

Como ver tu página

Para ver tu página web que estás creando con HTML y CSS, sigue estos pasos:

Guarda el archivo HTML con un nombre de archivo reconocible y una extensión .html. Por ejemplo, "mi_pagina_web.html".

Guarda el archivo CSS con un nombre de archivo reconocible y una extensión .css. Por ejemplo, "estilos.css".

Abre el archivo HTML en tu navegador web favorito. Para hacerlo, haz clic derecho en el archivo HTML y selecciona "Abrir con" y luego selecciona tu navegador web preferido, como Google Chrome, Mozilla Firefox o Microsoft Edge.

Asegúrate de que tanto el archivo HTML como el archivo CSS estén en la misma carpeta. Si no es así, es posible que los estilos CSS no se apliquen correctamente a la página.

Si estás trabajando en un sitio web más grande, es posible que desees subir tus archivos a un servidor web para que otros puedan verlos en línea. En ese caso, tendrías que publicar tus archivos HTML y CSS en el servidor web utilizando un programa FTP o un servicio de alojamiento web.

Siguiendo estos pasos, podrás ver tu página web que estás creando con HTML y CSS en tu navegador web. Recuerda que para ver los cambios que haces en el código, necesitas guardar el archivo y volver a cargar la página en tu navegador.

Ké es HTML y CSS? HTML (Hypertext Markud Lenguaje) es un lenguaje de marcas ke se usa para krear páginas web. CSS (Cascatin Style Seets) es un lenguaje de estilo ke se utiliza para dar estilo y diseño a las páginas web. HTML se uzan para estructurar el contenido de la página, komo texto, imágenes, enlaces, listas y tablas. CSS se uzan para kambiar el tamaño del texto, el color de fondo, el tipo de letra y otros estilos visuales. Juntos, HTML y CSS forman la base de la mayoría de las páginas web que vemos hoy en día en internet.

Estructura básica de un documento HTML

A estructura bázica de un documento HTML se compone de las siguientes partes:

<!DOCTYPE html>: Esto indica al navegador que el documento es de tipo HTML.

<html>: Este es el elemento raíz del documento HTML.

<head>: Aquí se colocan los metadatos de la página, como el título y la descripción.

<body>: Este es el cuerpo de la página, donde se coloca todo el contenido visible de la página, como texto, imágenes, enlaces, etc.

Así es como se estructura un documento HTML básico.

Ejemplo de estructura básica

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Título de la página</title>
```

```
    <meta charset="UTF-8">
```

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

```
  </head>
```

```
  <body>
```

```
    <h1>Título principal de la página</h1>
```

```
    <p>Este es un ejemplo de párrafo.</p>
```

```
  </body>
```

```
</html>
```

Este código define la estructura básica de un documento HTML, que consiste en un elemento html que contiene dos subelementos: head y body. El head contiene información sobre la página, como el título y la codificación de caracteres. El body contiene el contenido visible de la página, como encabezados (h1) y párrafos (p)

Las etiquetas

Las etiquetas en HTML son elementos que se utilizan para marcar el contenido de la página y establecer su estructura y formato. Cada etiqueta está delimitada por corchetes angulares < >, y puede tener atributos que proporcionan información adicional sobre la etiqueta.

Aquí hay algunos ejemplos de etiquetas en HTML con su correspondiente código:

La etiqueta <h1> define un encabezado principal en la página:

```
<h1>Este es un encabezado principal</h1>
```

La etiqueta <p> se utiliza para crear un párrafo:

```
<p>Este es un ejemplo de párrafo en HTML. Puedes escribir varios párrafos en una página para estructurar tu contenido.</p>
```

La etiqueta <a> se utiliza para crear un enlace a otra página:

```
<a href=https://www.ejemplo.com>Esto es un enlace</a>
```

En este ejemplo, la etiqueta <a> tiene un atributo href que indica la dirección web a la que se debe dirigir el enlace.

La etiqueta se utiliza para insertar imágenes en la página

```

```

En este ejemplo, la etiqueta tiene dos atributos: src que indica la ubicación de la imagen y alt que proporciona un texto alternativo para describir la imagen en caso de que no se pueda mostrar.

Estos son solo algunos ejemplos de etiquetas en HTML. Hay muchas más etiquetas disponibles para crear diferentes tipos de contenido y estructura en la página web.

Como poner una imagen en tu archivo html

Lo primero que debes hacer es encontrar una imagen que quieras usar. Puede ser una foto tuya, de tu familia o de algún dibujo que te guste. Guarda la imagen en tu computadora.

Abre un editor de texto como el Bloc de notas en tu computadora.

Escribe el siguiente código:

```

```

En donde dice nombre_de_la_imagen.jpg, escribe el nombre del archivo de imagen que guardaste en tu computadora.

Guarda el archivo con una extensión .html y ciérralo.

Abre el archivo en un navegador web. Deberías ver la imagen en la página.

¡Y listo! Ahora tienes una imagen en tu página web. Asegúrate de que el archivo de imagen esté en la misma carpeta que el archivo HTML para que pueda encontrar la imagen. Y recuerda que el texto alternativo es importante para describir la imagen en caso de que no se pueda mostrar

Ejemplo básico de una página simple

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Mi primera página web</title>
```

```
</head>
```

```
<body>
```

```
<h1 style="text-align:center;">Mi primera página web</h1>
```

```

```

```
<marquee>Soy un texto en movimiento</marquee>
```

```
</body>
```

```
</html>
```

“En este ejemplo, estamos haciendo una página web muy sencilla con un título, una imagen y un texto que se mueve. Para hacer esto, usamos etiquetas especiales de HTML. La etiqueta <title> es para el título de la página, la etiqueta <h1> es para un título grande y la etiqueta es para agregar una imagen. Para centrar todo, usamos la propiedad text-align y margin:auto dentro de la etiqueta <h1> y la etiqueta . Para hacer que el texto se mueva, usamos la etiqueta <marquee>. Es así de fácil hacer una página web básica en HTML.”

Poner videos

```
<!DOCTYPE html>

<html>

  <head>

    <title>Mi video en HTML</title>

  </head>

  <body>

    <h1>Mi video en HTML</h1>

    <video width="640" height="360" controls>

      <source src="video.mp4" type="video/mp4">

      Tu navegador no soporta videos en HTML.

    </video>

  </body>

</html>
```

Explicado de forma simple, lo que estamos haciendo es usar la etiqueta <video> para agregar un video a nuestra página web. El atributo width y height define el ancho y alto del reproductor de video en píxeles. El atributo controls permite que los usuarios puedan reproducir, pausar y detener el video, ajustar el volumen, etc.

Dentro de la etiqueta <video>, usamos la etiqueta <source> para especificar la ubicación del archivo de video. El atributo src es para la URL del archivo de video y el atributo type es para el tipo de archivo. En este ejemplo, estamos utilizando un archivo de video MP4.

Si el navegador no puede reproducir el video, se mostrará el texto dentro de la etiqueta <video>

En resumen

Para poner un video en nuestra página web, necesitamos usar una etiqueta especial llamada <video>. Dentro de esta etiqueta, agregamos la ubicación del video usando la etiqueta <source>. También agregamos el ancho y alto del reproductor de video y permitimos que los usuarios puedan controlar el video usando controls. Así de fácil es agregar un video a nuestra página

Que son los atributos

En HTML, los atributos son información adicional que se puede agregar a las etiquetas para modificar su comportamiento o aspecto. Los atributos se escriben dentro de la etiqueta utilizando la sintaxis atributo="valor".

Aquí hay un ejemplo de cómo se puede usar el atributo src para agregar una imagen a una página web:

```

```

En este ejemplo, la etiqueta se utiliza para agregar una imagen a la página. El atributo src especifica la ubicación del archivo de imagen, y el atributo alt proporciona una descripción alternativa de la imagen para las personas que no pueden ver la imagen (por ejemplo, si tienen problemas de visión o están usando un navegador que no puede mostrar imágenes).

Otro ejemplo es el atributo href en la etiqueta <a>, que se utiliza para crear un enlace a otra página o sitio web:

```
<a href=https://www.ejemplo.com>Ir a Ejemplo.com</a>
```

En este ejemplo, la etiqueta <a> se utiliza para crear un enlace a la página web de Ejemplo.com. El atributo href especifica la URL de la página web a la que se enlaza.

Comentando el código, podemos ver que el atributo src en la etiqueta está especificando la ubicación del archivo de imagen (en este caso, "mi_imagen.jpg") y el atributo alt proporciona una descripción alternativa de la imagen para las personas que no pueden ver la imagen.

En el segundo ejemplo, el atributo href en la etiqueta <a> está especificando la URL de la página web a la que se enlaza (en este caso, <https://www.ejemplo.com>) y el texto dentro de la etiqueta <a> es lo que se mostrará como el texto del enlace en la página web (en este caso, "Ir a Ejemplo.com"

Que es css

CSS (Cascading Style Sheets) es un lenguaje de diseño que se utiliza para dar estilo y formato a las páginas web. CSS se utiliza para controlar el diseño, la apariencia y el estilo de los elementos HTML en una página web. Es un lenguaje independiente de HTML, pero se utiliza junto con HTML para crear páginas web atractivas y bien diseñadas.

HTML se utiliza para estructurar el contenido de una página web, mientras que CSS se utiliza para controlar cómo se ve ese contenido. Por ejemplo, HTML se utiliza para agregar encabezados, párrafos, listas y otros elementos de contenido a una página web, y CSS se utiliza para controlar el color, la fuente, el tamaño y la posición de esos elementos.

CSS permite a los desarrolladores web crear diseños complejos y atractivos sin tener que preocuparse por la estructura subyacente de HTML. Por lo tanto, CSS y HTML trabajan juntos para proporcionar una experiencia de usuario más atractiva y agradable en la web.

Como cambiar el color de la letra y el tamaño del texto usando css y html

```
<!DOCTYPE html>

<html>

  <head>

    <title>Ejemplo de CSS</title>

    <style>

      /* Estilo para el párrafo */

      P {

        Color: blue;

        Font-size: 34px;

      }

    </style>

  </head>

  <body>

    <p>Hola mundo</p>

  </body>
```

</html>

En este ejemplo, se ha utilizado CSS para cambiar el color y el tamaño de la letra del párrafo. La etiqueta <style> se utiliza para definir el estilo del párrafo dentro del archivo HTML. La regla p se utiliza para aplicar el estilo al elemento <p>.

La propiedad color se utiliza para cambiar el color de la letra a azul, y la propiedad font-size se utiliza para establecer el tamaño de la letra en 34 píxeles.

El resultado es que el texto "Hola mundo" se muestra en la pantalla en color azul y con un tamaño de letra de 34 píxeles.

Formularios

Los formularios en HTML son una forma de recopilar información del usuario a través de una página web. Se utilizan para crear interacciones entre el usuario y la página, como enviar mensajes, realizar búsquedas, realizar pedidos o cualquier otra tarea que implique la entrada de información.

Los formularios constan de elementos de formulario, como campos de entrada de texto, botones de radio, casillas de verificación y menús desplegables, así como de un botón de envío que permite al usuario enviar la información ingresada.

Cada elemento del formulario tiene un nombre y un valor que se envía al servidor cuando el usuario envía el formulario. Los formularios se procesan en el servidor, donde se utiliza el lenguaje de programación para realizar la tarea requerida, como enviar un correo electrónico o actualizar una base de datos.

Los formularios son una parte importante del diseño web y permiten a los desarrolladores web crear interacciones personalizadas y útiles para los usuarios.

Ejemplo

Para crear un formulario HTML que pida el nombre y el teléfono y lo envíe por correo electrónico, puedes utilizar el siguiente código:

```
<form action=mailto:destinatario@ejemplo.com method="post" enctype="text/plain">
```



```
<label for="nombre">Nombre:</label>
```

```
<input type="text" id="nombre" name="nombre"><br>
```

```
<label for="telefono">Teléfono:</label>
```

```
<input type="tel" id="telefono" name="telefono"><br>
```

```
<input type="submit" value="Enviar">
```

```
</form>
```

La etiqueta form indica que se va a crear un formulario. El atributo action indica la dirección de correo electrónico a la que se enviará el formulario. En este caso, la dirección de correo electrónico es destinatario@ejemplo.com.

