Document Object Model
DOM

DEV.F::
DESARROLLAMOS(PERSONAS);

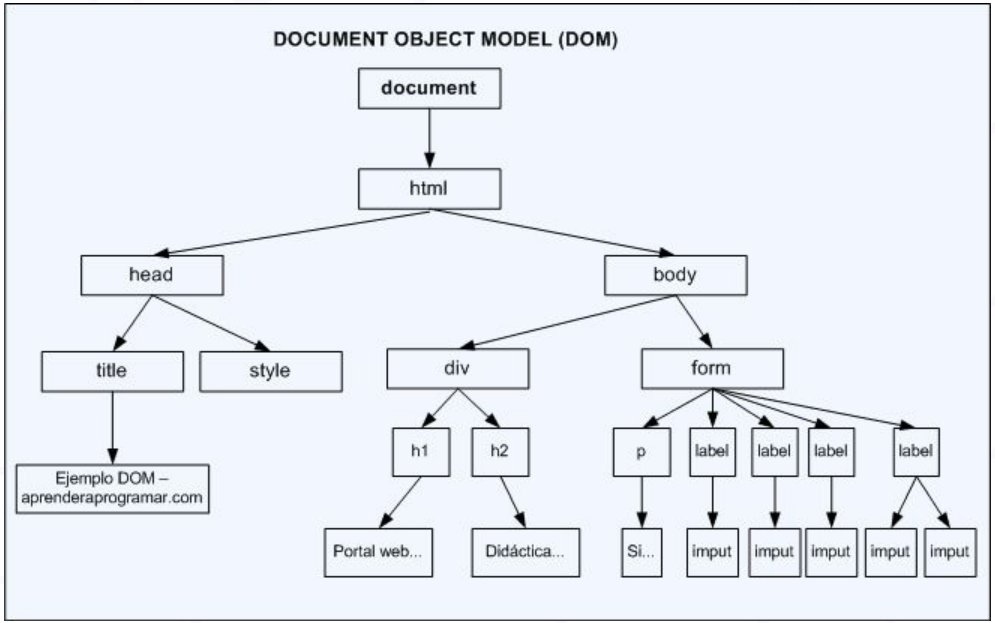
Document Object Model

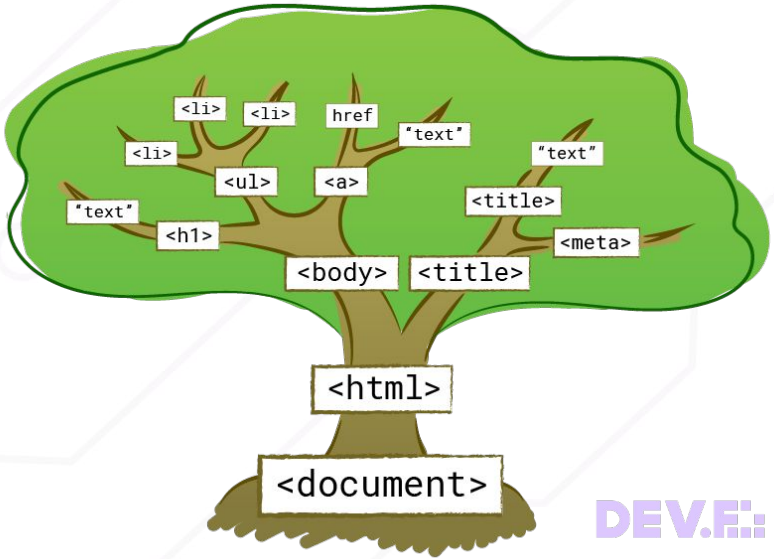
El **DOM** de un HTML es un **modelo de objetos** estándar y una **interfaz** de programación para HTML. Este define:

- Los elementos **HTML como objetos.**
- Las propiedades de todos los elementos HTML.
- Los métodos para acceder a todos los elementos HTML.
- Los eventos para todos los elementos HTML.

En otras palabras: El DOM de HTML es el estándar para cómo **obtener**, **modificar**, **cambiar** o **borrar** elementos HTML

Document Object Model





¿Para qué sirve?

