# Frontend developer

**Dom:** Document Object Model, me transforma mi código html en código legible para mi navegador.

**Csson:** Así como Dom, CSSON transforma mi codigo css en algo legible para el navegador.

**Render tree:** Une los dos procesos anteriores para ejecutarlos en mi browser.

**Elementos vacíos:** No tienen una etiqueta de cierre, por ejemplo la etiqueta img.

**<img src=”image.jpg” alt=”Mi casa”>**

**Etiquetas nuevas**

* **<aside></aside> : Gerneralmente aquí va la publicidad de nuestro sitio web.**
* <hr> : **Parecida a br, pero esta hace un corte entre dos párrafos**
* **<select> menú desplegable.**
* **<i> Pone mi font family en italic**
* **<em> Pone mi font family en italic con énfasis**

CÓDIGO SEMÁNTICO

Es estructurar muy bien nuestro HTML, es decir, poner la etiqueta correspondiente a cada uno de nuestros elementos, ya sea párrafo, header, main, footer, article, section, entre otros.

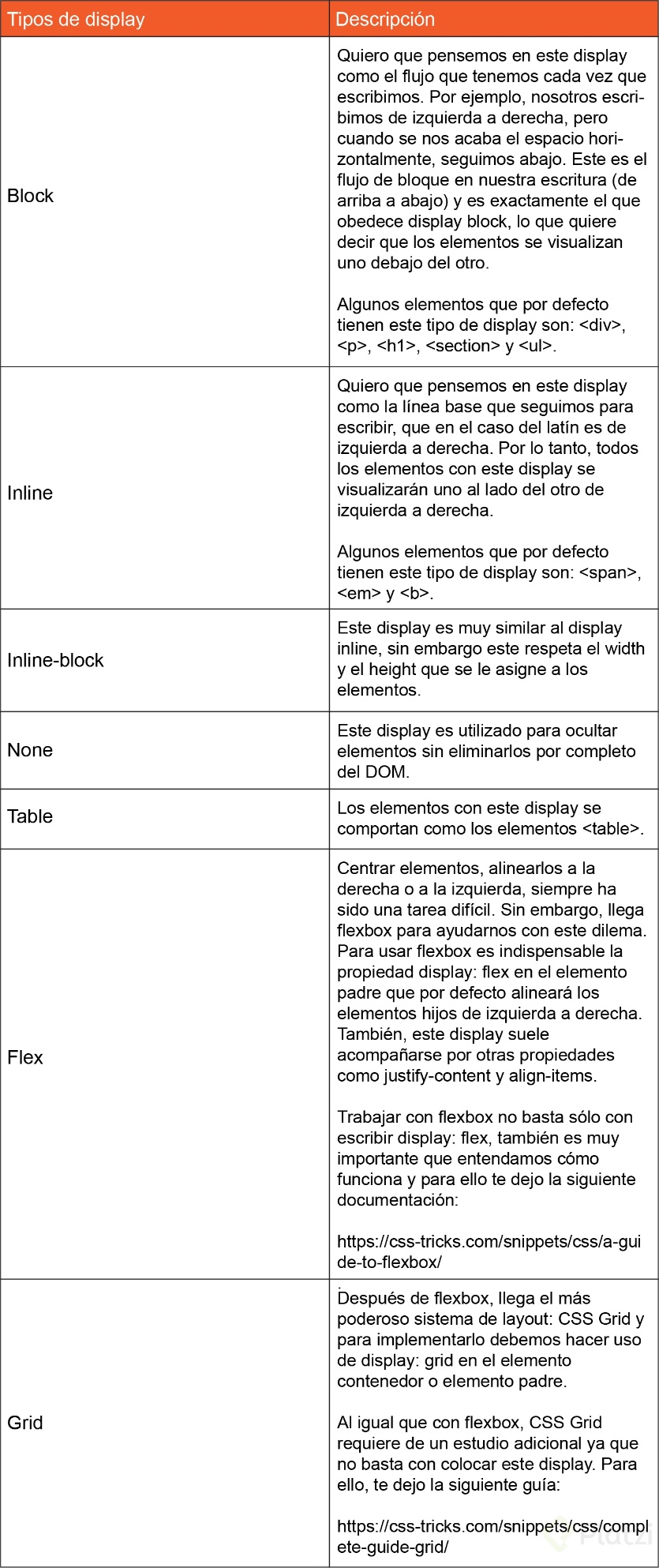
**PSEUDO CLASES Y PSEUDO ELEMENTOS**

\*pseudo clases: coge toda mi clase y le aplica las propiedades realizadas.

P:nth-first {}

\*pseudo elementos: Toma una parte de mi clase y le aplica las propiedades realizadas

**P::after {}**



# OOCSS

Cuando se repite mucho una línea de código y quiero englobarla para los elementos que la requieren, puedo poner al lado de la clase principal, otra clase para todos mis elementos que posean la misma propiedad de estios.

    <header class="header global">

      <img src="images/logo-platzi-video-BW2.png" alt="logo de platzi" />

    </header>

    <footer class="footer global">

    </footer>

# BEM

Es una metodología de nombramiento de clases para recordarlos mejor y que tengan una coherencia con mi estructura además que sea fácil deducir que estamos modificando en nuestro css.

BEM funciona como: bloque, elemento y modificador. Por ejemplo,

   <header class="header">

     <button class="header\_\_button-yellow">YELLOW</button>

     <button class="header\_\_button-black">BLACK</button>

   </header>

**QUÉ ES UN COMPONENTE**

Sección de código que cumple una función especifica en mi aplicación. Es importante que se pueda reutilizar.

“*identificar los componentes antes de programar, ósea, en el diseño*”

Esto nos permite que, cuando hagamos hover en un elemento, aparezca otro.

.header-menu:hover ul,

ul:hover {

*display*: block;

}

Bonus

.header-menu li a:hover {

*text-decoration*: underline;

}

El mayor que, me da a entender que las propiedades hechas van a ser aplicadas para todos los div de la clase ó el container.

.log-container-social-media > div {

*display*: flex;

*align-items*: center;

*margin-bottom*: 20px;

*font-size*: 14px;

}

Linear-gradient: Me hace un de3gradado que va desde un color hasta otro

.carousel-item\_\_details {

*background*: linear-gradient(

    to top,

    rgba(0, 0, 0, 0.7) 0%,

    rgba(0, 0, 0, 0) 100%

  );

Calc: Me permite realizar operaciones aritméticas en css.

.login {

*background*: linear-gradient(var(--purple), var(--light-green));

*display*: flex;

*align-items*: center;

*justify-content*: center;

*flex-direction*: column;

*padding*: 0 38px;

*min-height*: calc(100vh - 200px);

}