Appuntes curso fronted developer

Html lenguaje de marcado de hipertexto

Estructurar la página web. Esqueleto para colocar texto, imágenes, botones, etc.

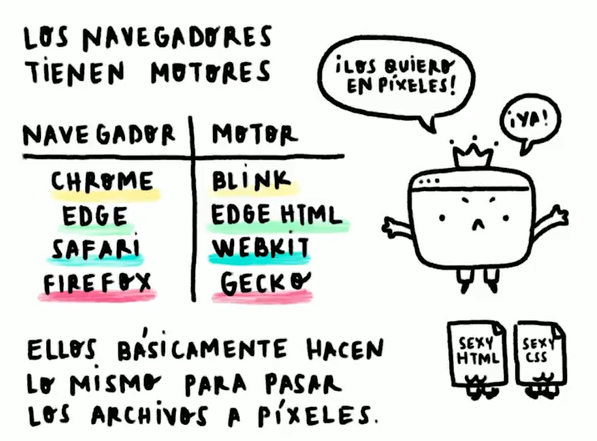
Css hojas de estilo en cascada

Lenguaje que da vida a las páginas. Estilizar

# Motores.

El navegador no entiende el lenguaje utilizado por los usuarios. Para eso tiene un motor que le sirve como traductor.

Navegadores y motores





1. Transforma los **archivos a un árbol de objetos** HTML o CSS, estos se denominan DOM (Document Object Model) y CSSDOM (Cascade Style Sheet Object Model), respectivamente. Cada nodo en el árbol es una representación de los elementos que contiene el archivo HTML o CSS.
2. **Calcula el estilo** correspondiente a cada nodo del DOM relacionado al CSSDOM.
3. **Calcula las dimensiones** de cada nodo y dónde va en la pantalla.
4. Pinta o renderiza los diferentes elementos como **cajas o contenedores**.
5. Agrupa todas las cajas en diferentes capas para **convertirlas en una imagen que se renderiza en pantalla**.

El primero de ellos es transformar el HTML

en objetos que él ya lo pueda entender.

Esto lo conocemos como DOM.

Este concepto de DOM, D-O-M,

lo vas a encontrar a lo largo de tu carrera

como Frontend Developer.

¿Esto qué significa?

Document Object Model,

que es básicamente ese mismo HTML,

pero transformado en objetos que el navegador puede comprender.

El segundo paso es que básicamente

calcula cada uno de los estilos

que tiene que ir en cada uno de esos nodos.

Ese DOM básicamente te va estructurando todas las etiquetas

desde la principal hasta las otras que vienen a continuación.

Puede ser, por ejemplo, Document,

luego el Head, luego el Body,

y de cada uno se va desprendiendo uno.

Entonces, en el segundo paso, lo que va a suceder

es que coge cada uno de los estilos

y los añade a cada uno de esos elementos

de ese arbolito que se va construyendo en ese DOM.

El tercer paso es básicamente calcular

las dimensiones de cada uno de esos nodos.

Va a decir en dónde van,

qué tamaño van a tener,

y básicamente va a empezar a estructurar toda esa página web.

El cuarto paso es pintar las diferentes cajas.

Entonces, básicamente, ya tiene toda nuestra estructura de HTML,

todo el CSS, ya tiene todas las dimensiones,

y aquí lo que hace es básicamente pintarlas.

Sin embargo, el quinto paso

es coger cada una de esas cosas que él ya pintó,

que puede ir en diferentes capas,

y lo que hace es como tomar una fotografía.

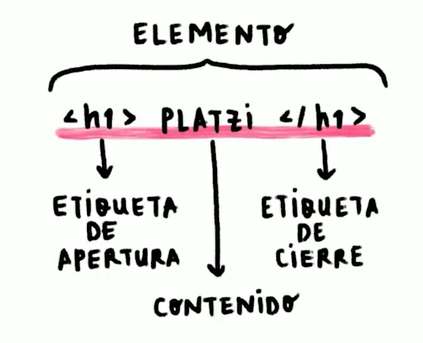
Toma una foto y esa foto es la que él va a renderizar,

es decir, que va a pintar finalmente en el navegador.

