Introducción a la web: JavaScript

IIC1005 - Computación: Ciencia y Tecnología del Mundo Digital

Fernando Florenzano Hernández
faflorenzano@ing.puc.cl

https://github.com/fdoflorenzano/intro-a-la-web-2

Contenidos

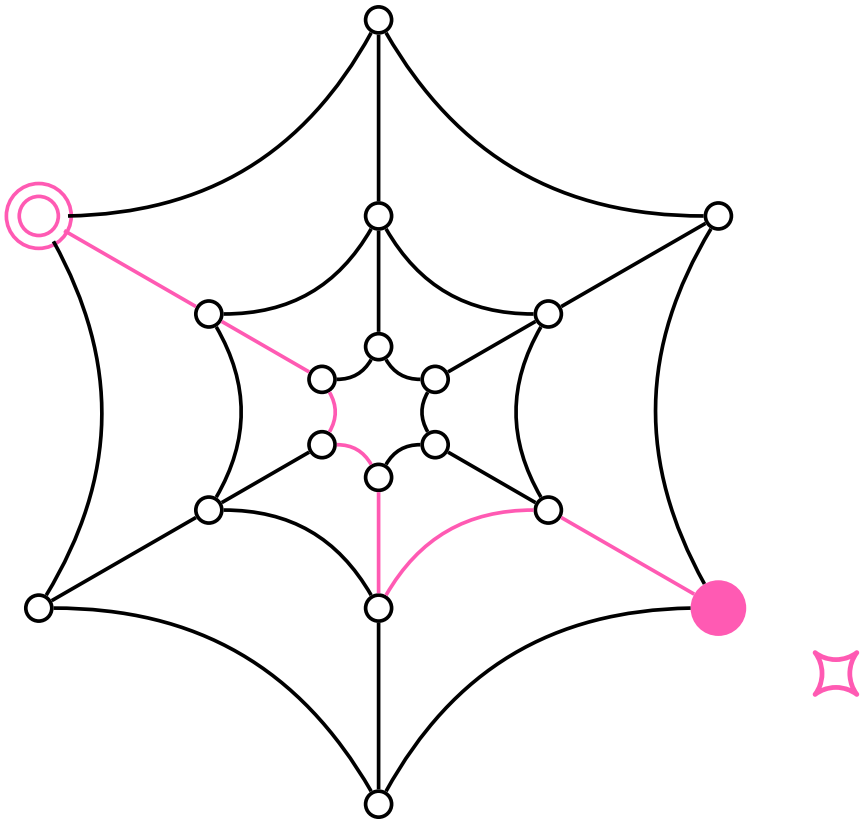
Motivación

Historia

JavaScript

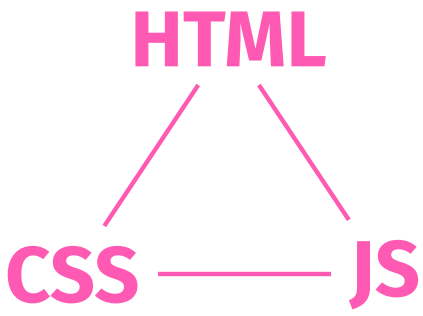
Contenidos

Recordando la Web



Desarrollo web

El **desarrollo web** define la creación de **sitios web**. Para conseguirlo se hace uso de tecnologías de **software** del lado del **servidor** y del **cliente**. Del lado del cliente, hay tres tecnologías base:



Hoy aprenderemos sobre **JavaScipt**.

Motivación

Ya aprendimos sobre **HTML** y **CSS**, ¿qué le falta a nuestras **páginas web**?

¡Interacción y dinamismo!

Otro **estándar de la web** hoy en día, es el agregar interacción y dinamismo a sus páginas. Le agrega **valor** de interfaz para el **mejor uso** de parte de su usuario.

