



# Curso de CSS

Aprende a diseñar y dar estilo a páginas web con CSS

Empezar

## Descripción general

En este curso aprenderás los fundamentos de CSS (Cascading Style Sheets) y cómo utilizarlo para controlar la presentación y el diseño de páginas HTML. Aprenderás a utilizar selectores, propiedades de estilo, el modelo de caja, diferentes técnicas de posicionamiento, pseudoclasas y pseudoelementos, media queries y diseño

responsivo. Al final del curso, serás capaz de crear sitios web atractivos y consistentes.

## 01 Selectores



# Selectores CSS

## 01 | Selectores CSS

Los selectores CSS son una parte fundamental del lenguaje de diseño CSS (Cascading Style Sheets) utilizado en el desarrollo web. Estos selectores nos permiten seleccionar y aplicar estilos a los diferentes elementos HTML de una página web.

## Selectores de tipo

Los selectores de tipo nos permiten seleccionar todos los elementos de un tipo específico en nuestro documento HTML. Utilizando el nombre del elemento como selector, por ejemplo `p`, seleccionaremos todos los elementos `<p>` (párrafos) de nuestra página.

---

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

En el ejemplo anterior, todos los párrafos ( `<p>` ) de nuestro documento tendrán texto de color azul y un tamaño de fuente de 16 píxeles.

## Selectores de clase

Los selectores de clase nos permiten seleccionar todos los elementos que tienen un atributo de clase específica. Para utilizar este selector, se antepone un punto ( `.` ) al nombre de la clase. Por ejemplo, si queremos seleccionar todos los elementos con la clase "boton", utilizamos el selector `.boton`.

```
.boton {  
  background-color: #f2f2f2;  
  padding: 5px;  
}
```

En el ejemplo anterior, todos los elementos con la clase "boton" tendrán un fondo de color gris claro ( `#f2f2f2` ) y un espacio de relleno interno de 5 píxeles.

## Selectores de ID

Los selectores de ID nos permiten seleccionar un elemento específico mediante su ID. Para utilizar este selector, se antepone un símbolo de almohadilla ( `#` ) seguido del nombre del ID. Por ejemplo, si queremos seleccionar un elemento con el ID "encabezado", utilizamos el selector `#encabezado`.

---

```
#encabezado {  
  font-family: Arial, sans-serif;  
  font-size: 16px;  
}
```

En el ejemplo anterior, el elemento con el ID "encabezado" tendrá una fuente de tipo Arial y un tamaño de fuente de 16 píxeles.

## Selectores descendientes

Los selectores descendientes nos permiten seleccionar elementos que están anidados dentro de otros elementos. Utilizamos un espacio para indicar que un elemento es descendiente de otro. Por ejemplo, si queremos seleccionar todos los párrafos `<p>` que están dentro de un `<div>`, utilizamos el selector `div p`.

```
div p {  
  color: blue;  
  font-size: 16px;  
}
```

En el ejemplo anterior, todos los párrafos que están dentro de un `<div>` tendrán texto de color azul y un tamaño de fuente de 16 píxeles.

## Selectores de atributo

Los selectores de atributo nos permiten seleccionar elementos que tienen un atributo específico. Por ejemplo, si queremos seleccionar todos los elementos que tienen el atributo `href`, utilizamos el selector `[href]`.

---

```
[href] {  
  text-decoration: underline;  
  color: purple;  
}
```

En el ejemplo anterior, todos los elementos que tienen el atributo `href` tendrán un subrayado en el texto y un color de texto púrpura.

Estos son solo algunos ejemplos de los selectores CSS que se pueden utilizar para aplicar estilos a los elementos HTML en una página web. Conociendo y utilizando estos selectores de manera adecuada, podemos lograr diseños atractivos y consistentes para nuestros sitios web.

#### Conclusión - Selectores CSS

En el curso de Selectores CSS, aprendimos a seleccionar y aplicar estilos a elementos específicos en una página web utilizando diferentes tipos de selectores como selectores de tipo, clase, ID, descendientes y atributos. Esto nos permite tener un mayor control sobre la apariencia de nuestro sitio web y hacerlo más dinámico y personalizado.



