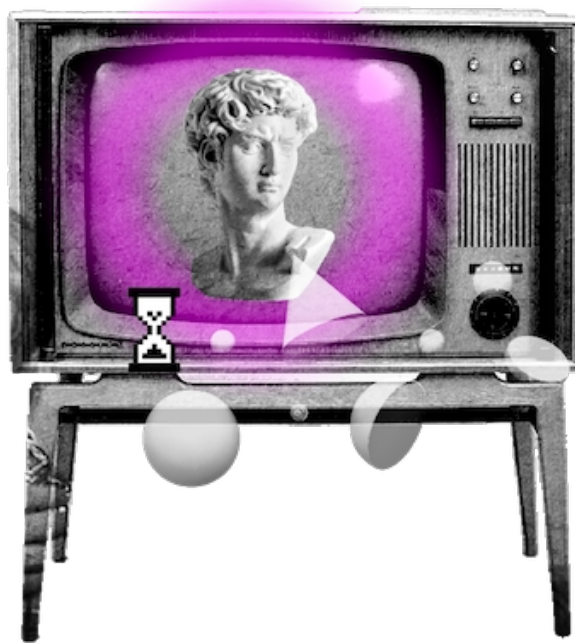




DISEÑO WEB

METODOLOGÍA BEM



APUNTES





Metodologías CSS

¿Qué son las metodologías de CSS?

Las **metodologías CSS** son una **forma estructurada y organizada de escribir CSS**, el lenguaje utilizado para dar estilo a páginas web. Las metodologías CSS son útiles para crear estilos coherentes, mantenibles y escalables en proyectos web grandes y complejos.

¿Por qué utilizar metodologías de CSS?

Existen varias razones por las cuales es recomendable utilizar metodologías CSS en nuestro proyecto, entre ellas podemos mencionar:

- **Coherencia:** Una metodología CSS proporciona reglas y convenciones claras para escribir estilos en una página web. Esto hace que los estilos sean coherentes y fáciles de entender, incluso si varios desarrolladores trabajan en el mismo proyecto.
- **Mantenibilidad:** Los proyectos web pueden volverse muy complejos con el tiempo, y el código CSS puede ser difícil de mantener si no se organiza correctamente. Las metodologías CSS proporcionan estructura y organización al código, lo que facilita su mantenimiento y actualización.
- **Escalabilidad:** Cuando un proyecto web crece, puede volverse difícil mantener el control sobre el código CSS. Las metodologías CSS ayudan a crear estilos escalables, lo que significa que el código puede adaptarse a medida que el proyecto crece sin perder su organización y coherencia.
- **Reutilización:** Las metodologías CSS a menudo se basan en la idea de reutilizar estilos existentes. Esto significa que no tienes que escribir código CSS nuevo cada vez que queramos agregar un nuevo elemento o componente a nuestra página web. En su lugar, podemos utilizar estilos ya existentes y modificarlos según sea necesario.

Desventajas de utilizar una metodología CSS

Aunque las metodologías CSS pueden ser útiles para mejorar la organización y la escalabilidad del código, también pueden tener algunas desventajas. Veamos las más importantes:

- **Curva de aprendizaje:** Aprender una nueva metodología CSS puede llevar tiempo y esfuerzo, especialmente si nunca hemos utilizado una antes. Podemos necesitar tiempo para acostumbrarnos a las reglas y convenciones de la metodología y esto puede ralentizar el proceso de desarrollo.



- **Rigidez:** Las metodologías CSS pueden ser muy estructuradas y rigurosas en cuanto a las reglas y convenciones que se deben seguir. Esto puede hacer que sea difícil hacer cambios o adaptarse a las necesidades específicas de un proyecto.
- **Complejidad:** Algunas metodologías CSS pueden ser muy complejas y difíciles de entender, especialmente para los desarrolladores novatos. La complejidad puede hacer que el código sea más difícil de mantener y actualizar, lo que puede ser una desventaja a largo plazo.
- **Sobrecarga:** Las metodologías CSS pueden requerir más tiempo y esfuerzo para escribir y mantener el código, lo que puede ser una sobrecarga para algunos proyectos web más pequeños o menos complejos. Si la metodología no se adapta a las necesidades del proyecto, puede ser más perjudicial que beneficiosa.