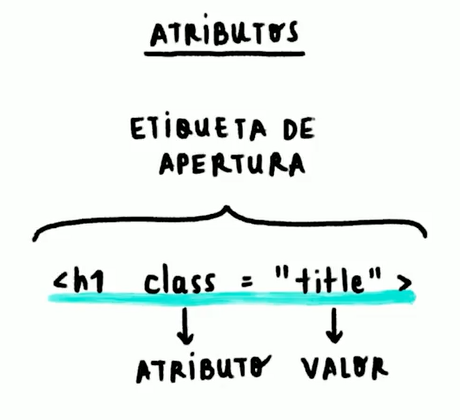
Anatomía de un elemento y documento en html

# Elementos

Etiquetas

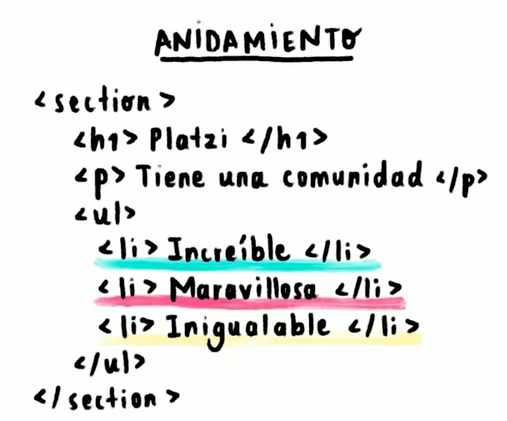


Atributos



Por ejemplo, el atributo class permite comunicarse con html

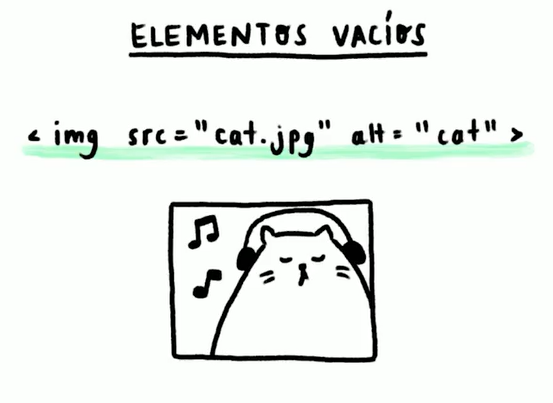
Anidamiento



Etiqueta dentro de otras etiquetas.

Section da una estructura principal.

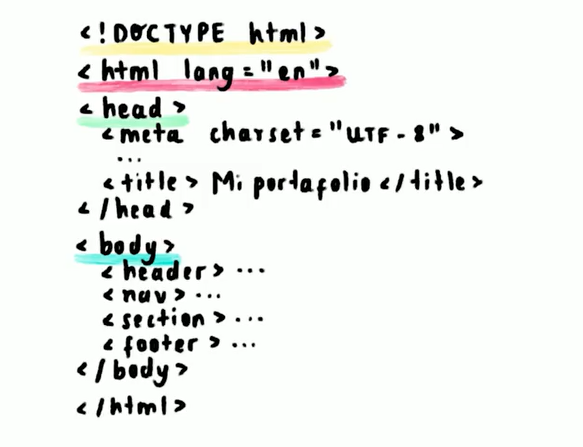
Elementos vacíos.



No todas las etiquetas tienen apertura y cierre.

# Documento

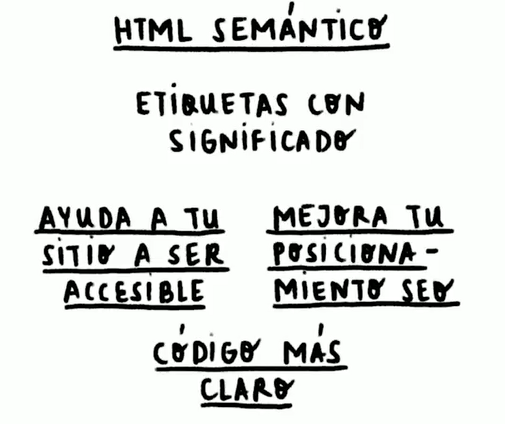
Estructura principal



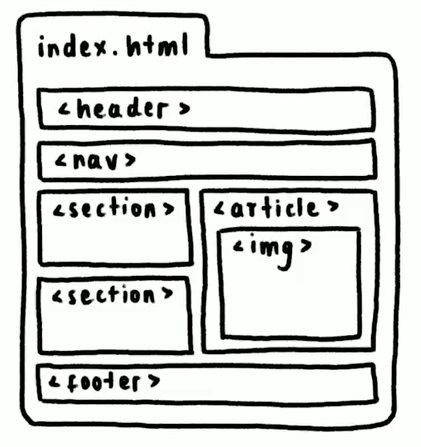
VS code la escribe por nosotros.