# Propiedades de Estilo

## 02 | Propiedades de Estilo

En CSS (Cascading Style Sheets), las propiedades de estilo son fundamentales para controlar la apariencia de los elementos HTML en una página web. Estas propiedades permiten definir características como colores, fondos, tipografía, márgenes, dimensiones, alineación y más. A continuación, se presentan algunas propiedades clave utilizadas en CSS:

## Colores y Fondos

La propiedad `color` define el color del texto de un elemento. Por ejemplo:

```
color: blue;
```

La propiedad `background-color` establece el color de fondo de un elemento. Por ejemplo:

```
background-color: #f2f2f2;
```

## Tipografía

Las propiedades de tipografía permiten personalizar la apariencia del texto en un elemento. La propiedad `font-family` especifica la familia de fuentes que se utilizará. Por ejemplo:

```
font-family: Arial, sans-serif;
```

La propiedad `font-size` define el tamaño del texto. Por ejemplo:

```
font-size: 16px;
```

## Márgenes, Bordes y Rellenos

Las propiedades de márgenes, bordes y rellenos son fundamentales para controlar el espaciado y la apariencia de los elementos. La propiedad `margin` define el espacio fuera del borde de un elemento. Por ejemplo:

```
margin: 10px;
```

La propiedad `padding` define el espacio dentro del borde de un elemento. Por ejemplo:

```
padding: 5px;
```

## Dimensiones

Las propiedades `width` y `height` determinan las dimensiones de un elemento. Por ejemplo:

---

```
width: 200px;  
height: 100px;
```

## Alineación y Espaciado

Las propiedades de alineación y espaciado controlan la posición y el espacio entre elementos. Por ejemplo, la propiedad `text-align` alinea el texto dentro de un elemento. Por ejemplo:

```
text-align: center;
```

La propiedad `line-height` controla la altura de línea dentro de un elemento. Por ejemplo:

```
line-height: 1.5;
```

Estas son solo algunas de las muchas propiedades de estilo que se pueden utilizar en CSS para personalizar la apariencia de los elementos HTML en una página web. Con estas propiedades, se puede lograr un diseño atractivo y coherente, separando la estructura del contenido (HTML) de su apariencia visual.

### Conclusión - Propiedades de Estilo

En el curso de Propiedades de Estilo, exploramos las diferentes propiedades que nos permiten definir el aspecto visual de los elementos HTML. Aprendimos a utilizar propiedades como



colores, fondos, tipografía, márgenes, bordes, dimensiones y posicionamiento para diseñar elementos atractivos y coherentes en nuestra página web.



# Modelo de Caja

## 03 | Modelo de Caja

En CSS, el Modelo de Caja se utiliza para describir cómo se estructura y se visualiza un elemento HTML. Cada elemento en una página web se representa como una caja rectangular, y el Modelo de Caja establece cómo se distribuye el contenido, se aplican los márgenes, los bordes y los rellenos, y se determinan las dimensiones del elemento.

El Modelo de Caja consta de cuatro componentes principales: margen, borde, relleno y contenido. Estos componentes se aplican a un elemento HTML y determinan cómo se posiciona y se muestra en la página.

## Margen

El margen es el espacio en blanco que se encuentra fuera del borde de un elemento. Se utiliza para controlar el espacio entre un elemento y otros elementos circundantes. El margen se puede ajustar utilizando la propiedad `margin` en CSS, y se pueden establecer valores para los cuatro lados (superior, derecho, inferior e izquierdo) o para cada lado de forma independiente.

## Borde

El borde es la línea que rodea el contenido y el relleno de un elemento. Define el límite visual del elemento y se utiliza para proporcionar una separación visual entre elementos adyacentes. El borde se puede personalizar estableciendo propiedades como el estilo del borde ( `border-style` ), el ancho del borde ( `border-width` ) y el color del borde ( `border-color` ).