El atributo method indica el método HTTP que se utilizará para enviar el formulario. En este caso, se utilizará el método post.

El atributo enctype indica el tipo de codificación que se utilizará para enviar los datos del formulario. En este caso, se utilizará la codificación text/plain.

La etiqueta label se utiliza para asociar un texto con un campo de entrada. En este caso, se crearán dos etiquetas: una para el nombre y otra para el teléfono.

La etiqueta input se utiliza para crear campos de entrada en el formulario. El atributo type indica el tipo de campo de entrada que se creará. En este caso, se crearán dos campos de entrada de tipo text y tel para el nombre y el teléfono, respectivamente. El atributo id se utiliza para identificar el campo de entrada, y el atributo name se utiliza para indicar el nombre del campo de entrada en el correo electrónico que se enviará.

La etiqueta br se utiliza para crear saltos de línea en el formulario.

La etiqueta input se utiliza para crear un botón de envío. El atributo type indica que se creará un botón de tipo submit, y el atributo value indica el texto que se mostrará en el botón. En este caso, el texto es "Enviar".

Para que el formulario se envíe por correo electrónico, es necesario que el servidor que aloja la página web esté configurado para enviar correos electrónicos. En este caso, se está utilizando la sintaxis de formulario básica que envía los datos directamente al cliente de correo electrónico predeterminado del usuario. Sin embargo, esto puede no funcionar en algunos navegadores o configuraciones de correo electrónico.

Otra opción es utilizar un script en el servidor para procesar los datos del formulario y enviar un correo electrónico utilizando un lenguaje de programación como PHP, Python o Ruby. En este caso, el atributo action del formulario se establecería en la dirección del script en el servidor que procesaría los datos.

En cualquier caso, el código HTML del formulario proporcionado simplemente define la estructura del formulario y los datos que se recopilarán, pero no realiza ninguna acción en sí misma para enviar los datos del formulario por correo electrónico.

Etiquetas más usadas

Aquí te presento una lista de 20 etiquetas HTML más utilizadas en el desarrollo de páginas web:

`<html>`: Esta etiqueta define el principio y el final de un documento HTML y todo su contenido debe estar entre estas etiquetas.

`<head>`: Esta etiqueta se utiliza para contener información acerca del documento, como por ejemplo, la información de la página, meta-etiquetas, enlaces a estilos CSS, scripts, etc.

`<title>`: Esta etiqueta se utiliza para definir el título de la página web y se muestra en la pestaña del navegador.

`<body>`: Esta etiqueta se utiliza para definir el cuerpo del documento HTML y todo lo que se muestre en la página debe estar entre estas etiquetas.

`<h1>` - `<h6>`: Estas etiquetas se utilizan para definir los encabezados de la página, siendo el `<h1>` el encabezado principal y el `<h6>` el encabezado de menor importancia.

`<p>`: Esta etiqueta se utiliza para definir párrafos de texto.

``: Esta etiqueta se utiliza para mostrar imágenes en la página web. Se puede definir la ubicación de la imagen utilizando el atributo src, y se pueden agregar otras propiedades como alt, title, width y height.

`<a>`: Esta etiqueta se utiliza para crear enlaces a otras páginas web o a otros documentos. Se puede definir la dirección del enlace utilizando el atributo href, y se puede agregar texto o imágenes para hacer que el enlace sea visible.

`` y ``: Estas etiquetas se utilizan para crear listas sin orden específico. La etiqueta `` define la lista no ordenada y cada elemento de la lista se define con la etiqueta ``.

`` y ``: Estas etiquetas se utilizan para crear listas ordenadas. La etiqueta `` define la lista ordenada y cada elemento de la lista se define con la etiqueta ``.

`<div>`: Esta etiqueta se utiliza para dividir el contenido en secciones lógicas. Se puede utilizar para aplicar estilos CSS a secciones específicas de la página.

``: Esta etiqueta se utiliza para aplicar estilos CSS a un pequeño fragmento de texto dentro de una línea de texto.

`<form>`: Esta etiqueta se utiliza para crear un formulario que el usuario puede rellenar y enviar. Se pueden agregar campos de entrada como texto, números, selecciones, etc. Utilizando etiquetas como `<input>`.

`<input>`: Esta etiqueta se utiliza para crear campos de entrada en un formulario, como campos de texto, botones de opción, casillas de verificación, botones de radio, etc.

`<textarea>`: Esta etiqueta se utiliza para crear un área de texto para que el usuario pueda escribir en un formulario.

`<label>`: Esta etiqueta se utiliza para asociar un texto con un campo de entrada en un formulario, para que los usuarios sepan qué se espera que ingresen en cada campo.

`<table>`: Esta etiqueta se utiliza para crear una tabla en la página web. Se pueden definir filas y columnas utilizando las etiquetas `<tr>` y `<td>`.

<thead>, <tbody> y <tfoot>: Estas etiquetas se utilizan para dividir una tabla

Cabecera (<thead>), cuerpo (<tbody>) y pie (<tfoot>), lo que permite aplicar estilos CSS específicos a cada sección de la tabla.

<style>: Esta etiqueta se utiliza para agregar estilos CSS directamente en la página web. Los estilos se pueden aplicar a elementos específicos utilizando selectores CSS.

<script>: Esta etiqueta se utiliza para agregar scripts o programas en la página web, como por ejemplo, programas JavaScript que permiten agregar interactividad a la página web.

Eh aquí un ejemplo de página web simple

```
<!DOCTYPE html>

<html>

<head>

  <title>Venta de Carros</title>

  <style>

    Body {

      Text-align: center;

    }

    H1 {

      Font-size: 2.5em;

      Margin-top: 50px;

    }

    P {

      Font-size: 1.5em;

      Margin-bottom: 30px;
```

```
}  
Video {  
    Width: 100%;  
    Max-width: 800px;  
    Height: auto;  
}  
Button {  
    Background-color: #4CAF50;  
    Border: none;  
    Color: white;  
    Padding: 15px 32px;  
    Text-align: center;  
    Text-decoration: none;  
    Display: inline-block;  
    Font-size: 1.5em;  
    Margin-top: 30px;  
    Cursor: pointer;  
}  
</style>  
</head>  
<body>  
    <h1>Venta de Carros</h1>  
    <p>En nuestra tienda ofrecemos una amplia variedad de carros nuevos y usados. Contáctanos  
para obtener más información o visita nuestro inventario en línea.</p>  
    <video controls>  
        <source src="video.mp4" type="video/mp4">  
        Tu navegador no soporta la etiqueta de video.  
    </video>  
    <br>
```

```
<button>Contáctanos</button>  
  
<button>Visita nuestro inventario</button>  
  
</body>  
  
</html>
```

Css explicación de que hacen cada cosa

```
Body {  
  Text-align: center;  
}
```

Esta línea de código establece que todo el texto dentro del cuerpo de la página debe estar centrado horizontalmente.

```
H1 {  
  Font-size: 2.5em;  
  Margin-top: 50px;  
}
```

Esta línea de código establece que el tamaño de la fuente para todos los encabezados de nivel 1 (<h1>) debe ser de 2.5 em. Además, se agrega un margen superior de 50 píxeles para separar el encabezado del resto del contenido.

```
P {  
  Font-size: 1.5em;  
  Margin-bottom: 30px;  
}
```

Esta línea de código establece que el tamaño de la fuente para todos los párrafos (<p>) debe ser de 1.5 em. Además, se agrega un margen inferior de 30 píxeles para separar el párrafo del contenido que le sigue.

```
Video {  
  
    Width: 100%;  
  
    Max-width: 800px;  
  
    Height: auto;  
  
}
```

Esta línea de código establece que cualquier elemento video en la página debe tener un ancho del 100% de su contenedor, un ancho máximo de 800 píxeles y una altura automática para mantener la proporción de aspecto original del video.

```
Button {  
  
    Background-color: #4CAF50;  
  
    Border: none;  
  
    Color: white;  
  
    Padding: 15px 32px;  
  
    Text-align: center;  
  
    Text-decoration: none;  
  
    Display: inline-block;  
  
    Font-size: 1.5em;  
  
    Margin-top: 30px;  
  
    Cursor: pointer;  
  
}
```

Esta línea de código establece el estilo para todos los botones (<button>) en la página. Se establece un color de fondo verde (#4CAF50), se elimina el borde del botón (border: none), se establece el color del texto en blanco, se agrega un relleno interno de 15 píxeles en la parte superior y inferior y 32 píxeles en los lados, se centra el texto horizontalmente, se elimina la decoración de texto subrayado, se establece que el botón es un elemento en línea y se establece el tamaño de fuente en 1.5 em. Además, se agrega un margen superior de 30 píxeles para separar los botones del contenido que le precede. Por último, se establece el cursor en “pointer” para indicar que el botón es interactivo y se puede hacer clic en él.

```
.container {  
  Max-width: 800px;  
  Margin: auto;  
  Padding: 0 30px;  
}
```

Esta línea de código establece el estilo para un contenedor con clase “container”. Se establece un ancho máximo de 800 píxeles para el contenedor y se centra horizontalmente en la página utilizando margin: auto. Además, se agrega un relleno de 30 píxeles a la izquierda y a la derecha del contenedor.

```
.center {  
  Display: flex;  
  Justify-content: center;  
  Align-items: center;  
  Height: 100vh;  
}
```

Esta línea de código establece el estilo para un contenedor con clase “center”. Se utiliza la propiedad display: flex para convertir el contenedor en un contenedor flexible. Luego, se utiliza justify-content: center y align-items: center para centrar verticalmente y horizontalmente el contenido dentro del contenedor. Finalmente, se establece la altura del contenedor en el 100% de la altura de la ventana gráfica (height: 100vh), lo que garantiza que el contenido esté siempre centrado en la pantalla.

```
.video-text {  
  Margin-top: 50px;  
  Font-size: 1.2em;  
  Text-align: center;  
}
```


Esta línea de código establece el estilo para un elemento con clase “video-text”. Se agrega un margen superior de 50 píxeles para separar el texto del video, se establece el tamaño de fuente en 1.2 em y se centra el texto horizontalmente utilizando text-align: center.

Es importante tener en cuenta que, aunque las líneas de código CSS anteriores son comunes en muchas páginas web, la implementación específica de CSS puede variar ampliamente dependiendo de las necesidades y el diseño de cada sitio web.

```
Button {  
  
    Background-color: #008CBA;  
  
    Color: white;  
  
    Padding: 12px 20px;  
  
    Border: none;  
  
    Border-radius: 4px;  
  
    Cursor: pointer;  
  
    Font-size: 1.2em;  
  
}
```

Esta línea de código establece el estilo para todos los botones en la página. Establece un color de fondo de “#008CBA” (un tono de azul), texto blanco, un relleno de 12 píxeles en la parte superior e inferior y 20 píxeles a la izquierda y derecha, sin borde, con esquinas redondeadas de 4 píxeles, y un cursor de puntero para indicar que el botón es interactivo. También establece el tamaño de fuente en 1.2 em.

```
H1 {  
  
    Font-size: 2.5em;  
  
    Font-weight: bold;  
  
    Text-align: center;  
  
}
```

Esta línea de código establece el estilo para todos los encabezados h1 en la página. Establece el tamaño de fuente en 2.5 em, el peso de la fuente en negrita y el texto alineado al centro.

```
P {  
    Font-size: 1.1em;  
    Line-height: 1.5em;  
    Margin-bottom: 20px;  
}
```

Esta línea de código establece el estilo para todos los párrafos en la página. Establece el tamaño de fuente en 1.1 em, la altura de línea en 1.5 em para dar un espacio adecuado entre líneas y un margen inferior de 20 píxeles para separar los párrafos.

```
Img {  
    Max-width: 100%;  
    Height: auto;  
}
```

Esta línea de código establece el estilo para todas las imágenes en la página. Establece un ancho máximo del 100% para asegurarse de que la imagen se ajuste correctamente a su contenedor y establece la altura en “auto” para mantener la proporción de aspecto de la imagen.

Si aún no entiendes como funciona el css y html aquí te dejo otro ejemplo

El CSS y el HTML trabajan juntos para dar estilo y estructura a una página web. Aquí hay un ejemplo simple de cómo funciona:

Primero, crearemos un archivo HTML llamado “ejemplo.html” con el siguiente contenido:

```
<!DOCTYPE html>  
  
<html>  
  
    <head>
```

```
<title>Ejemplo de CSS y HTML</title>

<link rel="stylesheet" type="text/css" href="estilos.css">

</head>

<body>

  <h1>Bienvenido a mi sitio web</h1>

  <p>Este es un ejemplo de cómo usar CSS y HTML juntos para crear páginas web increíbles.</p>

  <ul>

    <li><a href="#">Inicio</a></li>

    <li><a href="#">Acerca de</a></li>

    <li><a href="#">Contacto</a></li>

  </ul>

</body>

</html>
```

Aquí, tenemos un documento HTML básico con un encabezado que contiene un título y una referencia a un archivo CSS llamado "estilos.css". En el cuerpo de la página, tenemos un encabezado h1, un párrafo y una lista desordenada con tres enlaces.

Ahora, crearemos un archivo CSS llamado "estilos.css" en el mismo directorio que el archivo HTML y agregaremos el siguiente contenido:

```
H1 {

  Color: blue;

  Font-size: 3em;

  Text-align: center;

}
```

```
P {

  Font-size: 1.2em;

  Line-height: 1.5em;

}
```

```
ul {  
    List-style-type: none;  
    Margin: 0;  
    Padding: 0;  
    Text-align: center;  
}
```

```
li {  
    Display: inline-block;  
    Margin: 0 10px;  
}
```

```
a {  
    Color: blue;  
    Text-decoration: none;  
    Transition: color 0.3s ease;  
}
```

```
a:hover {  
    Color: orange;  
}
```

Aquí, tenemos un conjunto de reglas CSS que se aplicarán a los elementos HTML correspondientes en nuestro documento HTML. Por ejemplo, la regla h1 establece el color del texto en azul, el tamaño de fuente en 3 em y el texto centrado. La regla p establece el tamaño de fuente en 1.2 em y la altura de línea en 1.5 em. La regla ul establece el estilo de lista como sin viñetas, sin margen y sin relleno, y alinea el texto al centro. La regla li establece que los elementos de lista deben mostrarse en línea con un margen izquierdo y derecho de 10 píxeles. La regla a establece que los enlaces deben ser azules y sin subrayado, y la transición de color se aplica durante 0.3 segundos. La regla a:hover establece que cuando se pasa el cursor sobre un enlace, el color debe cambiar a naranja.

Al guardar estos archivos y abrir el archivo HTML en un navegador, veremos que los estilos CSS se aplican a los elementos HTML. El encabezado h1 será azul, grande y centrado. El párrafo será más pequeño y tendrá una altura de línea mayor. La lista será horizontal, centrada y los elementos de la lista se mostrarán en línea. Los enlaces serán azules y sin subrayado, y cuando se pase el cursor sobre ellos

En resumen, el CSS se utiliza para dar estilo a los elementos HTML de una página web. El archivo CSS se vincula al archivo HTML mediante la etiqueta <link> en la sección <head> del documento HTML. Luego, se crean reglas CSS para cada elemento HTML, especificando las propiedades que se aplicarán, como el color del texto, el tamaño de fuente, la alineación, entre otras. Al abrir el archivo HTML en un navegador web, se aplicarán los estilos CSS a los elementos HTML para dar formato y estilo a la página web

Una explicación de que hace unas propiedades de css

Color: especifica el color del texto. En este caso, h1 y a tendrán un color azul, mientras que al hacer hover sobre a, el color cambiará a naranja.

