



Sesión 14

Programación Nivel Básico



UNIVERSIDAD
LIBRE®

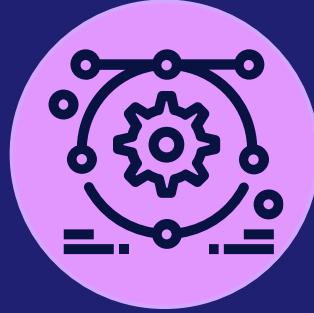
eTraining®



Sesión 14:

Desarrollo de sitios web, estáticos HTML, CSS JavaScript

Mi primera página web, diseño de navegación



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

Introducir conceptos clave

El objetivo principal es introducir a los estudiantes a los conceptos básicos de HTML, CSS y JavaScript. Es importante que comprendan la función de cada uno en la creación de páginas web.

Aplicar conocimientos en proyectos reales

El objetivo final es que los estudiantes puedan aplicar sus conocimientos en casos de uso reales. Esto incluye el diseño y desarrollo de sus propias páginas web.

Desarrollar habilidades prácticas

Otro objetivo es fomentar la práctica a través de ejercicios resueltos. Estos ejercicios ayudarán a los estudiantes a aplicar lo aprendido de manera efectiva.

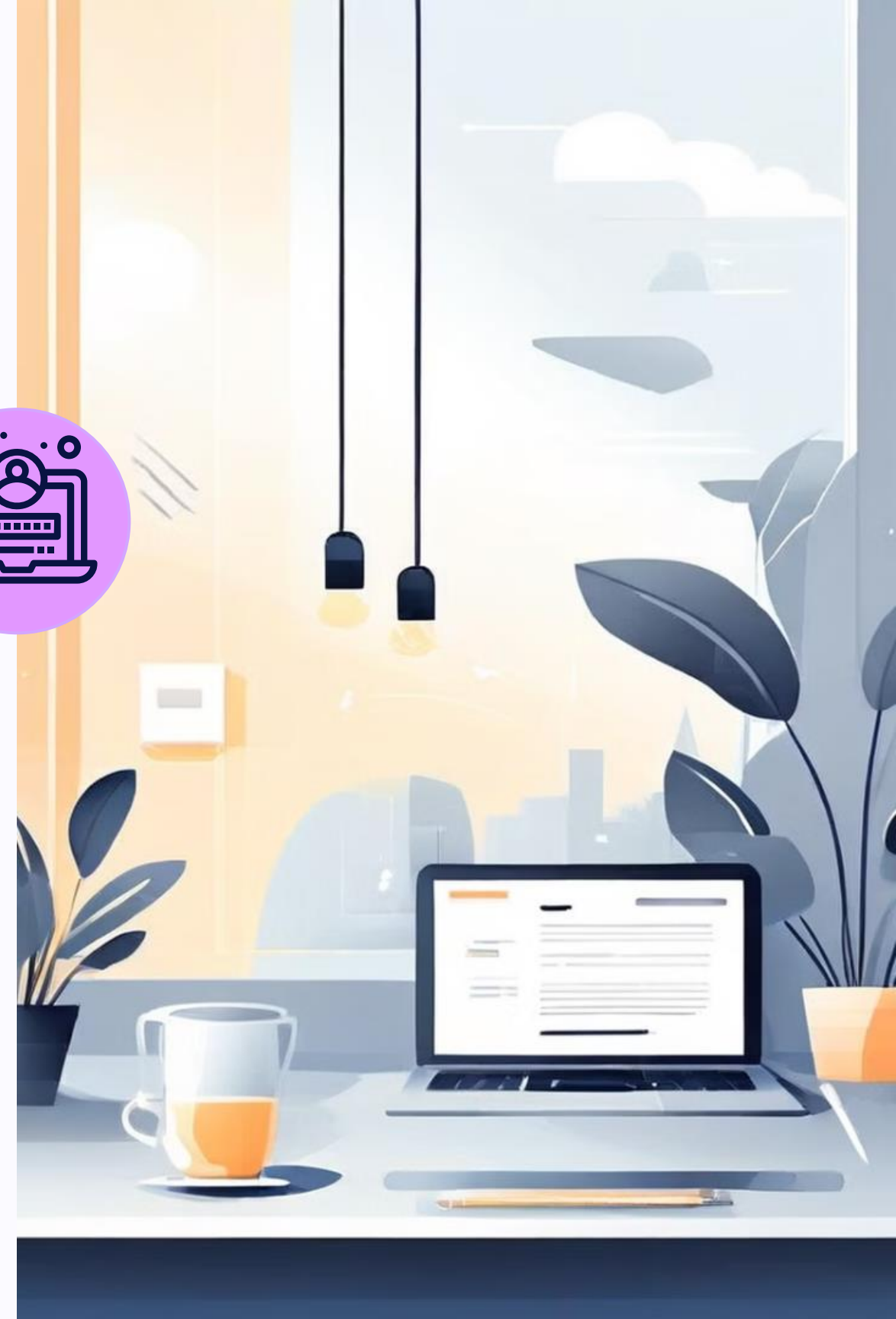
Facilitar el aprendizaje autónomo

Queremos que los estudiantes se sientan confiados para aprender de manera autónoma. Esto se logra mediante recursos y ejemplos que estimulen su curiosidad.

Mi primera página web

Esta es una introducción a la creación de páginas web usando HTML, CSS y JavaScript. Aprenderemos lo básico para comenzar a desarrollar. Crear una página web puede ser emocionante y muy gratificante.

