****

**CENTRO UNIVERSITARIO TECNOLÓGICO**

**CEUTEC**

**ASIGNATURA:**

**INTRODUCCIÓN AL DESARROLLO DE APLICACIONES WEB**

**DOCENTE:**

**CHRISTIAN SÁNCHEZ**

**TEMA DE LA TAREA:**

**¿QUÉ ES CSS?**

**ELAVORADO POR:**

**ALLAN HUMBERTO MENDOZA GOMEZ**

**SECCIÓN: T1026**

**08 DE SEPTIEMBRE DE 2024**

**TEGUCIGALPA, HONDURAS**

* **¿Qué es CSS?**

CSS (Cascading Style Sheets) es un lenguaje utilizado para describir el aspecto visual y el formato de una página web. Se usa junto con HTML (que define la estructura del contenido) para dar estilo a elementos como el texto, imágenes, tablas y otros componentes de una página web. Con CSS puedes controlar aspectos como:

* **Colores** (de texto, fondo, etc.)
* **Fuentes** (tipo, tamaño, estilo)
* **Espaciado** (márgenes, rellenos)
* **Distribución** (diseño de cuadrículas, flexbox)
* **Tamaños** de elementos (anchura, altura)
* **Posicionamiento** de los elementos en la página.

CSS facilita la separación del contenido y el diseño, permitiendo que los desarrolladores web mantengan y actualicen estilos de manera eficiente en una o múltiples páginas.

* **Logo**
* **Resumen breve de su historia y evolución**

CSS fue creado en 1996 por Håkon Wium Lie, mientras trabajaba en el CERN, con el propósito de mejorar la separación entre el contenido HTML y su presentación.

**Principales hitos en su evolución:**

1. **CSS1 (1996)**: La primera versión introdujo conceptos básicos como fuentes, colores y alineación de texto.
2. **CSS2 (1998)**: Añadió mejoras como posicionamiento absoluto, z-index y la posibilidad de aplicar estilos para distintos dispositivos.
3. **CSS3 (2001-actualidad)**: Introdujo un sistema modular con múltiples mejoras, como transiciones, animaciones, y nuevas propiedades para diseño flexible (flexbox y grid).

CSS sigue evolucionando, con nuevas características que mejoran su capacidad para crear interfaces más complejas y adaptables.

* **Muestre ejemplos de cómo dar estilos en CSS (usando ID, clases, o etiquetas)**

1. **Usando etiquetas**:

**HTML**

<p>Este es un párrafo.</p>

<p>Otro párrafo estilizado.</p>

**CSS**

/\* Seleccionar todos los párrafos <p> \*/

p {

color: blue;

font-size: 18px;

}

1. **Usando clases**:

Las clases se aplican a múltiples elementos y puedes reutilizarlas.

**HTML**

<p class="resaltado">Este párrafo está resaltado.</p>

<p>Este no tiene estilos especiales.</p>

<div class="resaltado">Este div también está resaltado.</div>

**CSS**

/\* Selecciona todos los elementos con la clase "resaltado" \*/

resaltado {

background-color: yellow;

font-weight: bold;

}

1. **Usando ID**:

Los IDs son únicos y solo deben aplicarse a un solo elemento en la página.

**HTML**

<h1 id="titulo">Este es el título principal</h1>

<p>Este párrafo no tiene estilos específicos de ID.</p>

**CSS**

/\* Selecciona el elemento con el ID "titulo" \*/

#titulo {

color: red;

text-align: center;

font-size: 24px;

}

* **¿Qué es la metodología BEM? Muestre ejemplos de cómo usarla.**

La **metodología BEM (Block, Element, Modifier)** es una convención para nombrar las clases CSS, que facilita la organización y la mantenibilidad del código, especialmente en proyectos grandes. BEM divide la interfaz en componentes y establece una estructura clara para el nombramiento de clases.

**BEM se basa en tres conceptos principales:**

1. **Bloque (Block)**: Es un componente independiente, como un botón, un menú o un formulario.
2. **Elemento (Element)**: Es una parte del bloque que cumple una función específica. Está relacionado al bloque, pero no puede existir por sí solo.
3. **Modificador (Modifier)**: Es una variación del bloque o del elemento que cambia su apariencia o comportamiento.

