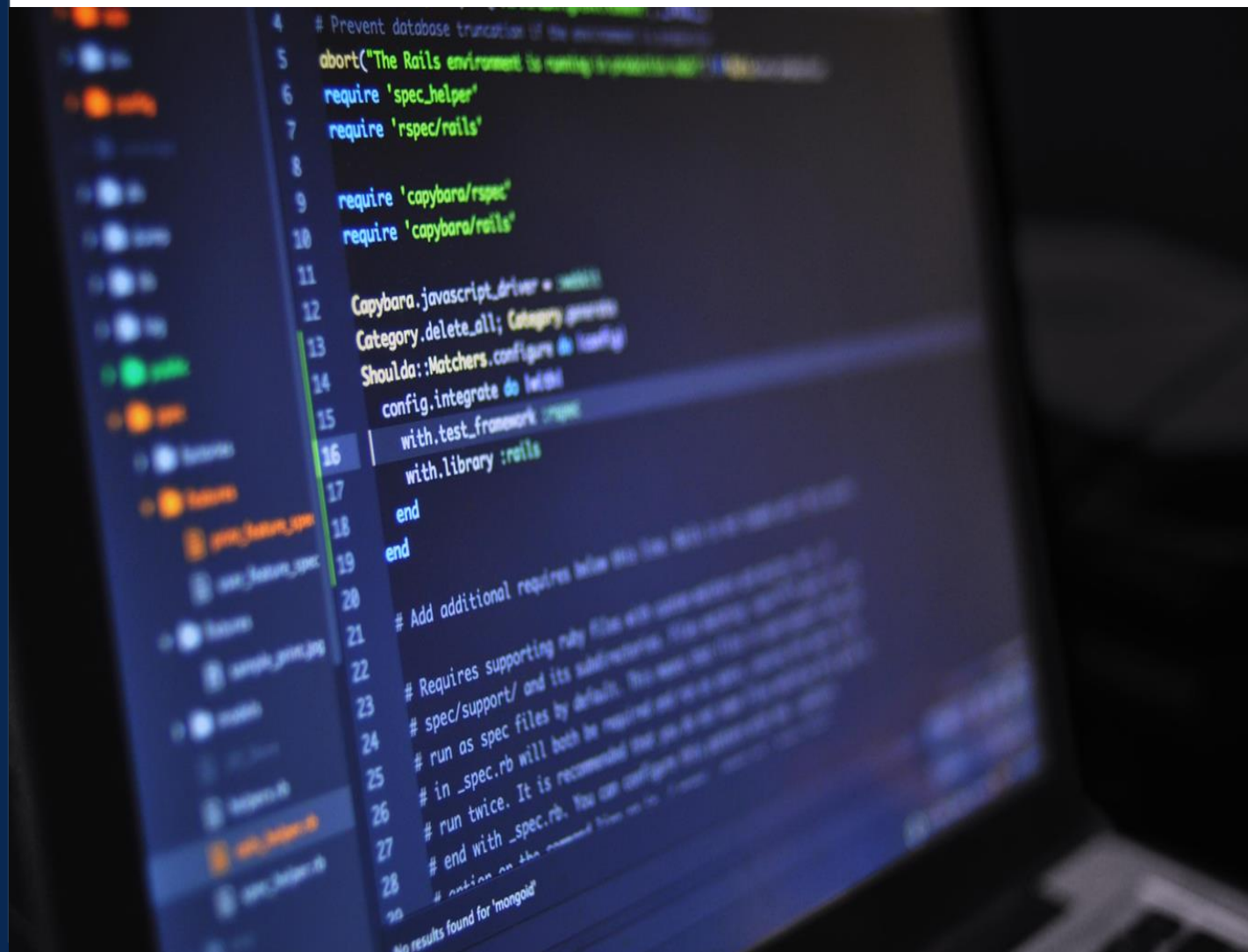


# Bloque TypeScript



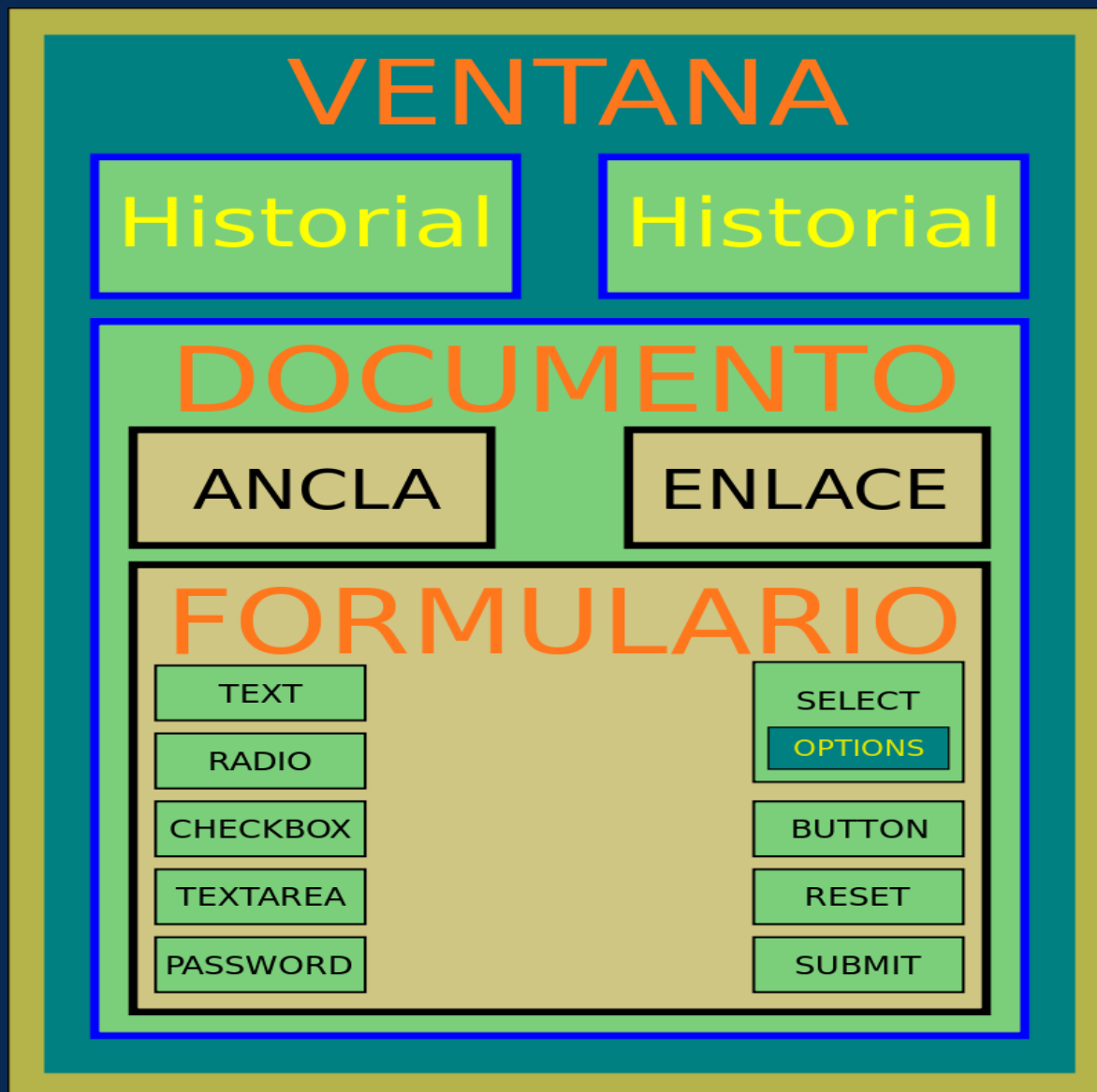
## Tema 5: DOM (Document Object Model) BOM (Browser Object Model)



# ¿QUÉ ES DOM?



El llamado DOM (Document Object Model) es un modelo que permite tratar un documento Web XHTML como si fuera XML, navegando por los nodos existentes que forman la página, pudiendo manipular sus atributos e incluso crear nuevos elementos.



A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos mencionados, que es para lo que se diseñó principalmente.

# El objeto DOCUMENT

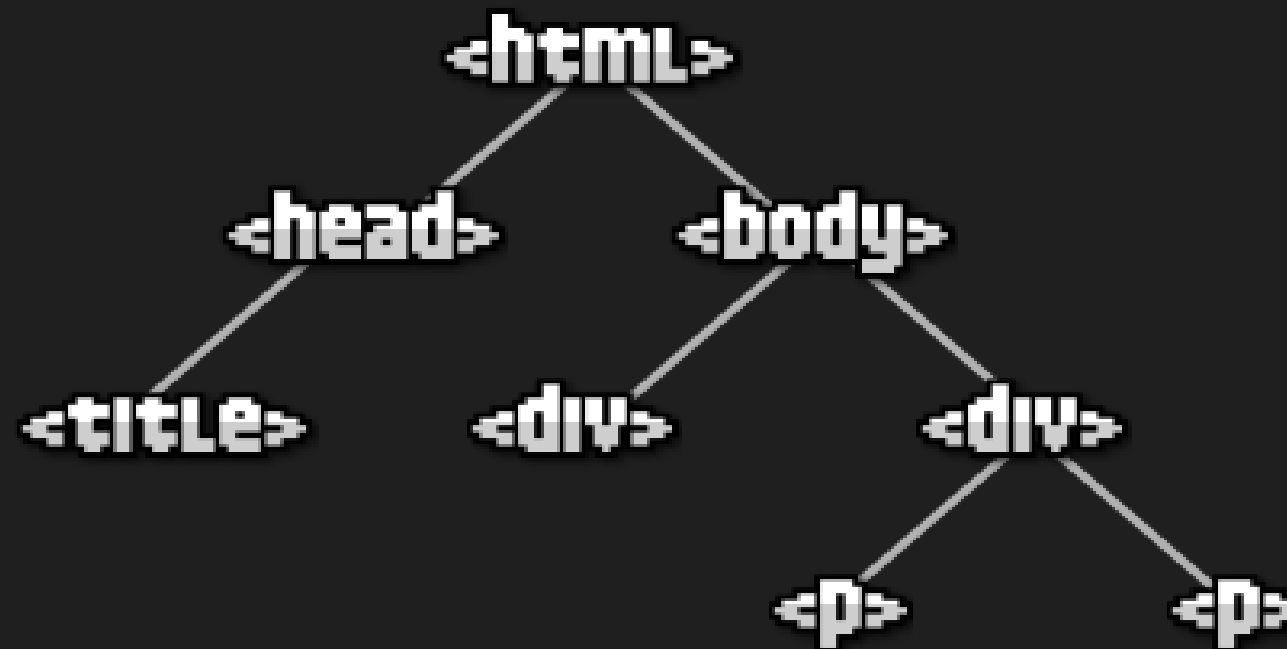
TS

En Javascript, la forma de acceder al DOM es a través de un objeto llamado document, que representa el árbol DOM de la página o, más concretamente, la página de la pestaña del navegador donde nos encontramos.

```
<html>
<head>
  <title>Title</title>
</head>
<body>
  <div></div>
  <div>
    <p>Párrafo 1</p>
    <p>Párrafo 2</p>
  </div>
</body>
</html>
```

HTML

DOM



# El objeto document



En su interior pueden existir varios tipos de elementos, pero principalmente serán :

- **Element:** no es más que la representación genérica de una etiqueta: `HTMLElement`.
- **Node:** es una unidad más básica, la cuál puede ser `Element` o un nodo de texto.

Tipo de dato genérico	Tipo específico	Etiqueta	Descripción
<b>ELEMENT</b> <code>HTMLElement</code>	<code>HTMLDivElement</code>	<code>&lt;div&gt;</code>	Etiqueta divisoria (en bloque).
<b>ELEMENT</b> <code>HTMLElement</code>	<code>HTMLSpanElement</code>	<code>&lt;span&gt;</code>	Etiqueta divisoria (en línea).
<b>ELEMENT</b> <code>HTMLElement</code>	<code>HTMLImageElement</code>	<code>&lt;img&gt;</code>	Imagen.
<b>ELEMENT</b> <code>HTMLElement</code>	<code>HTMLAudioElement</code>	<code>&lt;audio&gt;</code>	Contenedor de audio.