Contenidos

Historia de JavaScript

- **1995**: JavaScript fue diseñado en sólo **diez días**, por Brendan Eich, cuando trabajaba en **Netscape**.
- Inicialmente, se llamó **Mocha**, luego **LiveScript** y finalmente, por asuntos de marketing... **JavaScript**.
- Entre **1996-1999**: Se hace el intento de estandarizar JavaScript por ECMA International con el propósito de que todos los navegadores implementaran el mismo lenguaje. Comienza la ***browser wars***.
- Este estándar se llama **ECMAScript** (abreviado **ES**).

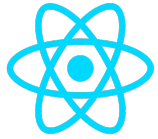
Historia de JavaScript

- En **2009** se libera **ES5**, versión con muchas mejoras que todos los navegadores soportan.
- En **2015** se libera **ES6** que introduce **muchas** nuevas funcionalidades. Esta es la versión que aprenderemos hoy.
- Desde entonces, cada año ha salido una especificación nueva con algunas novedades: **ES7** (**2016**), **ES8** (**2017**), **ES9** (**2018**) y **ES10** (**2019**).

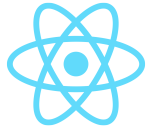
Historia de JavaScript

- **2010-2015**: Surgen los *frameworks* de JavaScript: **Backbone**, **Ember**, **AngularJS**, **React**, **Angular** y **Vue**.



- En el mismo tiempo, aparecen *frameworks* y librerías para desarrollo móvil y de escritorio basado en JavaScript como: **Ionic**, **React Native** y **Electron**.


Contenidos

Motivación

Historia

JavaScript

JavaScript

- Es un lenguaje de programación de **alto nivel**, **interpretado** y **multi-paradigma**, como **Python**.
- Es el **único** lenguaje que se puede ejecutar en un **navegador web**.
- Es un lenguaje por si solo, pero que también permite interactuar con la **estructura** de una página web y **alterarla**.

JavaScript

Variables

```
1  const numero = 1005; // una constante
2  const curso = 'Computación: Ciencia y ...';
3  let hora = 10; // variable que cambia
4  hora = 11;
```

```
1  const numero = 1005; // una constante
2  numero = 1006;
3  TypeError: Assignment to constant variable.
```

```
1  var numero = 1005; // funciona, pero no nos gusta var
```

JavaScript

Control de flujo

```
1  const cierto = true;
2  const falso = false;
3
4  if (cierto && falso) {
5      console.log('Esto no se imprimirá.');
6  } else if (cierto || falso) {
7      console.log('Esto sí se imprimirá.');
8  } else {
9      console.log('Esto tampoco se imprimirá.');
10 }
```

JavaScript

Strings

```
1  const cadena1 = 'Computación: ';
2  const cadena2 = 'Ciencia y Tecnología ';
3  const cadena3 = 'del Mundo Digital';
4  const cadenaCompleta = cadena1 + cadena2 + cadena3;
5  const largo = cadenaCompleta.length; // 51
```

JavaScript

Template strings

```
const texto = 'IIC1005 tiene largo: ' + largo;
console.log(texto);
```

```
// ES6
const texto = `IIC1005 tiene largo: ${largo}`;
console.log(texto);
```

JavaScript

Arrays

```
let listaDePi = [3, 1, 4, 1, 5];
listaDePi.push(9); // [3, 1, 4, 1, 5, 9]
listaDePi.pop(); // [3, 1, 4, 1, 5]
const largo = listaDePi.length; // 5
```

```
const primerElemento = listaDePi[0]; // 3
const ultimoElemento = listaDePi[listaDePi.length - 1];
// 5
const segundoYTercero = listaDePi.slice(1, 3);
// [1, 4]
```

JavaScript

For loops

```
for (let i = 0; i <= listaDePi.length; i++) {
    console.log(listaDePi[i]);
}
```

```
// ES6
for (const numero of listaDePi) {
    console.log(numero);
}
for (const caracter of 'Computación: Ciencia...') {
    console.log(caracter);
}
```

JavaScript

Objects

```
const profesor = {
    nombre: 'Denis',
    apellido: 'Parra',
    cursos: 3
};
const nombreProfe = profesor['nombre']; // 'Denis'
const cursosProfe = profesor.cursos; // 3
```

```
// ES6
for (const propiedad in profesor) {
    console.log(`${propiedad}:  ${profesor[propiedad]}`);
}
// nombre: Denis
// apellido: Parra
// cursos: 3
```

