

# Objeto document

Autor: Jesús Mejías Leiva



# Document



Representa cualquier **página web** cargada en el navegador y sirve como **punto de entrada** al contenido de la página (El árbol **DOM**).

Describe los **métodos** y **propiedades** comunes para cualquier clase de **documento** (ej. HTML, XML, SVG, ...)

**Document** cuelga del objeto padre **window**.

---

# Propiedades

## Generales

`Document.characterSet`

`Document.doctype`

`Document.styleSheets`

`Document.documentURI`

`ParentNode.childElementCount`

`ParentNode.children`

---

# Document.characterSet

Devuelve el **juego de caracteres** que utiliza el documento.

```
let character = document.characterSet
```

```
// Posible salida
```

```
"UTF-8"
```

# Document.doctype

Devuelve la **definición de tipo de documento** actual, comúnmente conocido como **DTD**.

```
let doctype = document.doctype
```

// Salida

El tipo de definición de documento, junto con sus propiedades.

# Document.styleSheets

Devuelve una **lista** de **objetos** de tipo **stylesheet** para las **hojas de estilo** que están específicamente **enlazadas** en el **documento**.

```
let styleSheets = document.styleSheets
```

```
// Salida
```

Lista de objeto de tipo stylesheet.

La propiedad **.length** - devuelve el número de objetos de tipo stylesheet contenidos en el objeto.

# Document.documentURI

Devuelve la **URI** del documento.

```
let uri = document.documentURI
```

```
// Posible salida
```

```
"https://github.com/susomejias"
```

# ParentNode.childElementCount

Devuelve el número de elementos del ParentNode.

```
let childElemCount = document.childElementCount
```

```
// Salida
```

El número de elementos hijos del padre



# ParentNode.children

Devuelve una **HTMLCollection**, con los **elementos hijos** del **ParentNode**.

```
let children = document.children
```

// Salida

Una collectionHTML, con los elementos hijos del padre.

# Propiedades

Específicas documentos html

Document.bgColor

Document.body

Document.location

Document.forms

Document.URL

Document.images

Document.lastModified

Document.readyState

---

# Document.bgColor

Extrae o define el **color** de **fondo** del documento.

```
// extrae el color de fondo  
  
let bgColor = document.bgColor  
  
// define el color de fondo  
  
document.bgColor = "#FFF"
```

```
// Posible salida en caso de extraer el color de fondo  
  
"#FFF"
```

# Document.body

Devuelve el elemento `<body></body>` del documento.

```
let body = document.body
```

// Salida

El elemento body del documento y sus propiedades.

# Document.location

Devuelve un objeto **location** con sus **propiedades**.

```
let location = document.location
```

```
// Salida
```

Un objeto location con sus propiedades

# Document.forms

Devuelve una **collectionHTML**, con los distintos **formularios** del **documento**.

```
let forms = document.forms
```

// Salida

Una collectionHTML con los distintos formularios.

# Document.URL

Devuelve la **URL** del documento.

```
let url = document.URL
```

```
// Posible salida
```

```
"https://www.google.com/"
```

# Document.images

Devuelve una **collectionHTML**, con los **elementos images**.

```
let images = document.images
```

// Salida

Una collectionHTML con los elementos images.



# Document.lastModified

Devuelve la última fecha de modificación del documento.

```
let lastModified = document.lastModified.
```

```
// Posible salida
```

```
"10/10/2018 09:18:19"
```

# Document.readyState

Devuelve el **estado de carga** del documento.

```
let readyState = document.readyState
```

// Posibles salidas

**loading** -> El documento está cargando.

**interactive** -> El documento ha terminado de cargar y ha sido analizado pero los sub-recursos como imágenes, estilos y frames aún siguen cargando. El estado indica que el evento **DOMContentLoaded** ha sido disparado.

**complete** -> El documento y todos los sub-recursos han cargado completamente. El estado indica que el evento **load** ha sido disparado.

# Métodos

## Generales

`getElementsByClassName(String  
className)`

`getElementsByTagName(String  
tagName)`

`getElementById(String id)`

`querySelector(String selector)`

`querySelectorAll(String selector)`

---

# getElementsByClassName(String className)

Devuelve una **collectionHTML** con los **elementos** identificados con la **clase/s** indicada.

```
// elementos que tengan las clase class
```

```
let elements = document.getElementsByClassName("class");
```

```
// elementos que tengan las clase class y red a la vez
```

```
let elements = document.getElementsByClassName("class red");
```

# getElementsByTagName(String tagName)

Devuelve una **collectionHTML** con los **elementos** identificados con el **tag** indicado.

```
let elements = document.getElementsByTagName("a");
```

// Salida

Una collection con los elementos identificados por su tag.

# getElementById(String id)

Devuelve una **collectionHTML** con los **elementos** identificados con el **id** indicado.

```
let elements = document.getElementById("elementId");
```

**// Salida**

Una collection con los elementos identificados por su id.

# querySelector(String selector)

Devuelve el **primer elemento**, que **cumpla** las **coincidencias** con los **selectores** **indicados**. Los **selectores** funcionan de **igual** manera que en **css**.

```
let elements = document.querySelector(".a"); // obtenemos el primer elemento con la clase .a
```

```
let elements = document.querySelector("#b"); // obtenemos el elemento con el id #b
```

// Salida

Elemento que cumplan los requisitos de los selectores

# querySelectorAll(String selector)

Devuelve un **listado de nodos** que **cumplan** las **coincidencias** de los **selectores** **indicados**

```
let elements = document.querySelectorAll(".a"); // obtenemos los elementos con la clase .a  
let elements = document.querySelectorAll("#b"); // no tiene mucho sentido aunque funcionaria
```

// Salida

NodeList de elementos que cumplan los requisitos de los selectores.



# Métodos

Específicas documentos html

`Document.hasFocus()`

`Document.getElementsByName(String name)`

`Document.getSelection()`

---

# hasFocus()

Devuelve **true** si el foco está en algún lugar del documento.

```
let hasFocus = document.hasFocus();
```

// Posibles salidas

**true** -> El foco está en el documento.

**false** -> El foco no está en ningún lugar del documento.

# getElementsByTagName(String name)

Devuelve un **listado** con los **nodos** que **cumplan** los **requisitos** indicado en el **name**.

```
let elements = document.getElementsByTagName("submit");
```

```
// Salida
```

NodeList de elementos que cumplan los requisitos del name.

# getSelection()

Devuelve el **texto seleccionado** del documento.

```
let getSelection= document.getSelection();
```

// Posibles salidas

Un objeto selection con el texto seleccionado en el documento.

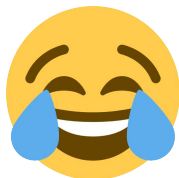
# Bibliografía

MDN WEB DOCS

---



Guarden silenciooo...



# ¡¡ Gracias por su atención !!