# Seleccionar elementos del DOM

TS

## getElementById()

Esta función devuelve un elemento DOM del subárbol cuyo identificador sea el indicado en la cadena "identificador".

```
<!DOCTYPE html>
<html>
<body>

<h1>Document Object Model</h1>
<h2>EL Metodo getElementById()</h2>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hola Mundo";
</script>

</body>
</html>
```

Document Object Model

EL Metodo getElementById()

Hola Mundo

# Seleccionar elementos del DOM

TS

## **getElementsByClassName()**

Permite buscar los elementos que tengan la clase especificada en class. Es importante darse cuenta del matiz de que el método tiene `getElements` en plural, es decir, puede devolver varias clases ya que al contrario que los `id`, pueden existir varios elementos con la misma clase.

# Seleccionar elementos del DOM

## getElementsByClassName()

TS

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
border: 1px solid black;
margin: 5px;
}
</style>
</head>
<body>

<div class="ejemplo">
  <p>P es el primer elemento en div con clase="ejemplo". Indice Div's es 0.</p>
</div>

<div class="ejemplo color">
  <p>P es el primer elemento en div con clase="ejemplo color". Indice Div's es 0.</p>
</div>

<div class="ejemplo color">
  <p>P es el segundo elemento en div con clase="ejemplo color". Indice Div's es 1</p>
</div>

  <p>Pulsar el botón para cambiar el color de fondo del primer elemento con clase "ejemplo" y "color".</p>

  <button onclick="miFuncion()">PULSA</button>

  <p><strong>Nota:</strong> El metodo getElementsByClassName() no es soportado pr Internet Explorer 8.</p>
<script>
function miFuncion() {
  var x = document.getElementsByClassName("ejemplo color");
  x[0].style.backgroundColor = "red";
}
</script>
</body>
</html>
```

# Seleccionar elementos del DOM



## getElementsByTagName()

P es el primer elemento en div con clase="ejemplo". Indice Div's es 0.

P es el primer elemento en div con clase="ejemplo color". Indice Div's es 0.

P es el segundo elemento en div con clase="ejemplo color". Indice Div's es 1

Pulsar el botón para cambiar el color de fondo del primer elemento con clase "ejemplo" y "color".

PULSA

**NOtA:** El metodo `getElementsByTagName()` no es soportado pr Internet Explorer 8.

P es el primer elemento en div con clase="ejemplo". Indice Div's es 0.

P es el primer elemento en div con clase="ejemplo color". Indice Div's es 0.

P es el segundo elemento en div con clase="ejemplo color". Indice Div's es 1

Pulsar el botón para cambiar el color de fondo del primer elemento con clase "ejemplo" y "color".

PULSA

**NOtA:** El metodo `getElementsByTagName()` no es soportado pr Internet Explorer 8.



# Seleccionar elementos del DOM



## getElementsByClassName()

```
<script>
function miFuncion() {
    var x = document.getElementsByClassName("ejemplo color");
    var i;
    for (i = 0; i < x.length; i++) {
        x[i].style.backgroundColor = "red";
    }
}
</script>
```

P es el primer elemento en div con clase="ejemplo". Indice Div's es 0.

P es el primer elemento en div con clase="ejemplo color". Indice Div's es 0.

P es el segundo elemento en div con clase="ejemplo color". Indice Div's es 1

P es el primer elemento en div con clase="ejemplo". Indice Div's es 0.

P es el primer elemento en div con clase="ejemplo color". Indice Div's es 0.

P es el segundo elemento en div con clase="ejemplo color". Indice Div's es 1

# Seleccionar elementos del DOM

TS

## getElementsByTagName()

Devuelve una colección de todos los elementos con un nombre de etiqueta especificado.

```
<!DOCTYPE html>
<html>
<body>

<h1>Document Object Model</h1>
<h2>El metodo getElementsByTagName() </h2>

<p>Esto es una elemento p</p>

<p>Este es otro elemento p</p>

<p>Este también es un elemento p.
Cambie el color de fondo de todos los p elementos de este documento</p>

<button onclick="miFuncion()">Pulsar</button>
<script>
function miFuncion() {
  const collection = document.getElementsByTagName("P");
  for (let i = 0; i < collection.length; i++) {
    collection[i].style.backgroundColor = "red";
  }
}
</script>

