Indice

**Intro a CSS**

Recuerda cómo es una página web sin color, sin estilos, sin personalidad. Simplemente un conjunto de textos y enlaces apilados uno tras otro. Así se ve una página cuando solo tiene HTML, que es como la estructura básica o los "huesos" de un sitio web. Aquí es donde entra en juego CSS, nuestro "maquillador" del mundo digital.

**CSS**, que significa Cascading Style Sheets (u "Hojas de Estilo en Cascada"), es el lenguaje que usamos para darle estilo a una página web. Piensa en CSS como la herramienta que te permite transformar una página web aburrida en algo atractivo y único, añadiendo colores, fuentes, espacios, y mucho más. Con CSS, puedes decidir que tu título sea de un azul vibrante, que el texto de tus párrafos esté centrado, o que tus botones cambien de color cuando alguien pasa el ratón sobre ellos.

**¿Para qué se utiliza CSS?**

CSS se utiliza para separar el contenido de una página web (que está definido en HTML) de su presentación o diseño. Esta separación es importante porque te permite modificar cómo se ve tu página web sin tocar su estructura o su contenido. Por ejemplo, si decides cambiar el color de fondo de todas tus páginas, solo necesitas cambiar una línea de código en tu archivo CSS, ¡y listo! Todos los cambios se aplican automáticamente.

Además, CSS es increíblemente poderoso porque te permite crear diseños responsivos. Es decir, tus páginas web se adaptarán perfectamente a diferentes tamaños de pantalla: desde la pantalla grande de una computadora hasta el pequeño espacio de un teléfono móvil.

**Ventajas de Usar CSS**

* **Estilos Consistentes**: Una sola hoja de estilo puede aplicarse a múltiples páginas web, asegurando que todas se vean de manera coherente.
* **Fácil Mantenimiento**: Cambia el estilo en un solo lugar y verás los cambios reflejados en todas las páginas vinculadas.
* Carga Rápida: Al separar el diseño del contenido, las páginas web cargan más rápido porque el código HTML es más ligero.
* Creatividad al Máximo: CSS te da la libertad de experimentar con colores, fuentes, y estilos para hacer que tu página sea única y atractiva.

CSS es la herramienta que te da el control total sobre cómo se ve tu página web, permitiéndote expresar tu creatividad sin límites. Es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

**Anatomía de una regla CSS**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Observa el código CSS de arriba, un poco más a detalle:

La estructura completa es llamada **regla predeterminada** (pero a menudo «regla» para abreviar). Nota también los nombres de las partes individuales:

**Selector**

El elemento HTML en el que comienza la regla. Esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos <p> ). Para dar estilo a un elemento diferente, solo cambia el selector.

**Declaración**

Una sola regla como color: red; especifica a cuál de las propiedades del elemento quieres dar estilo.

**Propiedades**

Maneras en las cuales puedes dar estilo a un elemento HTML. (En este caso, “color” es una propiedad del elemento <p> ). En CSS, seleccionas qué propiedad quieres afectar en tu regla.

**Valor de la propiedad**

A la derecha de la propiedad, después de los dos puntos (:), tienes el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (hay muchos valores para color además de red).

**Nota**: las otras partes importantes de la sintaxis:

* Cada una de las reglas (aparte del selector) deben estar encapsuladas entre llaves ({}).
* Dentro de cada declaración, debes usar los dos puntos (:) para separar la propiedad de su valor.
* Dentro de cada regla, debes usar el punto y coma (;) para separar una declaración de la siguiente.
* De este modo para modificar varios valores de propiedad a la vez, solo necesitas escribirlos separados por punto y coma (;), así:

Texto

Descripción generada automáticamente

**Cómo Vincular tu Archivo CSS Externo en HTML**

Una de las maneras más comunes y recomendadas de usar CSS es a través de un archivo CSS externo. Esto significa que tendrás un archivo separado (normalmente con la extensión .css) que contendrá todos los estilos (los selectores y las reglas de estilos) de tu página web. Luego, simplemente necesitas decirle a tu documento HTML dónde encontrar este archivo.

Para hacer esto, debes agregar una línea de código en la sección <head> de tu archivo HTML. Así es como lo haces:

1. Primero, crea tu archivo CSS, por ejemplo,styles.css.
2. Luego, dentro del archivo HTML (index.html), agrega el siguiente enlace dentro de la etiqueta<head>:

Texto

Descripción generada automáticamente

La etiqueta <link> es como una flecha que apunta hacia tu archivo CSS. Con el atributo rel="stylesheet", le decimos al navegador que este es un archivo de estilos, y con href="styles.css", especificamos la ruta o el nombre del archivo CSS que queremos usar.

