
 - Esto se llama “al pasar un **evento**”
- Para eso necesitamos darle un nombre a una parte del código
 - Esto se llama “declarar una **función**”

“Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento.”

Eventos

- Un evento es algo que ocurre en el sistema, originado por el usuario u otra parte del sistema y que se avisa al sistema.
- Ejemplos:
 - El usuario hace click.
 - Se terminó de cargar la página.
 - Pasó un segundo desde que se terminó de procesar.
- Las interfaces gráficas suelen programarse orientada a eventos.

Programación dirigida por eventos



Eventos

- Los eventos son capturados por manejadores (handlers).

Evento

- **HTML**

```
<button onclick="saludar()">Saludar</button>
```

- **JavaScript**

Función
(handler o
callback)

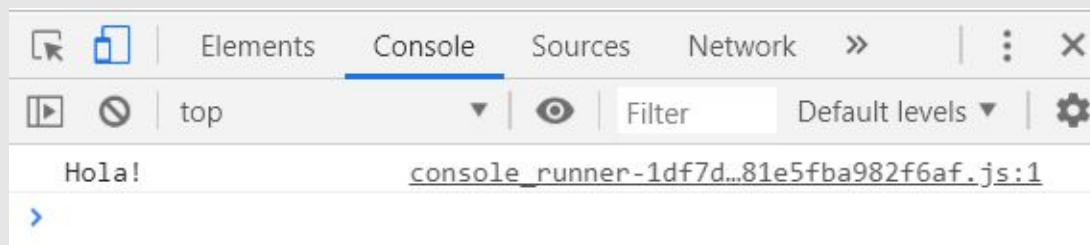
```
function saludar() {  
  console.log("Hola!");  
}
```

No recomendado
(por ahora se hace así)



Resultado

Saludar!



Podes ver el ejemplo aca:

<https://codepen.io/webUnicen/pen/VXWbWL>

Eventos

Ejemplos de eventos:

- [onclick](#)
- [onkeydown](#)
- [onload](#)
- [onfocus](#)
- [onchange](#) (para inputs)
- [ondrag](#)
- [oncopy](#)
- [onpause](#)(para media)

Hay 50~100 eventos:

http://www.w3schools.com/jsref/dom_obj_event.asp

Actividad en clase

Elegir 3 eventos de la siguiente lista, investigar cómo funcionan y hacer un sitio web utilizándolos

- [onclick](#)
- [onkeydown](#)
- [onload](#)
- [onfocus](#)
- [onchange](#) (para inputs)
- [ondrag](#)
- [oncopy](#)
- [onpause](#)(para media)
- Algun otro que les resulte interesante de usar

Id

```
<button type="submit" id="botonSaludar">Saludar</button>
```

```
let botonSaludar = document.getElementById("botonSaludar");  
botonSaludar.addEventListener("click", saludar);
```

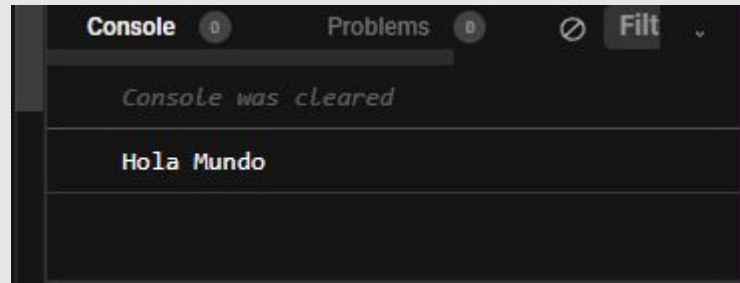
Vinculación del
elemento html
con variable en
ts y asignacion
de evento

```
function saludar(): void {  
  console.log("Hola Mundo");  
}
```



Resultado

Saludar!



Podes ver el ejemplo aca:

<https://codepen.io/webUnicen/pen/VXWbWL>

Contador de clicks

Carrera
Programador
full-stack

Resolver el problema

¿Qué vamos a aprender?

