¡Javascript!

guayerd beta hub

guayerd

Agenda del día

01

Introducción

Repaso métodos String, Number y Object.

02

DOM

Respuesta servidor

DOM

Consola y DOM

Nodos y Elementos

Métodos de acceso.

querySelector

querySelectorAll

Otros métodos de acceso

innerHTML

innerText

createElement

remove()

append()

03

Ejercitación

Mediante un prompt que el usuario vaya construyendo una página con elementos.

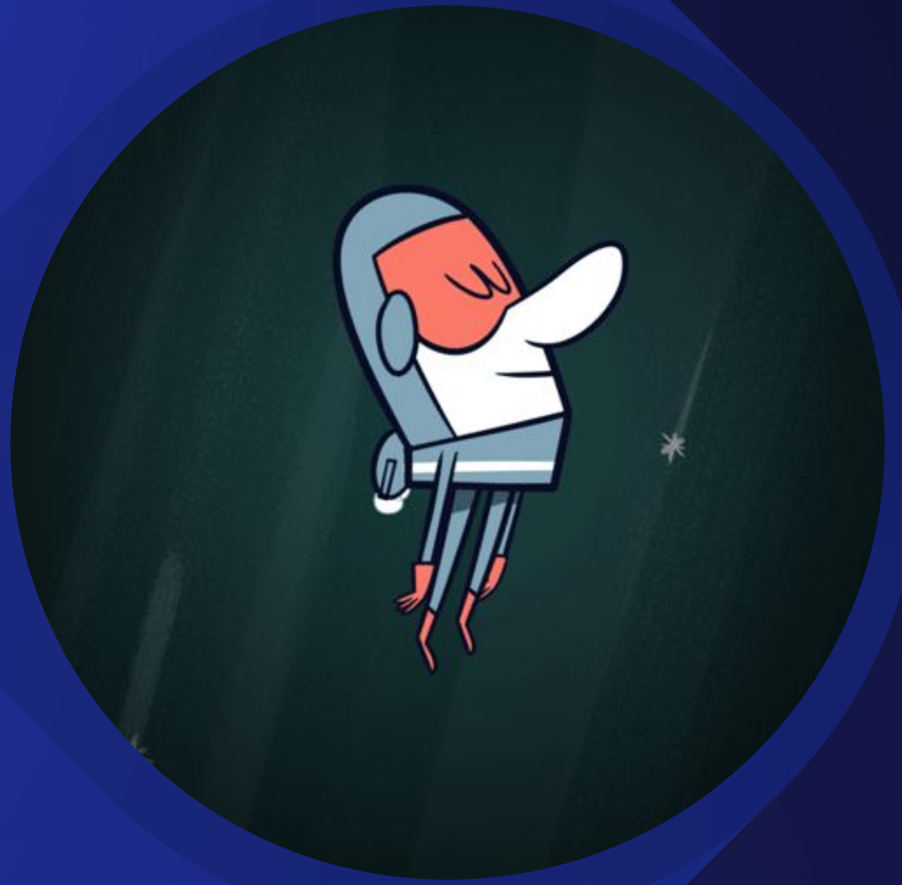
guayerd

daily

¿Cómo venimos?

¿Algo nos bloquea?

¿Cómo seguimos?



