



IT Academy
by KIBERNUM

Programa: Desarrollo de Aplicaciones Front-End

Módulo 2: Fundamentos de Desarrollo Front-End



Aprendizaje esperado

Aplicar hojas de estilo CSS básicas distinguiendo elementos de responsividad para personalizar la presentación de un documento HTML acorde a un requerimiento entregado.



¿Cómo crees que el diseño de una página web influye en la experiencia del usuario?





CSS: El Lenguaje de Estilo para la Web

CSS (Cascading Style Sheets) es el lenguaje que define el diseño y la apariencia de los documentos HTML. Permite separar el contenido y la estructura del estilo, facilitando la personalización visual de los elementos HTML.

¿Qué es CSS?

CSS controla aspectos como colores, márgenes, fuentes, disposición de los elementos y muchos otros detalles de presentación. A través de CSS, puedes crear páginas web visualmente atractivas, interactivas y dinámicas.



Historia de CSS

◆ Los primeros días de la web

El estilo se aplicaba directamente dentro de las etiquetas HTML.

Problemas:

- ❑ Modificar el estilo requería cambiar cada línea de código.
- ❑ Complicaba el mantenimiento y aumentaba los errores.

◆ La solución: CSS

En 1996, el W3C introdujo CSS (Cascading Style Sheets).

Objetivo: Separar la estructura y el contenido del diseño.

Ventajas:

- ❑ Definir estilos de manera centralizada.
- ❑ Facilidad para modificar y mantener el diseño sin alterar el HTML.

◆ Combinación con JavaScript

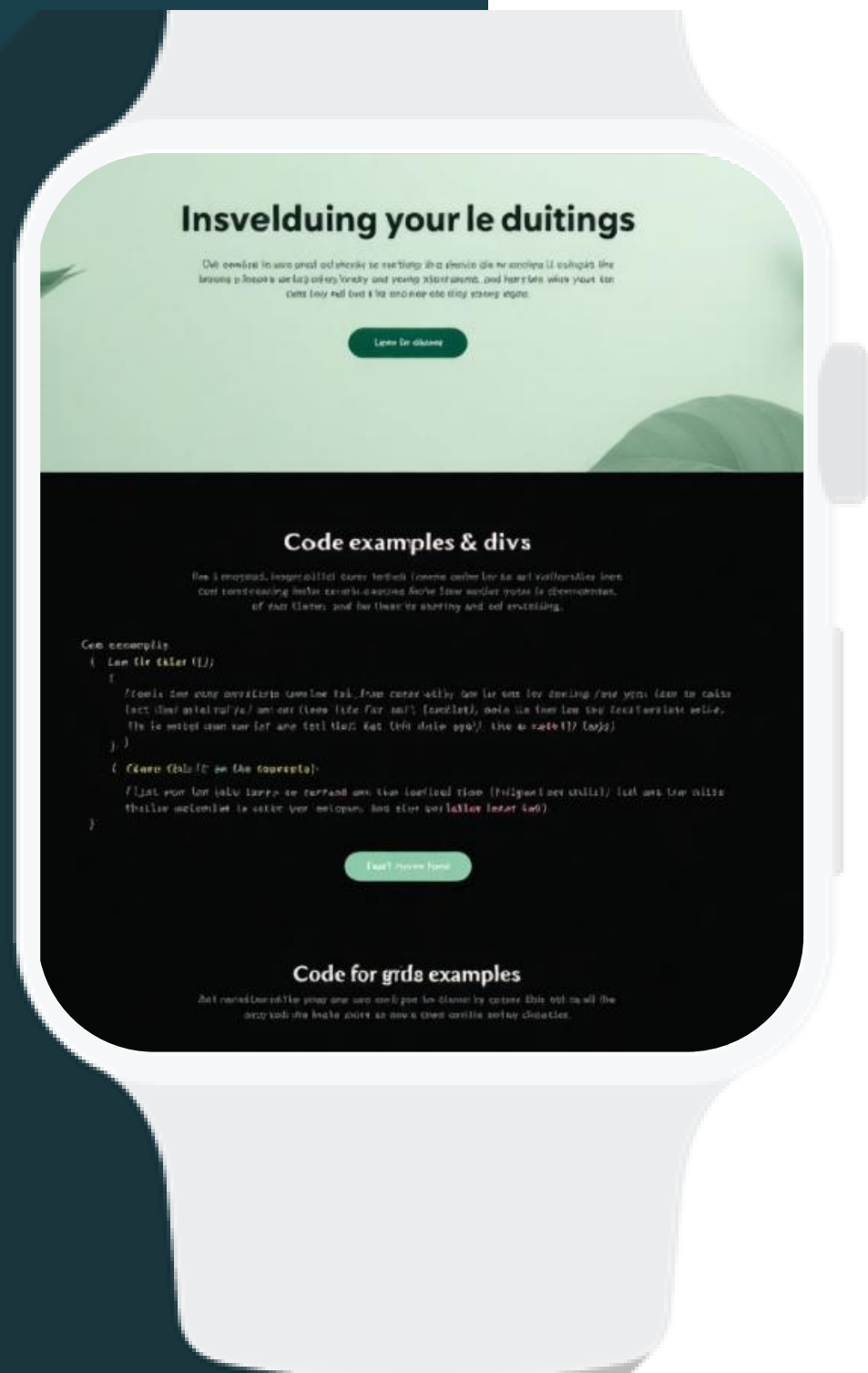
CSS permitió crear páginas interactivas y dinámicas al combinarse con JavaScript.