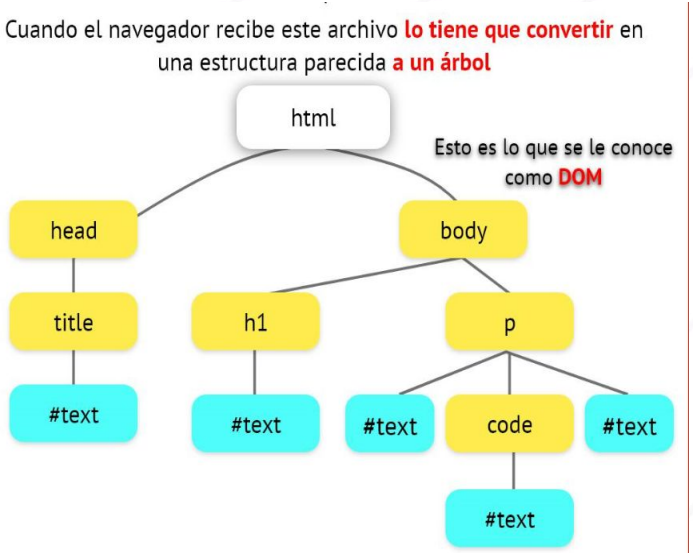
La interacción de una página web y los cambios en la interfaz gráfica son el centro de una web ya que son su funcionalidad, es decir, lo que le da valor al usuario. Para lograr esto, requerimos manipular el DOM.

- Agregar dinámicas a nuestra web (crear/modificar elementos del DOM).
- Navegar entre páginas.
- Validar formularios.
- Consultar APIs.

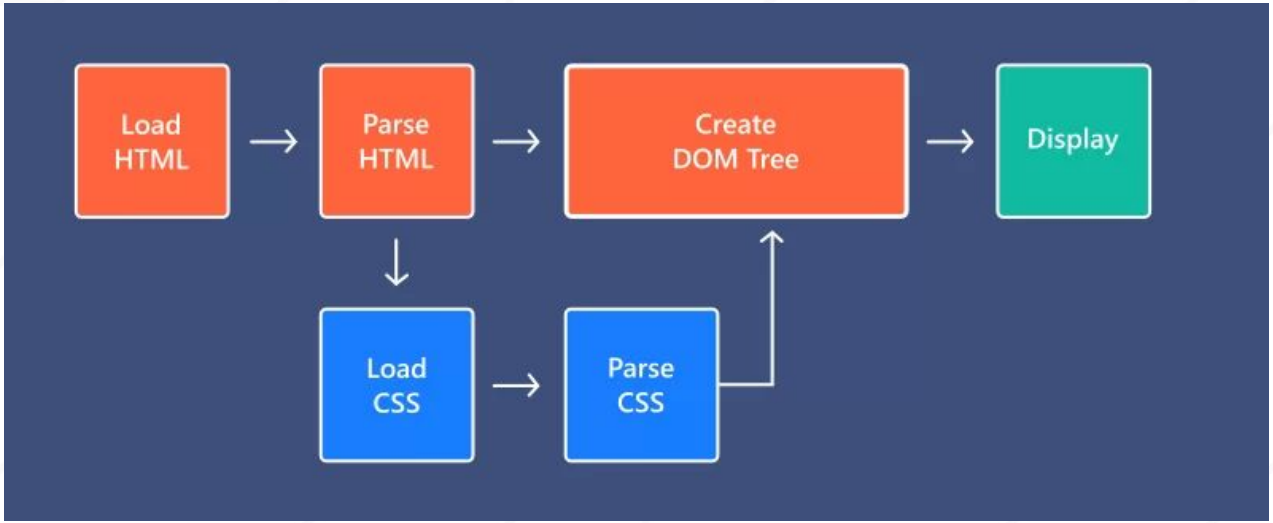
¿Para qué sirve?

Mediante el DOM, JavaScript puede acceder y modificar todos los elementos de un documento HTML.

Cuando se carga una página web, el navegador crea un Document Object Model de la página. El DOM se construye como un árbol de objetos:



¿Cómo funciona?



¿Cómo funciona?