**¿Por qué Vincular un Archivo CSS Externo?**

Usar un archivo CSS externo tiene muchas ventajas:

* **Centralización**: Todos los estilos de tu sitio web están en un solo lugar, lo que facilita su mantenimiento.
* **Reutilización**: Puedes usar el mismo archivo CSS en múltiples páginas de tu sitio, asegurando una apariencia coherente.
* **Limpieza**: Mantiene tu archivo HTML más ordenado y fácil de leer, ya que separa el contenido de la presentación.

¡Y eso es todo! Con solo una línea en tu etiqueta <head>, puedes cambiar el aspecto de toda tu página web. Ahora que sabes cómo vincular tu archivo CSS, estás listo para empezar a darle vida a tu sitio.

QUIZ

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Selectores CSS**

Ahora piensa en CSS como un gran salón de clases lleno de diferentes estudiantes (o elementos HTML). Cada estudiante tiene su propio nombre, su propia personalidad, y su propio estilo. Tu tarea como estilista es decidir qué estilo darle a cada uno. Para ello, utilizamos **selectores CSS**, que son como las etiquetas que usamos para elegir a quién queremos aplicar cada estilo.

Los **selectores** son palabras clave que te permiten decirle al navegador exactamente a qué parte de tu página web quieres aplicar un estilo. Aquí te presentamos los más básicos y útiles:

**Selectores de Elementos: Estilizando por Nombre**

Este es el tipo de selector más simple y directo. Imagina que tienes un grupo de elementos en tu página, como varios párrafos (<p>) o títulos (<h1>). Si quieres que todos los párrafos tengan el mismo color de texto, entonces se usa el nombre del elemento como selector.

p {

  color: blue;

}

¡Así de fácil! Con este pequeño bloque de código, todos los párrafos (<p>) de tu página tendrán texto de color azul.

**Selectores de Clase: Estilizando por Grupo**

A veces, no quieres estilizar todos los elementos de un mismo tipo. Quizás quieres que solo algunos párrafos tengan un color diferente. Aquí es donde entran los selectores de clase.

Las clases son como etiquetas que puedes asignar a cualquier elemento en tu página. Para crear un selector de clase en CSS, simplemente comienza con un punto (.) seguido del nombre de la clase.

.mi-clase {

  background-color: yellow;

}

Y en tu HTML, solo necesitas agregar la clase a los elementos que quieras estilizar a través del atributo **“class”** en la etiqueta de apertura de dichos elementos:

<p class="mi-clase">Este párrafo tiene un fondo amarillo.</p>

**Selectores de ID: Estilizando de Forma Única**

Ahora, imagina que hay un único estudiante en tu clase al que quieres darle un estilo especial, diferente de todos los demás. Aquí es donde usas los selectores de ID (identificador). Cada ID debe ser único dentro de tu página. En CSS, los selectores de ID comienzan con una almohadilla (#).

#mi-id {

  font-size: 24px;

}

En tu HTML, asignas este ID al elemento que quieras a través del atributo “id” en la etiqueta de apertura:

<h1 id="mi-id">¡Soy único y tengo un tamaño de fuente diferente!</h1>

**Selectores Universales: ¡Estiliza Todo!**

Finalmente, si quieres aplicar un estilo a todos los elementos de tu página (como cambiar el color de fondo), puedes usar el selector universal: \*.

\* {

  margin: 0;

  padding: 0;

}

Este código de ejemplo eliminaría todos los márgenes y rellenos de cada elemento de tu página al asignarles un valor 0.