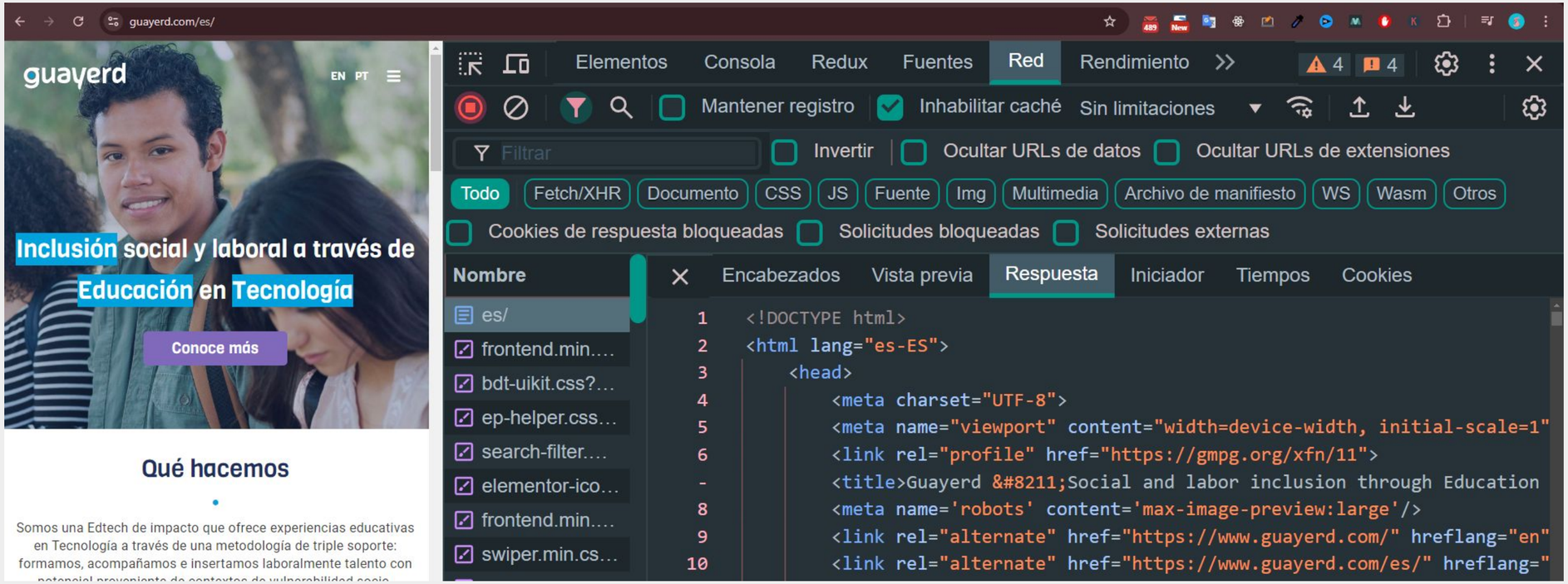
guayerd

DOM

Document Model Object

Respuesta del servidor a una petición

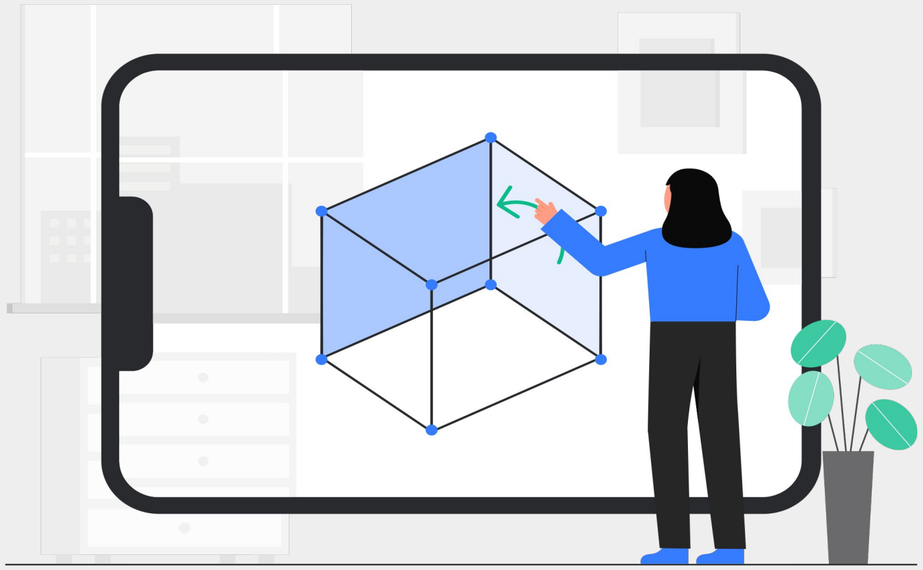
Cuando queremos entrar a una app web, hacemos una petición al servidor que nos devuelve un HTML. El navegador convierte esta respuesta en un objeto en forma de árbol, llamado DOM.