- Cómo recordar cosas
 - Vamos a usar una **variable** para contar
- Como concatenar strings

Variables y Constantes

Variables:

- Una variable es un nombre que le damos a un valor que puede cambiar (o no) con el tiempo (durante la ejecución del programa)
- El nombre no es el contenido, es como llamamos a ese valor, pero sin saber el valor exacto mientras escribimos

Constantes:

- Son un nombre que le damos a un valor
- Nunca cambia con el tiempo
- Se usan para aumentar la legibilidad del programa

Solución

Vamos a declarar una variable donde llevemos la cuenta de los clicks

```
let contador: number = 0
```

cada vez que el usuario hace click vamos a incrementar el valor del contador en 1

```
contador = contador + 1
```

que también se puede escribir como (abreviación para + 1)

```
contador++
```

Clicker!

```
<div class="container">
  <h1>Bienvenido</h1>
  <p>Llegaste a nuestro contador de click!</p>
  <button id="clickear">Contar click!</button>
</div>
<script src="src/index.js"></script>
```

```
let contador = 0;
function clickear() {
  //incrementa el valor de contador
  contador = contador + 1;
  //forma corta: contador++
  console.log("Hiciste " + contador + " clicks");
}
let botonContar = document.getElementById("clickear");
botonContar.addEventListener("click", clickear);
```

Use Strict

- Es una buena práctica escribir al comenzar un archivo JavaScript/TypeScript

`“use strict”;`

- Convierte en obligatoria la declaración de variables
- Restringe otros posibles errores de sintaxis

Clicker!

```
<div class="container">
  <h1>Bienvenido</h1>
  <p>
    Llegaste a nuestro contador de click!
  </p>
  <button onclick="clickear()">Contar click!</button>
</div>
<script type="text/JavaScript"
src="js/main.js"></script>
```



"use strict";

```
console.log("declarando funciones");
let contador = 0;
function clickear() {
  //incrementa el valor de contador
  contador++;
  console.log("Hiciste " + contador + " clicks");
  //es lo mismo que contador = contador + 1
}
```

DEMO

Debug

**Carrera
Programador
full-stack**

Hacer una página que salude al usuario al entrar

Mostrar un mensaje saludando al usuario al entrar a la página

¿Qué vamos a aprender?

- Incluir un archivo JavaScript y ejecutarlo
- Mostrar un cartelito por pantalla

Función Alert

- La función alert nos muestra una alerta en nuestro navegador
- La forma de usarla es:

```
alert([mensaje])
```

- No se suele usar en páginas reales, ya que no se integra visualmente con el resto del sitio

Buscá los cartelitos de Demo:

Recordá que en Codepen están todas las soluciones para experimentar, son lo mismo que hacemos en clase!



<http://codepen.io/webUnicen/pen/eZMvzo>

Pregunta

El código JavaScript incluido se ejecuta automáticamente al cargar la página

¿Qué pasa si hay dos alert?

1. Se muestran los dos mensajes
2. Se muestra uno y al aceptarlo recién se muestra el segundo

```
/* Mi codigo inicial de Javascript  
muestra un alert para comprobar que el codigo se esta ejecutando.  
*/  
alert("HOLA USUARIO!");
```


Resolver el problema

¿Qué vamos a aprender?

- Ejecutar un código al hacer click en un botón
 - Esto se llama “al pasar un **evento**”
- Para eso necesitamos declarar una **función**

Analizar el orden en la consola

```
"use strict";  
console.log("Paso 1: declarando funciones");  
let contador = 0;  
  
function clickear(){  
    //incrementa el valor de contador  
    console.log("Paso 3: Valor anterior del contador:" +  
contador);  
    contador++;  
    console.log("Paso 4: El contador ahora vale:" + contador);  
    alert("Hiciste " + contador + " clicks")  
    //es lo mismo que contador = contador + 1  
}  
console.log("Paso 2: continua ejecución");
```

A bright orange starburst graphic with a white border, containing the word "DEMO" in white capital letters.
DEMO

Saludo con nombre

Carrera
Programador
full-stack

Saludar

Consigna:

- Un lugar para escribir en la página web. A medida que escribo mi nombre la página me dice “Bienvenido {NOMBRE}”.
 1. Bienvenido J
 2. Bienvenido Ja
 3. Bienvenido Jav
 4. Bienvenido Javi
- En la consola mostrar el largo del nombre

¿Qué vamos a aprender?