**Convenciones de nombramiento:**

* **Bloque**: bloque
* **Elemento**: bloque\_\_elemento (doble guion bajo)
* **Modificador**: bloque--modificador (doble guion medio)

**Ejemplo en HTML y CSS usando BEM**

**HTML:**

<div class="menu">

<div class="menu\_\_item menu\_\_item--active">Inicio</div>

<div class="menu\_\_item">Servicios</div>

<div class="menu\_\_item">Contacto</div>

</div>

**CSS**

/\* Estilos para el bloque "menu" \*/

.menu {

background-color: #f0f0f0;

padding: 10px;

}

/\* Estilos para los elementos "menu\_\_item" \*/

.menu\_\_item {

padding: 10px;

font-size: 16px;

cursor: pointer;

}

/\* Modificador para el elemento activo "menu\_\_item--active" \*/

.menu\_\_item--active {

background-color: #d0d0d0;

font-weight: bold;

}

* **¿Qué son los preprocesadores CSS? Comparta al menos 3 preprocesadores.**

Los **preprocesadores CSS** son herramientas que extienden las funcionalidades de CSS, permitiendo escribir código más limpio, modular y eficiente. Estos preprocesadores ofrecen características avanzadas que CSS por sí solo no tiene, como variables, anidamiento de reglas, mixins, funciones y operaciones matemáticas. El código que escribes con un preprocesador se compila en CSS estándar, listo para ser interpretado por los navegadores.

**Beneficios de los preprocesadores CSS:**

* Uso de **variables** para reutilizar valores como colores o tamaños.
* **Anidamiento** de reglas para reflejar la estructura del HTML y mejorar la legibilidad.
* Creación de **funciones** y **mixins** que permiten reutilizar bloques de código.
* Soporte para **operaciones matemáticas.**

**Tres preprocesadores populares:**

1. **Sass (Syntactically Awesome Stylesheets)**:
   * Es uno de los preprocesadores más populares y poderosos.
   * Permite variables, anidamiento, mixins, funciones y más.
   * Sass tiene dos sintaxis: SCSS (similar a CSS) y la sintaxis original Sass (más concisa, sin llaves ni punto y coma).
2. **LESS (Leaner Style Sheets)**:

* Similar a Sass, pero con una sintaxis ligeramente diferente.
* Permite el uso de variables, mixins, funciones, y anidamiento.
* Fue muy popular en el ecosistema de JavaScript, ya que se integra fácilmente con herramientas como Node.js.

1. **Stylus**:

* Un preprocesador más flexible y minimalista.
* Ofrece una sintaxis opcionalmente sin llaves ni punto y coma, y tiene características avanzadas como interpolación, mixins automáticos y funciones.
* **¿Qué frameworks CSS existen? Muestre al menos 3 ejemplos.**

Un **framework CSS** es un conjunto de estilos predefinidos y componentes que facilitan el diseño de sitios web y aplicaciones. Los frameworks CSS ayudan a los desarrolladores a crear diseños responsive (adaptativos), organizados y estéticamente agradables sin tener que escribir todo el código desde cero.

**Tres ejemplos populares de frameworks CSS:**



* 1. **Bootstrap**:
* Es uno de los frameworks más populares y utilizados.
* Ofrece una amplia gama de componentes listos para usar, como botones, formularios, tarjetas, barras de navegación, entre otros.
* Tiene un sistema de **grid** flexible que permite crear diseños responsive fácilmente.
* Desarrollado por Twitter.
  1. **Foundation**:
* Creado por Zurb, es otro framework muy popular que ofrece un sistema de grid flexible y componentes.
* Similar a Bootstrap, pero con un enfoque ligeramente diferente en la personalización y escalabilidad.
* Incluye herramientas avanzadas como motion UI (para animaciones) y accesibilidad mejorada.



* 1. **Bulma**:
* Un framework ligero y moderno basado completamente en Flexbox, lo que lo hace muy flexible para crear diseños responsive.
* Enfocado en la simplicidad y la facilidad de uso sin depender de JavaScript.
* Ofrece un conjunto de clases predefinidas para diseño, tipografía, formularios, botones y más.