## Relleno

El relleno es el espacio entre el contenido de un elemento y su borde. Se utiliza para agregar espacio dentro de la caja del elemento y separar el contenido del borde. El relleno se puede ajustar utilizando la propiedad `padding` en CSS, y se pueden establecer valores para los cuatro lados o para cada lado de forma independiente.

## Contenido

El contenido es el interior de la caja del elemento y donde se coloca el contenido real, como texto, imágenes o elementos HTML anidados. El tamaño y la posición del

contenido se determinan por el Modelo de Caja. El ancho y alto del contenido se pueden establecer utilizando las propiedades `width` y `height`, respectivamente.

Existen dos tipos de Modelo de Caja: el Modelo de Caja estándar y el Modelo de Caja en modo de caja.

En el Modelo de Caja estándar, el tamaño del elemento se calcula sumando el contenido, el relleno, el borde y el margen. En cambio, en el Modelo de Caja en modo de caja, el tamaño del elemento se calcula únicamente a partir del contenido y el relleno, excluyendo el margen y el borde.

El Modelo de Caja es una parte fundamental del diseño en CSS, ya que permite controlar y ajustar la apariencia visual de los elementos HTML en una página web. Entender cómo funciona el Modelo de Caja es esencial para crear diseños atractivos y responsivos.

#### Conclusión - Modelo de Caja

En el curso de Modelo de Caja, comprendimos cómo funciona el modelo de caja en CSS y cómo afecta al diseño y la disposición de los elementos en una página. Aprendimos sobre los conceptos de margen, borde, relleno y contenido, y cómo utilizar estas propiedades para crear diseños flexibles y adaptables.



# Diseño de Página

## 04 | Diseño de Página

El diseño de página es una parte fundamental del desarrollo web utilizando CSS. Permite controlar la colocación y el posicionamiento de los elementos en una página HTML, creando una estructura visualmente atractiva y funcional. En este tema, exploraremos varios conceptos relacionados con el diseño de página en CSS.

## Posicionamiento

El posicionamiento es una técnica que determina cómo se ubicarán los elementos en una página web. En CSS, existen diferentes tipos de posicionamiento disponibles:

- **static:** es el valor por defecto. Los elementos con posicionamiento estático se colocan en el orden normal de la página, siguiendo el flujo del documento.
- **relative:** permite ajustar la posición de un elemento en relación con su posición original. Se puede utilizar las propiedades `top`, `right`, `bottom` y `left` para desplazar el elemento.
- **absolute:** posiciona un elemento de forma relativa a su contenedor padre más cercano que tenga un posicionamiento distinto de `static`.

- **fixed:** posiciona un elemento de forma fija en relación con la ventana del navegador, incluso cuando se desplaza la página.

## Modelo de Caja

El modelo de caja se refiere a cómo se representan y se distribuyen los elementos en una página web. Cada elemento HTML se representa como una caja rectangular, compuesta por cuatro componentes:

1. **Margen:** es el espacio transparente que rodea al elemento.
2. **Borde:** es la línea visible que rodea al elemento y separa el contenido del margen.
3. **Relleno:** es el espacio entre el contenido y el borde.
4. **Contenido:** es el área donde se encuentra el contenido del elemento.

Existen dos modelos de caja importantes en CSS:

- **Modelo de caja estándar:** en este modelo, el ancho y alto del elemento se refieren solo al contenido, sin incluir el relleno, margen o borde. Es el modelo de caja por defecto.
- **Modelo de caja en modo de caja:** en este modelo, el ancho y alto del elemento se calculan incluyendo el relleno, margen y borde, además del contenido.

## Diseño de Página Flexible (Flexbox)

Flexbox es un modelo de diseño en CSS que permite crear diseños flexibles y fluidos. Con flexbox, se puede organizar los elementos en una única dirección (fila o columna) y controlar fácilmente el tamaño y el espaciado de los elementos.

Algunas propiedades clave del flexbox son:

- `display: flex`: activa el modelo de flexbox en un elemento contenedor.
- `flex-direction`: establece la dirección de los elementos, puede ser `row` (fila) o `column` (columna).
- `justify-content`: controla la alineación horizontal de los elementos.
- `align-items`: controla la alineación vertical de los elementos.