**Selectores de atributo**

Este grupo de selectores te proporciona diferentes formas de seleccionar elementos según la presencia de un atributo determinado en un elemento:

a[title] {

    /\* reglas \*/

}

O incluso hacer una selección basada en la presencia de un atributo que tiene un valor particular asignado:

a[href="https://google.com"] {

    /\* reglas \*/

}

**Las pseudoclases y los pseudoelementos**

Este grupo de selectores incluye pseudoclases, que aplican estilo a ciertos estados de un elemento. La pseudoclase :hover, por ejemplo, selecciona un elemento solo cuando se le pasa el ratón por encima.

button: hover {    /\* reglas \*/}

También incluye pseudoelementos, que seleccionan una parte determinada de un elemento en vez del elemento en sí. Por ejemplo, ::first-line siempre selecciona la primera línea del texto que se encuentra dentro de un elemento (<p>, en el ejemplo siguiente), y actúa como si un elemento <span> hubiera delimitado la primera línea, seleccionado y aplicado estilo.

p::first-line {    /\* reglas \*/}

**Combinadores**

El último grupo de selectores combina otros selectores con el fin de delimitar elementos de nuestros documentos. El ejemplo siguiente selecciona los párrafos que son hijos directos del elemento <article> utilizando el operador de combinación hijo (>):

article > p {

    /\* reglas \*/

}

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Displays**

Cuando empiezas a diseñar una página web, te das cuenta de que no todos los elementos se comportan de la misma manera. Algunos se alinean uno tras otro, como las palabras en una frase, mientras que otros ocupan todo el ancho de la página, empujando a los demás hacia abajo. La forma en que cada elemento se comporta en la página depende de una propiedad “mágica” en CSS llamada display.

La propiedad display es como una instrucción que le dice al navegador cómo debe comportarse cada elemento en tu página. Vamos a descubrir algunos de los valores más importantes de esta propiedad para que puedas empezar a controlar el flujo y la disposición de tus elementos web.

**1. display: inline - Elementos en Línea**

Imagina que estás escribiendo un párrafo. Las palabras y las letras se colocan una tras otra, en la misma línea, hasta que llegan al final y saltan a la siguiente. Los elementos con display: inline funcionan de la misma manera. Son elementos que se alinean en la misma línea, uno tras otro, sin comenzar una nueva línea.

Por ejemplo, elementos como <span>, <a> (enlaces), y <em> (énfasis) son, por defecto, elementos en línea. Esto significa que ocupan solo el espacio que necesitan para su contenido, y puedes tener muchos de ellos en la misma línea.

<a href="#">Este es un enlace en línea</a>

<span> y este es un texto dentro de un span.</span>

Si aplicas un color de fondo a estos elementos, notarás que solo se colorea la parte donde está el texto, no toda la línea. Los elementos inline no respetan propiedades como width o height; solo se ajustan al tamaño de su contenido.

**2. display: block - Elementos en Bloque**

Ahora, imagina que estás organizando cajas en una estantería, pero cada caja es tan ancha que ocupa toda la estantería. Los elementos con display: block son como esas cajas. Ocupan todo el ancho disponible y comienzan en una nueva línea.

Elementos como <div>, <h1>, <p>, y <ul> son, por defecto, elementos de bloque. Esto significa que cada vez que los usas, empiezan en una nueva línea y ocupan todo el ancho disponible de su contenedor.

<div>Este es un elemento de bloque y ocupa toda la línea.</div>

<p>Este es otro elemento de bloque que también comienza en una nueva línea.</p>

Puedes ajustar su altura y ancho, agregar márgenes y rellenos, y controlar completamente su disposición en la página.

**3. display: inline-block - Lo Mejor de Dos Mundos**

Imagina que estás organizando elementos en una fila, como lo harías con inline, pero con la capacidad de darles un tamaño específico, como con block. Así es como funciona inline-block. Este valor de display combina lo mejor de ambos mundos: los elementos con inline-block se alinean uno tras otro en la misma línea (como inline), pero también respetan propiedades como width, height, margin, y padding (como block). Esto es ideal para crear menús de navegación horizontales, tarjetas de producto en una tienda en línea, o cualquier diseño donde necesites que los elementos estén en fila, pero con control sobre su tamaño y espacio.

**4. display: flex - Flexibilidad en la Distribución**

Cuando necesitas un poco más de control sobre cómo se alinean tus elementos, display: flex es tu mejor aliado. Piensa en flexbox como un organizador flexible para tu estantería. Puedes alinear tus cajas (elementos) de diferentes maneras, distribuir el espacio entre ellas, y asegurarte de que todo se vea bien, incluso cuando el tamaño de la pantalla cambia.

En lecciones posteriores entraremos en detalle de flexbox y de grid.

**5. display: grid - La Rejilla Maestra**

Si flexbox es un organizador flexible para tu estantería, display: grid es una rejilla maestra. Imagina una cuadrícula que te permite colocar elementos en filas y columnas. Con grid, puedes crear diseños complejos en dos dimensiones: filas y columnas al mismo tiempo.

display: grid es ideal cuando necesitas un control total sobre el diseño de tu página. Puedes definir cuántas columnas quieres, su tamaño, el espacio entre ellas, y exactamente dónde debe estar cada elemento.

