



# DOM+HTML+JS (parte I)

---

JUAN CARLOS CONDE RAMÍREZ

*WEB-TECHNOLOGIES*

# Introducción

---

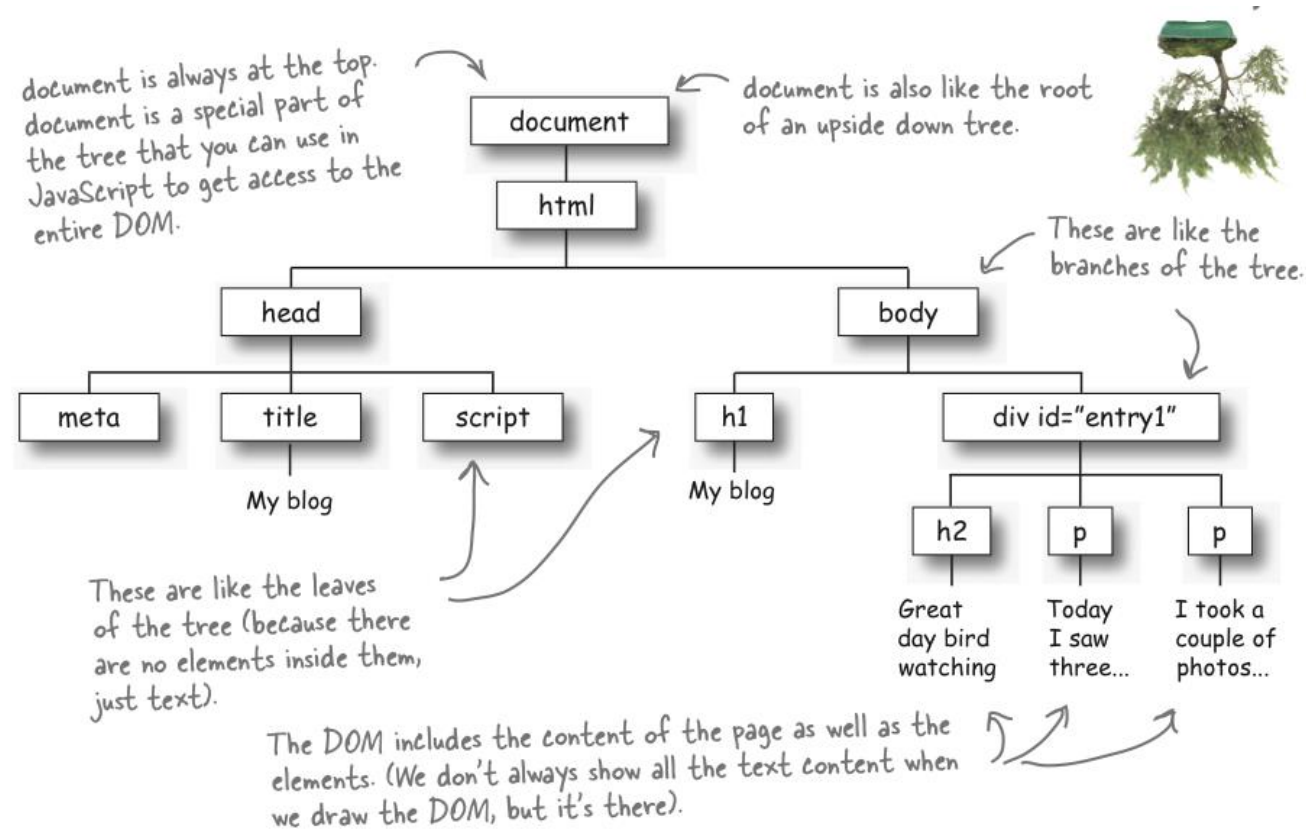
- Estas notas dan un breve recordatorio conceptual del DOM:
  - ¿Qué es?
  - ¿Cómo gestiona elementos para documentos HTML y XML?
  - ¿Cómo se accede a él usando JavaScript?
- También pretenden mostrar cómo esta "API" presenta la información, da referencias y ejemplos.