Evolución de CSS

Versión	Definición
CSS 1 (1997)	Apareció para separar contenido y estilo, pero tuvo falencias que limitaron su éxito.
CSS 2 (1998)	Introdujo cambios importantes, pero tuvo problemas de compatibilidad entre navegadores.
CSS 3 (2014)	Solucionó la mayoría de las necesidades de los diseñadores y cumplió el objetivo principal de CSS.

¿Por qué es útil CSS?



Consistencia visual

Asegura que todas las páginas de un sitio tengan un diseño uniforme.



Facilidad de mantenimiento

Modificar una hoja de estilo impacta en múltiples páginas.



Mejora la experiencia del usuario

Diseños más atractivos y accesibles.



Relación entre CSS y HTML

HTML estructura un sitio web, mientras que CSS define su apariencia visual. CSS controla cómo se presentan los elementos HTML, incluyendo su color, tamaño, alineación y espaciado.

Por ejemplo, si en HTML tenemos un encabezado y un párrafo:

```
<body>  
  <h1>Bienvenido a mi sitio web</h1>  
  <p>Este es un sitio de ejemplo para aprender HTML y CSS.</p>  
</body>
```




Podemos aplicar estilos con CSS para cambiar la apariencia de esos elementos, como hacer que el encabezado sea azul y centrado, y el párrafo sea negro con una fuente más pequeña:

Relación entre CSS y HTML

```
<style>
  h1 {
    color: blue;
    text-align: center;
  }

  p {
    color: black;
    font-size: 14px;
  }
</style>
```



¿Cómo Funciona CSS?

CSS funciona mediante selectores y bloques de declaración. Primero, se selecciona el elemento HTML que deseas estilizar. Luego, en el bloque de declaración, defines los estilos que deseas aplicar a ese elemento.

Regla CSS

Declaración

```
h1 { color: green; }
```

Selector Propiedad Valor

Manejo de Hojas de Estilo

El manejo de hojas de estilo en CSS se refiere a la forma en que se organizan y aplican los estilos a los elementos HTML para definir su apariencia visual en un navegador.



Estilos en Línea, Embebidos y en Archivo Externo

ESTILOS EN LÍNEA

Los estilos se aplican directamente a los elementos HTML mediante el atributo style. No es recomendable para proyectos grandes.

```
<p style="color: red; font-size: 16px;">  
Este es un texto en rojo con tamaño de 16px.</p>
```




Estilos en Línea, Embebidos y en Archivo Externo

ESTILOS EMBEBIDOS

Se utiliza la etiqueta **style** en la sección **head** del documento HTML. Permite aplicar estilos a múltiples elementos dentro de una misma página.

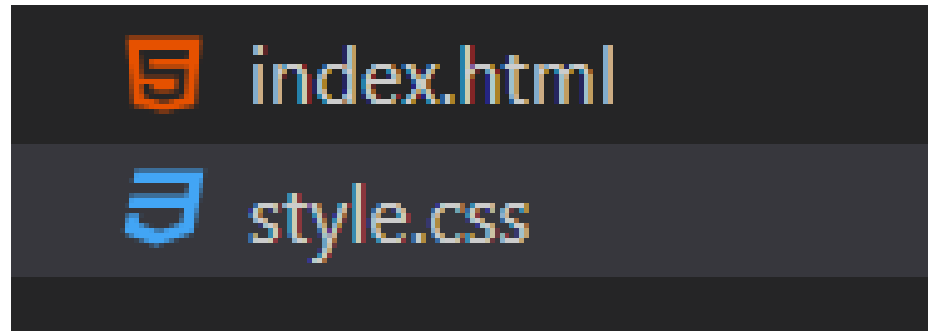
```
<head>
  <style>
    h1 {
      color: blue;
      text-align: center;
    }
    p {
      font-size: 14px;
      color: black;
    }
  </style>
</head>
```