- Usar otro evento (que no es onclick)
- Editar la página web desde JavaScript
- Calcular el largo de una cadena

Largo de una cadena

Existen muchas funciones que ya trae JavaScript
Para calcular el largo de una cadena puedo usar:

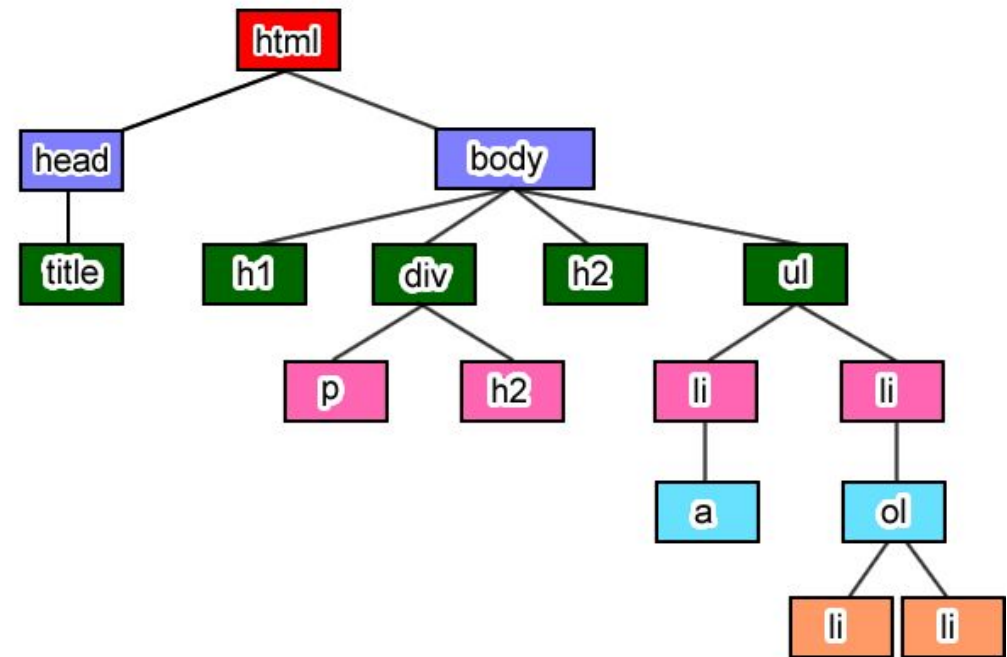
```
let cadena = "hola como estas?";  
let largo = cadena.length;  
console.log(largo);  
//Imprime 16
```

El valor calculado se **devuelve** y debe guardarse en una variable

Arbol HTML - DOM

Una manera de comprender las dependencias y relaciones entre elementos es mediante un diagrama de árbol.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



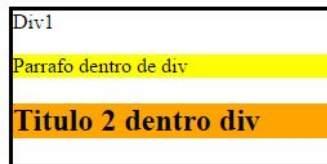
Introducción a DOM

El **D**ocument **O**bject **M**odel es una API (**A**pplication **P**rogramming **I**nterface) para documentos HTML y XML.

- Representación estructurada del documento
- Permite modificar el contenido
- Es lo que conecta las páginas web con JavaScript.

El DOM es un árbol de objetos...

Titulo 1



Titulo 2

- Elemento 1 [Link1](#)
- Elemento 2
 1. Primero
 2. Segundo

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Titulo 1</h1>
    <div>
      "Div1"
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 dentro div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>
        "Elemento 1 "
        <a href="#">Link1</a>
      </li>
      <li>
        "Elemento 2"
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```

Objetos en el DOM y JS

Existen muchos objetos ya predefinidos. Algunos son:

- **Window:** La ventana/pestaña del navegador. Es quien tiene el método “alert” que usamos antes.
- **History:** El historial, nos permite ir adelante, atrás, etc
- **Location:** La URL de la barra de navegación.
- **document:** El DOM de los elementos del body y header de este archivo HTML.