JavaScript

Functions

```
function sumaCinco(numero) {
    const resultado = numero + 5;
    return resultado;
}
sumaCinco(3); // 8
```

```
const suma = function (numero1, numero2) {
    const resultado = numero1 + numero2;
    return resultado;
}
suma(3, 5); // 8
```

JavaScript

Arrow functions (ES6)

```
const sumaCinco = (numero) => {
    const resultado = numero + 5;
    return resultado;
}
sumaCinco(3); // 8
```

```
const suma = (numero1, numero2) => numero1 + numero2;
suma(3, 5); // 8
```

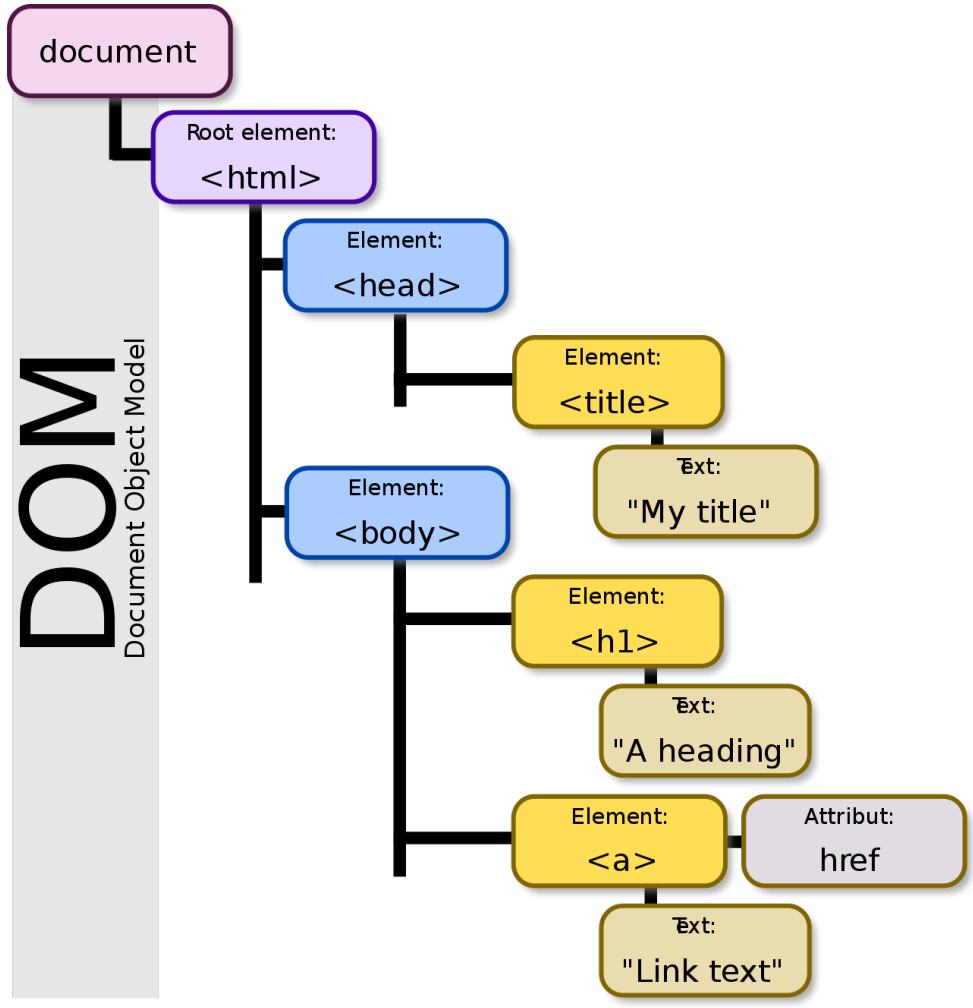
DOM

Document Object Model

El **DOM** es una interfaz que, a través de una representación estructurada, permite acceder y manipular un documento **HTML**.