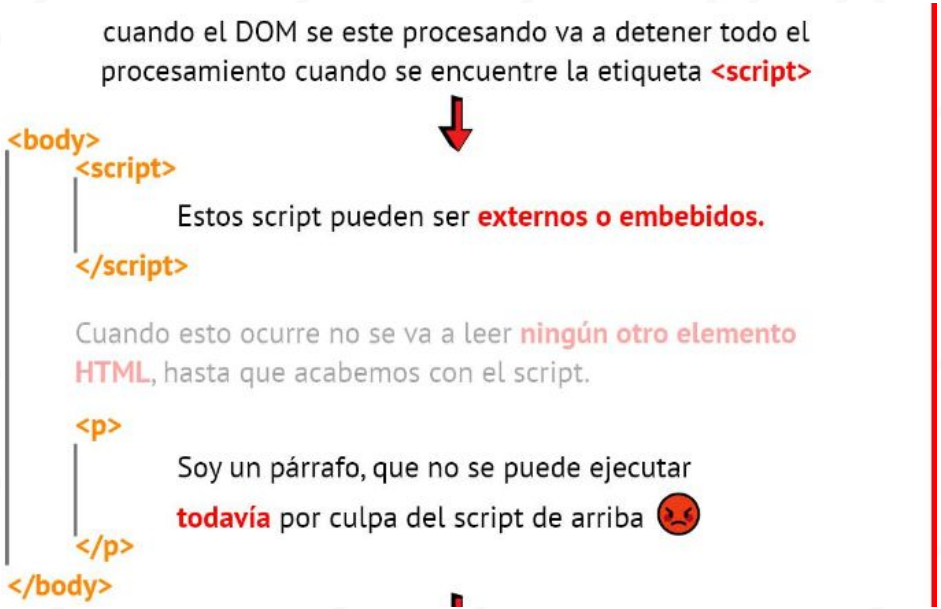
Cuando termina el **navegador** de convertirlo al **DOM** ocurre el **evento:**

DOMContentLoaded

A partir de este punto tenemos la garantía de que **todo nuestro documento se ha cargado**.

¿Cómo funciona?

Todo script que carguemos en nuestra página tiene un llamado y una ejecución.



¿Cómo funciona?

Cuando termina el **navegador** de convertirlo al **DOM** ocurre el **evento:**

DOMContentLoaded

A partir de este punto tenemos la garantía de que **todo nuestro documento se ha cargado**.

¿Cómo funciona?

por lo tanto, es importante en que lugar colocamos nuestros sciprt's.

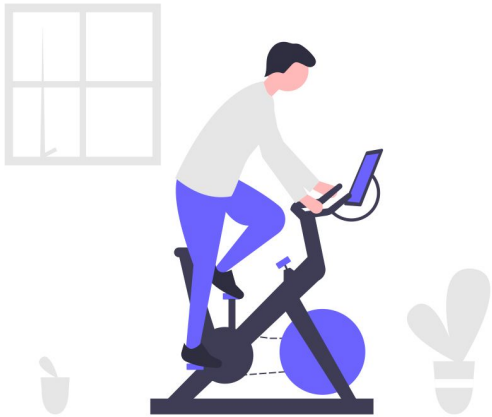
```
<body>
    <h1> titulo </h1>
    <p>  texto  </p>
    <script>El mejor lugar para colocarlo es al final del body<script>
</body>
```

Modificando el DOM

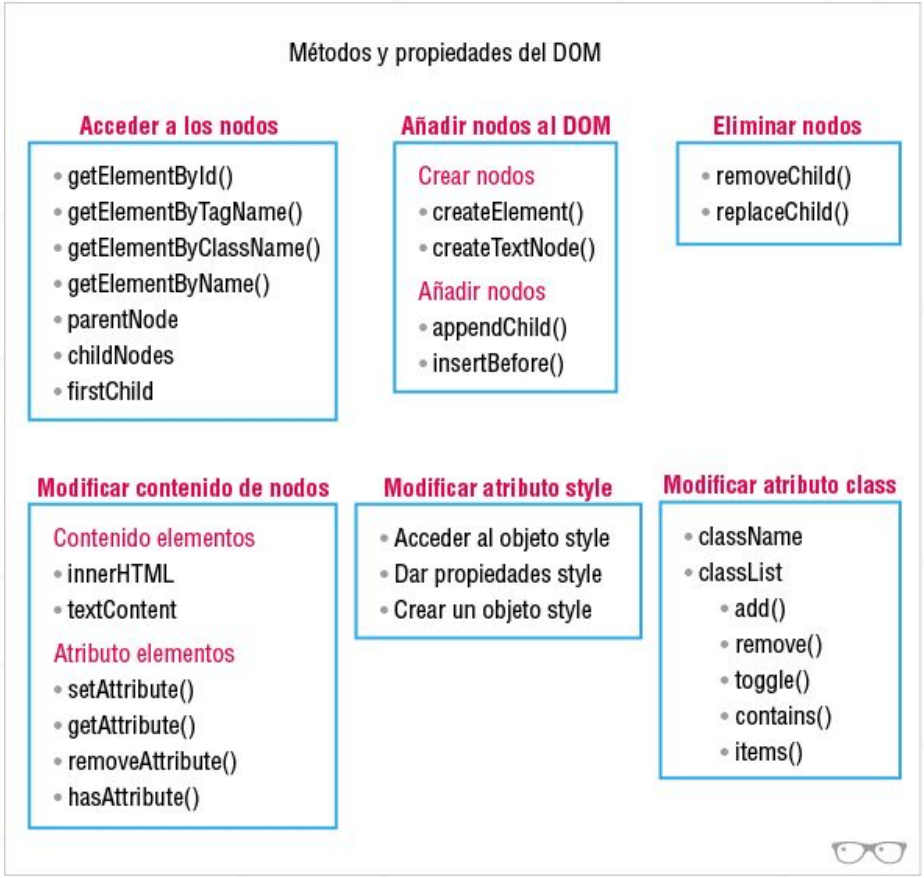
DEV.F:: DESARROLLAMOS(PERSONAS);