El body es lo que ve el usuario.

Es importante el html semántico. Utilizar las etiquetas en función de lo que significan. No sólo divs.



Dibujo estructura html



Las etiquetas semánticas para definir una interfaz de una página web son:

* <header>: define el **encabezado** de la página (no confundir con <head>).
* <nav>: define una **barra de navegación** que incluye enlaces.
* <section>: define una **sección** de la página.
* <footer>: define un **pie de página o de sección**.
* <article>: define un **artículo**, el cual puede tener su propio encabezado, navegación, sección o pie de página.

Ahora que ya conoces las etiquetas semánticas, evita el uso excesivo de <div>.

Las ventajas de utilizar un HTML semántico son:

* Ayuda a tu sitio a ser accesible
* Mejora tu posicionamiento (SEO)
* Código más claro, legible y mantenible
* Ayuda a buscadores (como Google) a encontrar tu página

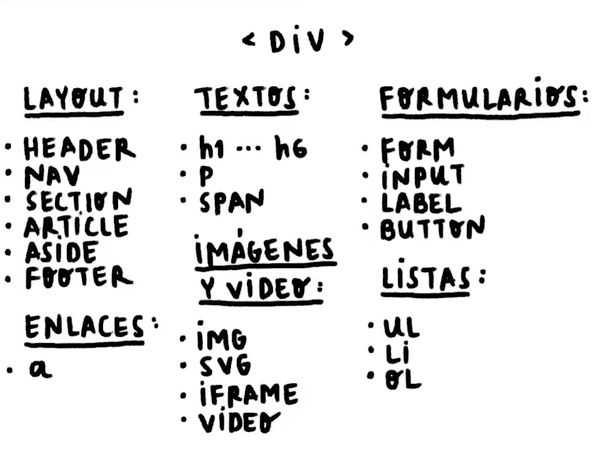
Importante tener en cuenta con la etiqueta div

Como ya vimos, tenemos varias etiquetas que pueden actuar como cajitas. A nivel de resultado visual obtendremos lo mismo, funcionan, me pintan lo que yo quiero que me pinten en la pantalla, ¿por qué debería preocuparme por ver si debo usar <div> o si debo usar <header> o <section>, etc.? Bueno, hay una razón muy específica para ello 👇.  
.  
Resulta que a nivel de significado, cada etiqueta tiene un significado diferente. Es decir, aunque hacen lo mismo, NO significan lo mismo, por eso se le llama “HTML semántico”.  
.  
¿Pero a qué nos referimos con “significado”? ¿Quién entiende ese significado? La respuesta es sencilla: Google

Ok, Google no es el único, básicamente el significado lo entiende cualquier motor de búsqueda, pero convengamos que Google es el más usado. Cuando tú buscas algo en Google ves que en los resultados te aparecen varias páginas web, y si tú vas a hacer una página web ¡pues también vas a querer que la tuya aparezca! ¿No lo crees? Esto hace parte de algo llamado SEO (Search Engine Optimization), es decir, Optimización para Motores de Búsqueda (que aparezcas en Google pues jaja).  
.  
Entonces, para poder aparecer en Google, Google necesita entender cómo está estructurada tu página web, de esa forma Google puede darle una navegación más fácil desde la vista de resultados a cualquier persona que encuentre tu página web 👇.

Y la forma en la que Google puede entenderlo es básicamente leyendo tu HTML. Y Google lo leerá por medio de HTML semántico 🤠.  
.  
¿Eso significa que nunca debería usar <div>? ¡No! Por supuesto que puedes usar <div>, esta etiqueta es muy útil cuando queremos hacer estilos con CSS, por ejemplo. Cuando decimos "no uses <div>" nos referimos a que no deberíamos usarlo para organizar el contenido de tu página, porque simplemente Google ignora cualquier <div> mientras lee tu página. Si tú pones un menú de navegación usando <div> Google lo va a ignorar, pero si tú pones un menú de navegación usando <nav> Google sabrá que tu página web tiene un menú de navegación (y te tomará en cuenta para posicionarte 7u7).  
.  
En cambio, si tú quieres mostrar un contenido dividido en 2 partes (izquierda y derecha) y usas <div> para hacerlo, entonces está súper bien, porque a Google no le importa cómo muestres (visualmente) tu contenido, así que en estos casos, <div> nos ayuda mucho cuando queremos hacer cosas visuales con nuestra página 😌.

