Javascript en front-end

CFL
Programador
full-stack

Ejemplo 1

CFL 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>
```

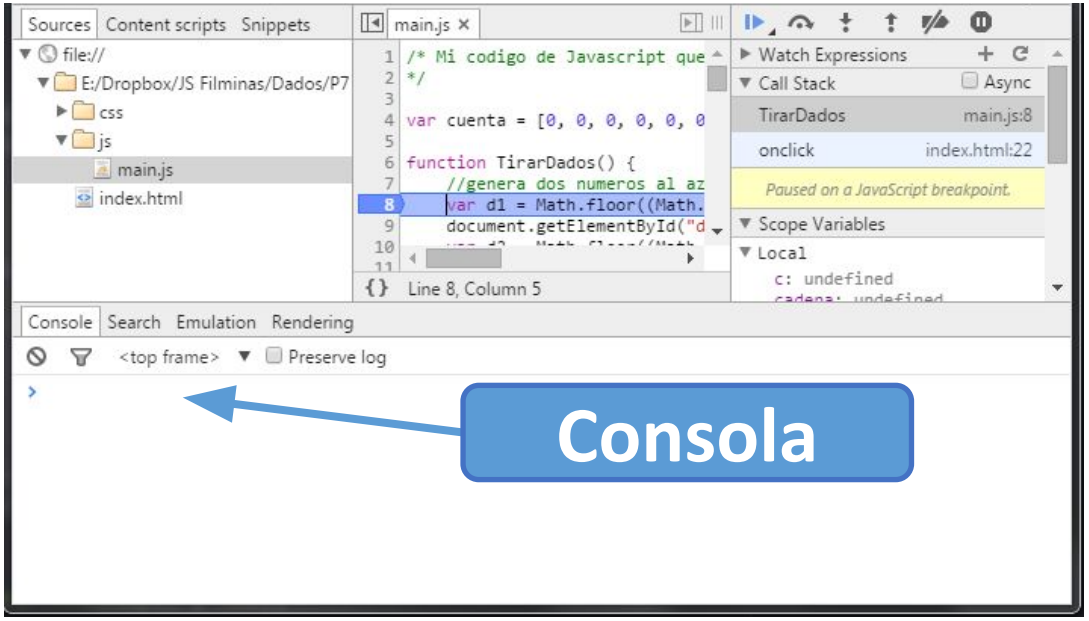
- Incluirlo al final del body, luego de que ya se cargo el html con **todos** sus elementos.

- **Se pueden agregar varios archivos .js**

Ver la consola

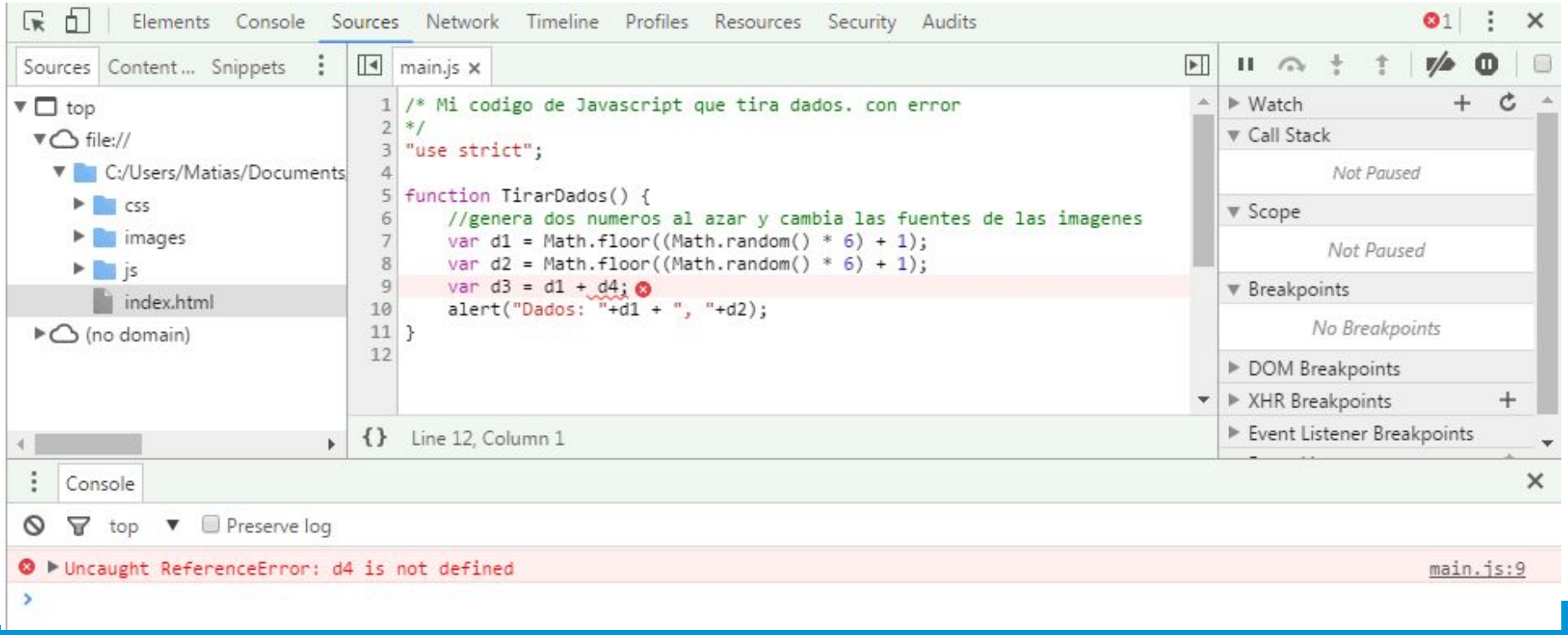
Menú Chrome > Más herramientas > Herramientas Desarrollador
Atajo de Teclado:

- Ctrl + Shift + I
- F12



Syntax Error

- Algunos errores del código se detectan al cargar el JS
- Otros errores luego de ejecutar esa línea
- Si no se tiene abierta a las “Herramientas de desarrollador”, no se ven los errores
- SIEMPRE ABRIRLA AL PROGRAMAR JS!



Botón para saludar

CFL
Programador
full-stack

Resolver el problema

¿Qué vamos a aprender?

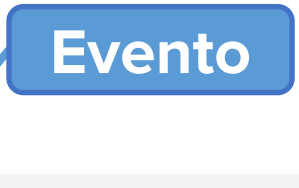
- Ejecutar un código al hacer click en un botón
  - Esto se llama “al pasar un **evento**”
- Para eso necesitamos darle un nombre a una parte del código
  - Esto se llama “declarar una **función**”

