1. CSS

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para describir la apariencia y el formato de un documento HTML. Se utiliza para definir la presentación visual de un sitio web, incluyendo aspectos como el diseño, los colores, las fuentes, los tamaños y la disposición de los elementos en la página.

La separación entre el contenido y la presentación es una de las principales ventajas de CSS. Permite mantener el contenido y la estructura del documento HTML separados de su estilo visual. Esto significa que puedes cambiar la apariencia de un sitio web aplicando diferentes hojas de estilo CSS sin tener que modificar el código HTML subyacente.

Con CSS, puedes aplicar reglas de estilo a elementos individuales, como párrafos, encabezados, enlaces o imágenes, utilizando selectores. Además, puedes agrupar elementos y aplicar estilos comunes a través de clases y ID. También puedes controlar la disposición de los elementos utilizando propiedades de diseño como el posicionamiento, el tamaño y el espaciado.

CSS se utiliza ampliamente en el desarrollo web para crear sitios web visualmente atractivos y con estilo. Permite personalizar la apariencia de una página web de acuerdo con los requisitos del diseño, mejorar la usabilidad y la accesibilidad, y proporcionar una experiencia de usuario más agradable.

Además de ser utilizado en la construcción de sitios web, CSS también se puede aplicar a otros tipos de documentos electrónicos, como documentos XML o aplicaciones de escritorio basadas en lenguajes de marcado.

2. VINCULACIÓN

2.1. EXTERNO

Para vincular un archivo CSS externo a tu documento HTML, debes utilizar el elemento <link> en la sección <head> de tu archivo HTML.

Crea un archivo CSS separado: Primero, crea un archivo CSS separado que contenga tus estilos. Guarda el archivo con una extensión ".css" (por ejemplo, "estilos.css"). En este archivo, puedes definir las reglas de estilo que deseas aplicar a tu página HTML.

Vincula el archivo CSS en tu HTML: Abre tu archivo HTML y agrega el siguiente código en la sección <head>:

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="ruta-del-archivo.css">
</head>
```

Asegurate de reemplazar "ruta-del-archivo.css" con la ruta relativa o absoluta de tu archivo CSS. Si ambos archivos (HTML y CSS) están en la misma carpeta, simplemente puedes especificar el nombre del archivo CSS.

Aplica los estilos en tu HTML: Una vez vinculado el archivo CSS, puedes aplicar los estilos a los elementos HTML utilizando selectores CSS en tu archivo CSS. Por ejemplo, si quieres aplicar un estilo de color rojo a todos los párrafos (<p>), puedes agregar la siguiente regla en tu archivo CSS:

```
p {
  color: red;
}
```

Cuando abras tu archivo HTML en un navegador web, los estilos definidos en el archivo CSS vinculado se aplicarán a los elementos HTML seleccionados.

2.2. INTERNO

La vinculación interna se utiliza para crear enlaces dentro del mismo documento HTML, para dirigir al usuario a diferentes secciones de la página. Se realiza utilizando el atributo href junto con el identificador (ID) de la sección de destino.

Por ejemplo, si tienes una sección con el ID "seccion1" en tu página, puedes crear un enlace interno utilizando el siguiente código:

```
<a href="#seccion1">Ir a la Sección 1</a>
```

Cuando se hace clic en este enlace, la página se desplazará automáticamente hacia la sección con el ID "seccion1". Puedes aplicar este enfoque a cualquier sección de tu página.

2.3. EN LINEA

La vinculación en línea se utiliza para aplicar estilos o atributos específicos directamente en una etiqueta HTML. No requiere un archivo externo ni una vinculación adicional. Se realiza utilizando el atributo style en la etiqueta.

Por ejemplo, si deseas aplicar un estilo de color rojo a un párrafo específico, puedes utilizar el siguiente código:

```
<p style="color: red;">Este es un párrafo en rojo</p>
```

En este caso, el estilo se define directamente en la etiqueta <p> mediante el atributo style.

Es importante tener en cuenta que, aunque la vinculación en línea puede ser conveniente en algunos casos simples, la vinculación externa y la vinculación interna con archivos CSS separados son generalmente preferibles para proyectos más grandes y complejos, ya que proporcionan una mejor separación de preocupaciones y facilitan el mantenimiento y la reutilización del código.

3. SELECTORES

3.1. BÁSICOS

3.1.1. ELEMENTO

El selector de elemento selecciona todos los elementos HTML del tipo especificado. Se utiliza el nombre del elemento como selector. Por ejemplo, para seleccionar todos los párrafos (<p>), se utiliza el siguiente selector:

```
p {
  /* Estilos aplicados a todos los párrafos */
  color: blue;
}
```

3.1.2. CLASE

El selector de clase selecciona todos los elementos que tienen un atributo de clase específico. Se utiliza un punto seguido del nombre de la clase como selector. Por ejemplo, para seleccionar todos los elementos con la clase "destacado", se utiliza el siguiente selector:

```
.destacado {
  /* Estilos aplicados a todos los elementos con la clase "destacado" */
  font-weight: bold;
}
```

En el HTML, se aplicaría la clase de la siguiente manera:

```
<p class="destacado">Este párrafo está destacado</p>
```

3.1.3. ID

El selector de ID selecciona un elemento específico según su atributo de ID. Se utiliza el símbolo de numeral (#) seguido del nombre del ID como selector. Por ejemplo, para seleccionar un elemento con el ID "encabezado", se utiliza el siguiente selector:

```
#encabezado {
  /* Estilos aplicados al elemento con el ID "encabezado" */
  background-color: gray;
}
```

En el HTML, se aplica el ID de la siguiente manera:

```
<h1 id="encabezado">Este es el encabezado</h1>
```

3.2. ATRIBUTOS

Los selectores de atributos en CSS permiten seleccionar elementos HTML en función de sus atributos y valores.

El selector de atributo selecciona elementos que tienen un atributo específico, independientemente de su valor. Se utiliza el siguiente formato: [atributo]. Por ejemplo, para seleccionar todos los elementos que tienen el atributo target, puedes usar el siguiente selector:

```
[target] {
  /* Estilos aplicados a elementos con el atributo "target" */
  color: red;
}
```

3.2.1. Selector de atributo con valor exacto:

El selector de atributo con valor exacto selecciona elementos que tienen un atributo con un valor específico. Se utiliza el siguiente formato: [atributo="valor"]. Por ejemplo, para seleccionar todos los enlaces (<a>) que tienen el atributo target con el valor _blank, puedes usar el siguiente selector:

```
[a target="_blank"] {
  /* Estilos aplicados a enlaces con el atributo "target" igual a "_blank" */
  text-decoration: underline;
```

```
}
```

3.2.2. Selector de atributo con valor que comienza por:

El selector de atributo con valor que comienza por selecciona elementos que tienen un atributo con un valor que comienza con una cadena específica. Se utiliza el siguiente formato: [atributo^="valor"]. Por ejemplo, para seleccionar todos los enlaces (<a>) que tienen el atributo href con un valor que comienza con "https://", puedes usar el siguiente selector:

```
[a href^="https://"] {
  /* Estilos aplicados a enlaces con el atributo "href" que comienza con "https://" */
  color: blue;
}
```

Estos son solo algunos ejemplos de los selectores de atributos en CSS. También existen otros selectores de atributos, como el selector de atributo con valor que termina por ($=), el selector de atributo con valor que contiene (*=), el selector de atributo con valor que no es igual (!=), entre otros. Estos selectores te permiten seleccionar elementos de manera más precisa y flexible en función de sus atributos y valores.

3.3. COMBINADOS

Los selectores combinados en CSS permiten seleccionar elementos basados en su relación con otros elementos o su posición dentro de la estructura del documento HTML.

3.3.1. Selector descendiente

El selector descendiente selecciona elementos secundarios que están anidados dentro de un elemento principal. Se utiliza el espacio para separar los elementos. Por ejemplo, para seleccionar todos los párrafos (<p>) que están dentro de un elemento , puedes usar el siguiente selector:

```
div p {
  /* Estilos aplicados a los párrafos dentro de un div */
  color: blue;
}
```

3.3.2. Selector hijo directo:

El selector hijo directo selecciona elementos que son hijos directos de un elemento principal. Se utiliza el símbolo mayor que (>). Por ejemplo, para

seleccionar todos los elementos <li> que son hijos directos de una lista no ordenada (<ul>), puedes usar el siguiente selector:

```
ul > li {
  /* Estilos aplicados a los elementos <li> hijos directos de un <ul> */
  font-weight: bold;
}
```

3.3.3. Selector de hermanos adyacentes:

El selector de hermanos adyacentes selecciona un elemento que es el siguiente hermano inmediato de otro elemento. Se utiliza el símbolo más (+). Por ejemplo, para seleccionar el primer párrafo (<p>) que sigue inmediatamente a un encabezado (<h1>), puedes usar el siguiente selector:

```
h1 + p {
  /* Estilos aplicados al primer <p> después de un <h1> */
  font-style: italic;
}
```

Estos son solo algunos ejemplos de los selectores combinados en CSS. También existen otros selectores combinados, como el selector de hermanos generales (~), el selector de pseudoelemento (::before, ::after), entre otros. Estos selectores te permiten seleccionar elementos de manera más específica y controlada, tomando en cuenta su relación con otros elementos dentro de la estructura del documento HTML.

3.4. ANIDADOS

Los selectores anidados en CSS te permiten seleccionar elementos específicos dentro de la estructura anidada de otros elementos. Esto se logra mediante la combinación de selectores para apuntar a elementos secundarios o descendientes dentro de un elemento principal. Los selectores anidados se utilizan comúnmente en lenguajes de preprocesadores CSS como Sass o Less.

Supongamos que tienes el siguiente código HTML:

```
<div class="contenedor">
  <h1>Título</h1>
  <p>Párrafo 1</p>
  <p>Párrafo 2</p>
```

```
</div>
```

En Sass, puedes utilizar los selectores anidados para aplicar estilos específicos a los elementos dentro del contenedor. Por ejemplo:

```
.contenedor {
  background-color: gray;

  h1 {
    color: blue;
  }

  p {
    color: red;
  }
}
```

En este ejemplo, el selector .contenedor establece un fondo gris para el contenedor. Dentro del contenedor, el selector h1 establece el color del título en azul, y el selector p establece el color de los párrafos en rojo.

```
.contenedor {
  background-color: gray;
}

.contenedor h1 {
  color: blue;
}

.contenedor p {
  color: red;
}
```

Como resultado, el título y los párrafos dentro del elemento con la clase .contenedor heredarán los estilos especificados en los selectores anidados.

Los selectores anidados en preprocesadores CSS proporcionan una forma más estructurada y legible de aplicar estilos a elementos específicos dentro de una jerarquía de elementos anidados. Esto ayuda a mantener el código organizado y facilita los cambios y ajustes en los estilos.

3.5. MULTIPLES

Un selector múltiple en CSS te permite seleccionar varios elementos y aplicar estilos a todos ellos con una sola regla de estilo. Esto te permite ahorrar código y facilita la aplicación de estilos similares a múltiples elementos. Puedes combinar selectores individuales separados por comas para crear un selector múltiple.

Supongamos que tienes los siguientes elementos HTML:

```
<p class="parrafo">Texto 1</p>

<p class="parrafo">Texto 2</p>

<p class="parrafo">Texto 3</p>

<h1 class="encabezado">Título</h1>
```

En CSS, si deseas aplicar estilos a todos los párrafos (<p>) y al encabezado (<h1>) al mismo tiempo, puedes utilizar un selector múltiple separando los selectores individuales por comas. Por ejemplo:

```
.parrafo, .encabezado {
  color: blue;
  font-size: 16px;
}
```

En este ejemplo, el selector múltiple .parrafo, .encabezado selecciona tanto los elementos con la clase .parrafo como los elementos con la clase .encabezado. Los estilos especificados en la regla se aplicarán a todos estos elementos seleccionados. En este caso, los elementos <p> con la clase .parrafo y el elemento <h1> con la clase .encabezado tendrán el color azul y un tamaño de fuente de 16 píxeles.

El uso de selectores múltiples en CSS te permite aplicar estilos a varios elementos de manera eficiente y concisa. Puedes combinar selectores individuales y agrupar elementos similares para aplicar estilos de forma coherente en tu documento HTML.

4. TEXTO

En CSS, existen varias propiedades que se utilizan para modificar la apariencia y el estilo del texto en un documento HTML.

4.1. COLOR

Esta propiedad establece el color del texto. Puedes especificar un color utilizando un nombre de color, un valor hexadecimal o un valor RGB.

```
p {
  color: red;
}
```

4.2. font-family:

Esta propiedad define la fuente o el tipo de letra a utilizar para el texto. Puedes especificar una lista de nombres de fuente, en orden de preferencia, separados por comas.

```
p {
  font-family: Arial, sans-serif;
}
```

4.3. font-size

Esta propiedad establece el tamaño de la fuente del texto. Puedes especificar un tamaño utilizando píxeles, porcentajes, em o rem. Por ejemplo:

```
p {
  font-size: 16px;
}
```

4.4. font-weight

Esta propiedad define el grosor o el peso de la fuente del texto. Puedes utilizar valores como normal, bold, bolder, lighter o números específicos para indicar el peso. Por ejemplo:

```
p {
  font-weight: bold;
}
```

4.5. text-align:

Esta propiedad alinea el texto dentro de su contenedor. Puedes utilizar valores como left, right, center o justify. Por ejemplo:

```
p {
  text-align: center;
}
```

4.6. text-decoration:

Esta propiedad establece una decoración de línea en el texto, como subrayado, tachado o ninguna. Por ejemplo:

```
p {
  text-decoration: underline;
}
```

5. MODELO CAJAS

El modelo de caja en CSS se refiere a cómo se estructura y se representa visualmente cada elemento HTML en una página web. Cada elemento en HTML se considera una "caja" rectangular que tiene contenido, relleno, borde y margen.

El modelo de caja en CSS se compone de cuatro componentes principales:

- Contenido (Content): Es el área donde se muestra el contenido real del elemento, como texto, imágenes, u otros elementos HTML anidados.
- Relleno (Padding): Es el espacio transparente entre el contenido y el borde. Se utiliza para agregar espacio adicional alrededor del contenido dentro de la caja. El tamaño del relleno se puede especificar utilizando la propiedad padding en CSS.
- Borde (Border): Es una línea que rodea el contenido y el relleno. El borde puede tener diferentes estilos, grosores y colores. Se puede controlar utilizando la propiedad border en CSS.
- Margen (Margin): Es el espacio transparente que rodea el borde. Se utiliza para crear espacio entre elementos adyacentes. El tamaño del margen se puede especificar utilizando la propiedad margin en CSS.

El tamaño total de una caja en el modelo de caja es la suma del contenido, el relleno, el borde y el margen. Esto significa que, al especificar dimensiones (como ancho y alto) de un elemento, debes tener en cuenta estos componentes.

Además de estas propiedades básicas del modelo de caja, CSS también proporciona formas de controlar el diseño y la posición de las cajas

utilizando propiedades de posicionamiento, como position y float, y propiedades de diseño, como display y box-sizing.

Comprender y aplicar el modelo de caja en CSS es fundamental para controlar y ajustar la apariencia y el diseño de los elementos en una página web.

5.1. BORDER

La propiedad border en CSS se utiliza para establecer el estilo, el grosor y el color del borde de un elemento. Puedes aplicarla a cualquier elemento HTML y controlar diferentes aspectos del borde.