# ¿Qué es DOM?, I

---

- El modelo de objetos de documento (DOM) es una interfaz de programación para los documentos HTML y XML.
- Facilita una representación estructurada del documento y define de qué manera los programas pueden acceder, al fin de modificar, tanto su estructura, estilo y contenido.
- El DOM da una representación del documento como un **grupo de nodos y objetos** estructurados que tienen propiedades y métodos.
- Esencialmente, **conecta las páginas web a scripts o lenguajes de programación.**

# Representación gráfica de DOM



# ¿Qué es DOM?, II

---

- Una página Web es un documento. Éste documento puede exhibirse en la ventana de un navegador o también como código fuente HTML.
- Pero, en los dos casos, es el mismo documento. El modelo de objetos de documento (DOM) proporciona otras formas de presentar, guardar y manipular este mismo documento.
- El DOM es una representación completamente orientada a los objetos de una página Web y puede ser modificado con un lenguaje de scripting como JavaScript.

# ¿Qué es DOM?, III

---

- El *W3C DOM estándar* forma la base del funcionamiento del DOM en muchos navegadores modernos.
- Varios navegadores ofrecen extensiones más allá del estándar W3C, por lo que hay que tener mucho cuidado al utilizarlas en la web, ya que los documentos pueden ser consultados por navegadores que tienen DOMs diferentes.
- Por ejemplo, el DOM de W3C especifica que el método `getElementsByTagName` en el código siguiente debe devolver una lista de todos los elementos `<p>` del documento:

# ¿Qué es DOM?, IV

---

```
1 paragraphs = document.getElementsByTagName ("p");  
2 // paragraphs[0] es el primer elemento <p>  
3 // paragraphs[1] es el segundo elemento <p>, etc.  
4 alert (paragraphs [0].nodeName);
```

- Todas las propiedades, métodos y eventos disponibles para la manipulación y la creación de páginas web está organizado dentro de objetos.
- Un ejemplo: el objeto `document` representa al documento mismo, el objeto `table` hace funcionar la interfaz especial `HTMLTableElement` del DOM para acceder a tablas HTML, y así sucesivamente.
- Ésta documentación procura una relación objeto-por-objeto del DOM que funciona con los navegadores basados en Gecko.

# DOM y JavaScript, I

---

- El DOM no es un lenguaje de programación pero sin él, el lenguaje JavaScript no tiene ningún modelo o noción de las páginas web, de las páginas XML ni de los elementos con los cuales es usualmente relacionado.
- Cada elemento; el documento entero, el título, las tablas, los títulos de las tablas, el texto dentro de las celdas de las tablas, etc., son parte de DOM para cada documento.
- Así se puede acceder y manipularlos utilizando el DOM y un lenguaje de escritura, como JavaScript.



# DOM y JavaScript, II

---

- En el comienzo, JavaScript y el DOM estaban herméticamente enlazados, pero después se desarrollaron como entidades separadas.
- El contenido de la página es almacenado en DOM y el acceso y la manipulación se hace vía JavaScript, lo que podría representarse de la siguiente manera:

*API (web o página XML) = DOM + JS(lenguaje de script)*

- El DOM fue diseñado para ser independiente de cualquier lenguaje de programación, hace que la presentación estructural del documento esté disponible desde un simple y consistente API.

# DOM y JavaScript, III

---

- Aunque nos centremos en JavaScript, las directrices del DOM pueden construirse para cualquier lenguaje, así como se muestra en el siguiente ejemplo de *Python*:

```
1  # Ejemplo DOM de Python
2  import xml.dom.minidom as m
3  doc = m.parse("C:\\Projects\\Py\\chap1.xml");
4  doc.nodeName # Propiedad DOM del objeto document;
5  p_list = doc.getElementsByTagName("para");
```

# ¿Cómo se accede a DOM?, I

---

- Los diferentes navegadores tienen directrices DOM distintas, y éstas directrices tienen diversos grados de conformidad al estándar actual, pero todos lo usan.
- Cuando se crea un script el API inmediatamente está disponible, permitiendo así el acceso a los elementos `document` o `window`, para manipular el documento y sus partes.
- La programación con DOM hace que algo tan simple como abrir un mensaje de alerta sea posible usando la función `alert()` del objeto `window`, o permite el uso de métodos más sofisticados para crear nuevo contenido, por ejemplo:

# ¿Cómo se accede a DOM?, II

```
1 | <body onload="window.alert('Bienvenido a mi página!');">
```

- Esta función crea un nuevo elemento h1, le pone texto y después lo agrega al árbol del documento:

```
<html>
  <head>
    <script>
      // ejecuta esta función cuando se cargue el documento
      window.onload = function() {

        // crea dinámicamente un par de elementos HTML en una página vacía
        var heading = document.createElement("h1");
        var heading_text = document.createTextNode("el texto que desee");
        heading.appendChild(heading_text);
        document.body.appendChild(heading);
      }
    </script>
  </head>
  <body>
  </body>
</html>
```