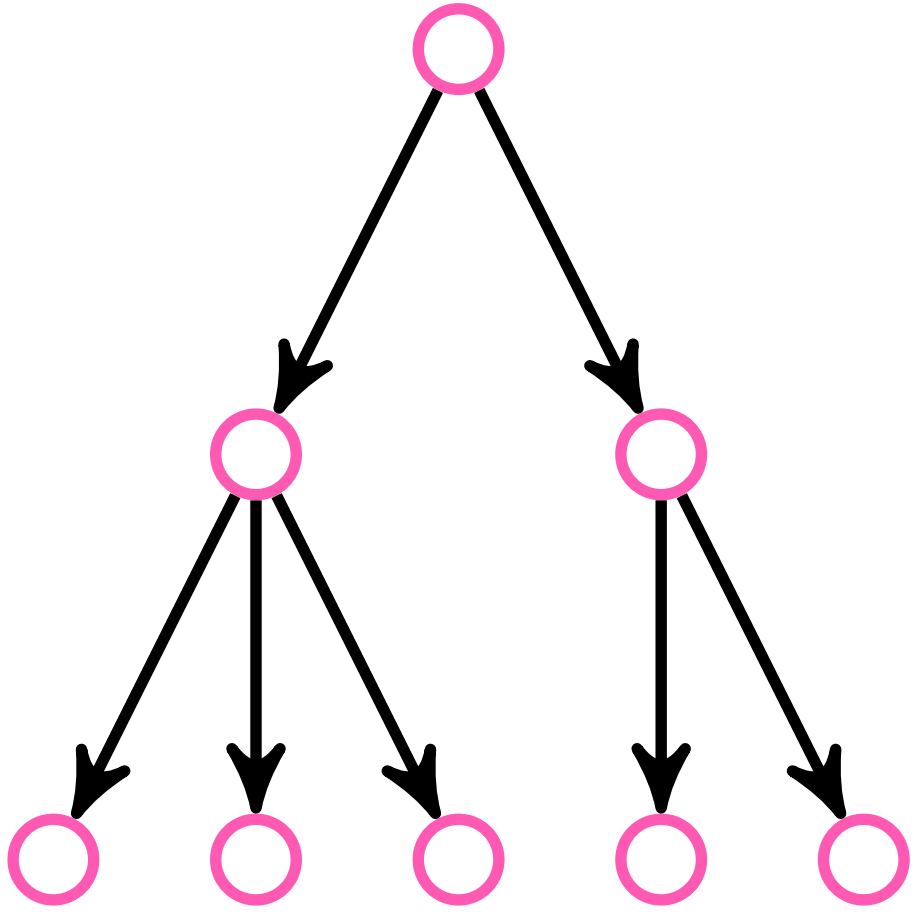
Esta representación se modela como un **árbol**, en donde cada **nodo** es un **objeto** del documento.