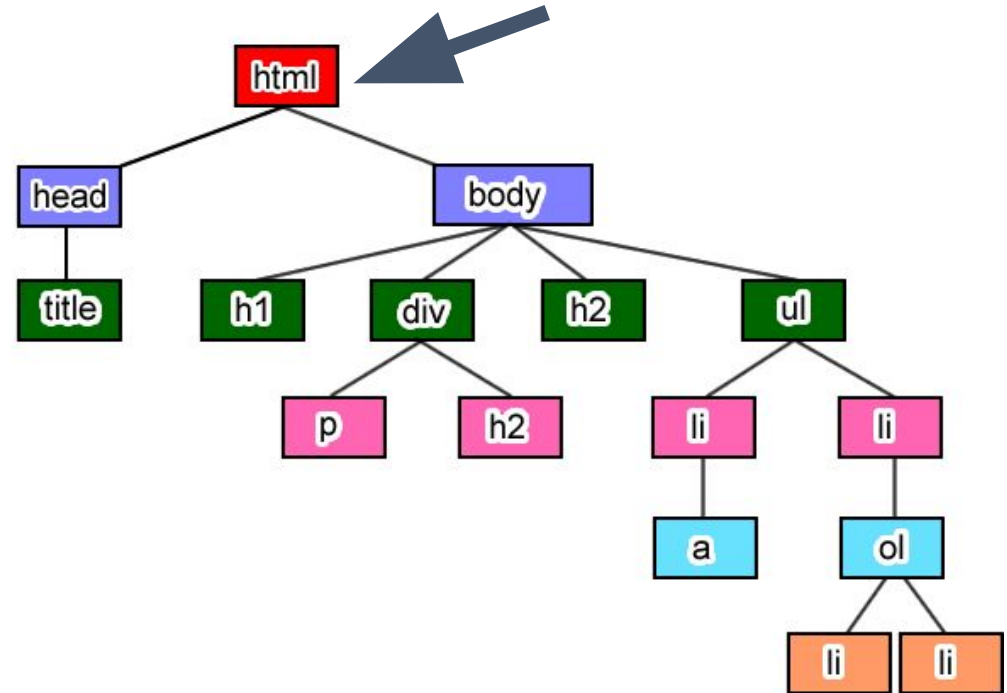
Como editar el DOM

1. Al documento le pedimos el nodo del elemento que queremos editar
2. A ese objeto (el nodo del arbol en cuestion) le modificamos los atributos que necesitemos con un nuevo valor

Dato: se le llama nodo a cada uno de los elementos de una lista enlazada, un árbol o un grafo en una estructura de datos■

Arbol HTML - DOM

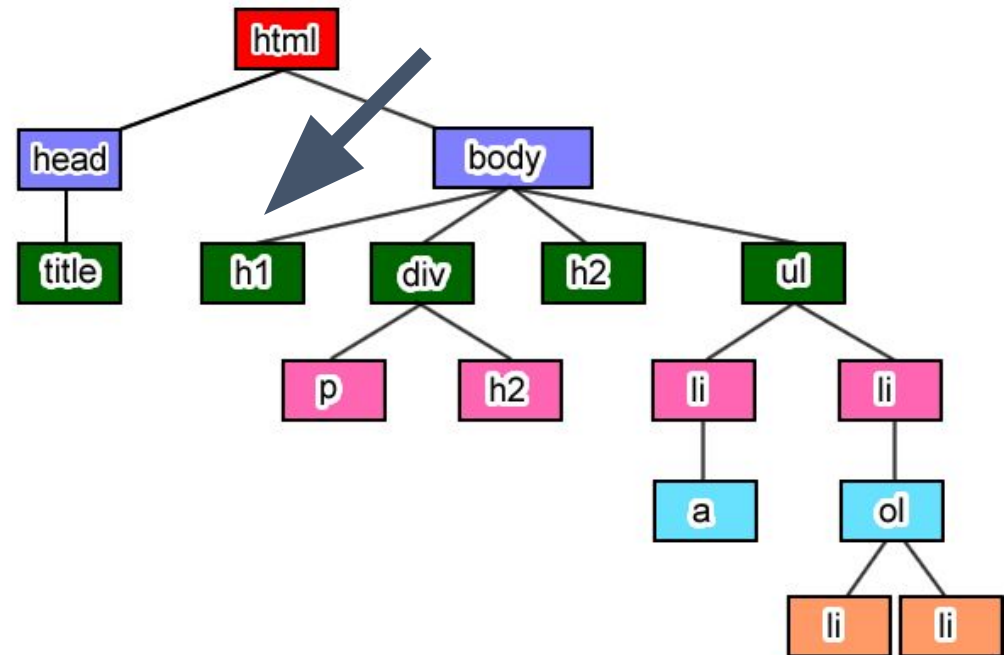
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