## Diseño de Página en Rejilla (Grid)

El diseño de página en rejilla (grid) es otro modelo de diseño en CSS que permite crear diseños más complejos y estructurados. Con el diseño en rejilla, se define una cuadrícula flexible con filas y columnas, y se colocan los elementos en las celdas de la cuadrícula.

Algunas propiedades clave del diseño en rejilla son:

- `display: grid`: activa el modelo de diseño en rejilla en un elemento contenedor.
- `grid-template-rows` y `grid-template-columns`: establecen el tamaño y la distribución de las filas y columnas de la cuadrícula.
- `grid-gap`: define el espacio entre las celdas de la cuadrícula.
- `grid-row` y `grid-column`: controlan la ubicación de los elementos en la cuadrícula.

## Centrado Vertical y Horizontal

El centrado vertical y horizontal de los elementos es una técnica común en el diseño de página. En CSS, existen varias formas de lograr este efecto:

- **Centrado vertical**: se puede utilizar la propiedad `display: flex` en un contenedor y la propiedad `align-items: center` para centrar verticalmente los elementos hijos.

- **Centrado horizontal:** se puede utilizar la propiedad `display: flex` en un contenedor y la propiedad `justify-content: center` para centrar horizontalmente los elementos hijos.
- **Centrado absoluto:** se puede utilizar el posicionamiento absoluto (`position: absolute`) en combinación con las propiedades `top`, `right`, `bottom` y `left` para centrar un elemento en relación con su contenedor.

## Conclusiones


El diseño de página en CSS es esencial para crear sitios web visualmente atractivos y funcionales. Con el control del posicionamiento, el modelo de caja, los modelos de diseño flexbox y grid, y las técnicas de centrado, se pueden crear diseños personalizados y adaptativos. A través de las propiedades y selectores CSS, se pueden aplicar estilos específicos a los elementos y lograr los resultados deseados en el diseño de página.

Recuerda practicar y experimentar con estos conceptos para lograr un dominio efectivo del diseño de página en CSS.

### Conclusión - Diseño de Página

En el curso de Diseño de Página, exploramos diferentes técnicas de posicionamiento y disposición de elementos en una página web. Aprendimos a utilizar el posicionamiento estático, relativo, absoluto y fijo para controlar la ubicación de los elementos.

También aprendimos sobre los modelos de caja flexibles y en rejilla para crear diseños más complejos y estructurados.



# Pseudoclases y Pseudoelementos

05 | Pseudoclases y Pseudoelementos

## Pseudoclases y Pseudoelementos en CSS

En CSS, existen las pseudoclases y los pseudoelementos, que permiten seleccionar y aplicar estilos a elementos en estados específicos o a partes específicas de los elementos HTML. Estos son útiles para crear interacciones y efectos visuales en nuestras páginas web.



# Pseudoclases

Las pseudoclases son palabras clave que se agregan a un selector CSS y seleccionan elementos HTML en estados específicos. Algunas pseudoclases comunes son:

- `:hover`: Aplica estilos cuando el cursor pasa sobre el elemento.
- `:active`: Aplica estilos cuando el elemento está siendo clicado.
- `:focus`: Aplica estilos cuando el elemento tiene el foco de atención, como cuando se hace clic en un campo de formulario.

Estas pseudoclases se utilizan para cambiar la apariencia de los elementos en respuesta a la interacción del usuario. Por ejemplo, podemos cambiar el color de fondo de un botón cuando el cursor está sobre él utilizando la pseudoclase `:hover`.

```
.boton:hover {  
  background-color: blue;  
}
```

# Pseudoelementos

Los pseudoelementos, por otro lado, son elementos ficticios que se generan mediante CSS y se agregan al contenido existente de un elemento HTML. Estos se utilizan para aplicar estilos a partes específicas de los elementos. Algunos pseudoelementos comunes son:

- `::before`: Agrega contenido antes del contenido principal del elemento seleccionado.
- `::after`: Agrega contenido después del contenido principal del elemento seleccionado.