Eventos

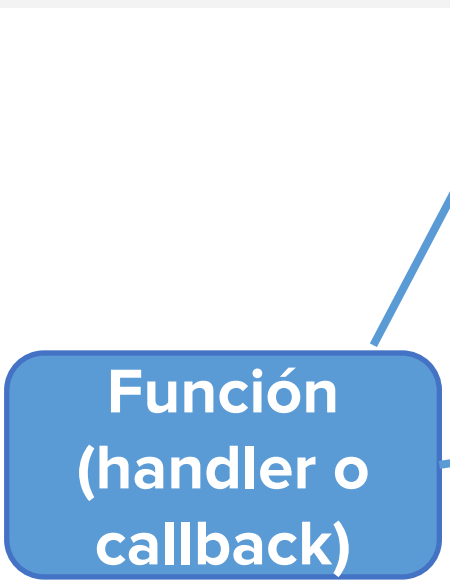
- Un evento es algo que ocurre en el sistema, originado por el usuario o otra parte del sistema y que se avisa al sistema.
- Ejemplos:
  - El usuario hace click.
  - Se terminó de cargar la página.
  - Pasó un segundo desde que se terminó de procesar.
- Las interfaces gráficas suelen programarse orientada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

Eventos

- Los eventos son capturados por manejadores (handlers).
  - **HTML**




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

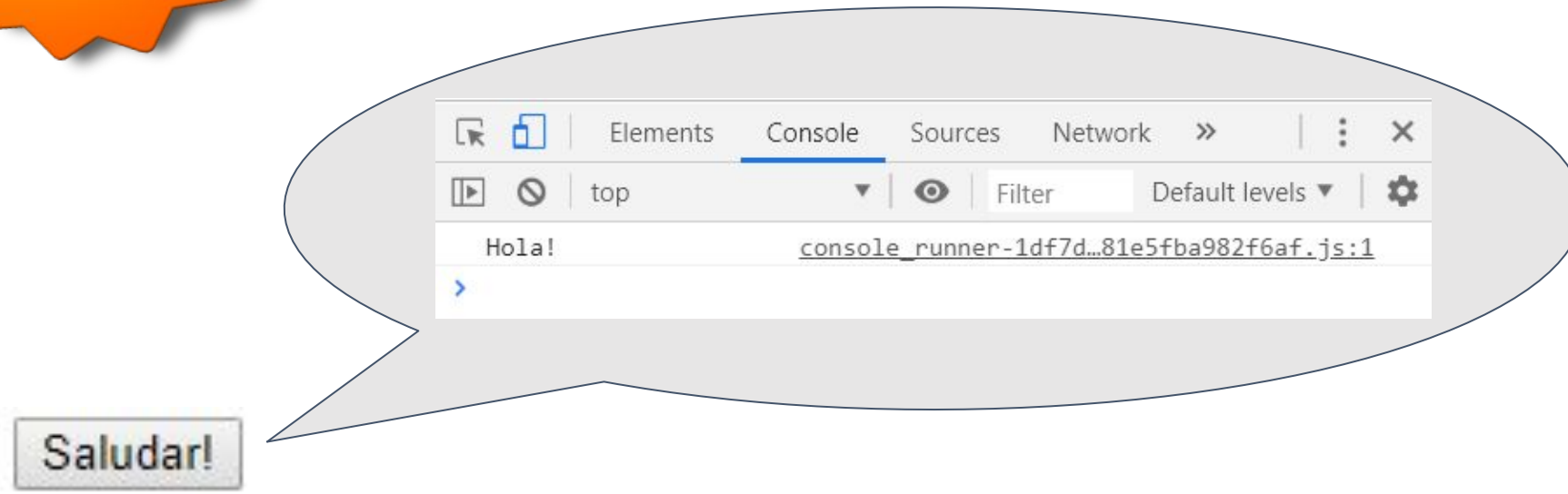


  - **Javascript**

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

No recomendado
(por ahora se hace así)

Resultado



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

Contador de clicks

CFL
Programador
full-stack

Resolver el problema

¿Qué vamos a aprender?

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

Variables: Variables y Constantes

- 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