Con JS se puede modificar el DOM

1. Leer los elementos.
2. Obtener valores, atributos y contenido de los elementos.
3. Crear nuevos elementos.
4. Eliminar elementos.
5. Agregar contenido, atributos, clases a los elementos.
6. **Volver dinámica nuestra web.**



Métodos y propiedades del DOM

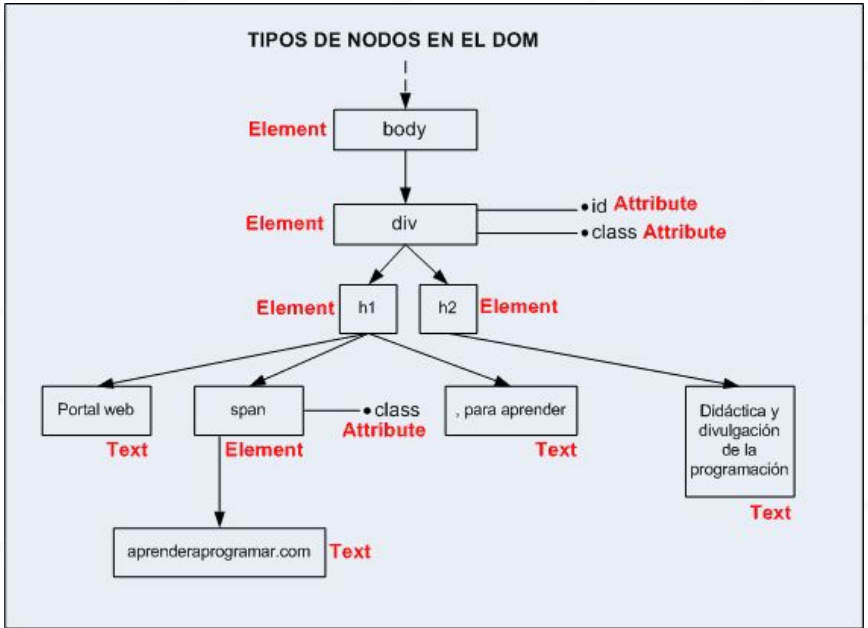


Nodos y HTML List

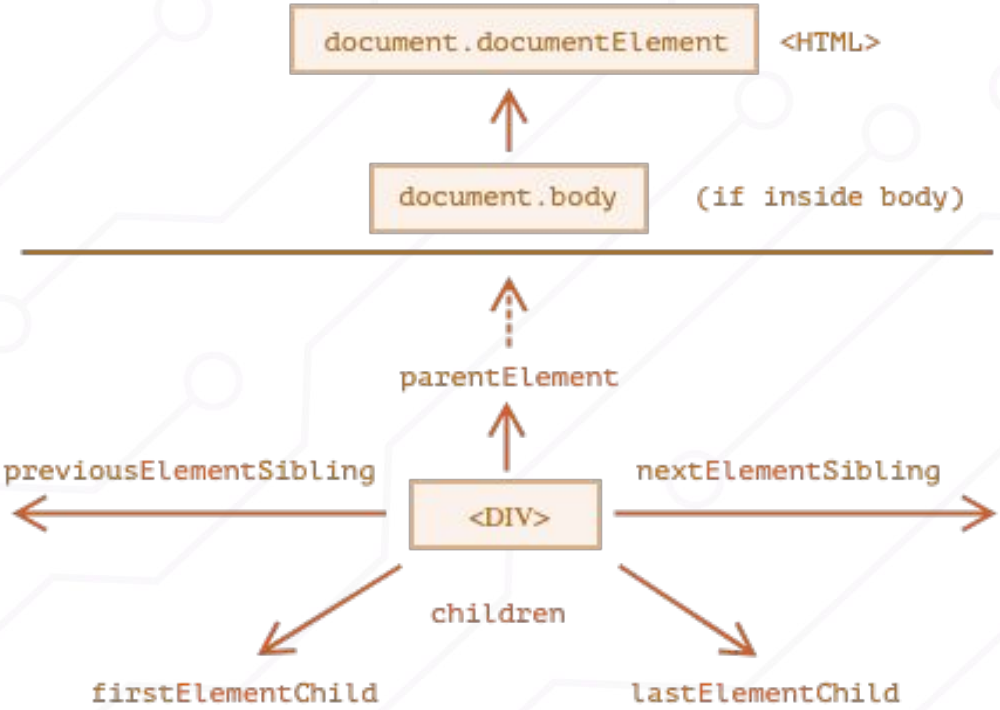