# Etiquetas más utilizadas



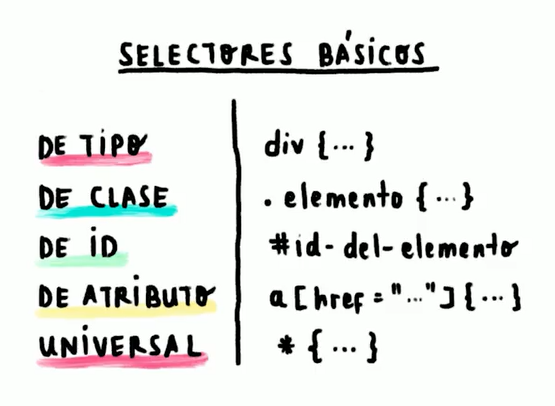
Ojo. El for del label debe tener el mismo nombre que el id del input.

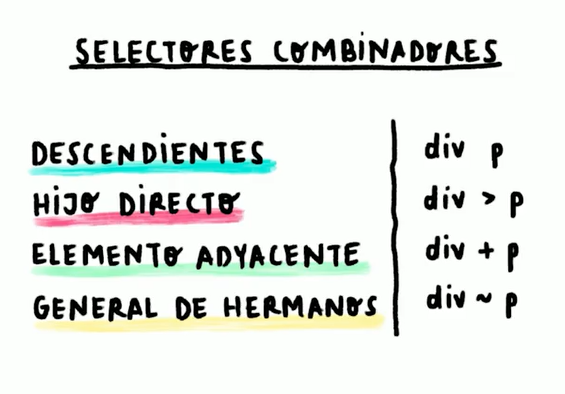
<label for="name">Nombre perro</label>

        <input type="text" id="name">

# Selectores CSS







Desafíos selectors

Tipo

<https://codi.link/PGRpdj5Tb3kgZWwgZGl2IDE8L2Rpdj4NCjxkaXY+U295IGVsIGRpdiAyPC9kaXY+DQo8ZGl2PlNveSBlbCBkaXYgMzwvZGl2Pg0KPGRpdj5Tb3kgZWwgZGl2IDQ8L2Rpdj4NCjxkaXY+U295IGVsIGRpdiA1PC9kaXY+DQo8ZGl2PlNveSBlbCBkaXYgNjwvZGl2Pg0KPGRpdj5Tb3kgZWwgZGl2IDc8L2Rpdj4NCjxkaXY+U295IGVsIGRpdiA4PC9kaXY+DQo8ZGl2PlNveSBlbCBkaXYgOTwvZGl2Pg0KPGRpdj5Tb3kgZWwgZGl2IDEwPC9kaXY+|LypBZ3JlZ2EgZWwgc2VsZWN0b3IgYXF1w60gKi8NCg0KLypBZ3JlZ2EgZWwgc2VsZWN0b3IgYXF1w60gKi8NCg0KDQovKiBJZ25vcmEgZXN0bywgcG9yIGFob3JhICovDQoqIHsNCiAgZm9udC1zaXplOiAxLjVyZW07DQogIG1hcmdpbi1ib3R0b206IDEwcHg7DQp9|>