document

Arbol HTML - DOM

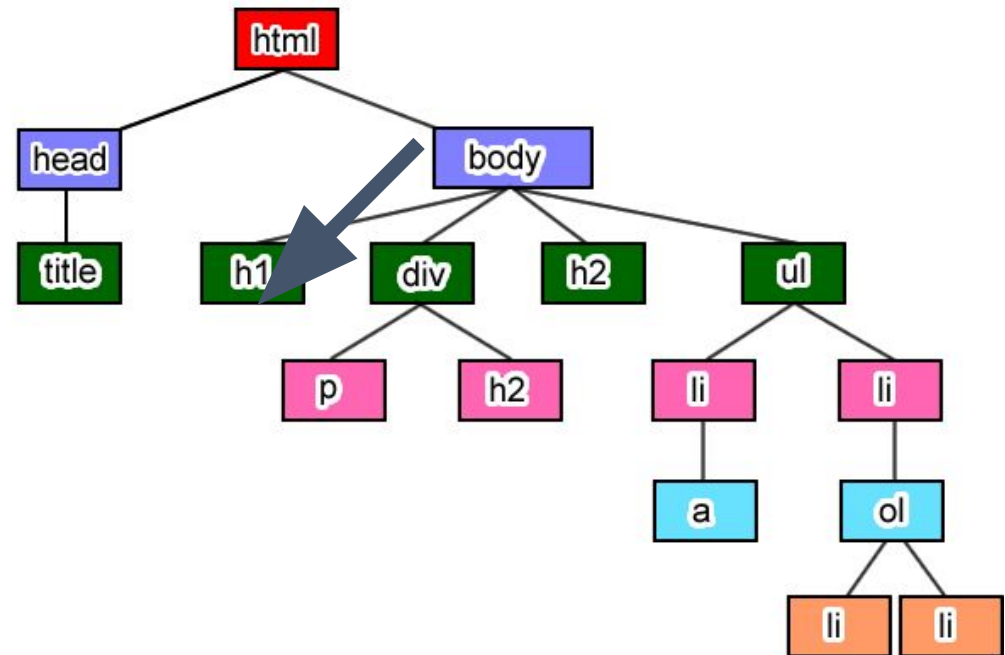
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



document.querySelector("h1")

Arbol HTML - DOM

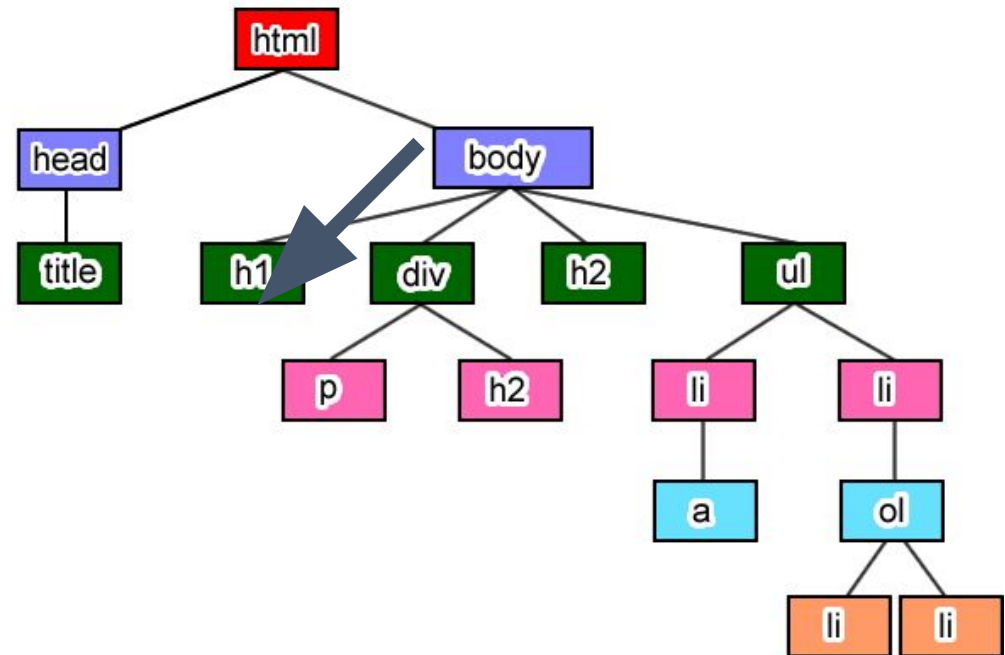
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div 1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