A lo largo de esta experiencia, exploraremos los objetivos de estos lenguajes de programación. También analizaremos conceptos clave que te ayudarán a entender mejor el desarrollo web.





Introducción

La Importancia de la Web

La web es una herramienta esencial en nuestra vida diaria. Facilita la comunicación, provee información y permite el comercio.

Creación de Páginas Web

Aprender a crear una página web es un proceso emocionante. Con HTML, CSS y JavaScript, puedes dar vida a tus ideas.

Conceptos básicos



HTML

HTML es el lenguaje estándar para crear páginas web. Proporciona la estructura y contenido de la web. Permite a los navegadores mostrar texto, imágenes y enlaces correctamente.



CSS

CSS se utiliza para definir el estilo de las páginas web. Permite personalizar colores, fuentes y diseño. Mejora la presentación y la experiencia del usuario en la web.



JavaScript

JavaScript es un lenguaje de programación que aporta interactividad. Permite crear elementos dinámicos en las páginas. Es esencial para mejorar la funcionalidad de la web.



Aprendizaje

Entender estos conceptos es crucial para el desarrollo web. Facilita la creación de proyectos más complejos. Fomenta un aprendizaje continuo en tecnología.

Definiciones

1 HTML

HTML, o Lenguaje de Marcado de Hipertexto, es el estándar para crear páginas web. Define la estructura de la información mediante etiquetas que organizan el contenido.