```
contador = 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)

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>
```

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

Use Strict

- Es una buena práctica escribir al comenzar un archivo Javascript

```
"use strict";
```

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

Más info : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode

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
}
```

https://codepen.io/webUnicen/pen/XqWqwp

Debug

CFL 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**:*
Recorda 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");
```

https://codepen.io/webUnicen/pen/pLRRZP

Saludo con nombre

CFL
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 largo = str.length("cadena");
```

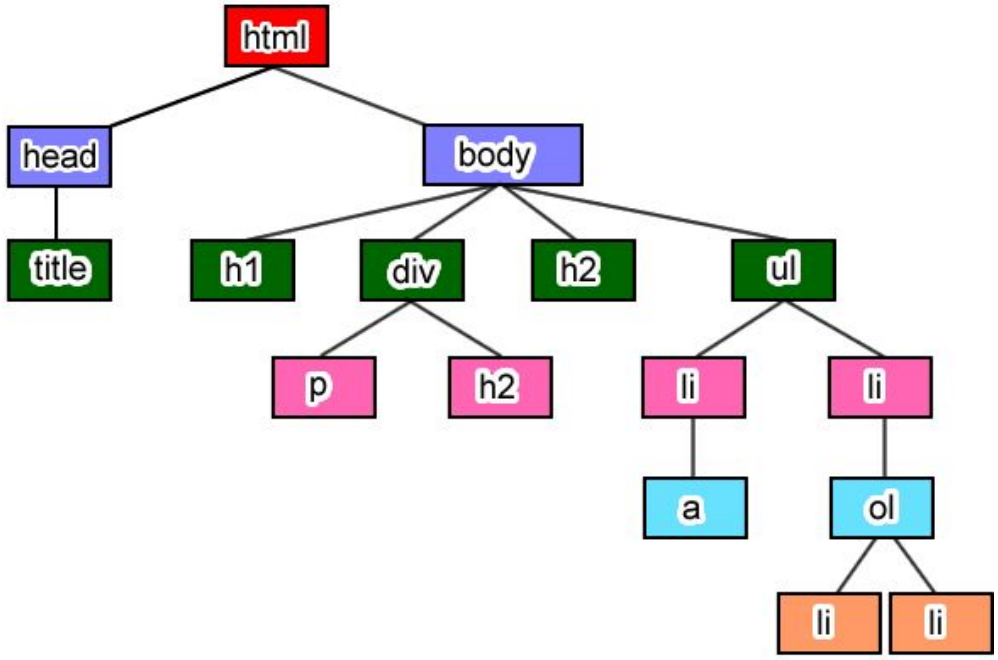
El valor calculado se **devuelve** y debe guardarse en una variable