guayerd

SPRINT 2

DOM

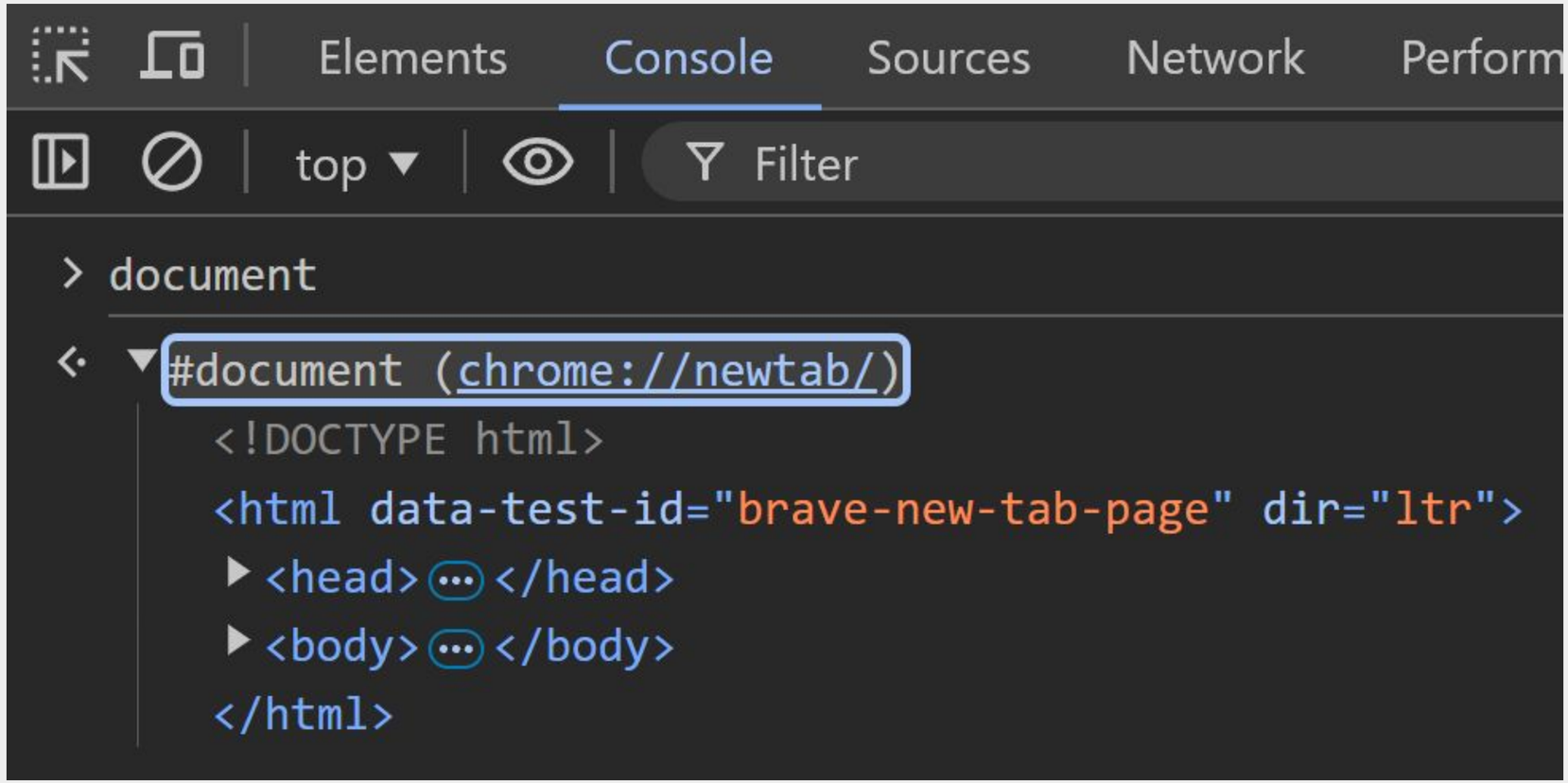


DOM es el acrónimo de Document Object Model. Es la representación en memoria de lo que vemos en la interfaz de usuario (UI).

Es un objeto anidado que puede ser accedido y modificado por Javascript. Por lo cual, al editarlo, cambia lo que vemos en la UI.

guayerd

En Javascript DOM se escribe así



guayerd

Acceder a las propiedades del DOM

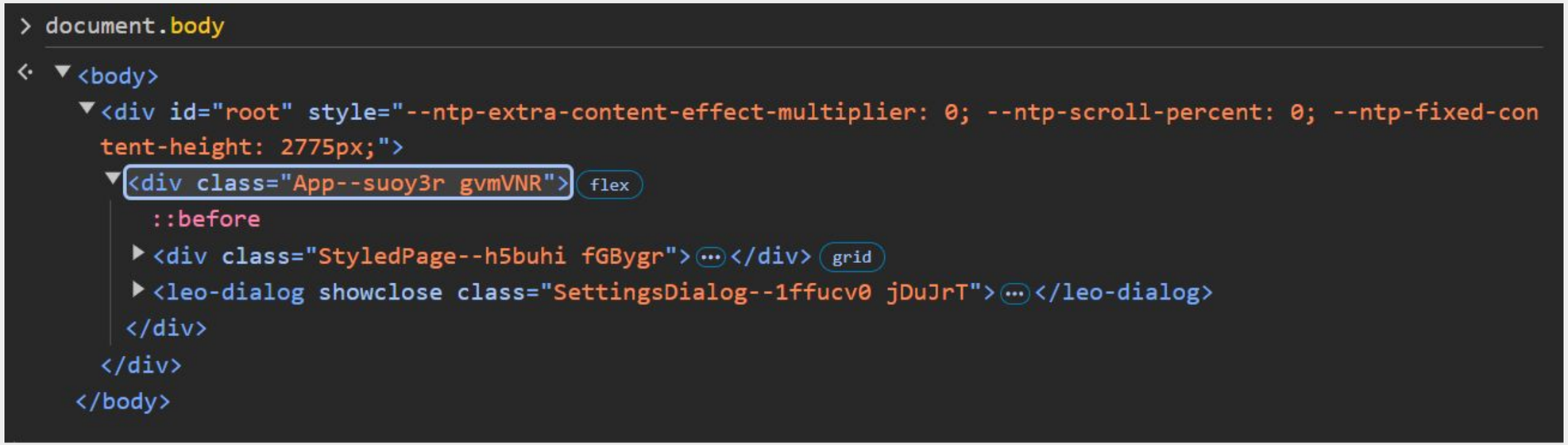
Debemos usar console.dir.

```
> console.dir(document)
                                                                    VM186:1
▼ #document
  ▶ location: Location {ancestorOrigins: DOMStringList, href: 'chrome://newtab/', origin: 'chrome://newtab'
    _reactListening6709p8qjjmt: true
    URL: "chrome://newtab/"
  ▶ activeElement: body
  ▶ adoptedStyleSheets: Proxy(Array) {}
    alinkColor: ""
  ▶ all: HTMLAllCollection(290) [html, head, meta, meta, title, link, link, link, link, link, link, script,
  ▶ anchors: HTMLCollection []
  ▶ applets: HTMLCollection []
    baseURI: "chrome://newtab/"
    bgColor: ""
  ▶ body: body
    characterSet: "UTF-8"
    charset: "UTF-8"
    childElementCount: 1
  ▶ childNodes: NodeList(2) [<!DOCTYPE html>, html]
  ▶ children: HTMLCollection [html]
    compatMode: "CSS1Compat"
    contentType: "text/html"
    cookie: ""
    currentScript: null
  ▶ defaultView: Window {window: Window, self: Window, document: document, name: '', location: Location, …}
    designMode: "off"
    dir: "ltr"
  ▶ doctype: <!DOCTYPE html>
```