2 CSS

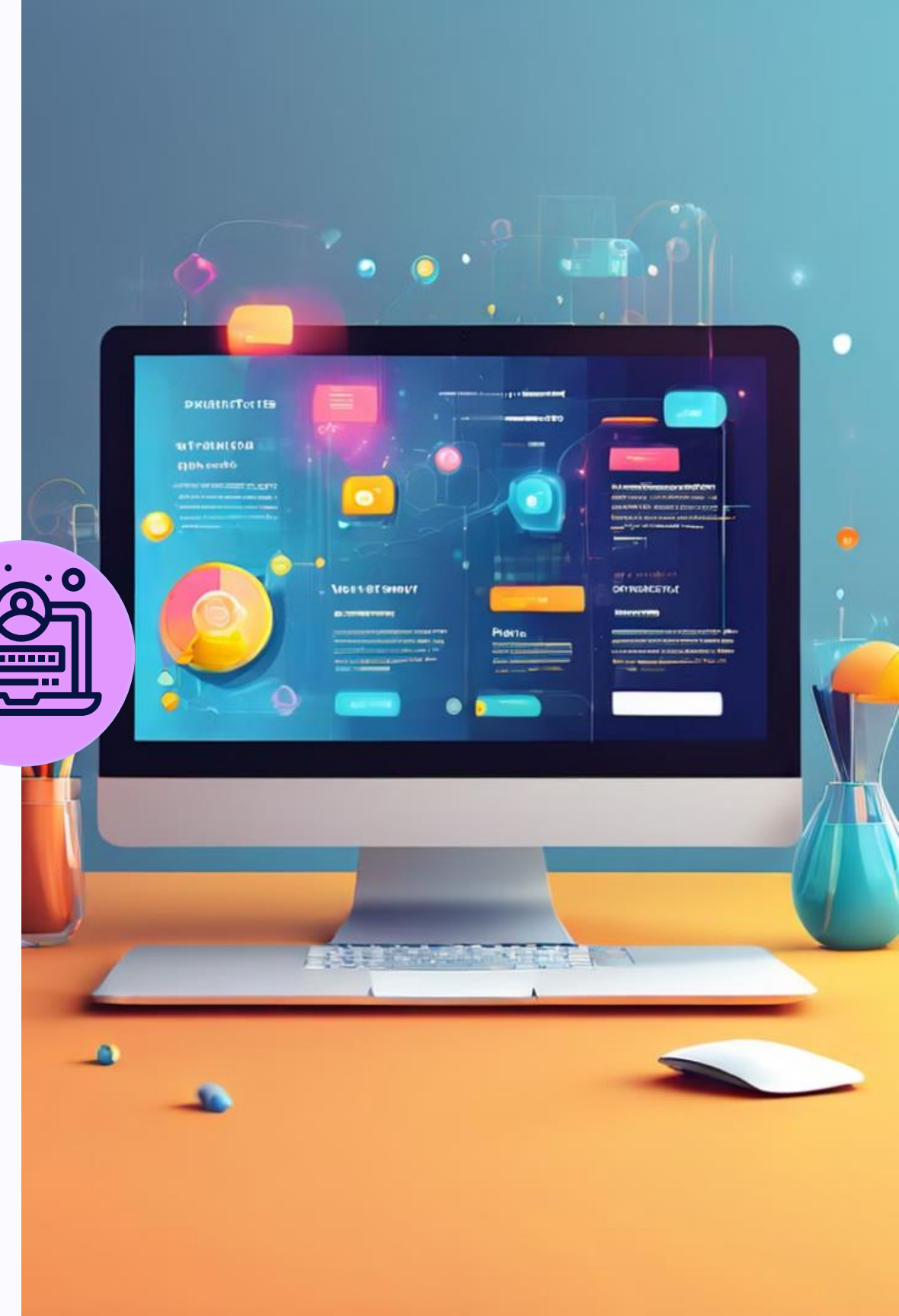
CSS, o Hojas de Estilo en Cascada, se utiliza para definir el estilo visual de una página web. Permite aplicar colores, fuentes y layouts que mejoran la presentación del contenido.

3 JavaScript

JavaScript es un lenguaje de programación que otorga interactividad a las páginas web. Permite crear elementos dinámicos y responder a acciones del usuario en tiempo real.

4 Ejercicio Resuelto

Un ejercicio resuelto es un ejemplo práctico que ilustra el uso de HTML, CSS o JavaScript. Incluye el código correspondiente y explica cada parte para facilitar el aprendizaje.





Estructura básica de una página web

1

HTML

HTML es el lenguaje fundamental para crear una página web. Define la estructura y el contenido, incluyendo encabezados, párrafos y enlaces. Sin HTML, no hay una base sobre la cual construir.

2

CSS

CSS se utiliza para estilizar y diseñar la página. Permite modificar colores, fuentes y layout. Un buen uso de CSS mejora la estética y la usabilidad del sitio web.

3

JavaScript

JavaScript agrega interactividad y dinamismo a la web. Permite crear efectos visuales y manejar eventos de usuario. Sin JavaScript, la interfaz sería estática y menos envolvente.

Etiquetas HTML



Etiquetas Comunes

Las etiquetas HTML son esenciales para crear estructuras en una página web. Cada etiqueta tiene un propósito específico. Las etiquetas comunes incluyen `<div>`, `<p>` y `<a>`.



Estructura Fundamental

La estructura básica de un documento HTML comienza con la etiqueta `<html>`. Dentro, encontrarás las secciones `<head>` y `<body>`. Cada parte juega un papel crucial en la representación del contenido.



