document: generación de elementos HTML

Javier Sánchez López
Hugo Vicente Peligro

Índice

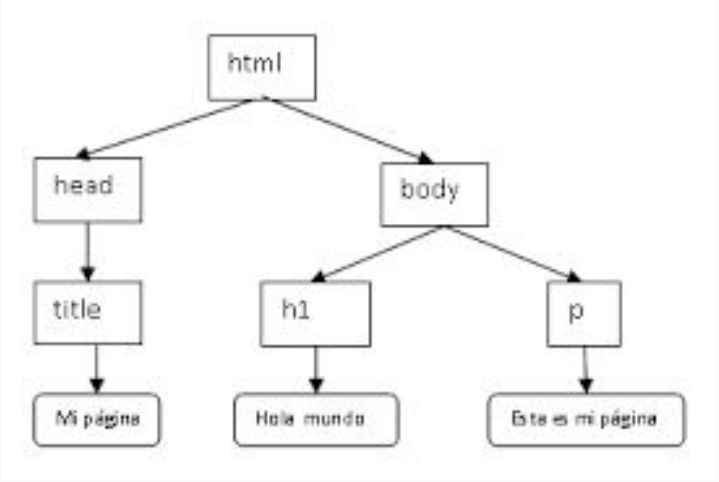
- Utilidad
- Constantes
- Propiedades
- Métodos

Utilidad

Utilidad

Es un objeto que representa cualquier página web cargada en el navegador y sirve como punto de entrada al contenido de la página web, que es el árbol DOM, permite obtener la URL de la página y crear nuevos elementos en el documento, también describe las propiedades y métodos comunes para cualquier tipo de documento.

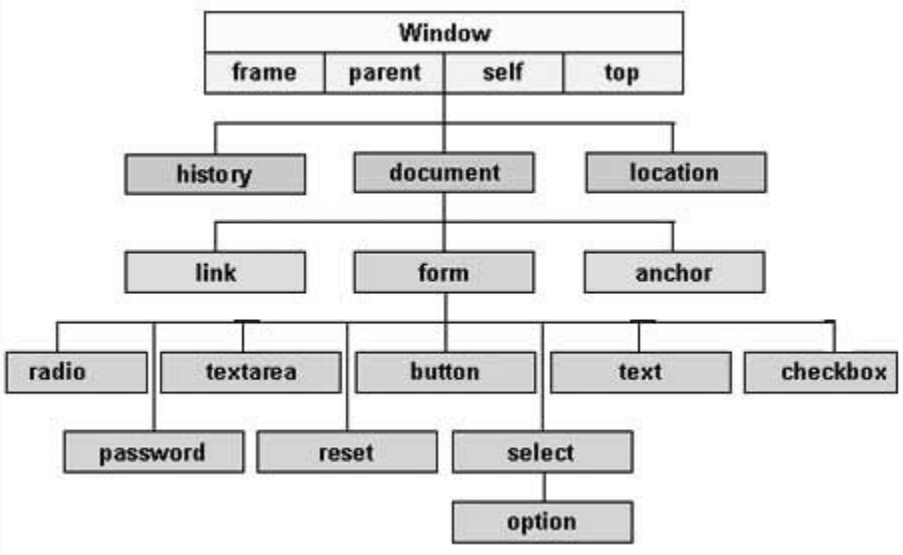
Constructor → new Document();



Constantes

Constantes

No tiene constantes pero utiliza instance properties que son variables heredadas de las interfaces Node y EventTarget, que devuelve valores que van cambiando dependiendo del documento



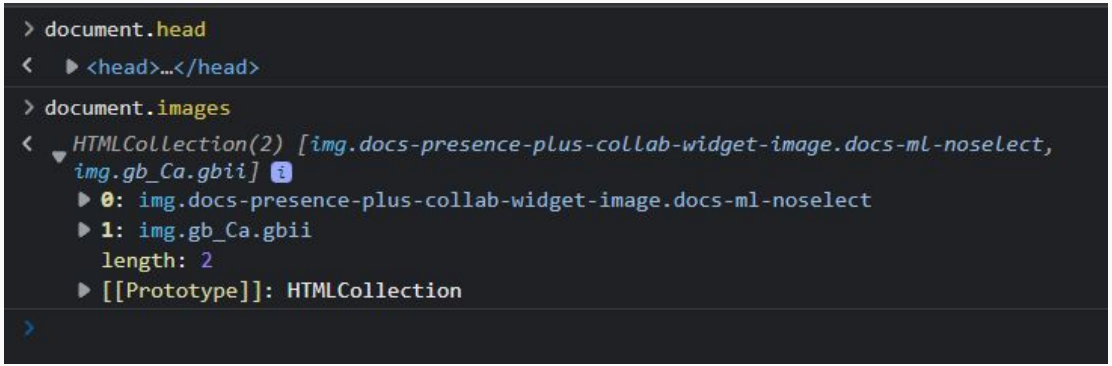
Propiedades

- **document.body** = devuelve el nodo body del documento actual.
- **document.activeElement** = Devuelve el Element actualmente activo.
- **document.doctype** = Devuelve el tipo de documento que es, por ejemplo <!DOCTYPE html>

```
> document.activeElement
<   ▶<button aria-haspopup="menu" type="button" class="button action has-icon
    theme-switcher-menu small" aria-expanded="true">…</button>
> document.body
<   ▶<body>…</body>
> document.doctype
<   <!DOCTYPE html>
```

Propiedades

- **document.head** = Devuelve el elemento <head> del documento actual.
- **document.images** = Devuelve un HTMLCollection de las imágenes del documento.