guayerd

Elementos

El DOM está conformado por una gran cantidad de **tipos de nodos**. Entre ellos hay unos determinantes para nosotros, los elementos. Los elementos son el equivalente de las etiquetas HTML.
También hay jerarquías de elementos, HTML es padre de body y los divs son elementos descendientes.



guayerd

Métodos de acceso

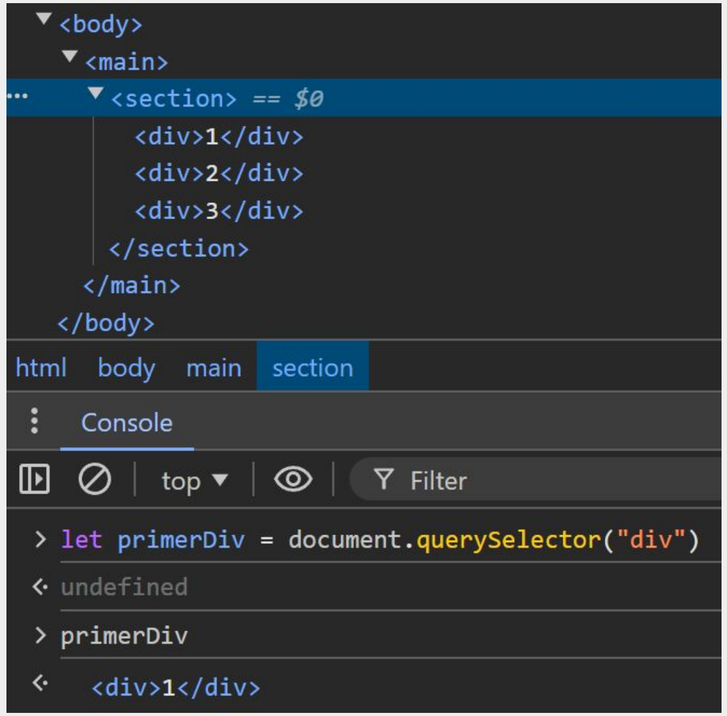
DOM

guayerd

querySelector

La sintaxis del argumento es como la de los selectores CSS.

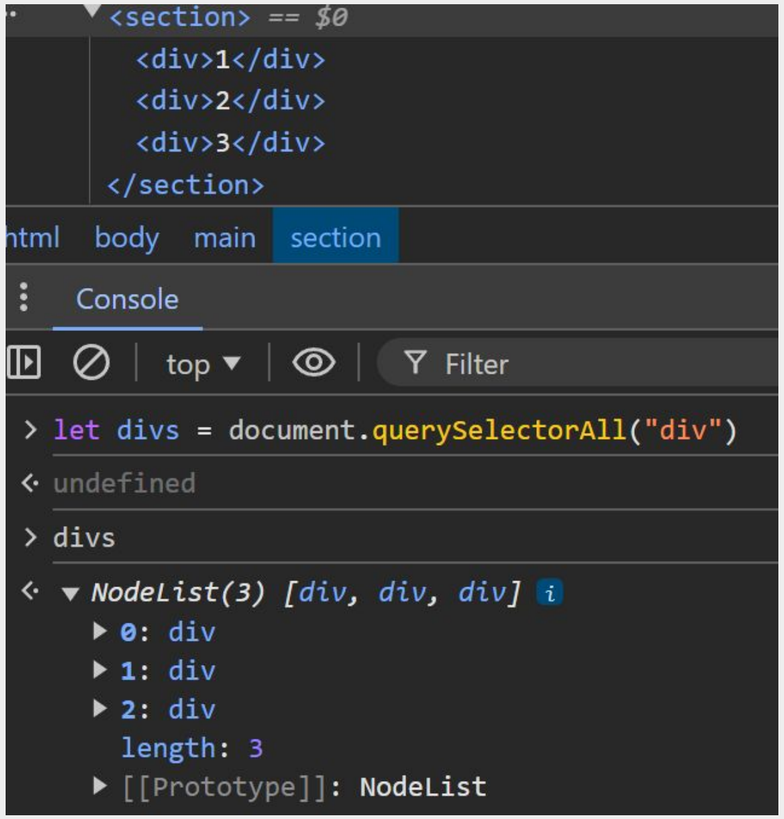
Devuelve el primer elemento que coincide con el argumento pasado al método.