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length

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>
```

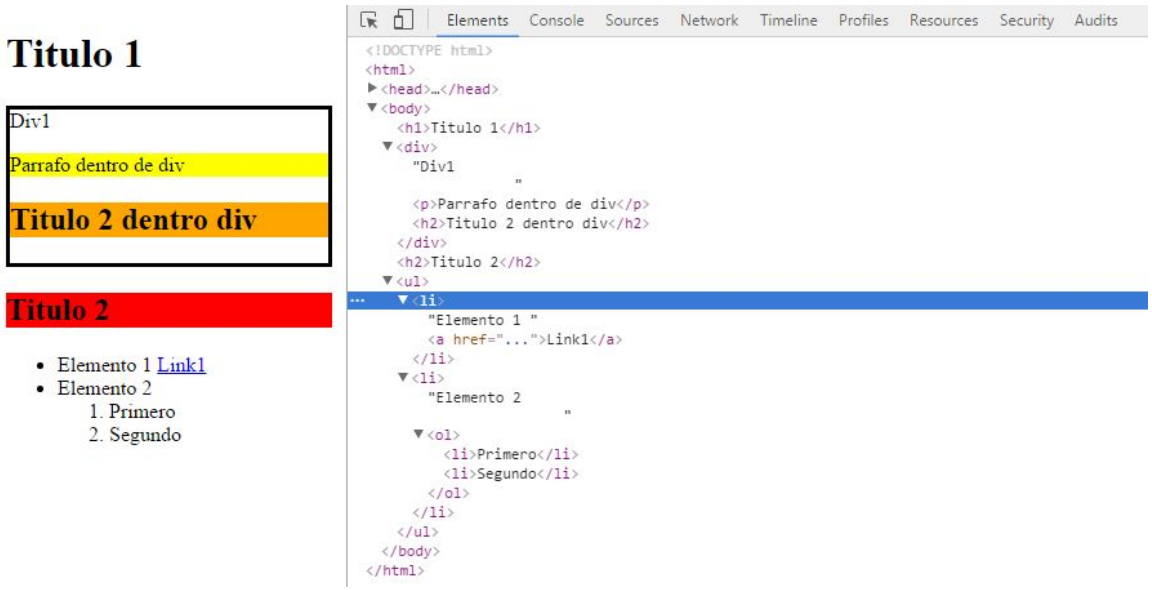


Introducción a DOM

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...



Objetos en el DOM y JS

Existen muchos objetos ya predefinidos. Algunos son:

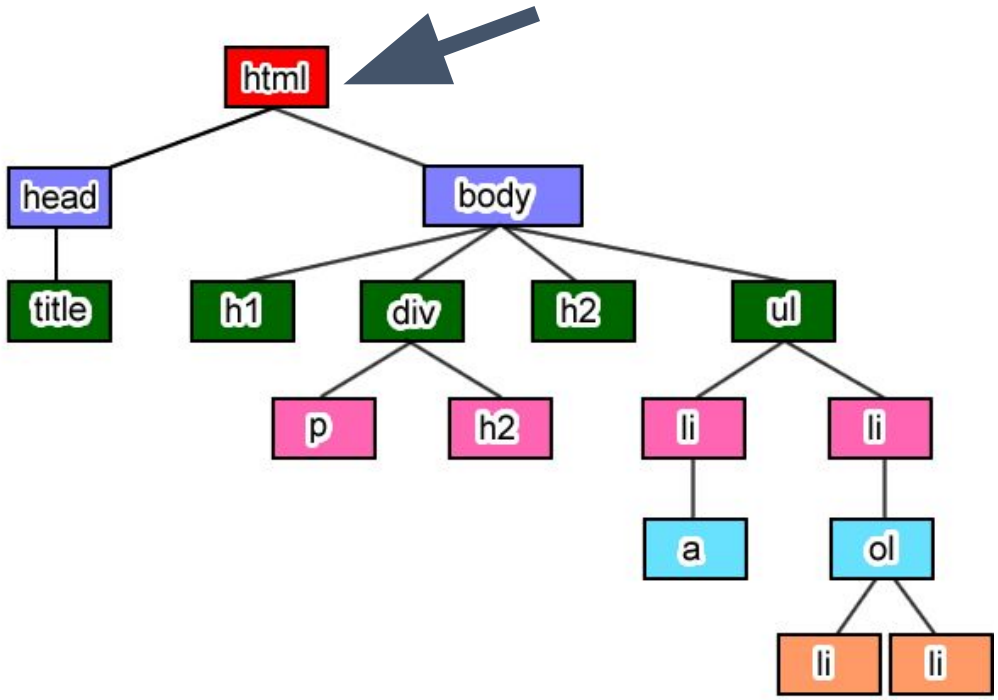
- **Window**: La ventana/pestaña del navegador. Es quien tiene el método “alert” que usamos antes.