Metodología BEM

La **metodología BEM** es una forma popular de organizar el código CSS que utilizamos en nuestros proyectos web. BEM significa Block, Element, Modifier, que se refiere a los tres componentes principales de esta metodología.

La idea detrás de BEM es que cada clase CSS debe tener un significado claro y descriptivo. En lugar de utilizar nombres genéricos para nuestras clases CSS, como "button" o "header", debemos utilizar nombres descriptivos basados en la función y el propósito de los elementos que estamos estilizando. Por ejemplo, en lugar de usar una clase genérica como "button", podríamos usar una clase descriptiva como "button--primary" para indicar que es el botón principal del sitio web.

Al utilizar nombres descriptivos y significativos para nuestras clases CSS, podemos hacer que nuestro código sea más fácil de entender y mantener. Además, BEM también fomenta la reutilización de código y la modularidad, lo que puede ser muy útil en proyectos web más grandes y complejos.

Bloque

En BEM, un **bloque** es un **componente independiente de la interfaz de usuario** que puede ser reutilizado en todo el sitio web.

Un ejemplo de bloque podría ser una sección de la página que contiene una lista de productos. Podríamos nombrar este bloque como "product-list". La clase CSS correspondiente sería ".product-list".



Elemento

Los **elementos** son **partes de un bloque que tienen un significado específico y están estrechamente relacionados con él**.

Siguiendo con el ejemplo anterior, dentro de nuestro bloque "product-list", podemos tener elementos específicos, como una imagen del producto y el nombre del producto. Podríamos nombrar estos elementos como **"product-list_image"** y **"product-list_name"** respectivamente. Las clases CSS correspondientes serían **".product-list_image"** y **".product-list_name"**.

Modificador

Los **modificadores** son **características opcionales que se pueden agregar a un bloque o elemento para cambiar su apariencia o comportamiento**.

Por ejemplo, si queremos destacar un producto específico en nuestra lista, podríamos agregar un modificador a ese elemento. Podríamos nombrar este modificador como **"product-list_item--featured"**. La clase CSS correspondiente sería **".product-list_item--featured"**. Este modificador podría ser utilizado para aplicar estilos especiales a ese producto en particular, como un borde o fondo diferente.

Otras metodologías

Además de BEM, existen otras metodologías importantes de CSS, que si bien no veremos durante la cursada, vamos a ver de qué se tratan las más relevantes:

Smacss

La metodología **SMACSS (Scalable and Modular Architecture for CSS)** es una forma de organizar y estructurar nuestro código CSS para que sea más escalable y fácil de mantener. En SMACSS, utilizamos el concepto de **módulos para dividir nuestro sitio web en partes más pequeñas y manejables**. Cada módulo tiene un propósito específico y contiene los estilos necesarios para ese propósito.

Los módulos se dividen en cinco categorías: **base, layout, módulos, estados y temas**. La categoría **base** contiene los estilos predeterminados para todos los elementos HTML, como fuentes y estilos de texto. La categoría **layout** contiene los estilos que afectan la estructura general del sitio web, como el ancho y la altura de los contenedores. La categoría **módulos** contiene los estilos específicos para cada módulo individual, como el estilo para un botón o un formulario de contacto. La categoría **estados** contiene los estilos que cambian en función del estado del elemento, como el color de un botón al pasar el cursor sobre él. Finalmente, la categoría **temas** contiene los estilos que cambian en función del tema general del sitio web.