```
.mi-elemento {
  border: 2px solid red;
}
```

En este ejemplo, se aplica un borde de 2 píxeles de grosor, sólido y de color rojo al elemento con la clase .mi-elemento.

La propiedad border se compone de varios subpropiedades que puedes utilizar para controlar aspectos específicos del borde:

- border-width: Establece el grosor del borde. Puedes especificar un valor en píxeles, em, rem o porcentajes.
- border-style: Define el estilo del borde. Puedes utilizar valores como solid (sólido), dashed (discontinuo), dotted (punteado), double (doble línea), entre otros.
- border-color: Establece el color del borde. Puedes utilizar nombres de colores, valores hexadecimales o valores RGB.
- border-radius: Define el radio de los bordes, creando esquinas redondeadas. Puedes especificar un valor en píxeles o porcentaje para todos los bordes o para cada borde individualmente.

Por ejemplo, puedes aplicar diferentes propiedades para controlar el borde:

```
.mi-elemento {
  border-width: 2px;
  border-style: dashed;
  border-color: blue;
  border-radius: 5px;
```

```
}
```

En este caso, se establece un borde de 2 píxeles de grosor, de estilo discontinuo (dashed), de color azul y con esquinas redondeadas de 5 píxeles al elemento con la clase .mi-elemento.

Recuerda que también puedes especificar propiedades individuales para cada borde (como border-top, border-bottom, border-left, border-right) si deseas aplicar diferentes estilos o colores a bordes específicos del elemento.

5.2. PADDING

La propiedad padding en CSS se utiliza para establecer el espacio entre el contenido de un elemento y su borde. Puedes aplicarla a cualquier elemento HTML y controlar el espaciado interno. El padding aumenta el espacio entre el contenido y el borde del elemento.

```
.mi-elemento {
  padding: 10px;
}
```

En este ejemplo, se aplica un espacio de relleno de 10 píxeles en todos los lados del elemento con la clase .mi-elemento.

La propiedad padding se puede especificar de varias formas:

- padding: Establece el espacio de relleno para todos los lados del elemento.
- padding-top, padding-bottom, padding-left, padding-right: Establecen el espacio de relleno para cada lado individualmente.

Puedes especificar valores en píxeles, em, rem o porcentajes para controlar el tamaño del relleno. Además, también puedes especificar múltiples valores separados por espacios para establecer diferentes valores de relleno para cada lado. Por ejemplo:

```
.mi-elemento {
  padding: 10px 20px 10px 20px;
}
```

En este caso, se establece un espacio de relleno de 10 píxeles en la parte superior e inferior y de 20 píxeles en los lados izquierdo y derecho del elemento con la clase .mi-elemento.

El uso de la propiedad padding te permite controlar el espaciado interno de un elemento y separar el contenido del borde. Puedes ajustar el valor de padding según tus necesidades de diseño para lograr el espaciado deseado.

5.3. MARGIN

La propiedad margin en CSS se utiliza para establecer el espacio alrededor de un elemento, es decir, el espacio entre el borde del elemento y los elementos adyacentes. Puedes aplicarla a cualquier elemento HTML y controlar el espaciado externo. La propiedad margin crea espacio entre elementos adyacentes.

```
.mi-elemento {
  margin: 10px;
}
```

En este ejemplo, se aplica un margen de 10 píxeles en todos los lados del elemento con la clase .mi-elemento.

La propiedad margin se puede especificar de varias formas:

- margin: Establece el margen para todos los lados del elemento.
- margin-top, margin-bottom, margin-left, margin-right: Establecen el margen para cada lado individualmente.

Puedes especificar valores en píxeles, em, rem o porcentajes para controlar el tamaño del margen. Además, también puedes especificar múltiples valores separados por espacios para establecer diferentes valores de margen para cada lado. Por ejemplo:

```
.mi-elemento {
  margin: 10px 20px 10px 20px;
}
```

En este caso, se establece un margen de 10 píxeles en la parte superior e inferior y de 20 píxeles en los lados izquierdo y derecho del elemento con la clase .mi-elemento.

El uso de la propiedad margin te permite controlar el espaciado externo de un elemento y crear separación visual entre elementos adyacentes. Puedes ajustar el valor de margin según tus necesidades de diseño para lograr el espaciado deseado.

5.4. BOX SIZING

Es una propiedad de CSS que determina cómo se calculan las dimensiones totales de una caja (elemento) en el modelo de caja. Establece si el tamaño declarado de ancho y alto de un elemento incluye o no el relleno (padding) y el borde.

Por defecto, el valor de box-sizing es content-box, lo que significa que el ancho y alto declarados de un elemento se aplican solo al contenido, sin tener en cuenta el relleno y el borde adicionales. Esto puede hacer que el tamaño total de la caja sea más grande que el tamaño declarado.

Sin embargo, si se establece box-sizing en border-box, el ancho y alto declarados se aplicarán al tamaño total de la caja, incluyendo el contenido, el relleno y el borde. En este caso, el tamaño total de la caja será igual al tamaño declarado y el contenido se ajustará dentro de esa área.

```
.mi-elemento {
  box-sizing: content-box;
  width: 200px;
  padding: 20px;
  border: 2px solid black;
}

.mi-otro-elemento {
  box-sizing: border-box;
  width: 200px;
  padding: 20px;
  border: 2px solid black;
}
```

En este ejemplo, .mi-elemento y .mi-otro-elemento tienen el mismo ancho declarado de 200 píxeles, pero con diferentes valores de box-sizing. En el primer caso, el ancho total de la caja incluirá el tamaño del contenido, el relleno y el borde, lo que resultará en una caja más ancha. En el segundo caso, el ancho total de la caja será de 200 píxeles, y el contenido se ajustará dentro de ese espacio.

El uso de box-sizing te permite controlar cómo se calculan y se comportan las dimensiones totales de una caja en el modelo de caja. Puedes elegir la opción que mejor se adapte a tus necesidades de diseño y facilita el manejo de tamaños y espaciados en tus elementos HTML.

6. RESPONSIVIDAD

Para manejar la responsividad en CSS y adaptar tus diseños a diferentes tamaños de pantalla, puedes utilizar las siguientes técnicas:

6.1. Diseño fluido (Fluid Layout):

Utiliza porcentajes o unidades relativas (como em o rem) en lugar de valores fijos (como píxeles) para establecer el ancho y el alto de los elementos. De esta manera, los elementos se ajustarán automáticamente según el tamaño de la ventana del navegador. Por ejemplo:

```
.container {
  width: 100%;
}

.box {
  width: 50%;
  height: 200px;
}
```

En este caso, el ancho del contenedor se establece en el 100% del ancho de su contenedor padre, lo que hace que se ajuste al ancho de la ventana. El ancho de la caja dentro del contenedor se establece en el 50% del ancho del contenedor, lo que permite que la caja se redimensione proporcionalmente en diferentes tamaños de pantalla.

6.2. Media Queries:

Las Media Queries te permiten aplicar estilos diferentes según las características de la pantalla, como el tamaño de la ventana, la resolución o la orientación. Puedes utilizar Media Queries para ajustar los estilos en diferentes puntos de interrupción (breakpoints) para adaptar el diseño a diferentes dispositivos. Por ejemplo:

```
@media (max-width: 768px) {
```

```
  .box {
    width: 100%;
  }
}
```

En este ejemplo, cuando el ancho de la ventana sea igual o menor a 768 píxeles, el ancho de la caja se establecerá en el 100%, lo que permitirá que la caja ocupe todo el ancho disponible en pantallas más pequeñas.

6.3. Unidades de vista relativa (Viewport Units)

Puedes utilizar unidades de vista relativa, como vw (viewport width) y vh (viewport height), para establecer el tamaño de los elementos en relación con el tamaño de la ventana del navegador. Por ejemplo:

```
.box {
  width: 50vw;
  height: 50vh;
}
```

En este caso, la caja ocupará el 50% del ancho y el 50% del alto de la ventana del navegador, lo que garantiza que se ajuste proporcionalmente al tamaño de la pantalla.

Estas son solo algunas técnicas para manejar la responsividad en CSS. Puedes combinar estas técnicas y ajustar los estilos según tus necesidades específicas. Recuerda probar y ajustar tu diseño en diferentes dispositivos y tamaños de pantalla para asegurarte de que se vea bien en todas las situaciones.