DOM

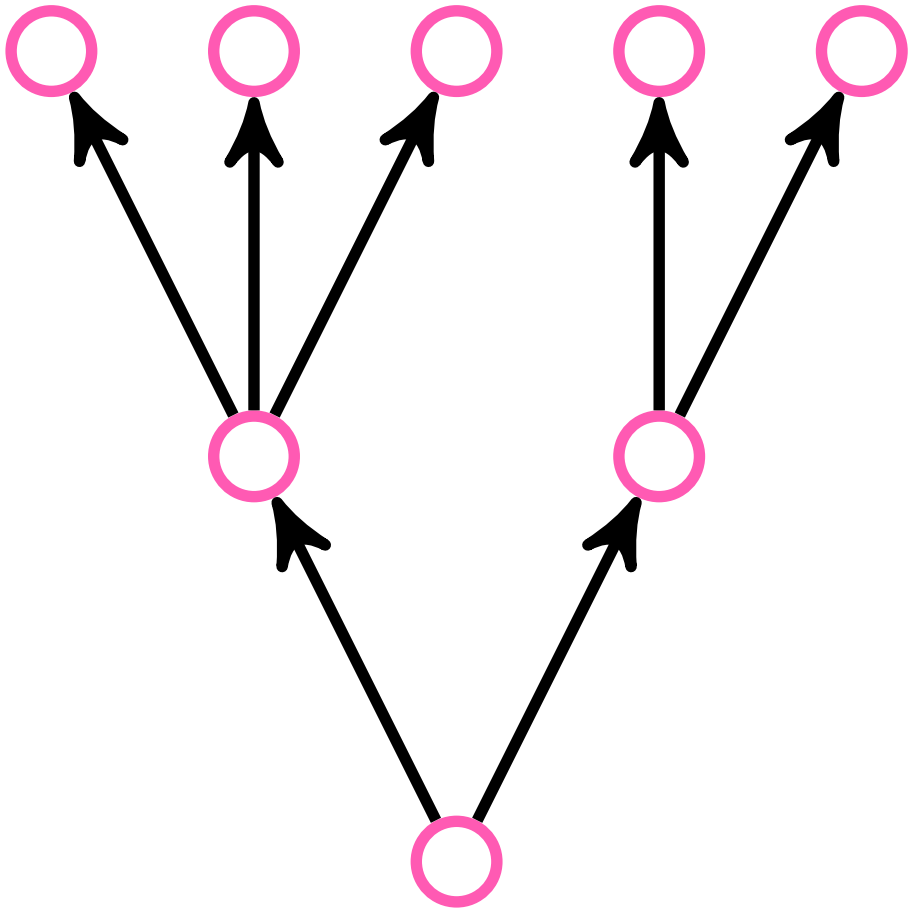


Fuente: **Wikipedia: DOM**

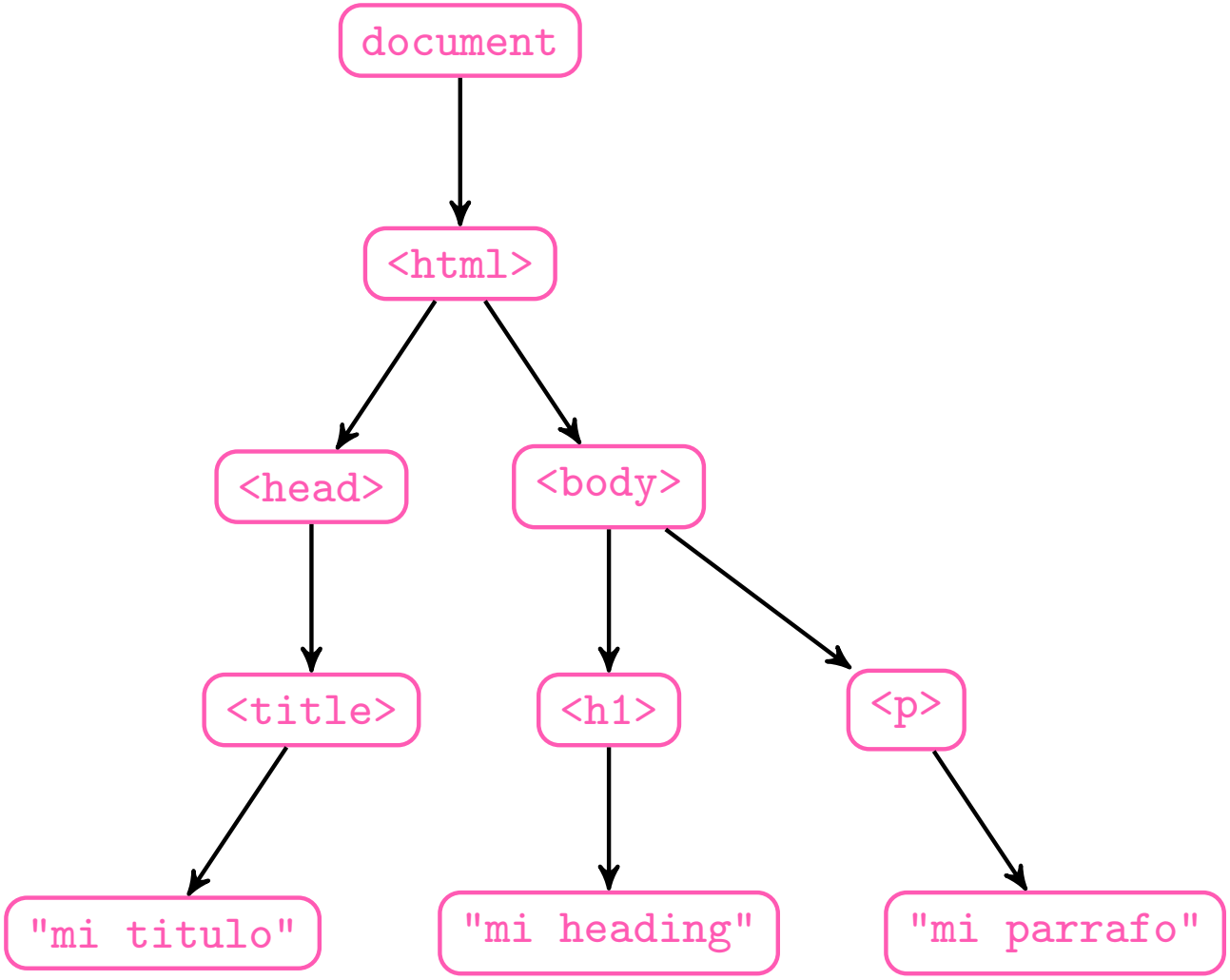
Árboles y el DOM



Árboles y el DOM



Árboles y el DOM



Importar JS en HTML

De forma similar a CSS, hay etiquetas para escribir directamente o para importar código **JavaScript** al documento HTML:

```
<script>
const elProyectorEs = 'malo';
const vecesQueHaPestanado = 1304;
</script>
```

```
<script src='script.js' charset='utf-8'></script>
```