Propiedades

- **document.forms** = Devuelve un HTMLCollection de los elementos <form> del documento.

```
document.forms
▼HTMLCollection(2) [form.edition-selector__menu-preferences, form.js-search-portada]
  ▶0: form.edition-selector__menu-preferences
  ▶1: form.js-search-portada
    length: 2
  ▶[[Prototype]]: HTMLCollection
```

- **document.URL** = Devuelve la ubicación del documento como una cadena.

```
> document.URL
< 'https://www.marca.com/'
```

Métodos

Métodos

- addEventListener(): Añade un evento, pasado como parámetro, que afecta a todo el DOM.
- append(): Inserta un conjunto de objetos Node u objetos DOMString después del último elemento hijo del documento.
- removeChild(): Elimina el elemento pasado como parámetro, además devuelve el elemento borrado.
- replaceChild(): reemplaza el elemento segundo pasado por parámetro, por el primero pasado a dicho parámetro.

Métodos

- createElement(): Crea un nuevo elemento con el nombre pasado al parámetro.
- createTextNode(): Añade un nodo de texto
- createAttribute(): Crea un atributo para añadirlo a un nodo.

```
const node = document.getElementById("div1");
const a = document.createAttribute("my_attrib");
a.value = "newVal";
node.setAttributeNode(a);
console.log(node.getAttribute("my_attrib")); // "newVal"
```

Métodos

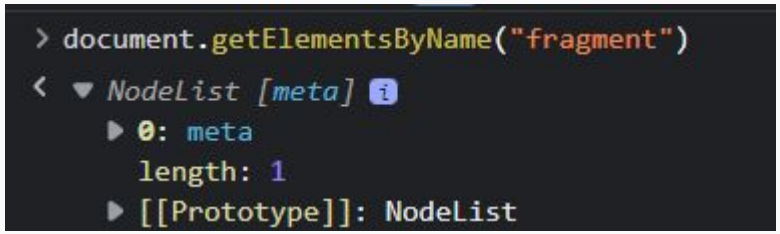
- open(): Permite la escritura en un documento, además te permite abrir urls si le pasas tres parámetros.
- write(): Escribe una cadena de texto dentro del hilo de un document abierto por document.open().
- close(): Cierra el flujo de escritura del documento.

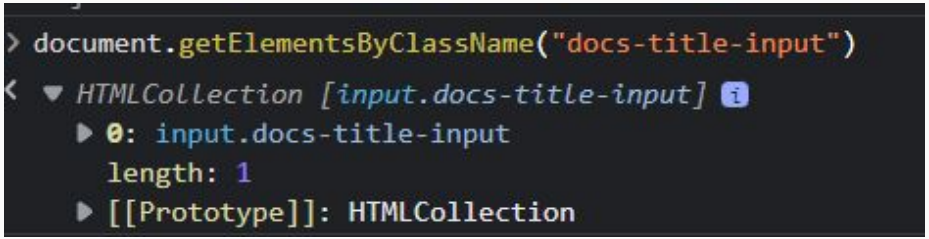
```
<span id="s1" style="color: red;"> Some text </span>

<script>
  const elem = document.getElementById("s1");
  alert(elem.style.cssText); // "color: red;"
</script>
```

Métodos

- getElementById(): Devuelve la referencias al elemento HTML, pasando como parámetro el id.
- getElementsByName(): Devuelve todas las referencias a elementos HTML, pasando como parámetro.
- getElementsByClassName(): Devuelve todas las referencias a elementos HTML, pasando como parámetro su clase.





Métodos

- getElementsByTagName(): Devuelve todas las referencias a elementos HTML, pasando como parámetro su nombre de etiqueta HTML.
- querySelector(): Devuelve el primer elemento del grupo de elementos del DOM que coincidan con el selector pasado como parámetro.
- querySelectorAll(): Devuelve un array de todos los elementos del árbol DOM que coincidan con el selector pasado como parámetro.

```
> document.getElementsByTagName("a")
< HTMLCollection(5) [a, a.gb_A.gb_Ma.gb_f,
  a#punch-html-viewer-link.punch-hidden-link,
  ▶ a#docs-a11y-aria-enable.docs-offscreen-z-index, a, punch-html-viewer-link:
  a#punch-html-viewer-link.punch-hidden-link, docs-a11y-aria-enable:
  a#docs-a11y-aria-enable.docs-offscreen-z-index]
```

Eventos

blur(): Cuando el elemento pierde el foco.

click(): El usuario hace clic sobre el elemento.

dblclick(): El usuario hace doble clic sobre el elemento.

focus(): El elemento gana el foco.

keydown(): El usuario presiona una tecla.

Eventos

keyup(): El usuario libera la tecla.

load(): El documento termina su carga.

mousedown(): El usuario presiona el botón del ratón en un elemento.

mousemove(): El usuario mueve el puntero del ratón sobre un elemento.

mouseout(): El usuario mueve el puntero fuera de un elemento.

mouseover(): El usuario mantiene el puntero sobre un elemento.

Eventos

mouseup(): El usuario libera el botón pulsado del ratón sobre un elemento.

unload(): El documento se descarga, bien porque se cierra la ventana, bien porque se navega a otra página.

Webgrafía

Document() - Web APIs | MDN (mozilla.org)

W3Schools Online Web Tutorials

JSxD Eventos (mywonderland.es)

¡Gracias por su atención espero que hayan aprendido!