Elementos HTML

Los elementos HTML se componen de etiquetas y contenido. Pueden incluir listas, tablas, y enlaces. Comprender cómo se estructuran es clave para dominar HTML.



Atributos HTML

Definición de Atributos

Los atributos son propiedades que se añaden a las etiquetas HTML. Permiten especificar detalles como el comportamiento o la apariencia de los elementos. Cada atributo tiene un nombre y un valor, separados por un signo igual.

Tipos Comunes de Atributos

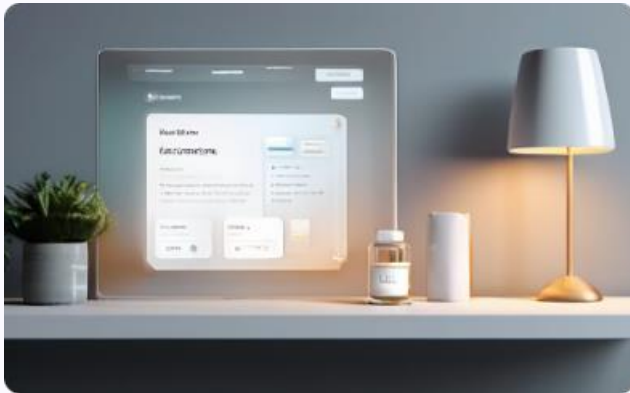
Existen varios atributos comunes como **class**, **id** y **style**. El atributo **class** se utiliza para agrupar elementos, mientras que **id** identifica uno específico. **style** se usa para aplicar CSS directamente.

Uso de Atributos

Los atributos se añaden dentro de las etiquetas de apertura. Es importante no olvidar el uso de comillas para los valores de los atributos. Así, se garantiza que se interpreten correctamente por el navegador.



Elementos HTML



Elementos Estructurales

Los elementos estructurales son fundamentales en HTML. Estos incluyen etiquetas como `<html>`, `<head>` y `<body>`.

Establecen la base para cualquier documento HTML.



Elementos de Texto

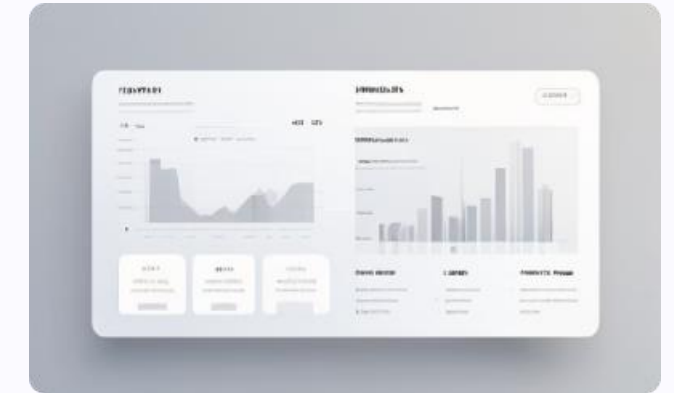
Los elementos de texto permiten dar formato al contenido. Incluyen etiquetas como `<h1>` a `<h6>`, `<p>` y ``. Son esenciales para una buena estructura de la información.



Elementos Multimedia

HTML también permite la inclusión de multimedia. Las etiquetas como `` y `<video>` facilitan este proceso.

Contribuyen a hacer contenido más atractivo y dinámico.



Elementos de Formularios

Los formularios son fundamentales para la interacción con el usuario. Usan etiquetas como `<input>`, `<select>` y `<button>`. Facilitan la recopilación de datos de manera eficiente.



Estilos CSS

¿Qué son los Estilos CSS?

Los estilos CSS son reglas que se aplican para definir la presentación de un documento HTML. Permiten personalizar colores, fuentes y diseño en la web. Con los estilos CSS, se puede lograr un diseño atractivo y funcional fácilmente.



Importancia de CSS

CSS mejora la experiencia del usuario al hacer la información más accesible. Además, proporciona control total sobre el diseño y la disposición. Esto facilita la creación de sitios web que se ven bien en diferentes dispositivos.