guayerd

querySelectorAll

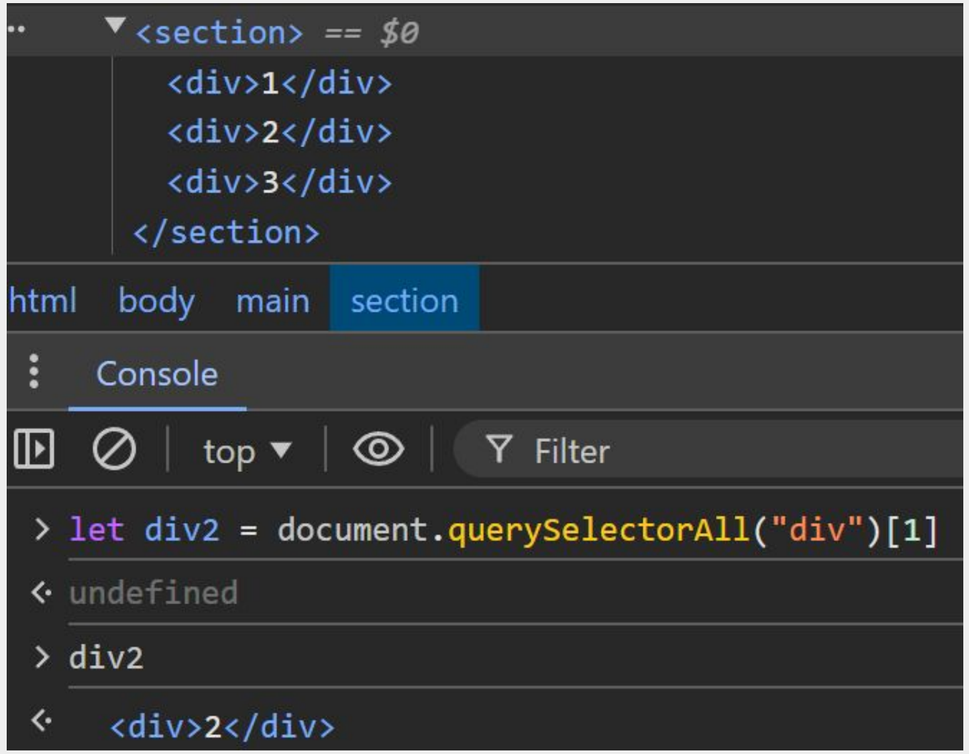
Devuelve una estructura parecida a un array que contiene todas las coincidencias.



guayerd

querySelectorAll por índice

Recuperar el segundo elemento de la lista.



guayerd

Métodos antiguos

Pero que aún se usan.

getElemetById(“id”)	Recupera un elemento con el id pasado.
getElementsByClassName(“clase”)	Recupera la lista de elementos que tengan la clase.
getElementsByTagName(“etiqueta”)	Recupera la lista de elementos que tengan la etiqueta.