</body>
</html>
```

## Document Object Model

### El metodo getElementsByTagName()

Esto es una elemento p

Este es otro elemento p

Este también es un elemento p. Cambie el color de fondo de todos los p elementos de este documento

Pulsar

## Document Object Model

### El metodo getElementsByTagName()

Esto es una elemento p

Este es otro elemento p

Este también es un elemento p. Cambie el color de fondo de todos los p elementos de este documento

Pulsar

# Funciones para crear/borrar nodos



## removeChild(nodo)

```
<!DOCTYPE html>
<html>
<body>

<!-- Tenga en cuenta que los elementos <li> dentro de <ul> no tienen sangría (espacios en blanco).
Si lo fueran, el primer nodo hijo de <ul> sería un nodo de texto.
-->
<ul id="miLista"><li>Cafe</li><li>Te</li><li>leche</li></ul>

<p>Haga clic en el botón para eliminar el primer elemento de la lista.</p>

<button onclick="miFuncion()">Pulsar</button>

<script>
function miFuncion() {
    var lista = document.getElementById("miLista");
    lista.removeChild(lista.childNodes[0]);
}
</script>

</body>
</html>
```

- Cafe
- Te
- leche

Haga clic en el botón para eliminar el primer elemento de la lista.

Pulsar

- Te
- leche

Haga clic en el botón para eliminar el primer elemento de la lista.

Pulsar

# Seleccionar elementos del DOM

TS

## appendChild()

Recibe un nodo y lo incluye como nodo hijo del padre.

```
<html>
<body>
  <h1>El Elemento Objeto</h1>
  <h2>Metodo appendChild()</h2>

  <ul id="miLista">
    <li>Cafe</li>
    <li>Te</li>
  </ul>

  <p>Pulsa "Añadir" para añadir un elemto al final de la lista:</p>

  <button onclick="miFuncion()">AÑADIR</button>

<script>
function miFuncion() {

  // Crear un elemento en la lista:
  const node = document.createElement("li");

  // Crear el texto del nuevo elemento:
  const textnode = document.createTextNode("Water");

  // Crear el nuevo elemento en la lista:
  node.appendChild(textnode);

  // Añadir el nuevo elemento en la lista:
  document.getElementById("miLista").appendChild(node);
}
</script>