Estilos en Línea, Embebidos y en Archivo Externo

HOJAS DE ESTILO EXTERNAS

Se utiliza un archivo `.css` separado, que se enlaza al documento HTML mediante la etiqueta `link`. Es el método más recomendable para proyectos grandes.



index.html
style.css

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




Selectores por etiqueta

- ◆ Aplica el estilo a todos los elementos del mismo tipo o etiqueta.

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




Selectores por clase

- ◆ Aplica el estilo a todos los elementos que tengan la clase especificada.
- ◆ Se denota con el prefijo ".".

```
.mi-clase {  
  background-color: yellow;  
}
```

- ◆ Luego en HTML

```
<body>  
  <div class="mi-clase">Contenido con fondo amarillo</div>  
</body>
```




Selectores por id

- ◆ Aplica el estilo a un solo elemento que tenga el ID especificado.
- ◆ Los IDs son únicos dentro de un documento.
- ◆ Se denota con el prefijo #.

```
#mi-id {  
  font-size: 20px;  
}
```

- ◆ Luego en HTML

```
<body>  
  <p id="mi-id">Este párrafo tiene un tamaño de fuente de 20px.</p>  
</body>
```


Reto

DADO EL SIGUIENTE CÓDIGO HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Ejercicio de Selectores en CSS</title>
</head>
<body>