- **History**: El historial, nos permite ir adelante, atrás, etc
- **Location**:La URL de la barra de navegación.
- **document**: El DOM de los elementos del body y header de este archivo HTML.

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

Arbol HTML - DOM

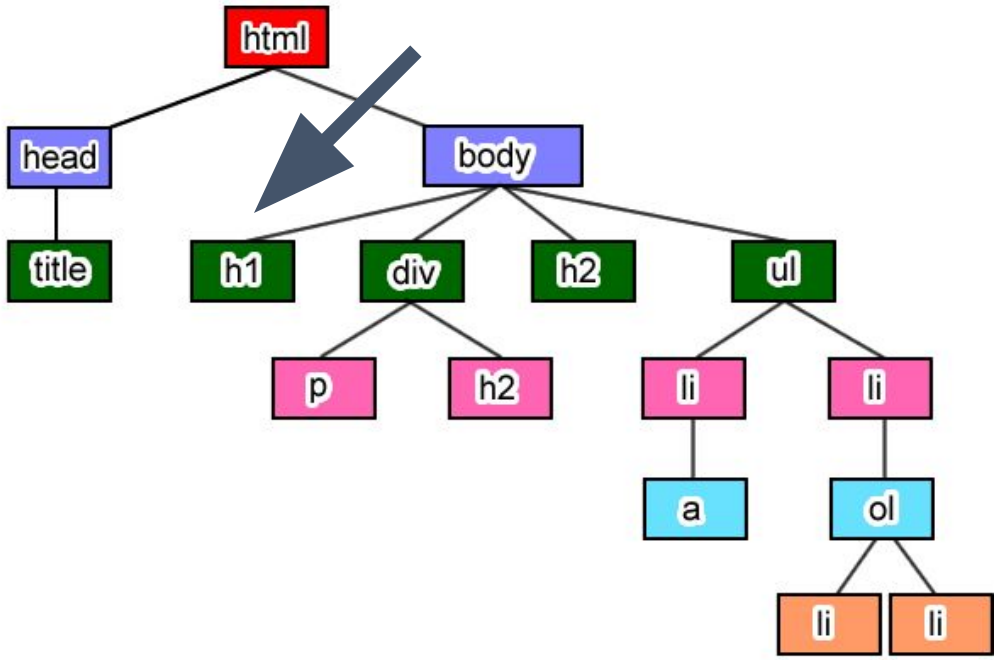
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a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