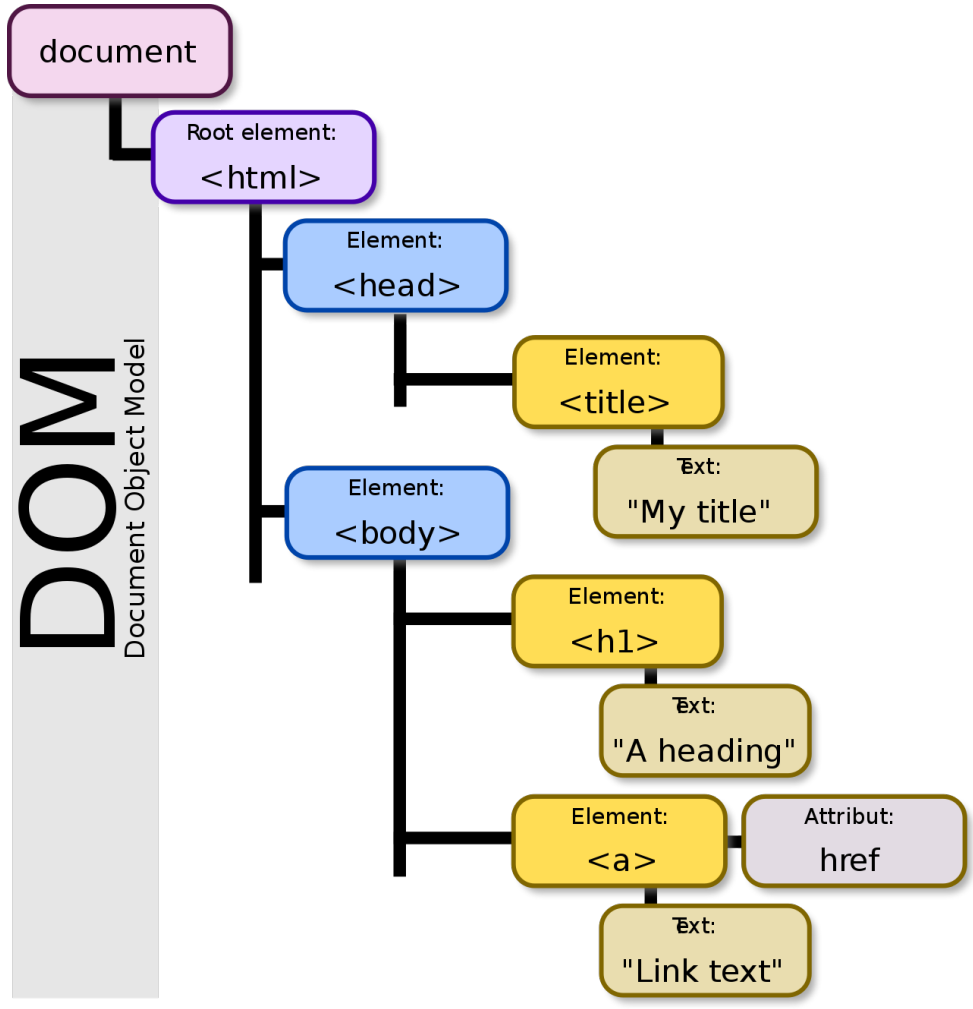
JavaScript y el DOM

Podemos acceder al **DOM** a través del objeto `document`.

```
1  console.log(document.URL); // imprime el URL actual
```

```
1  document.children; // contiene los elementos hijos
2  // en este caso: [<html>]
```

JavaScript y el DOM



Fuente: **Wikipedia: DOM**

JavaScript: elementos

```
const elemento = document.getElementById('idElemento');
// retorna elemento en el documento de id='idElemento'
```

```
const nuevoP = document.createElement('p');
// retorna nuevo elemento de etiqueta <p>
```

```
elemento.appendChild(nuevoP);
// ahora, <p> es hijo del elemento inicial
```

```
elemento.removeChild(nuevoP);
// <p> fue eliminado como hijo del elemento
// y del documento entero :(
```

JavaScript: elementos de texto

```
1  const nuevoP = document.createElement('p');
2  const nodoDeTexto = document.createTextNode('¿Qué tal?');
3  nuevoP.appendChild(nodoDeTexto);
4  // Eso entregaría una elemento de la forma:
5  // <p>¿Qué tal?</p>
```

JavaScript: clases

```
elemento.className = 'magenta';
// ahora el elemento tiene clase: 'magenta'
// en realidad, tiene clases: ['magenta']
```

```
elemento.classList.add('azul');
// ahora el elemento tiene clases: ['magenta', 'azul']
```

```
elemento.classList.remove('magenta');
// ahora el elemento tiene clases: ['azul']
```

JavaScript: eventos

```
elemento.addEventListener('click', () => {
    // código que se ejecuta cada vez que
    // se hace clic sobre elemento
});
```

JavaScript: ejemplo

```
const contenedor = document.getElementById('contenedor');
const boton = document.getElementById('boton');
boton.addEventListener('click', () => {
    const nuevoH1 = document.createElement('h1');
    const textoH1 = document.createTextNode('¡Hola!');
    nuevoH1.appendChild(textoH1);
    contenedor.appendChild(nuevoH1);
});
// ¿Qué ocurriría aquí?
```

Aprender practicando

Para el material:

- **Descarga** el repositorio `https://github.com/fdoflorenzano/intro-a-la-web-2`.
- En la carpeta `ejemplo` encontrarán el ejemplo de la clase de hoy.
- En la carpeta `practico` encontrarán un ejercicio para ustedes.
- **Edita** los archivos y revisa los cambios que se producen.

Para trabajar:

- Usa un navegador web **moderno**, como **Firefox** o **Chrome**.
- Descarga un editor de texto con muchas funcionalidades. Opciones populares son **Atom**, **Visual Studio Code**, **Notepad ++** y **Subime Text**.

Herramientas útiles y referencias de JS

- Los ***developer tools*** del navegador, en este caso, revisar la **consola** por cualquier `console.log()`.
- **Can I Use... ?**
- **Tutoriales web** del **MDN**.
- **Tutoriales web de JavaScript** del **MDN**.