DEV.F::
DESARROLLAMOS(PERSONAS);

Nodos

Es un objeto que devuelve un conjunto de nodos. Podríamos pensar que se parece a un array, por ser un conjunto de datos (Array: lista ordenada de datos) pero no lo es. De entrada, porque no podemos trabajar con él como si lo fuese.



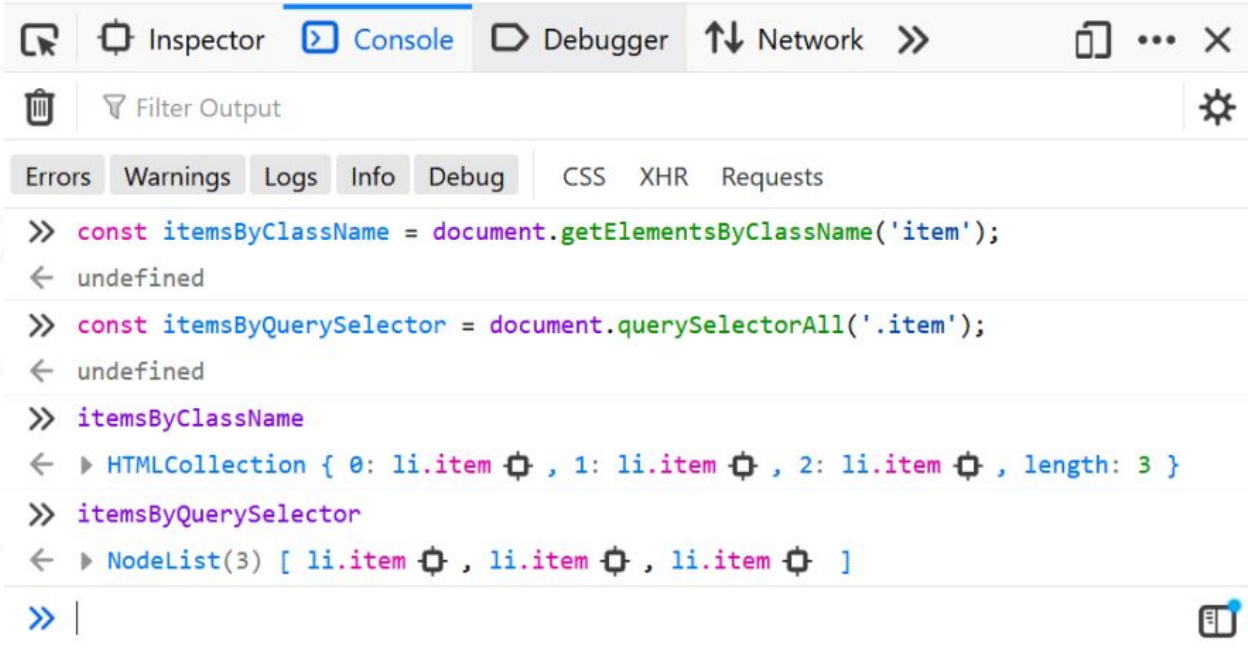
Parent, siblings and children



Node list

- Documentación.