Al utilizar estas categorías, podemos mantener nuestro código CSS organizado y fácil de entender. Además, SMACSS también nos proporciona una forma de nombrar nuestras clases CSS de manera coherente y descriptiva, lo que facilita la lectura y el mantenimiento del código.

Otro aspecto importante de SMACSS es la separación de la estructura y la presentación de nuestro código. En lugar de incluir estilos específicos en nuestra hoja de estilos principal, utilizamos clases para aplicar estilos específicos a elementos HTML. Esto nos permite separar la estructura del sitio web de su estilo y facilita la modificación y actualización de nuestro código CSS.

OOCSS

La metodología **OOCSS** es una forma de escribir código CSS que se basa en la **creación de objetos reutilizables**. Un objeto en OOCSS es cualquier parte del diseño que tenga una función o propósito específico. En lugar de escribir CSS específico para cada elemento individual en nuestro sitio web, escribimos CSS para objetos que se pueden reutilizar en diferentes partes del sitio.

Un objeto en OOCSS se compone de dos partes: **la estructura y la piel**. La **estructura** es la parte del objeto que determina su función o propósito, mientras que la **piel** es la parte que determina su estilo visual. Al separar la estructura y la piel, podemos reutilizar objetos con diferentes estilos visuales en diferentes partes del sitio.

Otra técnica importante en OOCSS es la creación de clases CSS reutilizables. En lugar de crear clases CSS específicas para cada elemento en nuestro sitio, creamos clases que se pueden aplicar a diferentes elementos para proporcionar una función o estilo específico. Por ejemplo, en lugar de crear una clase `.button-rojo` y otra clase `.button-azul` para botones rojos y azules, creamos una clase `.button` que se puede aplicar a cualquier botón para proporcionar un estilo específico.

Al utilizar la metodología OOCSS, podemos escribir código CSS más reutilizable y mantenible. Al escribir CSS para objetos en lugar de elementos individuales, podemos reutilizar objetos en diferentes partes del sitio web y reducir la cantidad de código CSS que necesitamos escribir. Además, al separar la estructura y la piel, podemos cambiar fácilmente el estilo visual de un objeto sin tener que cambiar su estructura.

Atomic CSS

La metodología **ATOMIC CSS** se basa en la **creación de clases CSS atómicas que se pueden combinar para crear estilos visuales complejos**. Cada clase atómica representa un estilo visual específico, como un tamaño de fuente o un color de fondo. Al combinar clases atómicas, podemos crear estilos más complejos sin tener que escribir CSS específico para cada elemento en nuestro sitio.



Por ejemplo, en lugar de escribir un CSS específico para cada botón en nuestro sitio web, podemos crear clases atómicas para tamaños de fuente, colores de fondo y bordes. Luego, podemos combinar estas clases atómicas para crear estilos de botón específicos. Esto nos permite crear estilos visuales complejos de manera eficiente y reducir la cantidad de código CSS que necesitamos escribir.

Otra técnica importante en ATOMIC CSS es la creación de clases con nombres significativos. En lugar de crear clases con nombres como "boton1" y "boton2", creamos clases con nombres que describen el estilo visual que representan. Esto hace que nuestro código CSS sea más fácil de leer y entender, lo que a su vez hace que sea más fácil de mantener.

Al utilizar la metodología ATOMIC CSS, podemos crear estilos visuales complejos de manera eficiente y reducir la cantidad de código CSS que necesitamos escribir. Además, al crear clases atómicas y combinarlas para crear estilos visuales específicos, podemos reutilizar nuestro código CSS en diferentes partes del sitio web. Además, al utilizar nombres significativos para nuestras clases, podemos hacer que nuestro código sea más fácil de leer y entender.

Nucba tip: *Sigan investigando sobre los temas aquí expuestos. Practiquen, apoyense en la documentación oficial y sobre todo mantengan la constancia y la curiosidad a medida que vayan aprendiendo cosas nuevas.*

#HappyCoding 🚀