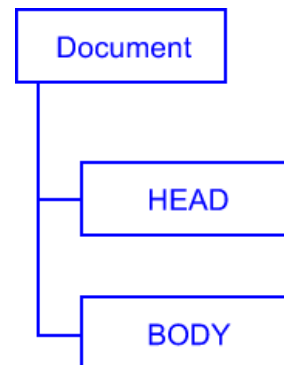


DOM Document

Document es la raíz del árbol de nodos de cualquier página XHTML. Es un nodo de tipo especial.

- A partir de ese nodo raíz, cada etiqueta XHTML se transforma en un nodo de tipo "Elemento".
- La conversión de etiquetas en nodos se realiza de forma jerárquica.
- De esta forma, del nodo raíz solamente pueden derivar los nodos HEAD y BODY.
- La interfaz Document describe los métodos y propiedades comunes para cualquier clase de documento.
- Dependiendo del tipo de documento (ej. HTML, XML, SVG, ...), hay disponible una API más extensa: los documentos HTML, que se sirven con el tipo de contenido text/html, también implementan la interfaz HTMLDocument, mientras que los documentos XML y SVG implementan la interfaz XMLDocument.
- La interfaz Document también hereda de las interfaces Node y EventTarget.



Propiedades

- **activeElement**
Devuelve el elemento actualmente enfocado en el documento.
- **anchors**
Devuelve una colección de todos los elementos <a> en el documento que tienen un atributo de nombre
- **applets**
Devuelve una colección de todos los elementos <applet> en el documento.
- **baseURI**
Devuelve el URI base absoluto de un documento.
- **body**
Establece o devuelve el cuerpo del documento (el elemento <body>)
- **cookie**
Devuelve todos los pares de cookies de nombre / valor en el documento.
- **characterSet**
Devuelve la codificación de caracteres del documento.
- **defaultView**
Devuelve el objeto de ventana asociado con un documento, o nulo si no hay ninguno disponible.
- **designMode**
Controla si todo el documento debe ser editable o no.
- **Doctype**
Devuelve la Declaración de tipo de documento asociada con el documento.
- **documentElement**
Devuelve el elemento de documento del documento (el elemento <html>)
- **documentMode**
Devuelve el modo utilizado por el navegador para representar el documento.
- **documentURI**
Establece o devuelve la ubicación del documento
- **domain**
Devuelve el nombre de dominio del servidor que cargó el documento.
- **embeds**
Devuelve una colección de todos los elementos <embed> del documento.

- **forms**
Devuelve una colección de todos los elementos <form> en el documento.
- **fullscreenElement**
Devuelve el elemento actual que se muestra en modo de pantalla completa
- **head**
Devuelve el elemento <head> del documento.
- **images**
Devuelve una colección de todos los elementos en el documento.
- **implementation**
Devuelve el objeto DOMImplementation que maneja este documento.
- **inputEncoding**
Devuelve la codificación, el juego de caracteres, utilizado para el documento.
- **lastModified**
Devuelve la fecha y hora en que se modificó por última vez el documento.
- **links**
Devuelve una colección de todos los elementos <a> y <area> del documento que tienen un atributo href
- **readyState**
Devuelve el estado (cargando) del documento
- **referrer**
Devuelve la URL del documento que cargó el documento actual.
- **scripts**
Devuelve una colección de elementos <script> en el documento.
- **strictErrorChecking**
Establece o devuelve si la verificación de errores se aplica o no
- **title**
Establece o devuelve el título del documento.
- **URL**
Devuelve la URL completa del documento HTML.

Métodos

- **addEventListener()**
Adjunta un controlador de eventos al documento
- **adoptNode()**
Adopta un nodo de otro documento
- **close()**
Cierra el flujo de salida previamente abierto con document.open ()
- **createAttribute()**
Crea un nodo de atributo
- **createComment()**
Crea un nodo Comentario con el texto especificado
- **createDocumentFragment()**
Crea un nodo DocumentFragment vacío
- **createElement()**
Crea un nodo Elemento
- **createEvent()**
Crea un nuevo evento

- **createTextNode()**
Crea un nodo de texto
- **execCommand()**
Invoca la operación de portapapeles especificada en el elemento que tiene el foco actualmente.
- **fullscreenEnabled()**
Devuelve un valor booleano que indica si el documento se puede ver en modo de pantalla completa.
- **getElementById()**
Devuelve el elemento que tiene el atributo ID con el valor especificado
- **getElementsByClassName()**
Devuelve un HTMLCollection que contiene todos los elementos con el nombre de clase especificado
- **getElementsByName()**
Devuelve una colección HTMLC que contiene todos los elementos con un nombre específico
- **getElementsByTagName()**
Devuelve un HTMLCollection que contiene todos los elementos con el nombre de etiqueta especificado
- **hasFocus()**
Devuelve un valor booleano que indica si el documento tiene el foco.
- **importNode()**
Importa un nodo de otro documento
- **normalize()**
Elimina los nodos de texto vacíos y une los nodos adyacentes
- **normalizeDocument()**
Elimina los nodos de texto vacíos y une los nodos adyacentes
- **open()**
Abre una secuencia de salida HTML para recopilar la salida de document.write()
- **querySelector()**
Devuelve el primer elemento que coincide con un selector o selectores CSS especificados en el documento.
- **querySelectorAll()**
Devuelve una NodeList estática que contiene todos los elementos que coinciden con un selector o selectores CSS especificados en el documento.
- **removeEventListener()**
Elimina un controlador de eventos del documento que se ha adjuntado con el método addEventListener()
- **renameNode()**
Cambia el nombre del nodo especificado
- **write()**
Escribe expresiones HTML o código JavaScript en un documento.
- **writeln()**
Igual que write(), pero agrega un carácter de nueva línea después de cada declaración