- First Item
- Second Item
- Third Item



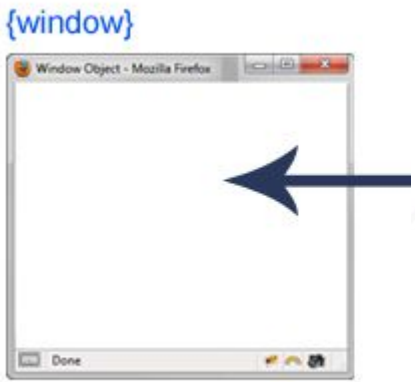
Window y Document

DEV.F
DESARROLLAMOS(PERSONAS);

window y document

window representa la **ventana que contiene** al **DOM**.

document es la propiedad que **apunta al DOM**, es decir es el objeto de **acceso al árbol html.**



Eventos

DEV.F::
DESARROLLAMOS(PERSONAS);

Eventos

Es una acción que ocurre sobre un elemento HTML, dicha acción es disparada por el usuario quien espera una respuesta. Esto se basa en el paradigma de programación orientada a eventos.

- Lista de eventos.
- Events w3Schools.
- Resumen de eventos más utilizados.

Ejemplo eventos

Con JavaScript puedes escribir pequeñas funciones, llamadas **event handlers** *(manejadores de eventos)* y hacer que éstas se activen empleando **atributos HTML**.

```html
<html>

   <head>
      <title>Vinculando mi JavaScript Externo</title>
      <script src = "/miproyecto/script.js" type = "text/javascript"></script>
   </head>

   <body>
      <input type = "button" onclick = "Saludar();" name = "ok" value = "Haz click" />
   </body>

</html>
```

```javascript
function Saludar() {
   alert("¡Hola Mundo!");
}
```

jQuery

DEV.F::
DESARROLLAMOS(PERSONAS);

jQuery

jQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, animaciones y agregar funciones AJAX.

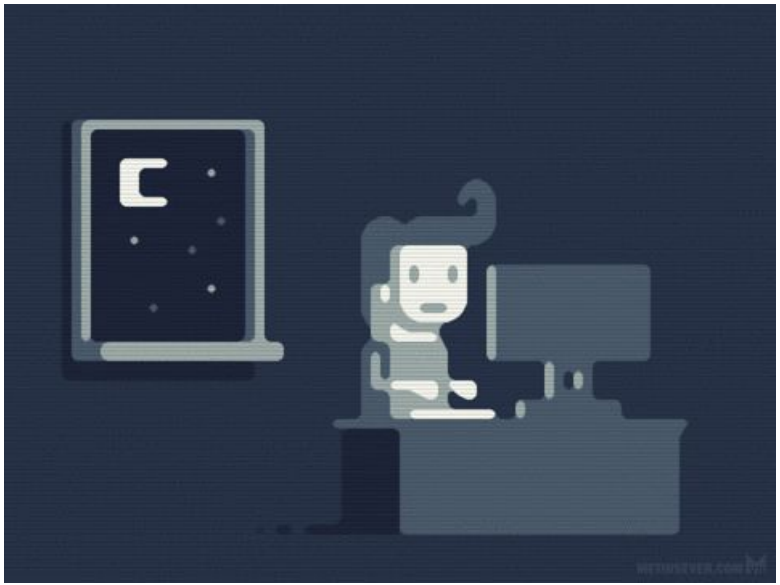
```
// js
document.getElementById("boton").style.display = 'none';


// jQuery

$("#boton").hide();
```

DEV.F

Docs

DEV.F::
DESARROLLAMOS(PERSONAS);

Documentación



- ¿Cómo se usa?
  - MDN.
  - W3Schools.
- ¿Puedo usarlo?
  - Can I Use
- Best practices
  - Code Guide html/css.
  - Clean code JS.

DEV.F::

DEV.F::

Too much?

Consejos

1. Seleccionar la información que quieren adquirir.
2. Vivir una línea de código a la vez.
3. Tener más cosas que hacer y momentos de desconexión.
4. Trabajar en sí mismos y su inteligencia emocional.
5. Aceptar que no siempre es divertido.