clase

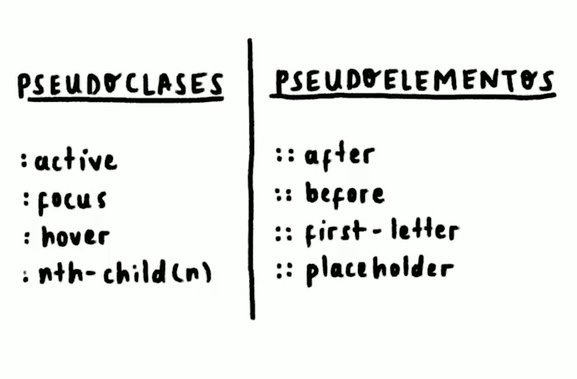
<https://codi.||>

id

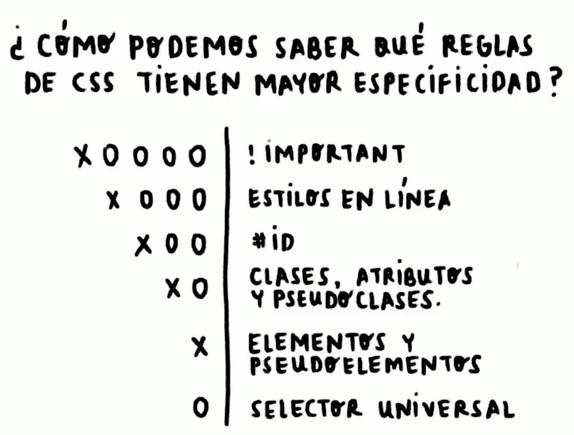
<https://codi.||>

atributo

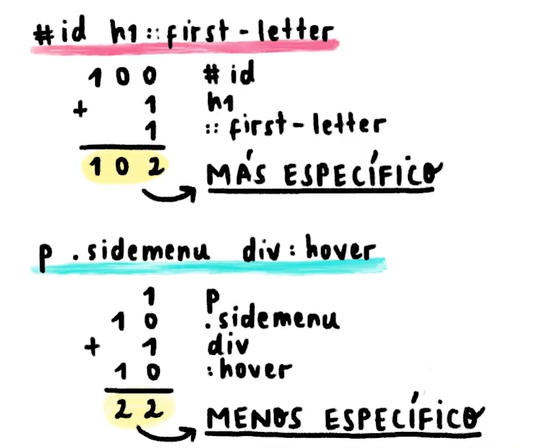
<https://codi.|LypBZ3JlZ2EgbG9zIHNlbGVjdG9yZXMgYXF1w60gKi8NCg0KLypBZ3JlZ2EgbG9zIHNlbGVjdG9yZXMgYXF1w60gKi8NCmFbaHJlZj0iaHR0cHM6Ly9wbGF0emkuY29tIl0gew0KICBiYWNrZ3JvdW5kLWNvbG9yOiBncmVlbjsNCn0NCg0KLyogSWdub3JhIGVzdG8sIHBvciBhaG9yYSAqLw0KKiB7DQogIGZvbnQtc2l6ZTogMS4ycmVtOw0KICBwYWRkaW5nOiA1cHg7DQogIG1hcmdpbi1ib3R0b206IDEwcHg7DQp9DQoNCmEgew0KICBkaXNwbGF5OiBibG9jazsNCn0=|>



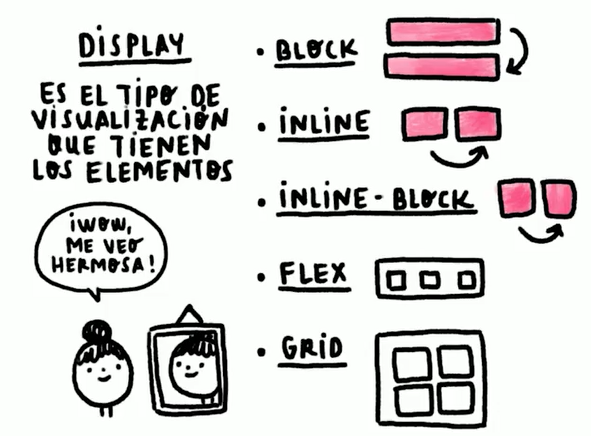
Preudoclases nos ayuda a realizar cosas cuando el usuario haga alguna acción, como dar click a algo.



Ejemplo especificidad



Tipos de display



Al block se le puede modificar width y height. Se le puede poner margin y padding.

Al inline sólo se le puede colocar padding y margin, este último sólo horizontal.

El inline block se visualiza como un inline, pero se le pueden aplicar las características de un block.

# Links de interés

Imágenes.

<https://www.pexels.com/es-es/>

Códigos de colores

<https://htmlcolorcodes.com/es/>

índice de pseudoclases

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes#indice_de_las_pseudo-clases_est%C3%A1ndar>

lista de pseudoelementos

<https://developer.mozilla.org/es/docs/web/css/pseudo-elements#lista_de_pseudoelementos>

Para ver código html y css online

<https://codepen.io/>

juego selectores css

<https://flukeout.github.io/>

specifity calculator

<https://specificity.keegan.st/>

ver etiquetas html. Muestra si su visualización es inline o block

<https://htmlreference.io/>

juegos css

* <https://flexboxfroggy.com/#es>
* <https://cssgridgarden.com/#es>
* <https://flukeout.github.io/>
* <https://cssbattle.dev/>
* <https://mastery.games/flexboxzombies/>
* <https://codingfantasy.com/games/css-grid-attack>