Curso sobre Arquitectura CSS.

Clase 1.

¿Qué es una arquitectura CSS?

Es una metodología que se enfoca en la estructura visual de un proyecto de desarrollo web, estéticamente hablando.

Objetivos de una arquitectura CSS: Es la organización del código y carpetas junto a una metodología de clases para que el trabajo en equipo esté estandarizado por partes iguales.

Una arquitectura CSS debe ser:

* Predecible: sus reglas hacen lo que se espera. Añadir o actualizar una regla no tiene repercusiones inesperadas.
* Reutilizable: las reglas son abstractas y están desacopladas. Así se podrán crear nuevos componentes de forma rápida y sencilla.
* Mantenible: es la capacidad de hacer cambios en el CSS sin necesidad de tener que refactorizar el código existente.
* Escalable: significa que a medida que las CSS aumentan de tamaño, siguen siendo fáciles de mantener durante el tiempo.

Relación entre arquitectura CSS y metodología:

* Metodología 🡪 conjunto de reglas o parte lógica que estudia los pasos a seguir en el desarrollo de un proyecto tecnológico.
* Arquitectura 🡪 La metodología en la práctica. Conjunto de reglas aplicadas en la estructura de un proyecto.

Tipos de metodologías existentes:

* OOCSS
* SMACSS
* BEM
* ITCSS
* ACSS (Atomic Design)

Siempre se deben añadir clases a las etiquetas HTML, para así en dado caso se llegue a cambiar alguna de las etiquetas, el estilo no se pierda y sea más fácil dar mantenimiento.

List-style:none; : Quita los estilos por defecto que tienen los elementos de una lista.

Al escribir el CSS, pueden crearse selectores de varias formas, ya sea:

* Tag
* Id
* Clase
* Atributos

**Cuanto más específico sea el selector, más problemas tendremos con las sobreescrituras, el mantenimiento y la legibilidad del código.**

**En general, evitamos usar id como selector, ya que podemos romper fácilmente el CSS con algún cambio.**

**Al usar selectores de etiquetas, dejamos los estilos muy atados al HTML.**

**El uso de demasiados selectores en línea puede dañar la legibilidad y el rendimiento del código.**

***Metodología BEM***

BEM 🡪 (Bloque, Elemento, Modificador)

Es una convención de nomenclatura popular para clases en HTML y CSS

La idea detrás de esto es dividir la interfaz de usuario en bloques independientes. Esto hace que el desarrollo de la interfaz sea fácil y sencilla, incluso con una interfaz de usuario compleja, permite crear componentes de interfaz extensibles y reutilizables. La importancia principal de esta metodología es permitirle dar un significado más informativo al nombre de las clases.

***Bloque:*** Es un componente de página funcionalmente independiente que se puede reutilizar. En HTML, los bloques están representados por el class=”atributo”. Este bloque debe considerarse como un padre.

**<div class= “card”></div>**

***Elemento***: Una parte compuesta de un bloque que no se puede usar por separado.

<div class= “card”>

<input class=”**contacto\_\_input**>”

<button class=”**contacto\_\_button”**>Enviar</button>”

</div>

Modificadores: Una entidad que define la apariencia, el estado o el comportamiento de un bloque o elemento.

<div class= “card”>

<a class=”card\_\_link” href=”#”>Nombre</a>

<a class=”card\_\_link**--resaltar**” href=”#”>Precio</a>

</div>

Clase 2

Atomic design: El atomic design o diseño atómico es una metodología para crear sistemas de diseño. Hay cinco niveles distintos en el sistema del diseño atómico:

* Atomos: Son los componentes básicos de la materia. En las interfaces, serían las etiquetas HTML.
* Moléculas: Son grupos de átomos unidos entre sí, son las unidades fundamentales más pequeñas de un compuesto.
* Organismos: Son grupos de moléculas unidas para formar una sección distinta y relativamente compleja de una interfaz.
* Template: Consisten principalmente en grupos de organismos unidos para formar páginas.
* Páginas: Son instancias específicas de plantillas. Es donde se prueba la efectividad del sistema de diseño.

**El atomic design entonces consiste en separar en archivos las secciones del proyecto e ir pasandolas a los archivos específicos en caso de ser necesario.**

Clase 3

Propiedad CSS ease-in-out: