

Conceptos básicos de la escritura CSS

El HTML se encarga de definir la estructura del contenido en una página web, mientras que CSS (Cascading Style Sheets) es responsable del diseño, la apariencia y la disposición de ese contenido. Hasta ahora, hemos aprendido la estructura básica de HTML. Ahora, es momento de explorar cómo aplicar estilos a tu página web utilizando CSS.

Observa el siguiente ejemplo:

HTML



Estructura

CSS



Apariencia

En la imagen se muestra que HTML es el esqueleto del personaje, es decir, su estructura, mientras que CSS da el estilo a esta estructura, por lo que nos sirve para agregar aspectos visuales a la página como colores, fuentes, tipografías, etc



CSS, sigla de "Cascading Style Sheets" (Hojas de Estilo en Cascada).

Existen tres maneras principales de aplicar CSS a un documento HTML:

1- Estilos en línea

Consiste en escribir estilos directamente dentro de las etiquetas HTML utilizando el atributo

style. Es útil para aplicar estilos rápidos y específicos a elementos individuales.

```
1 <p style="color: red;">Este texto será rojo.</p>
```

2- Estilos en la Etiqueta **<style>**

- **Descripción:** Consiste en definir los estilos dentro de una etiqueta **<style>** ubicada en la sección **<head>** del documento HTML. Es útil para aplicar estilos a múltiples elementos dentro de un mismo documento.

```
<head>
  <style>
    p {
      color: blue;
      font-size: 16px;
    }
  </style>
</head>
```

3- Cargar un Archivo CSS Externo

- **Descripción:** Consiste en crear un archivo separado con la extensión **.css** y enlazarlo a tu documento HTML. Este método es ideal para proyectos más grandes, ya que permite separar el contenido del estilo.
- **Ejemplo de Código:** (para referencia futura)

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

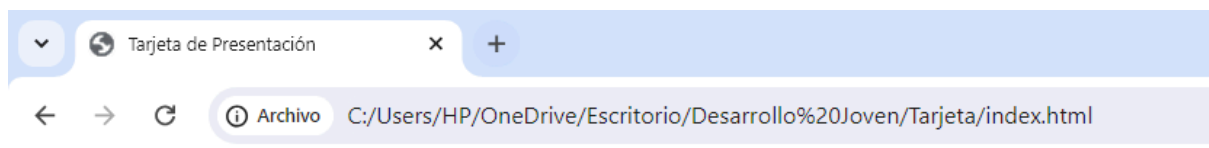
Aplicación de Estilos en Línea

Para empezar con CSS en tu tarjeta de presentación, puedes aplicar estilos en línea para cambiar el color del texto directamente dentro del código HTML. Este método es sencillo y útil para cambios rápidos.

Ejemplo de Aplicación:

Supongamos que quieres cambiar el color del título "Mi Tarjeta de Presentación" a rojo. Aquí está cómo hacerlo utilizando estilos en línea:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta de Presentación</title>
7 </head>
8 <body>
9   <h1 style="color: red;">Mi Tarjeta de Presentación</h1>
10  <p>Nombre: Juan Pérez</p>
11  <p>Profesión: Desarrollador Full Stack</p>
12  <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y
13  atractivos.</p>
14 </body>
15 </html>
```



Mi Tarjeta de Presentación

Nombre: Juan Pérez

Profesión: Desarrollador Full Stack

Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.

[vista del navegador].

En este ejemplo, el atributo `style="color: red;"` cambia el color del texto del título a rojo. Puedes utilizar esta técnica para aplicar estilos rápidos y específicos a otros elementos en tu tarjeta de presentación.

Este atributo de estilo es una forma de escribir CSS directamente en los elementos HTML, y esta forma de escribir se llama estilo en línea.

Las reglas de escritura son las siguientes.

```
<p style="color: red">
```

Como se muestra arriba, una propiedad y un valor se utilizan para especificar lo que se va a hacer.

En este caso significa "hacer que el color del texto (propiedad llamada color) sea rojo (valor llamado red) ". Es decir.

Aquí tenemos. De este modo, hay muchas propiedades y valores que se pueden utilizar para establecer el color, el tamaño y la ubicación de visualización, etc.

¡Vamos a aprender mucho a partir de ahora!

La propiedad [color](#) es una propiedad que permite establecer el color del texto.

El valor de color puede especificarse en notación RGB como se muestra a continuación, además de nombres de colores comunes como negro, rojo y naranja como palabras clave.

- Valor hexadecimal: `color: #ff0001`
- Notación funcional: `color: rgb(255,0,1)`

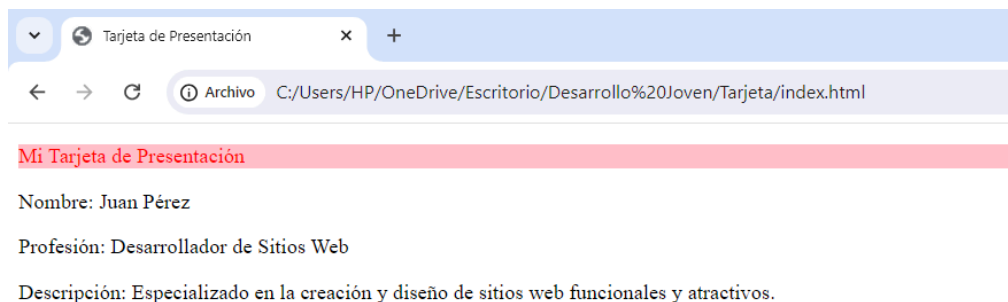
Ajustes de Propiedades en CSS: Cambiar Color del Texto y del Fondo

En esta sección, aprenderás cómo modificar tanto el color del texto como el color de fondo de los elementos en tu tarjeta de presentación. Vamos a utilizar estilos en línea para demostrar estos cambios.

Ejemplo de Código HTML y CSS

En el siguiente ejemplo, hemos cambiado el color del texto a rojo y el color de fondo a rosa para el título de la tarjeta. También hemos incluido un párrafo con texto normal para mostrar cómo queda el diseño.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta de Presentación</title>
7 </head>
8 <body>
9   <div class="tarjeta">
10    <!-- Cambia lo siguiente. -->
11    <p style="color: red; background-color: pink;">Mi Tarjeta de Presentación</p>
12    <p>Nombre: Juan Pérez</p>
13    <p>Profesión: Desarrollador Full Stack</p>
14    <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y
15    atractivos.</p>
16  </div>
17 </body>
</html>
```



[vista del navegador].

Explicación del Código

- Etiqueta `<p>`:
 - `style="color: red; background-color: pink;"`: Aquí estamos utilizando estilos en línea para cambiar el color del texto a rojo (color: red;) y el color de fondo a rosa (background-color: pink;).

Personalización Adicional

- Cambiar el Color del Texto:
 - Prueba con diferentes colores, como blue, green, black, etc.
 - Ejemplo: `style="color: blue;"`
- Cambiar el Color de Fondo:
 - Experimenta con otros colores de fondo, como lightgray, yellow, lightblue, etc.
 - Ejemplo: `style="background-color: lightgray;"`

Este ejercicio te ayudará a comprender cómo los colores afectan la apariencia de tu tarjeta y a experimentar con diferentes combinaciones para lograr el diseño deseado.

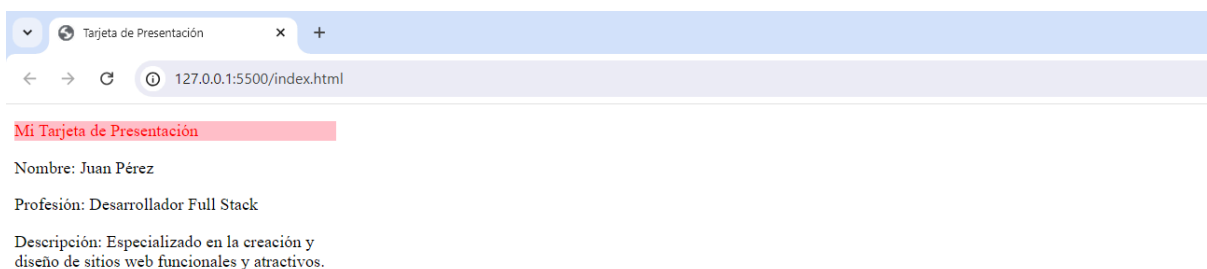
Añadir Estilo a la Etiqueta `<div>`: Fijar Ancho

En esta sección, aprenderás a fijar el ancho de un contenedor `<div>`, lo cual afecta el tamaño de los elementos hijos dentro de él.

Ejemplo de Código HTML y CSS

En el siguiente ejemplo, hemos añadido el estilo `width` a la etiqueta `<div>` para establecer un ancho fijo. Esto afectará a los elementos `<p>` dentro de ese contenedor.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta de Presentación</title>
7 </head>
8 <body>
9   <div style="width: 300px;">
10     <!-- Contenido de la tarjeta -->
11     <p style="color: red; background-color: pink;">Mi Tarjeta de Presentación</p>
12     <p>Nombre: Juan Pérez</p>
13     <p>Profesión: Desarrollador Full Stack</p>
14     <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
15   </div>
16 </body>
17 </html>
```



[vista del navegador].

Explicación

- **<div style="width: 300px;">**: Este código establece el ancho del contenedor <div> a 300 píxeles. Todos los elementos <p> dentro de este contenedor se ajustan a este ancho.
- **style="color: red; background-color: pink;"**: Aquí se aplican estilos en línea al texto para cambiar su color a rojo y el fondo a rosa.

Resultado

Al visualizar este código en un navegador, verás que la tarjeta de presentación está contenida dentro de un área de 300 píxeles de ancho. Esto demuestra cómo el ajuste del ancho del contenedor afecta la disposición y la presentación de los elementos que contiene.

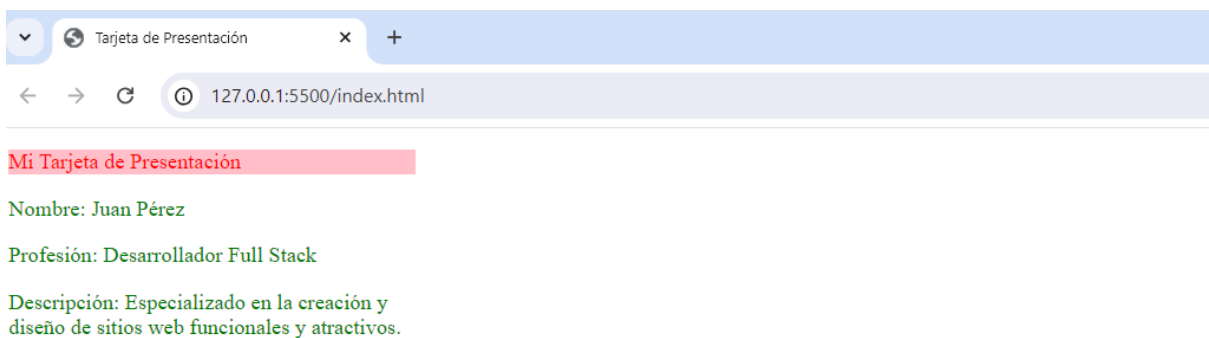
Herencia de Estilo: Añadir Color al Contenedor <div>

En esta sección, aprenderás cómo la herencia de estilo funciona en HTML y CSS, aplicando un color al contenedor <div> que afectará a los elementos dentro de él.

Ejemplo de Código HTML

En el siguiente ejemplo, hemos añadido el estilo `color: verde;` al contenedor <div>. Este estilo se aplicará a los elementos <p> dentro de la tarjeta de presentación:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta de Presentación</title>
7 </head>
8 <body>
9   <div style="width: 300px; color: green;">
10     <!-- Contenido de la tarjeta -->
11     <p style="color: red; background-color: pink;">Mi Tarjeta de Presentación</p>
12     <p>Nombre: Juan Pérez</p>
13     <p>Profesión: Desarrollador Full Stack</p>
14     <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
15   </div>
16 </body>
17 </html>
```



[vista del navegador].

Explicación

- `<div style="width: 300px; color: green;">`: Aquí se ha añadido `color: green;` al estilo del contenedor `<div>`. Este color verde se aplicará a todos los elementos de texto `<p>` dentro del `<div>`, a menos que se sobrescriba con otro estilo específico.
- `style="background-color: pink;"`: Aunque el color de texto es heredado del `<div>`, el color de fondo de este párrafo se mantiene rosa debido a la propiedad de estilo en línea aplicada directamente.

Resultado

Al visualizar este código en un navegador, notarás que el texto de la tarjeta de presentación es verde, excepto donde se haya especificado un color diferente. Este ejemplo demuestra cómo los estilos aplicados a un contenedor pueden heredarse por sus elementos hijos, simplificando la aplicación de estilos en HTML y CSS.

Especificar Estilos mediante la Etiqueta `<style>`

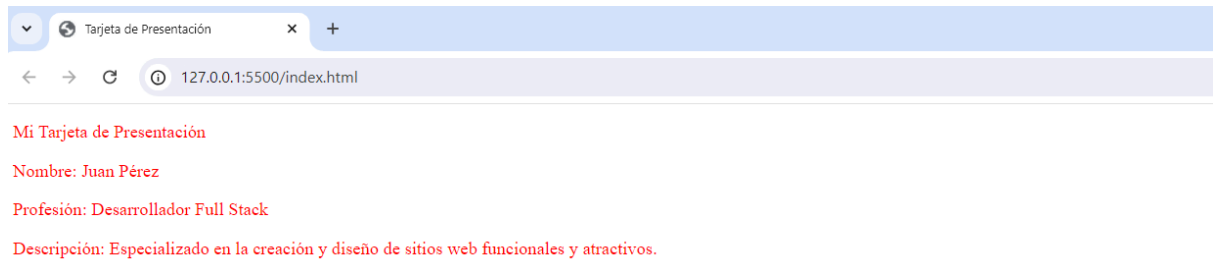
El uso del selector de elementos en CSS permite aplicar estilos a todos los elementos de un tipo específico en un documento HTML. En este caso, vamos a ver cómo aplicar estilos a todos los elementos `<p>` usando la etiqueta `<style>` dentro de la sección `<head>` del documento.

Descripción del Proceso

Hasta ahora, hemos visto cómo aplicar estilos en línea directamente dentro de las etiquetas HTML usando el atributo `style`. Sin embargo, un método más limpio y eficiente es definir todos los estilos de una vez dentro de una etiqueta `<style>` en la sección `<head>` del documento.

Ejemplo de Código HTML con CSS en `<style>`

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Tarjeta de Presentación</title>
6      <!-- Definición de estilos en la etiqueta <style> -->
7      <style>
8          /* Todos los elementos <p> tendrán texto de color rojo */
9          p {
10             color: red;
11         }
12     </style>
13 </head>
14 <body>
15     <p>Mi Tarjeta de Presentación</p>
16     <div>
17         <p>Nombre: Juan Pérez</p>
18         <p>Profesión: Desarrollador Full Stack</p>
19         <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
20     </div>
21 </body>
22 </html>
```



[vista del navegador].

Explicación:

- **Selector p:** El selector **p** en el CSS se utiliza para aplicar estilos a todos los elementos `<p>` en el documento. En este caso, hemos decidido cambiar el color del texto de todos los párrafos a rojo.
- **Propiedad color:** La propiedad **color** en CSS especifica el color del texto. En este ejemplo, hemos configurado el color del texto en rojo (`red`).

Resultado:

Cuando este código se visualiza en un navegador, todos los párrafos definidos con la etiqueta `<p>` tendrán su texto en color rojo. Esto incluye el título "Mi Tarjeta de Presentación" y los detalles de la tarjeta como el nombre, la profesión y la descripción.

Beneficios de Usar Selectores de Elemento en `<style>`:

- **Organización:** Mantiene el HTML más limpio y organizado, separando la estructura del contenido (HTML) del diseño (CSS).
- **Reutilización:** Al aplicar estilos a través de selectores de elementos, se evita la repetición de código y se asegura que los cambios se aplican de manera consistente en todo el documento.

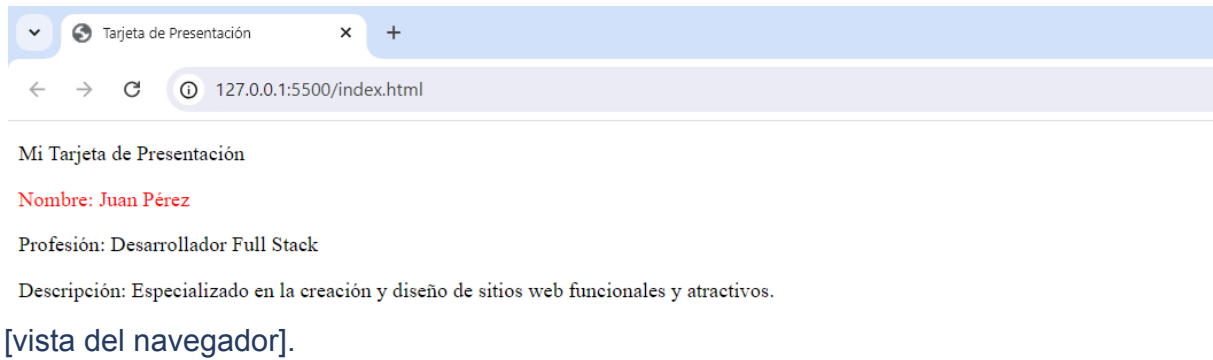
Designación mediante la Clase (class)

Cuando necesitas aplicar un estilo a un elemento específico dentro de un documento HTML, los selectores de elementos por sí solos pueden no ser suficientes. En estos casos, puedes utilizar los selectores de clase (class) para aplicar estilos únicos a ciertos elementos sin afectar a otros.

Ejemplo de Código HTML y CSS

En este ejemplo, usaremos la clase sub para aplicar un estilo específico a uno de los párrafos (<p>) dentro de la tarjeta de presentación.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tarjeta de Presentación</title>
6   <!-- Definición de estilos en la etiqueta <style> -->
7   <style>
8     /* Estilo aplicado a cualquier elemento con la clase "sub" */
9     .sub {
10       color: red;
11     }
12   </style>
13 </head>
14 <body>
15   <p>Mi Tarjeta de Presentación</p>
16   <div>
17     <!-- Aplicación del estilo mediante la clase "sub" -->
18     <p class="sub">Nombre: Juan Pérez</p>
19     <p>Profesión: Desarrollador Full Stack</p>
20     <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
21   </div>
22 </body>
23 </html>
```



Cuando visualices este código en un navegador, sólo el párrafo con la clase `sub`, que contiene el nombre "Juan Pérez", tendrá el texto en color rojo. Los otros párrafos dentro de la `<div>` no se verán afectados por este estilo.

Explicación:

- **Clase `.sub`:** La clase `sub` se define en la sección `<style>` y se aplica a elementos específicos mediante el atributo `class`. En este caso, se aplica al primer párrafo dentro de la `<div>`, que contiene el nombre "Juan Pérez".
- **Selector `.sub`:** En CSS, los selectores de clase se definen con un punto (.) seguido del nombre de la clase. En este ejemplo, `.sub` cambia el color del texto a rojo solo para el párrafo que tiene la clase `sub`.

Ejemplo de Uso de Clases Significativas

Las clases son extremadamente útiles cuando necesitas aplicar el mismo estilo a varios elementos sin tener que repetir las reglas de estilo en cada uno. Esto resulta particularmente conveniente cuando se definen múltiples propiedades de estilo.

A continuación, te muestro un ejemplo donde se define una clase `.sub` con varias propiedades:

```
1  .sub {  
2    color: red;  
3    font-size: 20px;  
4    font-weight: bold;  
5  }
```

Aplicación en el Proyecto de Tarjeta de Presentación

Supongamos que queremos aplicar esta clase `.sub` a uno de los párrafos de la tarjeta de presentación para destacar el nombre del desarrollador:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tarjeta de Presentación</title>
6   <!-- Definición de estilos en la etiqueta <style> -->
7   <style>
8     /* Estilo aplicado a cualquier elemento con la clase "sub" */
9     .sub {
10       color: red;
11       font-size: 20px;
12       font-weight: bold;
13     }
14   </style>
15 </head>
16 <body>
17   <p>Mi Tarjeta de Presentación</p>
18   <div>
19     <!-- Aplicación del estilo mediante la clase "sub" -->
20     <p class="sub">Nombre: Juan Pérez</p>
21     <p>Profesión: Desarrollador Full Stack</p>
22     <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
23   </div>
24 </body>
25 </html>
```



Mi Tarjeta de Presentación

Nombre: Juan Pérez

Profesión: Desarrollador Full Stack

Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.

[vista del navegador].



Importante:

El nombre de una clase (class) puede ser cualquier palabra que elijas, pero es crucial que el nombre de la clase que asignas a una etiqueta coincida exactamente con el nombre que utilizas en el selector CSS. Sin embargo, debes evitar utilizar nombres sin sentido como abc o 123. En lugar de eso, utiliza nombres significativos que sean fáciles de recordar y que otros desarrolladores puedan entender rápidamente cuando revisen tu código.

Ventajas de Utilizar Clases en CSS

1. **Eficiencia:** Evita la repetición de reglas de estilo, lo que reduce el tamaño del archivo CSS y facilita su mantenimiento.
2. **Flexibilidad:** Puedes aplicar la misma clase a múltiples elementos, asegurando un diseño consistente en todo el documento.
3. **Claridad:** Al utilizar nombres significativos para las clases, el código es más comprensible y fácil de mantener.

Resumen

Al definir una clase en CSS, no solo facilitan la aplicación de estilos consistentes en el documento, sino que también hace que el código sea más limpio y manejable. Asegúrate de elegir nombres de clase que tengan sentido en el contexto de su uso, lo que te ayudará a ti y a otros desarrolladores a entender rápidamente el propósito de cada estilo.

Designación mediante el id

El atributo id se utiliza para aplicar estilos a un elemento específico en un documento HTML. A diferencia de class, un id debe ser único dentro de un documento, lo que significa que sólo puedes usar un id una vez por página.

Veamos un ejemplo de cómo usar el atributo id en un documento HTML.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tarjeta de Presentación</title>
7   <!-- Cambia lo siguiente. -->
8   <style>
9     #titulo {
10       color: red;
11       font-size: 24px;
12       font-weight: bold;
13     }
14   </style>
15 </head>
16 <body>
17   <!-- Cambia lo siguiente. -->
18   <p id="titulo">Mi Tarjeta de Presentación</p>
19   <div>
20     <p>Nombre: Juan Pérez</p>
21     <p>Profesión: Desarrollador de Sitios Web</p>
22     <p>Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.</p>
23   </div>
24 </body>
25 </html>
```



Mi Tarjeta de Presentación

Nombre: Juan Pérez

Profesión: Desarrollador de Sitios Web

Descripción: Especializado en la creación y diseño de sitios web funcionales y atractivos.

[vista del navegador].

Explicación

1. **Atributo id:** En el ejemplo, se utilizó el atributo id para el primer párrafo (`<p id="titulo">Mi Tarjeta de Presentación</p>`). Esto permite aplicar estilos específicos a este elemento, cuando se utiliza un id como selector de esta forma, se describe con un prefijo # (numeral).
2. **Selector de id:** En la sección de `<style>`, se define un estilo para el id usando `#titulo`, que es el identificador único para ese elemento.
3. **Estilos aplicados:** Los estilos definidos en `#titulo` hacen que el texto sea rojo, con un tamaño de fuente de 24px y en negrita.

Consideraciones Importantes

- **Unicidad del id:** Recuerda que un id debe ser único dentro de un documento HTML, lo que significa que no puedes reutilizarlo en otros elementos en la misma página.
- **Nombres significativos:** Al igual que con class, es importante dar nombres significativos a los id para que sean fáciles de recordar y entender.

Este ejemplo te muestra cómo utilizar id para personalizar elementos individuales en tu página web, como la tarjeta de presentación que estamos creando.



Resumen de Aprendizaje

Hemos recorrido paso a paso el proceso de creación de una tarjeta de presentación utilizando HTML y CSS. A continuación, un resumen de los puntos clave:

1. Estructura Básica de HTML

- **Declaración DOCTYPE:** Define el tipo de documento y es el primer paso en la construcción de una página web.
- **Etiquetas HTML:** Exploramos las etiquetas fundamentales como `<html>`, `<head>`, `<body>`, `<p>` y su importancia en la estructura del contenido.
- **Contenido de la Tarjeta:** Utilizamos etiquetas `<p>` para estructurar la información de la tarjeta, como el nombre, profesión y descripción.

2. Introducción a CSS

- **Tres formas de aplicar CSS:**
 - **Estilos en línea:** Aplicación de estilos directamente dentro de las etiquetas HTML mediante el atributo `style`.
 - **Etiqueta `<style>`:** Inserción de reglas CSS en la sección `<head>` del documento para aplicar estilos a múltiples elementos.
 - **Archivo CSS externo:** Mencionamos la posibilidad de enlazar un archivo CSS externo, aunque nos enfocamos en los estilos en línea y en la etiqueta `<style>` para mayor simplicidad.

3. Ajustes de Estilos

- **Múltiples Propiedades:** Aprendimos a modificar varias propiedades como el color de fondo y el tamaño de los elementos utilizando CSS.
- **Fijar Ancho:** Vimos cómo fijar la anchura de un contenedor `<div>` y cómo esto afecta a los elementos hijos dentro de él.
- **Herencia de Estilos:** Explicamos cómo los estilos aplicados a un contenedor pueden ser heredados por sus elementos hijos.

4. Uso de Clases y IDs para Estilizar Elementos

- **Clases (class):** Utilizamos clases para aplicar estilos a múltiples elementos de manera eficiente y consistente.
- **IDs (id):** Exploramos cómo los IDs permiten aplicar estilos a elementos únicos, lo que resulta útil para personalizar componentes específicos en una página.

5. Aplicación Práctica: Creación de la Tarjeta de Presentación

- **Ejemplos Prácticos:** A través de ejemplos de código, construimos una tarjeta de presentación con los conocimientos adquiridos, aplicando estilos básicos.
- **Personalización:** Incorporamos el uso de clases e IDs para lograr un diseño más flexible y modular.

Conclusión

Continúa practicando con diferentes combinaciones de HTML y CSS para reforzar lo aprendido y experimentar con nuevos estilos y estructuras.

¡El desarrollo web es un campo lleno de creatividad y oportunidades!