document

Arbol HTML - DOM

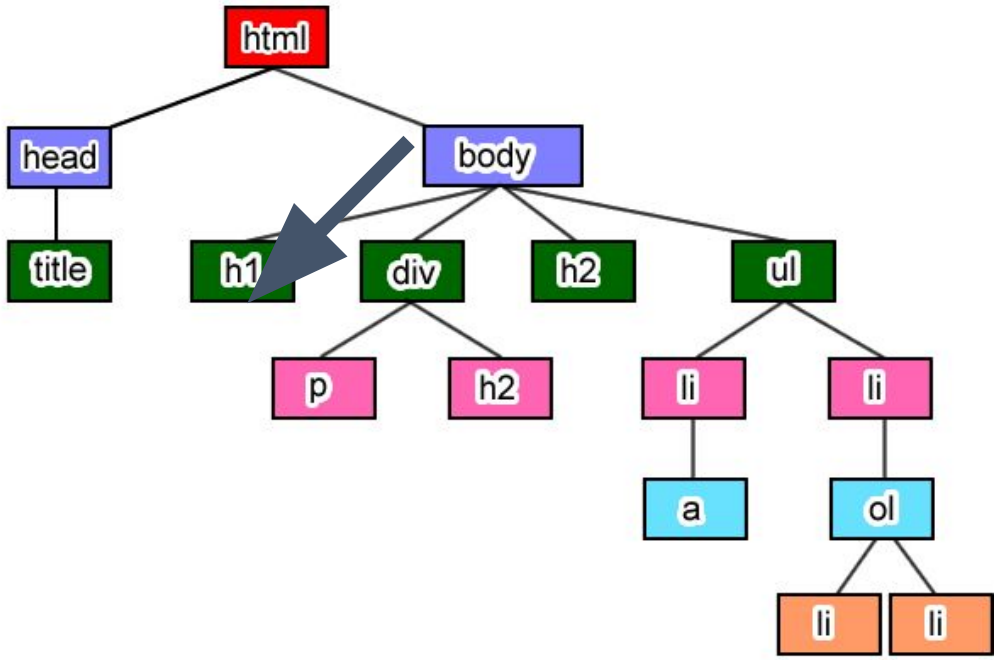
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a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

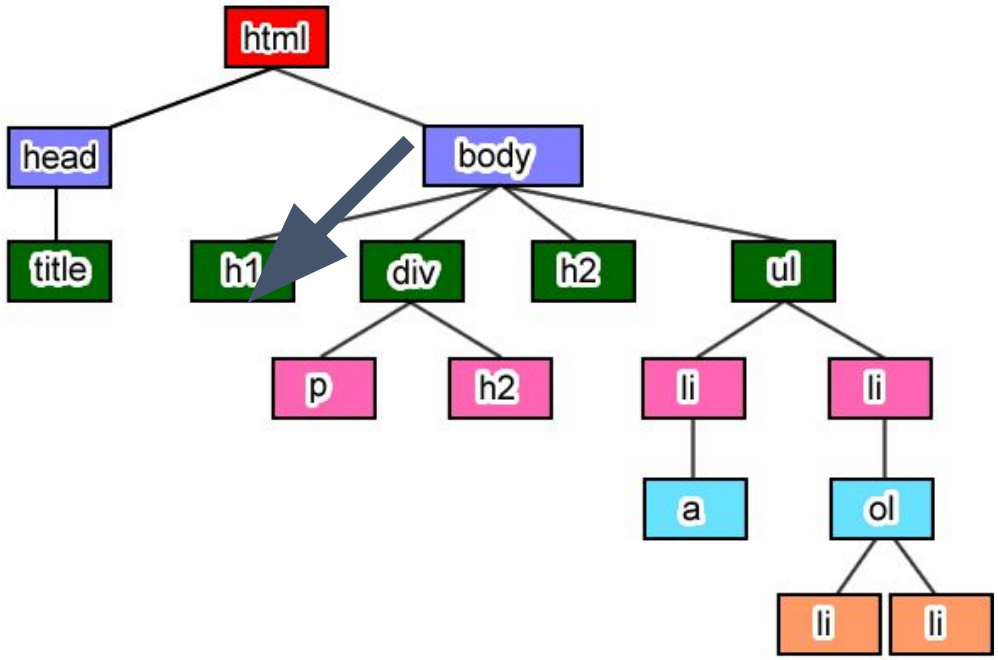


document.querySelector("h1").innerHTML

Arbol HTML - DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>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>
```