`document.querySelector("h1").innerHTML`

Arbol HTML - DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div 1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



```
document.querySelector("h1").innerHTML = "H1";
```

Obtener nodos del DOM

- Se pueden obtener elementos del DOM consultando por un ID, nombre, clase o un selector.
- Por ahora solo vamos a acceder a elementos mediante IDs

```
let elem = document.getElementById("identificador");
```

Leer/Editar el DOM

Las propiedades del DOM (HTML) se pueden leer/editar desde JavaScript.

```
let lampImg = document.getElementById("lamp");  
let lampImgAnterior = lampImg.src;  
lampImg.src = "foto.png";  
  
let unDiv = document.getElementById("unDiv");  
unDiv.innerHTML = "Cambiar contenido";
```

Resultado

En negrita marcado lo nuevo (Template Static)

EVENTO INPUT

```
<input type="text" id="txtNombre" oninput="ActualizarSaludo()" />  
<p id="txtSaludo">ACA VA EL SALUDO</p>
```

DA NOMBRE A LOS NODOS

```
function ActualizarSaludo() {
```

PIDE NODO

```
  //lee el nombre
```

```
  let nodoInput = document.getElementById("txtNombre");
```

```
  let nombre = nodoInput.value;
```

LEE VALOR

```
  //lo muestra en consola (opcional, para debug)
```

```
  console.log(nombre);
```

```
  //lo muestra en el DOM
```

PIDE NODO

```
  let nodoSaludo = document.getElementById("txtSaludo");
```

```
  nodoSaludo.innerHTML = "Hola " + nombre;
```

ESCRIBE VALOR

```
}
```

DEMO

Resumen DOM

Las propiedades del DOM (HTML) se pueden leer/editar desde JavaScript.

Estas tres líneas resumen todo lo que van a necesitar en esta etapa.

```
let lampImg = document.getElementById("lamp");  
let lampImgAnterior = lampImg.src;  
lampImg.src = "foto.png";
```

Objetos en el DOM y JS

Al DOM podemos:

- Agregarle nodos (es como escribir nuevas etiquetas en el HTML).
- Editar nodos (es como cambiar el HTML) para alterar propiedades o el contenido interno (el HTML que contiene).
- Borrar nodos (es como borrar las etiquetas).