</body>
</html>
```

## El Elemento Objeto

### Metodo appendChild()

- Cafe
- Te

Pulsa "Añadir" para añadir un elemto al final de la lista:

AÑADIR

## El Elemento Objeto

### Metodo appendChild()

- Cafe
- Te
- Water

Pulsa "Añadir" para añadir un elemto al final de la lista:

AÑADIR

# Introducción a BOM



Mediante BOM, es posible redimensionar y mover la ventana del navegador, modificar el texto que se muestra en la barra de estado y realizar muchas otras manipulaciones no relacionadas con el contenido de la página HTML.

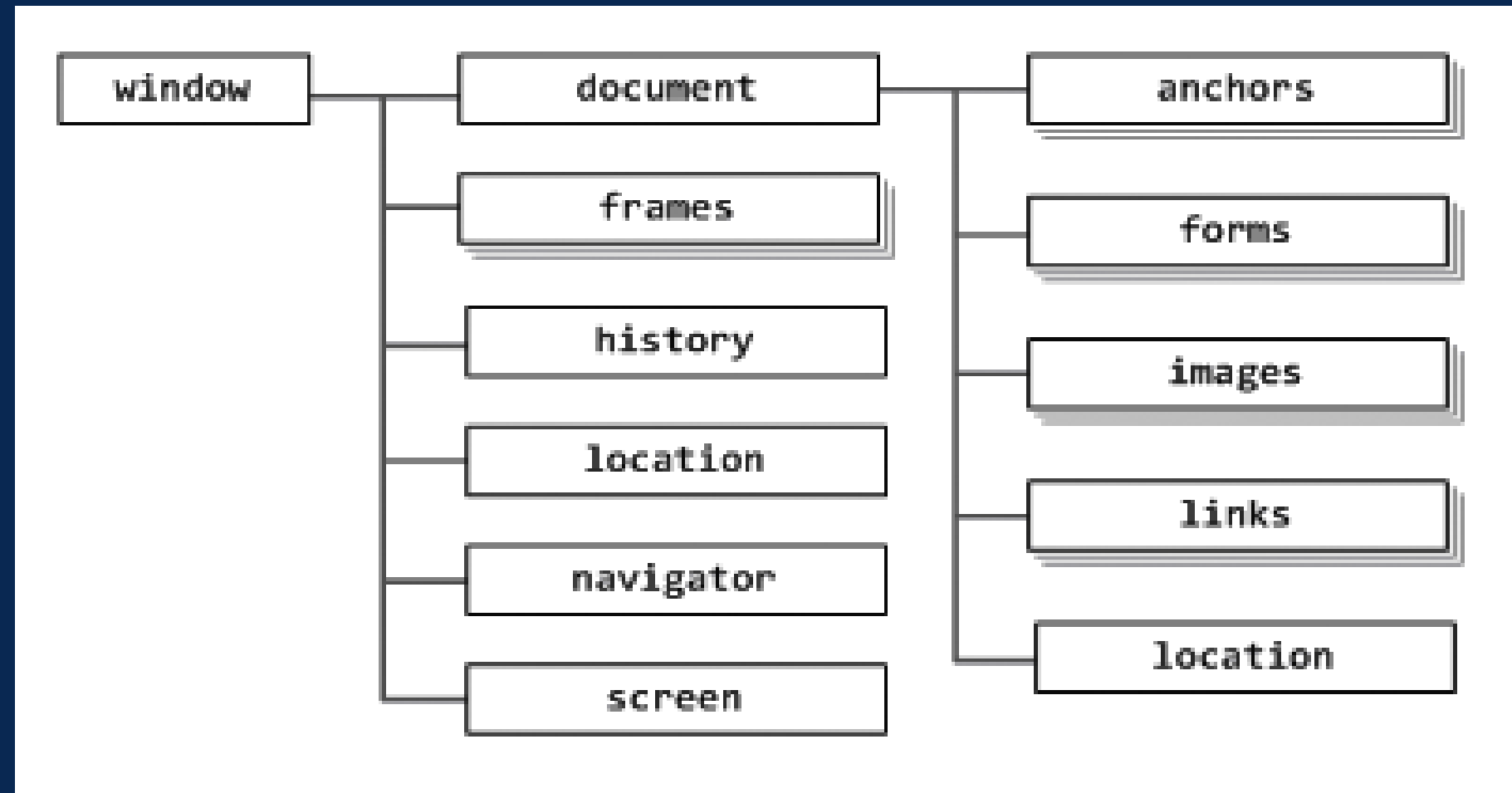
Algunos de los elementos que forman el BOM son los siguientes:

- Crear, mover, redimensionar y cerrar ventanas de navegador.
- Obtener información sobre el propio navegador.
- Propiedades de la página actual y de la pantalla del usuario.
- Gestión de cookies.
- Objetos ActiveX en Internet Explorer.

# Introducción a BOM



El BOM está compuesto por varios objetos relacionados entre sí. El siguiente esquema muestra los objetos de BOM y su relación:



# O b j e t o   w i n d o w



El objeto **window** representa una ventana abierta en un navegador. Si un documento contiene marcos (`<frame>` o `<iframe>`), el navegador crea un objeto **window** para el documento HTML, y un objeto **window** adicional para cada marco.



# Objeto window



Propiedades del objeto Window	
Propiedad	Descripción
<code>closed</code>	Devuelve un valor <code>Boolean</code> indicando cuando una ventana ha sido cerrada o no.
<code>defaultStatus</code>	Ajusta o devuelve el valor por defecto de la barra de estado de una ventana.
<code>document</code>	Devuelve el objeto <code>document</code> para la ventana.
<code>frames</code>	Devuelve un array de todos los marcos (incluidos <code>iframes</code> ) de la ventana actual.
<code>history</code>	Devuelve el objeto <code>history</code> de la ventana.
<code>length</code>	Devuelve el número de <code>frames</code> (incluyendo <code>iframes</code> ) que hay en dentro de una ventana.
<code>location</code>	Devuelve la Localización del objeto ventana (URL del fichero).
<code>name</code>	Ajusta o devuelve el nombre de una ventana.
<code>navigator</code>	Devuelve el objeto <code>navigator</code> de una ventana.
<code>opener</code>	Devuelve la referencia a la ventana que abrió la ventana actual.
<code>parent</code>	Devuelve la ventana padre de la ventana actual.
<code>self</code>	Devuelve la ventana actual.
<code>status</code>	Ajusta el texto de la barra de estado de una ventana.



# Objeto window

TS

Métodos del objeto Window	
Método	Descripción
<code>alert()</code>	Muestra una ventana emergente de alerta y un botón de aceptar.
<code>blur()</code>	Elimina el foco de la ventana actual.
<code>clearInterval()</code>	Resetea el cronómetro ajustado con <code>setInterval()</code> .
<code>setInterval()</code>	Llama a una función o evalúa una expresión en un intervalo especificado (en milisegundos).
<code>close()</code>	Cierra la ventana actual.
<code>confirm()</code>	Muestra una ventana emergente con un mensaje, un botón de aceptar y un botón de cancelar.
<code>focus()</code>	Coloca el foco en la ventana actual.
<code>open()</code>	Abre una nueva ventana de navegación.
<code>prompt()</code>	Muestra una ventana de diálogo para introducir datos.

# Objeto window



## Open() / closed()