H1




```
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

EVENTO INPUT

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

DA NOMBRE A LOS NODOS

```
function ActualizarSaludo() {
    //lee el nombre
    let nodoInput = document.getElementById("txtNombre");
    let nombre = nodoInput.value;
    //lo muestra en consola (opcional, para debug)
    console.log(nombre);
    //lo muestra en el DOM
    let nodoSaludo = document.getElementById("txtSaludo");
    nodoSaludo.innerHTML = "Hola " + nombre;
}
```

PIDE NODO

LEE VALOR

PIDE NODO

ESCRIBE VALOR

DEMO

https://codepen.io/webUnicen/pen/geRRLe

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

CFL Programador full-stack

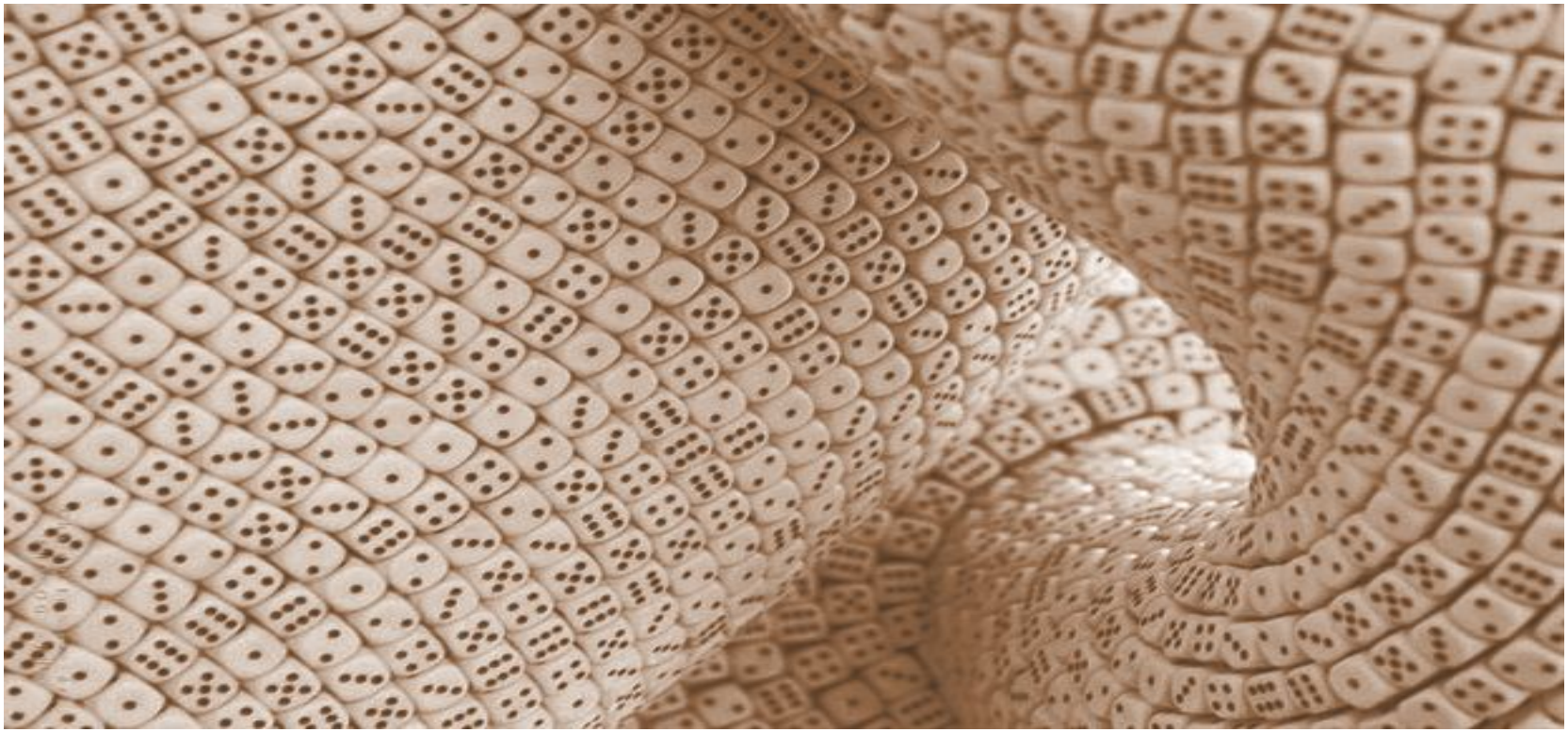
Ejercicio

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
- Hay una caja de texto para sumar muchos elementos X en una sola acción

Ejemplo

Hacer una aplicación web que al apretar un botón simule el lanzamiento de dos dados 1000 veces, sumarlos y muestre en el HTML la cantidad de veces que salió 7.



Ejemplo

Dar un form para cargar compras del cliente y un sumar que nos diga el total (dos botones, un agregar y otro calcular, arreglo global)

Ejemplos

- Two.js
  - Framework de dibujo en 2 dimensiones con Javascript.
  - GitHub: https://github.com/jonobr1/two.js
- Bootstrap
  - Dropdowns, carrousels, algunos comportamientos responsive (menu), etc…
- Paper.js
  - Framework para dibujos vectoriales
  - http://paperjs.org/
- 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