Estos pseudoelementos son útiles para agregar elementos decorativos o contenido adicional a los elementos HTML. Por ejemplo, podemos agregar un ícono antes del contenido de un enlace utilizando el pseudoelemento `::before`.

```
.enlace::before {  
  content: "\f105";  
  font-family: "FontAwesome";  
  margin-right: 5px;  
}
```

En el ejemplo anterior, utilizamos el pseudoelemento `::before` para agregar un ícono del conjunto de iconos "FontAwesome" antes del contenido de un elemento con la clase `.enlace`. El contenido se define con la propiedad `content`, y se puede utilizar cualquier código unicode o caracteres especiales para representar el contenido deseado.

## Conclusion

Las pseudoclasas y pseudoelementos en CSS nos permiten aplicar estilos a elementos en estados específicos o partes específicas de los elementos, lo que agrega interactividad y personalización a nuestras páginas web. Al utilizar estas herramientas, podemos crear efectos visuales y mejorar la experiencia del usuario en nuestros sitios web.

En el curso de Pseudoclasas y Pseudoelementos, descubrimos cómo aplicar estilos a elementos en diferentes estados y cómo agregar contenido antes y después de los elementos HTML utilizando pseudoclasas y pseudoelementos. Estas técnicas nos permiten darle interactividad y dinamismo a nuestros sitios web.



# Media Queries y Diseño Responsivo

06 | Media Queries y Diseño Responsivo

En el desarrollo web, es esencial que los sitios sean accesibles y adaptables a diferentes dispositivos y pantallas. Esto se logra mediante el uso de técnicas de diseño responsivo, que permiten que el contenido se ajuste y se vea correctamente en una amplia variedad de tamaños de pantalla.

# ¿Qué son las Media Queries?

Las Media Queries son una característica en CSS que nos permite aplicar estilos específicos a diferentes dispositivos o condiciones. Mediante el uso de Media Queries, podemos adaptar el diseño de nuestro sitio web según las características del dispositivo en el que se está visualizando, como por ejemplo, el tamaño de la pantalla.

Un ejemplo simple de una Media Query sería:

```
@media (max-width: 768px) {  
  /* estilos para pantallas con un ancho máximo de 768px */  
  /* aquí irían los estilos específicos para dispositivos móviles */  
}
```

En el ejemplo anterior, se aplicarían los estilos dentro de la Media Query cuando el ancho de la pantalla sea igual o menor a 768 píxeles. Esto nos permite crear diseños adaptativos que se verán correctamente en dispositivos con pantallas más pequeñas.

## Diseño Responsivo

El diseño responsivo consiste en adaptar el diseño y la apariencia de una página web según las características del dispositivo en el que se está visualizando. Esto se logra utilizando Media Queries, que nos permiten aplicar diferentes estilos en base a ciertos puntos de quiebre.

Un enfoque común en el diseño responsivo es el principio "móvil primero", que implica diseñar inicialmente para dispositivos móviles y luego ir agregando estilos conforme

aumenta el tamaño de la pantalla. Esto garantiza que el diseño se vea bien en dispositivos pequeños y se adapte de manera adecuada a pantallas más grandes.

## Adaptable a diferentes tamaños de pantalla

Cuando diseñamos de manera responsiva, debemos asegurarnos de que nuestro sitio se vea y se comporte correctamente en diferentes tamaños de pantalla. Esto implica ajustar el tamaño y el diseño de los elementos para que se adapten a pantallas más pequeñas o más grandes.