    <h1>Ejercicio de Selectores CSS</h1>

    <p>Este es un párrafo con un selector por etiqueta.</p>
    <p>Este es un párrafo con un selector por clase.</p>
    <p>Este es un párrafo con un selector por ID.</p>
    <div>
        <p>Este es otro párrafo dentro de un div con clase
"destacado".</p>
    </div>
    <p>Otro párrafo sin estilos aplicados.</p>
</body>
</html>
```


Reto

Instrucciones:

1. Copia y guarda el código HTML que se te entregará.
2. Crea un archivo styles.css en la misma carpeta.
3. Aplica estilos usando selectores:
 - Por etiqueta: Aplicar un estilo a todos los elementos de un tipo.
 - Por clase: Aplicar un estilo a elementos con una clase específica.
 - Por ID: Aplicar un estilo a un elemento único identificado con un ID.
4. Tienen libertad total para diseñar como deseen. ¡Sean creativos! 🎨

¿Cuál será el diseño más original? 🚀





El Modelo de Cajas en CSS

El modelo de cajas es un concepto fundamental en CSS que describe cómo los elementos HTML se organizan en términos de su contenido, relleno (padding), borde (border) y margen (margin).



Componentes del Modelo de Cajas

Los Contenido

El texto o imagen dentro del elemento.

Relleno (padding)

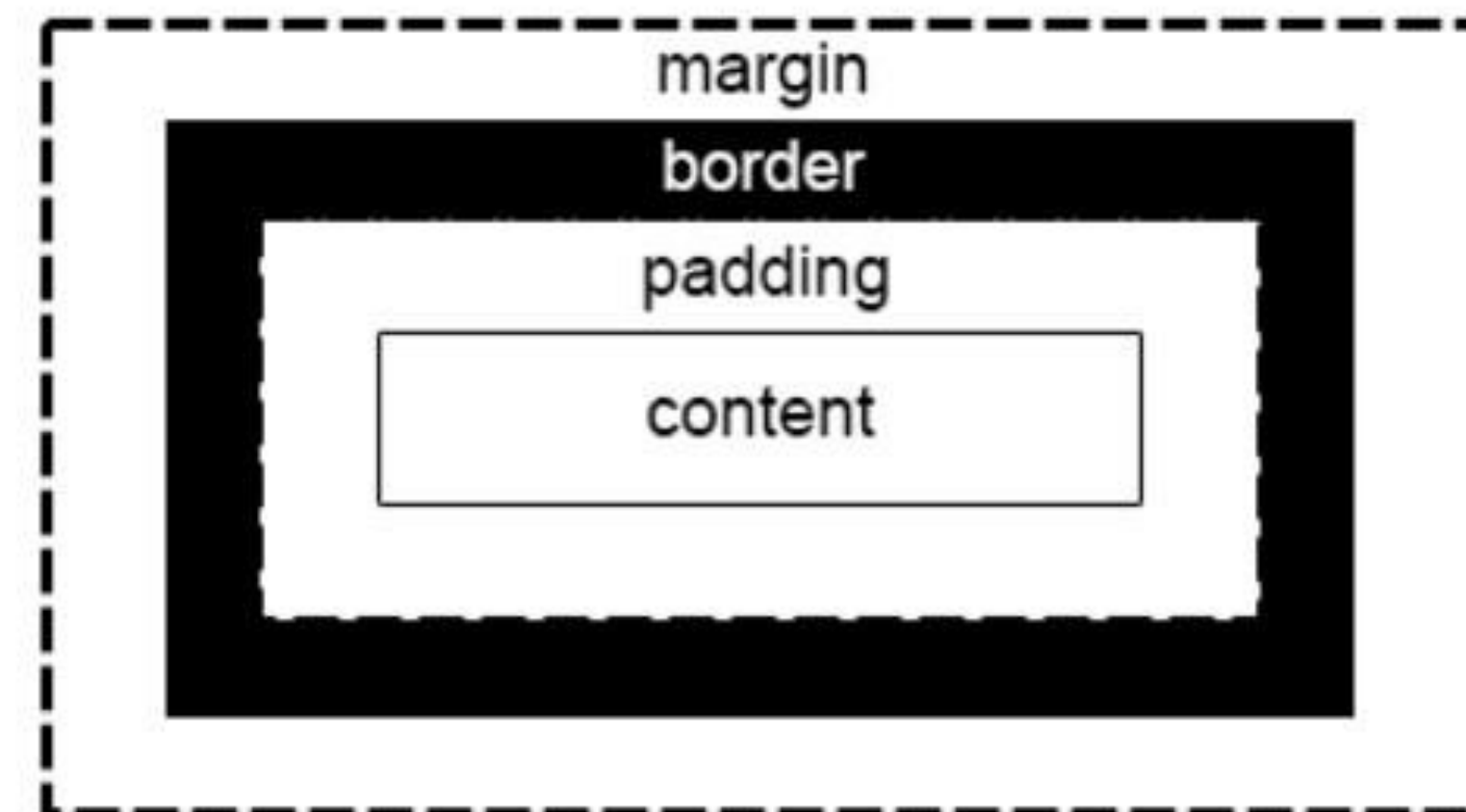
Espacio entre el contenido y el borde del elemento.

Borde (border)

El contorno alrededor del elemento.

Margen (margin)

El espacio externo al borde que separa el elemento de otros.



Ejemplo: CSS para el modelo de cajas

- ❑ width: 300px: Define el ancho del contenido del elemento div (área para texto o elementos internos).
- ❑ padding: 20px: Añade 20 píxeles de espacio interno entre el contenido y el borde del div.
- ❑ border: 5px solid black: Añade un borde de 5 píxeles, sólido y de color negro alrededor del div.
- ❑ margin: 10px: Añade 10 píxeles de espacio externo entre el div y los elementos adyacentes.