```
<html>
<body>
  <h1>El objeto Window</h1>
  <h2>Propiedad closed</h2>
  <p><button onclick="openWin()">Abrir "mi ventana"</button></p>
  <p><button onclick="closeWin()">Cerrar "mi ventana"</button></p>
  <p><button onclick="checkWin()">Esta "mi ventana" cerrada?</button></p>
  <div id="demo"></div>
  <script>
    let miWindow;
    function openWin() {
      miWindow = window.open("", "mi ventana", "width=400,height=200");
    }
    function closeWin() {
      if (miWindow) {
        miWindow.close();
      }
    }
    function checkWin() {
      let text = "";
      if (!miWindow) {
        text = "Nunca se ha abierto!";
      } else {
        if (miWindow.closed) {
          text = "Esta cerrada.";
        } else {
          text = "Esta abierta.";
        }
      }
      document.getElementById("demo").innerHTML = text;
    }
  </script>
</body>
</html>
```

# Objeto window



## Open() / closed()

### El objeto Window

#### Propiedad closed

Abrir "mi ventana"

Cerrar "mi ventana"

Esta "mi ventana" cerrada?

### El objeto Window

#### Propiedad closed

Abrir "mi ventana"

Cerrar "mi ventana"

Esta "mi ventana" cerrada?

Esta abierta.

## El objeto Window

### Propiedad closed

Abrir "mi ventana"

Cerrar "mi ventana"

Esta "mi ventana" cerrada?

Nunca se ha abierto!

# Objeto location



El objeto **location** contiene información referente a la URL actual. Este objeto, es parte del objeto **window** y accedemos a él a través de la propiedad **window.location**.

Propiedades del objeto Location

Propiedad	Descripción
<code>hash</code>	Cadena que contiene el nombre del enlace, dentro de la URL.
<code>host</code>	Cadena que contiene el nombre del servidor y el número del puerto, dentro de la URL.
<code>hostname</code>	Cadena que contiene el nombre de dominio del servidor (o la dirección IP), dentro de la URL.
<code>href</code>	Cadena que contiene la URL completa.
<code>pathname</code>	Cadena que contiene el camino al recurso, dentro de la URL.
<code>port</code>	Cadena que contiene el número de puerto del servidor, dentro de la URL.
<code>protocol</code>	Cadena que contiene el protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
<code>search</code>	Cadena que contiene la información pasada en una llamada a un script, dentro de la URL.

Métodos del objeto Location

Método	Descripción
<code>assign()</code>	Carga un nuevo documento.
<code>reload()</code>	Vuelve a cargar la URL especificada en la propiedad <code>href</code> del objeto <code>location</code> .
<code>replace()</code>	Reemplaza el historial actual mientras carga la URL especificada en <code>cadenaURL</code> .

# Objeto location

TS