Utilizando Media Queries, podemos establecer diferentes estilos para pantallas con diferentes tamaños. Por ejemplo:

```
@media (max-width: 600px) {  
  /* estilos para pantallas con un ancho máximo de 600px */  
}  
@media (min-width: 601px) and (max-width: 900px) {  
  /* estilos para pantallas con un ancho mínimo de 601px y un ancho máximo de  
  900px */  
}  
@media (min-width: 901px) {  
  /* estilos para pantallas con un ancho mínimo de 901px */  
}
```

En el ejemplo anterior, se establecen diferentes estilos para tres rangos de tamaño de pantalla: pantallas con un ancho máximo de 600px, pantallas con un ancho mínimo de 601px y un ancho máximo de 900px, y pantallas con un ancho mínimo de 901px. Esto nos permite adaptar el diseño a diferentes tamaños de pantalla de manera fluida y mejorar la experiencia del usuario.

## Uso de @media para aplicar estilos específicos

Para utilizar Media Queries, debemos hacer uso de la regla `@media` en nuestro archivo CSS. Esta regla nos permite definir los estilos que se aplicarán cuando se cumpla una determinada condición.

```
@media (condición) {  
    /* estilos que se aplicarán cuando se cumpla la condición */  
}
```

La condición se especifica dentro de los paréntesis y puede ser basada en varias características, como el ancho de la pantalla (`width`), la orientación (`orientation`), etc. Al cumplirse la condición, los estilos dentro de la Media Query se aplicarán a los elementos seleccionados.

## Diseño móvil primero vs. diseño de escritorio primero

Al diseñar de manera responsiva, se pueden seguir dos enfoques principales: el enfoque móvil primero y el enfoque de escritorio primero.

El enfoque móvil primero implica diseñar y desarrollar inicialmente para dispositivos móviles, utilizando Media Queries para agregar estilos y adaptar el diseño a pantallas más grandes. Con este enfoque, nos aseguramos de que el sitio se vea y funcione correctamente en dispositivos móviles y luego mejoramos la experiencia en pantallas más grandes.

Por otro lado, el enfoque de escritorio primero implica diseñar y desarrollar inicialmente para pantallas de escritorio y luego utilizar Media Queries para adaptar el diseño a pantallas más pequeñas. Este enfoque puede ser útil cuando el diseño inicial

está muy enfocado en la experiencia de escritorio y se requiere optimizar el diseño para dispositivos móviles.

Ambos enfoques tienen sus ventajas y desventajas y la elección depende del proyecto y las necesidades específicas.

## Conclusiones

Las Media Queries y el diseño responsivo son herramientas esenciales en el desarrollo web actual. Permiten que los sitios web se adapten y se vean correctamente en diferentes dispositivos y tamaños de pantalla. Utilizando Media Queries, podemos aplicar estilos específicos según las características del dispositivo, creando una experiencia de usuario óptima y mejorando la accesibilidad del sitio.

Recuerda que el diseño responsivo implica pensar en diferentes tamaños de pantalla y adaptar el diseño de manera fluida. Tanto el enfoque móvil primero como el enfoque de escritorio primero tienen sus ventajas y desventajas, y la elección del enfoque depende de las necesidades del proyecto.

¡Utiliza las Media Queries y el diseño responsivo para crear sitios web atractivos y accesibles!

### Conclusión - Media Queries y Diseño Responsivo

En el curso de Media Queries y Diseño Responsivo, aprendimos cómo adaptar nuestros diseños CSS a diferentes tamaños de

pantalla utilizando media queries. También exploramos el enfoque de diseño móvil primero y diseño de escritorio primero, y cómo utilizar media queries para aplicar estilos específicos según las características del dispositivo.



# Ejercicios Practicos

Pongamos en práctica tus conocimientos

07 | Ejercicios Practicos

En esta lección, pondremos la teoría en práctica a través de actividades prácticas. Haga clic en los elementos a continuación para verificar cada ejercicio y desarrollar habilidades prácticas que lo ayudarán a tener éxito en el tema.



## Selector de Clase



Crea una página HTML con varios elementos de tipo párrafo. Agrega un selector de clase a uno de los párrafos y aplica un estilo diferente a través de CSS.

## Estilo de Fondo



Crea una página HTML con una sección de contenido. Establece un estilo de fondo personalizado para la sección usando una propiedad de estilo en CSS.

## Tamaño de Caja



Crea una página HTML con un elemento de imagen. Ajusta el tamaño de la caja de la imagen utilizando propiedades de modelo de caja en CSS.