- ❑ box-sizing: border-box: Hace que el ancho total del div (300px) incluya el padding y el borde, evitando que el contenido se expanda más allá de los 300px.

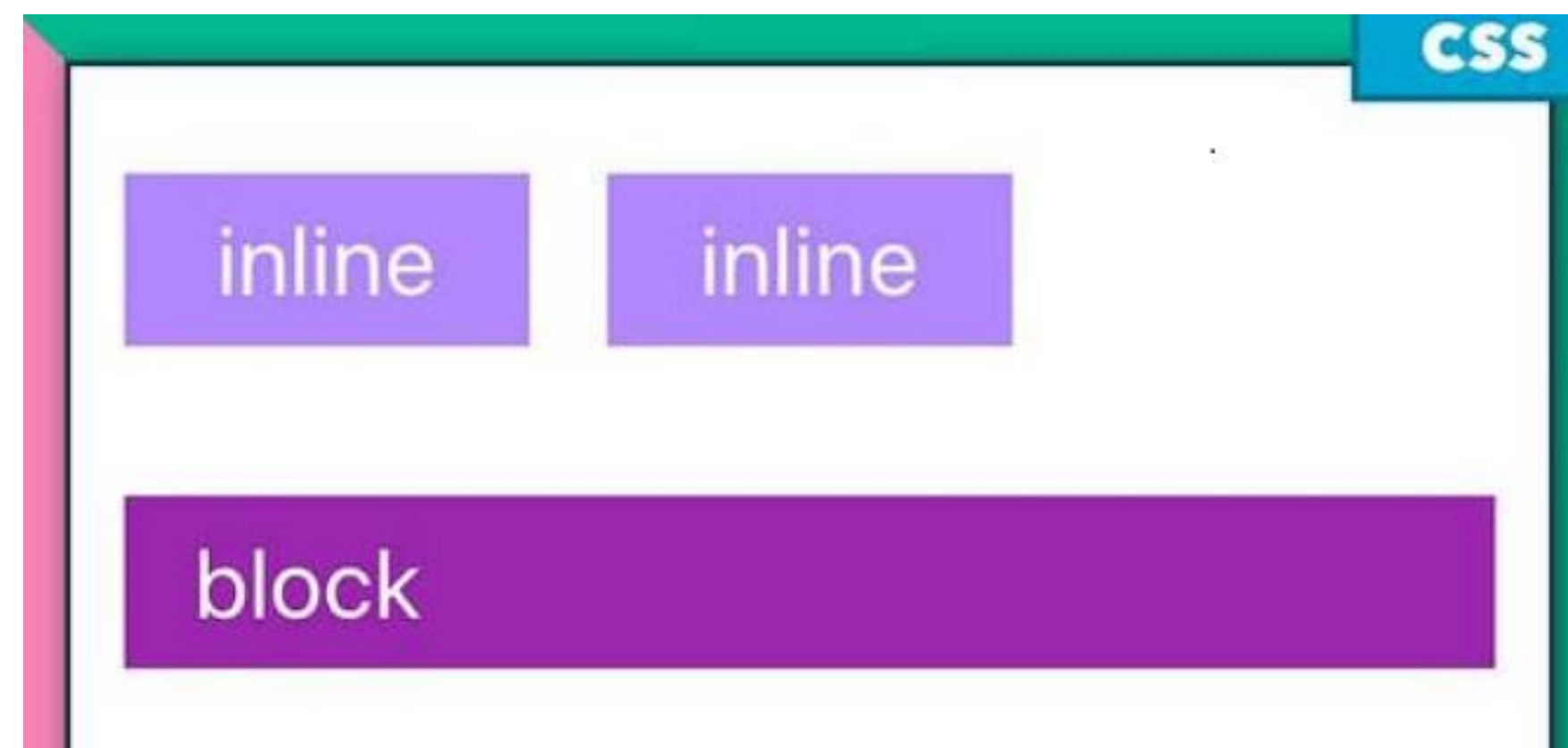
```
div {  
  width: 300px;  
  padding: 20px;  
  border: 5px solid black;  
  margin: 10px;  
  box-sizing: border-box;  
}
```




Block vs Inline



- ◆ Los conceptos de block e inline están directamente relacionados con el modelo de cajas en CSS.
- ◆ Comprender estos conceptos es clave para entender cómo se disponen y comportan los elementos en una página web.
- ◆ La clasificación de un elemento como block o inline afecta su representación dentro del modelo de cajas.





Elementos de Bloque (block)



Características:

- ◆ Ocupan todo el ancho disponible.
- ◆ Siempre comienzan en una nueva línea.
- ◆ Su ancho se extiende a lo largo del contenedor.
- ◆ Márgenes, rellenos y bordes afectan a los elementos que están por encima y por debajo.

Ejemplos comunes.

``, `<a>`, ``, `<input>`, ``.



Elementos en Línea (inline)

Características:

- ◆ Ocupan solo el ancho del contenido que contienen.
- ◆ Se colocan uno al lado del otro en la misma línea.
- ◆ Márgenes y rellenos afectan solo a su espacio interno.

Ejemplos comunes.

`<div>`, `<p>`, encabezados (`<h1>`, `<h2>`, `<h3>`, etc.), ``, `<section>`.



Comparación Block vs Inline

Característica	Block	Inline
Ancho	Ocupa todo el ancho disponible.	Ocupa solo el ancho del contenido.
Línea	Comienza en una nueva línea.	Se coloca en la misma línea.
Márgenes y rellenos	Afectan a elementos adyacentes.	Afectan solo al espacio interno.
Ejemplos	<div>, <p>, <h1>, .	, <a>, .

Reto: Aplicando el modelo de Cajas

Instrucciones:

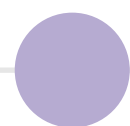
Crea una caja con las siguiente propiedades

- Ancho: 200px
- Alto: 100px
- Borde: 4px sólido púrpura
- Padding: 10px
- Margen: 20px

Tiempo límite:

¡Tienen 5 minutos para resolverlo! ⌚

¡Esta es una caja!





IT Academy

by KIBERNUM