guayerd

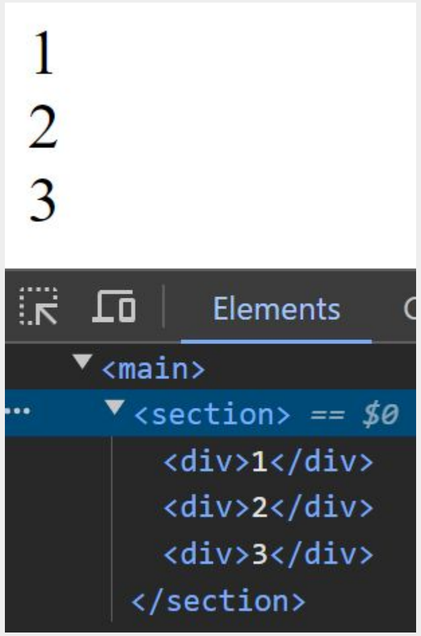
Modificación de elementos

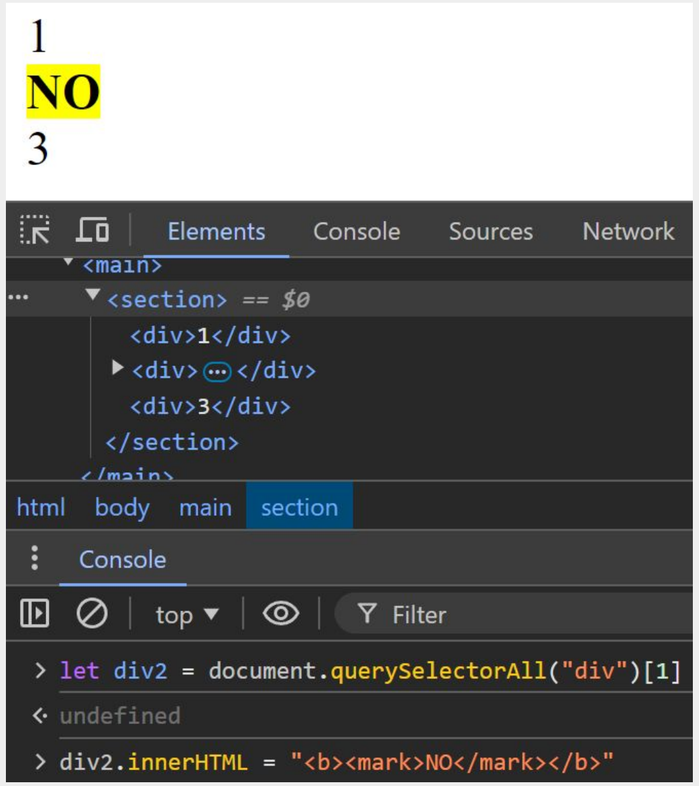
DOM

guayerd

InnerHTML

Permite ver o recrear el contenido de un elemento o etiqueta. Recibe un string.