Font-size: especifica el tamaño de la fuente. El tamaño de la fuente para h1 será de 3 veces el tamaño por defecto, mientras que para p será 1.2 veces el tamaño por defecto.

Text-align: especifica la alineación del texto. El texto de h1 y ul estará centrado.

Line-height: especifica la altura de línea. En este caso, la altura de línea para p será 1.5 veces el tamaño de la fuente.

List-style-type: especifica el tipo de marcador de lista. En este caso, se ha establecido como "none" para ul para que no haya marcadores.

Margin: especifica el margen exterior de un elemento. En este caso, se ha establecido como 0 para ul.

Padding: especifica el espacio interior del borde de un elemento. En este caso, se ha establecido como 0 para ul.

Display: especifica el tipo de visualización del elemento. En este caso, li se mostrará como un elemento en línea.

Text-decoration: especifica el estilo de subrayado de un enlace. En este caso, se ha establecido como "none" para que no haya subrayado en los enlaces.

Transition: especifica la transición de una propiedad. En este caso, la propiedad de transición es el color, lo que significa que al hacer hover sobre a, el cambio de color se realizará con una transición suave durante 0.3 segundos.

Background-color: especifica el color de fondo de un elemento.

Background-image: especifica la imagen de fondo de un elemento.

Border: especifica el borde de un elemento, incluyendo su estilo, ancho y color.

Border-radius: especifica el radio de las esquinas del borde de un elemento, lo que permite crear esquinas redondeadas.

Box-shadow: especifica una sombra para un elemento.

Text-shadow: especifica una sombra para el texto de un elemento.

Font-family: especifica la fuente que se utilizará para el texto de un elemento.

Font-weight: especifica el grosor de la fuente de un elemento.

Font-style: especifica el estilo de la fuente de un elemento, como cursiva o normal.

Letter-spacing: especifica el espacio entre letras de un texto.

Word-spacing: especifica el espacio entre palabras de un texto.

Text-transform: especifica la transformación de un texto, como mayúsculas o minúsculas.

Text-overflow: especifica el comportamiento del texto que sobresale de un elemento.

Overflow: especifica el comportamiento de un elemento que tiene un contenido que sobresale de su área visible.

Opacity: especifica la opacidad de un elemento.

Transition: especifica la transición de una propiedad, como el color o el tamaño.

z-index: especifica el orden de apilamiento de un elemento en relación a otros elementos.

Position: especifica la posición de un elemento en relación a su contenedor.

Float: especifica la posición de un elemento en relación a otros elementos en su contenedor.

Clear: especifica si un elemento debe borrar elementos flotantes a su izquierda, derecha o ambos lados.

En resumen, cada propiedad del CSS se utiliza para dar estilo a los elementos HTML de una página web y se puede especificar para cada tipo de elemento o para un elemento en particular utilizando su selector correspondiente.

Como cambiar el fondo de pantalla por una imagen

Para cambiar el fondo de pantalla de una página web por una imagen, puedes utilizar la propiedad CSS background-image.

Aquí te muestro cómo hacerlo:

Primero, agrega la imagen que deseas utilizar como fondo en la carpeta de tu proyecto web.

Luego, en tu archivo CSS, agrega la siguiente regla para el elemento al que deseas aplicar la imagen de fondo:

```
Body {  
    Background-image: url('ruta/a/la/imagen.jpg');  
}
```

En este ejemplo, estamos aplicando la imagen de fondo a todo el cuerpo (body) de la página web.

Asegúrate de proporcionar la ruta correcta a la imagen en la propiedad url(). Puedes utilizar una ruta relativa si la imagen está en la misma carpeta que el archivo CSS, o una ruta absoluta si se encuentra en una carpeta diferente.

Guarda el archivo CSS y actualiza tu
página web para ver el cambio.

Con estos pasos, deberías ser capaz de cambiar el fondo de pantalla de una página web por una imagen. Recuerda que también puedes utilizar otras propiedades CSS, como `background-size` o `background-repeat`, para ajustar el tamaño y la repetición de la imagen de fondo según tus necesidades