## Posicionamiento



Crea una página HTML con varios elementos de contenido. Utiliza diferentes valores de posicionamiento (static, relative, absolute, fixed) para controlar la ubicación de los elementos en la página.

## Estilo de Enlace

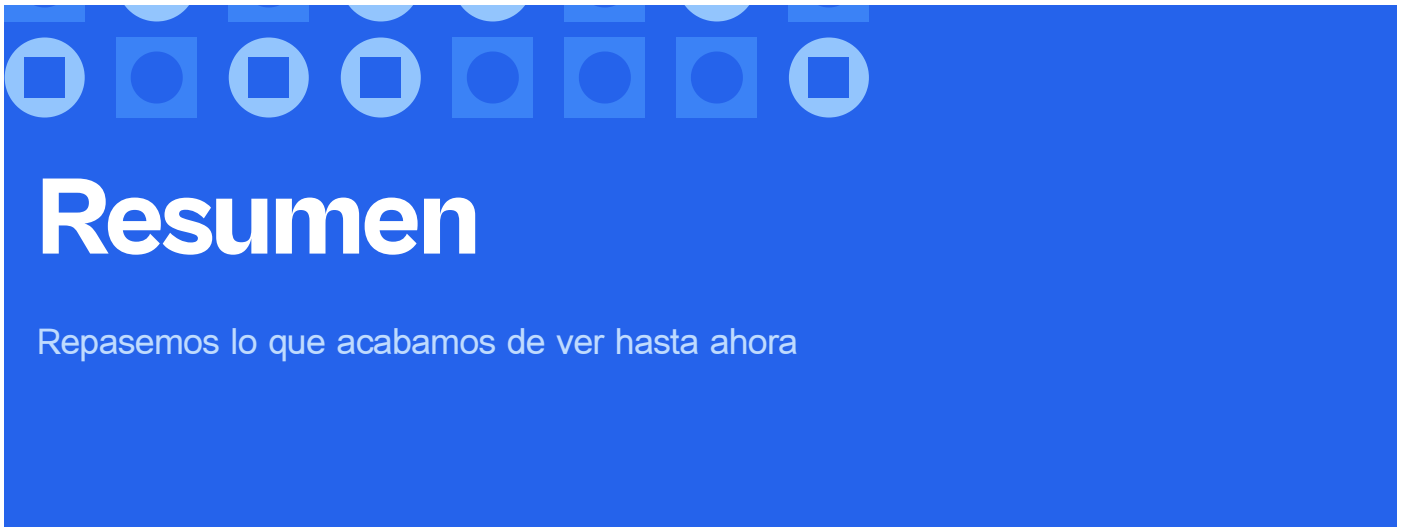


Crea una página HTML con varios enlaces. Aplica estilos diferentes a los enlaces cuando el cursor pasa sobre ellos utilizando pseudoclases en CSS.

## Diseño Responsivo



Crea una página HTML con una sección de contenido. Utiliza media queries en CSS para hacer que la sección se adapte y se vea diferente en diferentes tamaños de pantalla.



- ✓ En el curso de Selectores CSS, aprendimos a seleccionar y aplicar estilos a elementos específicos en una página web utilizando diferentes tipos de selectores como selectores de tipo, clase, ID, descendientes y atributos. Esto nos permite tener un mayor control sobre la apariencia de nuestro sitio web y hacerlo más dinámico y personalizado.
- ✓ En el curso de Propiedades de Estilo, exploramos las diferentes propiedades que nos permiten definir el aspecto visual de los elementos HTML. Aprendimos a utilizar propiedades como colores, fondos, tipografía, márgenes, bordes, dimensiones y posicionamiento para diseñar elementos atractivos y coherentes en nuestra página web.
- ✓ En el curso de Modelo de Caja, comprendimos cómo funciona el modelo de caja en CSS y cómo afecta al diseño y la disposición de los elementos en una página. Aprendimos sobre los conceptos de margen, borde, relleno y contenido, y cómo utilizar estas propiedades para crear diseños flexibles y adaptables.
- ✓ En el curso de Diseño de Página, exploramos diferentes técnicas de posicionamiento y disposición de elementos en una página web. Aprendimos a utilizar el posicionamiento estático, relativo, absoluto y fijo para controlar la ubicación de los elementos. También aprendimos sobre los modelos de caja flexibles y en rejilla para crear diseños más complejos y estructurados.
- ✓ En el curso de Pseudoclases y Pseudoelementos, descubrimos cómo aplicar estilos a elementos en diferentes estados y cómo agregar contenido antes y después de los elementos HTML utilizando pseudoclases y pseudoelementos. Estas técnicas nos permiten darle interactividad y dinamismo a nuestros sitios web.