**6. display: none - Desapareciendo en Acción**

A veces, necesitas que un elemento simplemente… desaparezca. Tal vez es un mensaje de error que quieres ocultar después de un tiempo, o una sección de tu página que solo debería mostrarse en dispositivos móviles. Para estos casos, usamos display: none.

Cuando aplicas display: none a un elemento, este desaparece por completo de la página. No ocupa espacio ni es visible para los usuarios. Es como si nunca hubiera existido.

.error-message {

  display: none; /\* Oculta el mensaje de error \*/

}

Recuerda:**display: none** es **diferente** de **visibility: hidden**, que solo hace invisible el elemento, pero sigue ocupando su espacio en la página.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Hover**

**Lectura: Usando :hover para Mejorar tu Página Web**

En CSS, hay maneras de cambiar el estilo de un elemento cuando ocurre una acción específica. Estas formas especiales de aplicar estilos se llaman pseudoclases.

**¿Qué es una Pseudo-clase?**

Una pseudo-clase en CSS es una palabra clave que se añade al selector para aplicar estilos en un estado específico de un elemento. Es una forma de decirle al navegador que queremos cambiar el aspecto de algo cuando pasa algo en particular. Por ejemplo, **cuando pasas el ratón sobre un botón o un enlace**. La pseudo-clase más común y útil para esto es :hover.

La pseudo-clase :hover en CSS permite cambiar el estilo de un elemento cuando el usuario pasa el ratón por encima de él. Es una forma sencilla de hacer que tu página sea más interactiva y atractiva.

**Cómo Usar :hover**

Para usar :hover, necesitas agregar una regla CSS que cambie el estilo de un elemento cuando el ratón está encima de él. Aquí tienes un ejemplo simple:

**Código HTML:**

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Ejemplo de Hover</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <button>¡Pasa el ratón por aquí!</button>

</body>

</html>

**Código CSS (styles.css):**

button {

  background-color: grey; /\* Color inicial del botón \*/

  color: white;

  padding: 10px 20px;

  border: none;

  cursor: pointer;

}

button:hover {

  background-color: blue; /\* Color del botón cuando el ratón pasa por encima \*/

}

**Qué Hace Este Código**

* El botón tiene un fondo gris y texto blanco al inicio.
* Cuando el ratón pasa por encima del botón (hover), el fondo cambia a azul.

**Prueba este Ejemplo**

1. Copia y pega el código HTML en un archivo llamadoindex.html.
2. Crea un archivo CSS llamadostyles.cssy copia el código CSS en él.

Abre el archivo index.html en tu navegador y pasa el ratón sobre el botón para ver el efecto.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Introducción a CSS**

En este proyecto, practicarás nuevamente los comandos de git así como las prácticas recomendadas para crear un pproyecto básico de HTML y poder publicarlo en Gituhb Pages. La idea es que este repositorio sea el que trabajes con cada nuevo tema aprendido para darle forma a tu proyecto de este módulo.

**Objetivo**

El objetivo es crear un repositorio local con los archivos de una página web sencilla que incluya la configuración de una hoja de estilos externa a HTML (archihvo css). También se pretende practicar la creación y publicación de dicho repositorio a través de Github Pages.

**Actividades**

1. Crea un repositorio en tu equipo local en la carpeta donde se ubicarán los archivos que forman parte de tu proyecto.
2. En dicha carpeta recuerda incluir un archivo index.html y un archivo de tipo .css para que contenga los estilos de tu página.
3. Crea un repositorio (remoto) en tu cuenta de Github.
4. Sigue la secuencia de comandos de git para vincular tu repositorio local con el repositorio recién creado (remoto) en Github y así poder subir tus archivos.
5. Realiza la configuración necesaria para poder hacer público tu proyecto desde Github con la configuración de Github Pages.
6. Comparte la liga de tu repositorio creado en Github así como la URL pública de tu proyecto.

**Entregables**

Para este proyecto, debes compartir la URL de tu repositorio recién creado así como la URL pública de tu proyecto en Github Pages.

1. Comparte la liga de tu repositorio creado en Github.
2. Comparte la URL pública de tu proyecto en Github Pages.

Captura de pantalla de computadora

Descripción generada automáticamente