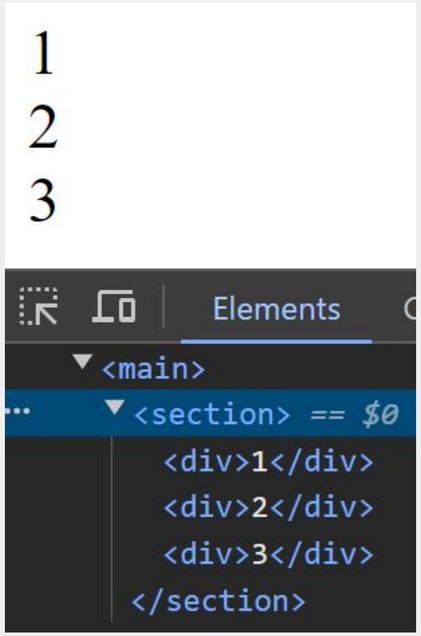


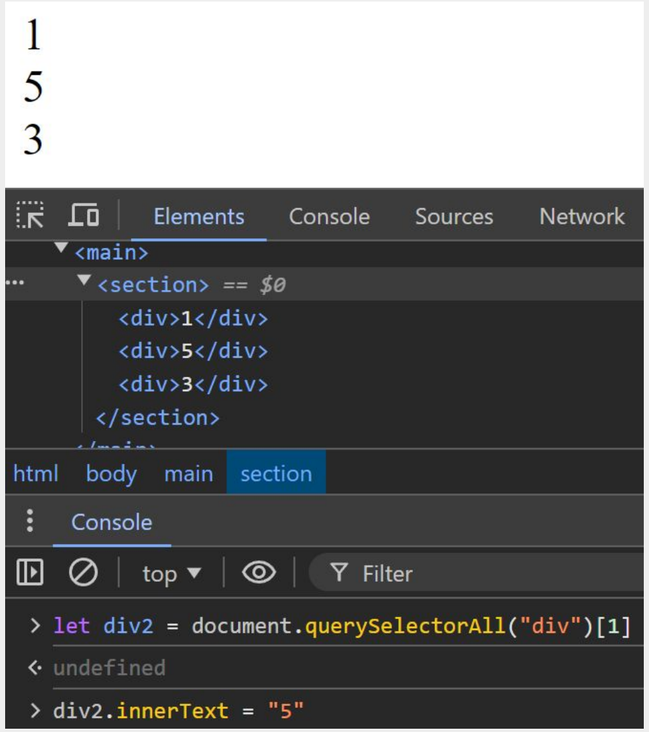


guayerd

InnerText

Permite ver o recrear el texto de un elemento o etiqueta. Recibe un string.





guayerd

Crear, agregar y eliminar elementos

DOM

guayerd

CreateElement

Crear un elemento, pero no lo agrega al DOM.

guayerd

Append

Inserta el elemento creado en el DOM en la parte inferior del elemento de destino.

Usamos prepend para insertar al inicio de la etiqueta.

guayerd

Remove

Elimina elementos del DOM.

```
let h2 = document.querySelector("h2");

h2.remove();
```

guayerd

¡DEMO SPRINT 2!

📅 15/10/2025

Tienen que presentar:

- Navbar desde el array recorrido con for of (Ejercicio de la clase 13).
- Array de productos y mostrar datos en la card (Ejercicio de la clase 16).
- Ejercicio de la clase 18.

⚠️**Que el css sea decente.**

guayerd

¡Manos a la obra!

SPRINT 2

Ejercicio 1

No entregable.

Subir en una nueva rama llamada ejercicios (o el nombre que eligieron)

- Crear un HTML de práctica vinculado a un Js.
- Mediante un prompt que el usuario vaya construyendo una página con elementos.
- Los elementos serán:
  - button
  - input
  - textarea
  - h1
  - p
- Si los elementos contienen texto agregarlos por prompt.

guayerd

SPRINT 2

Ejercicio 1

De clase 15. Entregable.

- Crear un array js de productos con un prompt AI.
- Introducirlo en main.js asignándolo a la variable data.
- Asignar data.map a la variable que habíamos asignado antes un array vacío, implementando las propiedades de los objetos del array en el marcado de una card.
- Crear la lista de productos e insertarlos en el elemento <main />.

guayerd

SPRINT 2

Ejercicio 2

De clase 15. Entregable..

- En los ver más de la Home, en el <a href, luego de .html poner ?prod=${producto.id}
- Usar el array de productos.
- Filtrarlo por el número de producto obtenido con window.location.
- Lograr que el producto mostrado cambie dependiendo del número de producto en la URL.

guayerd

¡DEMO SPRINT 2!

📅 15/10/2025

Tienen que presentar:

- Navbar desde el array recorrido con for of (Ejercicio de la clase 13).
- Array de productos y mostrar datos en la card (Ejercicio de la clase 16).
- Ejercicio de la clase 18.

⚠️**Que el css sea decente.**

guayerd

retro

¿Cómo nos fué?

¿Qué cosas no quedaron claras y necesitamos repasar la próxima?