```
<HTML>
<HEAD>
<title>Ejemplo de JavaScript</title>
</HEAD>
<BODY>
<script LANGUAGE="JavaScript">
<!--
document.write("Location <b>href</b>: " + location.href + "<br>");
document.write("Location <b>host</b>: " + location.host + "<br>");
document.write("Location <b>hostname</b>: " + location.hostname + "<br>");
document.write("Location <b>pathname</b>: " + location.pathname + "<br>");
document.write("Location <b>port</b>: " + location.port + "<br>");
document.write("Location <b>protocol</b>: " + location.protocol + "<br>");
//-->
</script>
</BODY>
</HTML>
```

Location **href**: file:///K:/2DAW/00\_TYPESCRIPT/EJEMPLOS\_MIIOS/02\_ventana\_LOCALIZACION.html  
Location **host**:  
Location **hostname**:  
Location **pathname**: /K:/2DAW/00\_TYPESCRIPT/EJEMPLOS\_MIIOS/02\_ventana\_LOCALIZACION.html  
Location **port**:  
Location **protocol**: file:

# Objeto navigator



Este objeto simplemente nos da información relativa al navegador que esté utilizando el usuario.

## Propiedades

- ✓ `appName`. Cadena que contiene el nombre del código del cliente.
- ✓ `appName`. Cadena que contiene el nombre del cliente.
- ✓ `appVersion`. Cadena que contiene información sobre la versión del cliente.
- ✓ `language`. Cadena de dos caracteres que contiene información sobre el idioma de la versión del cliente.
- ✓ `mimeType`. Array que contiene todos los tipos MIME soportados por el cliente. A partir de NS 3.
- ✓ `platform`. Cadena con la plataforma sobre la que se está ejecutando el programa cliente.
- ✓ `plugins`. Array que contiene todos los plug-ins soportados por el cliente. A partir de NS 3.
- ✓ `userAgent`. Cadena que contiene la cabecera completa del agente enviada en una petición HTTP. Contiene la información de las propiedades `appName` y `appVersion`.

## Métodos

- ✓ `javaEnabled()`. Devuelve `true` si el cliente permite la utilización de Java, en caso contrario, devuelve false.

# Objeto navigator

TS

```
<html>
<body>

<div id="demo"></div>

<script>
var txt = "";
txt += "<p>Browser CodeName: " + navigator.appCodeName + "</p>";
txt += "<p>Browser Name: " + navigator.appName + "</p>";
txt += "<p>Browser Version: " + navigator.appVersion + "</p>";
txt += "<p>Cookies Enabled: " + navigator.cookieEnabled + "</p>";
txt += "<p>Browser Language: " + navigator.language + "</p>";
txt += "<p>Browser Online: " + navigator.onLine + "</p>";
txt += "<p>Platform: " + navigator.platform + "</p>";
txt += "<p>User-agent header: " + navigator.userAgent + "</p>";

document.getElementById("demo").innerHTML = txt;
</script>

</body>
</html>
```

Browser CodeName: Mozilla

Browser Name: Netscape

Browser Version: 5.0 (Windows)

Cookies Enabled: true

Browser Language: es-ES

Browser Online: true

Platform: Win32

User-agent header: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0

# Objeto screen



Contiene información acerca de la pantalla del visitante.

**No existe una norma pública que se aplica al objeto de imagen, pero todos los navegadores soportan.**

Propiedad	Descripción
availHeight	Devuelve la altura de la pantalla (excluding the Windows Taskbar)
availWidth	Devuelve el ancho de la pantalla (excluding the Windows Taskbar)
colorDepth	Devuelve la profundidad de bits de la paleta de colores para la visualización de imágenes
height	Devuelve la altura total de la pantalla
pixelDepth	Devuelve la resolución de color (in bits per pixel) de la pantalla
width	Devuelve el ancho total de la pantalla



# Objeto screen

TS

```
<html>
<body>

  <h3>Your Screen:</h3>

  <div id="demo"></div>

<script>
  var txt = "";
  txt += "<p>Total width/height: " + screen.width + "*" + screen.height + "</p>";
  txt += "<p>Available width/height: " + screen.availWidth + "*" + screen.availHeight + "</p>";
  txt += "<p>Color depth: " + screen.colorDepth + "</p>";
  txt += "<p>Color resolution: " + screen.pixelDepth + "</p>";

  document.getElementById("demo").innerHTML = txt;
</script>

</body>
</html>
```

## Your Screen:

Total width/height: 1920\*1080

Available width/height: 1920\*1040

Color depth: 24

Color resolution: 24