Ejercicios

Carrera Programador full-stack

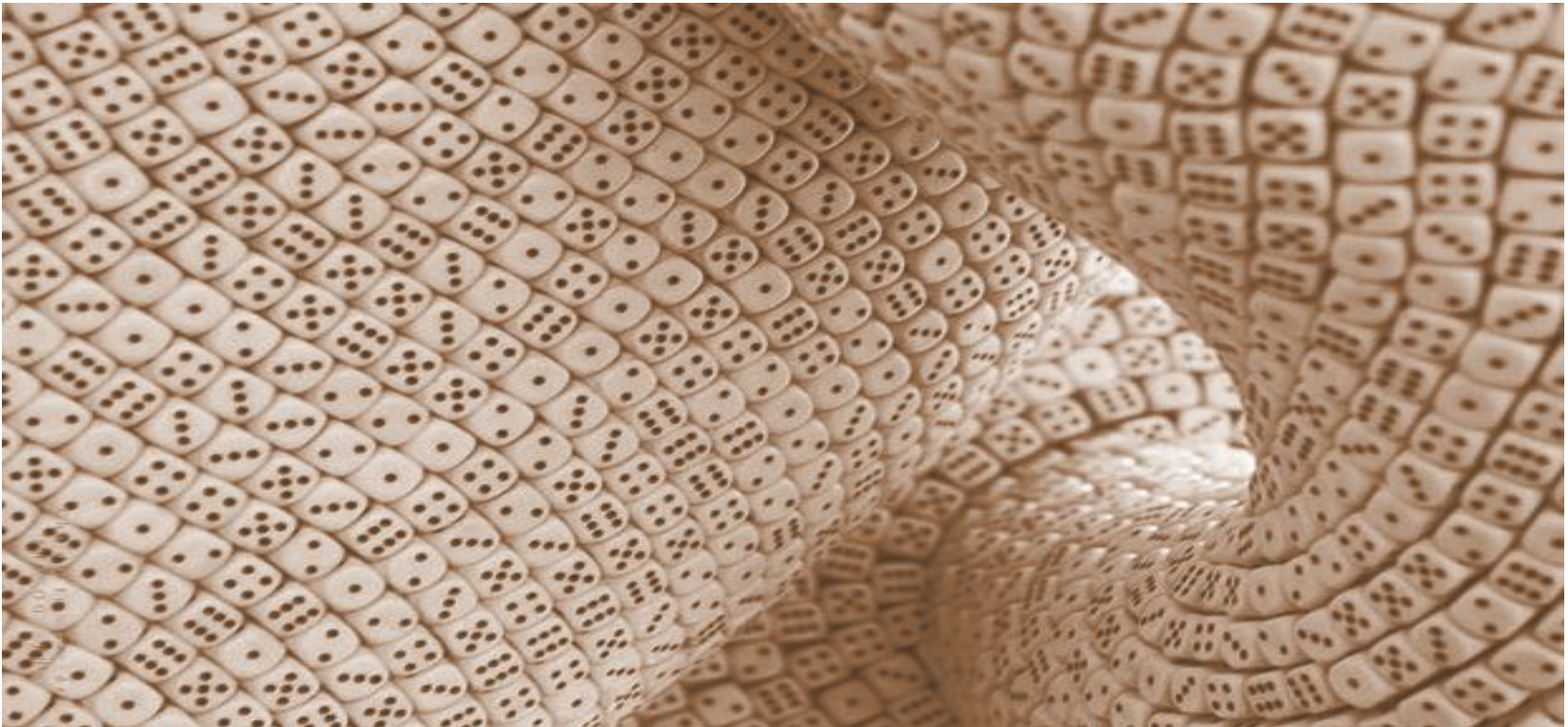
Ejercicio

En su página web, agregar un campo para llevar un contador de X elemento que siempre arranque en 0

- En la página web se muestra la cantidad de X
- Hay un botón para incrementar la cantidad
- Hay un botón para decrementar la cantidad

Ejercicio

Hacer una aplicación web que al apretar un botón simule el lanzamiento de dos dados 10 veces, sumarlos y muestre en el HTML la cantidad de veces que salió 7.



Ejemplos

- Two.js
 - Framework de dibujo en 2 dimensiones con JavaScript.
 - GitHub: <https://github.com/jonobr1/two.js>
- Pacman
 - <http://www.masswerk.at/JavaPac/JS-PacMan2.html>

Más Información

Libros

- JavaScript and JQuery : Interactive Front-End Web Development, Jon Duckett Willey 2014
- Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, Jennifer Niederst Robbins O'Reilly Media 2012
- Standard: <http://standardjs.com/rules.html>
- Tutorial W3 Schools: <http://www.w3schools.com/js/>
- [JavaScript from birth to closure](#)

Eventos

- <http://www.elcodigo.net/tutoriales/JavaScript/JavaScript5.html>
- <http://dev.opera.com/articles/view/handling-events-with-JavaScript-es>