- ✓ En el curso de Media Queries y Diseño Responsivo, aprendimos cómo adaptar nuestros diseños CSS a diferentes tamaños de pantalla utilizando media queries. También exploramos el enfoque de diseño móvil primero y diseño de escritorio primero, y cómo utilizar media queries para aplicar estilos específicos según las características del dispositivo.



# Prueba

Comprueba tus conocimientos respondiendo unas preguntas

09 | Prueba

1. ¿Cuál es el objetivo principal del lenguaje CSS?

- ☐ Manipular la estructura del contenido HTML
  - ☐ Interactuar con bases de datos en línea
  - ☐ Controlar la presentación y el diseño de las páginas HTML
-

2. ¿Cuál de los siguientes selectores de CSS selecciona todos los elementos de un tipo específico?

- ☐ Selector de clase
  - ☐ Selector de tipo
  - ☐ Selector de ID
- 

3. ¿Cuál es la propiedad de CSS que define el color del texto?

- ☐ Font-family
  - ☐ Color
  - ☐ Background-color
- 

4. ¿Cuál de los siguientes modelos de caja de CSS incluye el margen, borde, relleno y contenido?

- ☐ Modelo de caja flexible
  - ☐ Modelo de caja estándar
  - ☐ Modelo de caja en modo de caja
- 

5. ¿Cuál es el posicionamiento de CSS que se utiliza para elementos que no tienen una posición especificada?

- ☐ Static
- ☐ Absolute

☐ Relative

---

6. ¿Cuál es el selector de CSS que selecciona elementos antes del contenido?

☐ ::after

☐ :hover

☐ ::before

---

7. ¿Cuál es la técnica utilizada en CSS para adaptar el diseño a diferentes tamaños de pantalla?

☐ Media queries

☐ Pseudoclases

☐ Flexbox

---

8. ¿Cuál es la propiedad de CSS que controla cómo se posiciona un elemento en la página?

☐ Width

☐ Color

☐ Position

---

9. ¿Cuál de los siguientes selectores de CSS selecciona elementos anidados dentro de otro elemento?

☐ Selector descendiente

- ☐ Selector de tipo
  - ☐ Selector de clase
- 

10. ¿Cuál es la propiedad de CSS que establece el estilo, ancho y color del borde de un elemento?

- ☐ Border
  - ☐ Width
  - ☐ Background-color
- 

11. ¿Cuál de los siguientes selectores de CSS selecciona un elemento con un ID específico?

- ☐ Selector de clase
  - ☐ Selector de ID
  - ☐ Selector de tipo
- 

12. ¿Cuál es la propiedad de CSS que determina las dimensiones de un elemento?

- ☐ Color
  - ☐ Font-size
  - ☐ Width
- 

13. ¿Cuál es la técnica utilizada en CSS para crear efectos suaves de cambio entre diferentes estados de un elemento?

- ☐ Transiciones
- ☐ Pseudoclases
- ☐ Media queries

Entregar

Conclusión

# Felicidades!

¡Felicitaciones por completar este curso! Has dado un paso importante para desbloquear todo tu potencial. Completar este curso no se trata solo de adquirir conocimientos; se trata de poner ese conocimiento en práctica y tener un impacto positivo en el mundo que te rodea.



Comparte este curso

Created with [LearningStudioAI](#)