Selectores CSS



Selectores de clase

Los selectores de clase se utilizan para aplicar estilos a elementos específicos. Se definen con un punto seguido del nombre. Esto permite dar formato a varios elementos con la misma clase fácilmente.



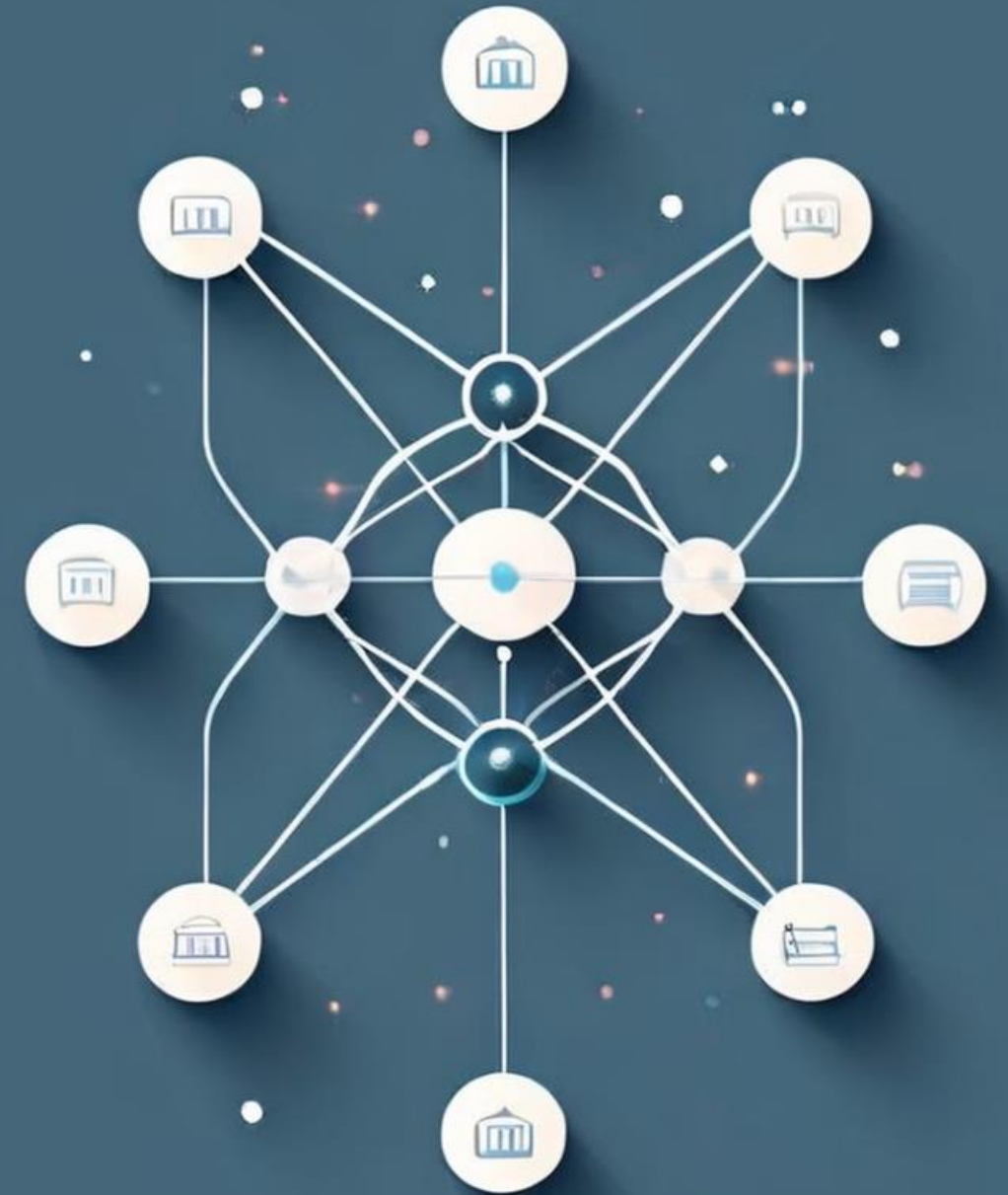
Selectores de ID

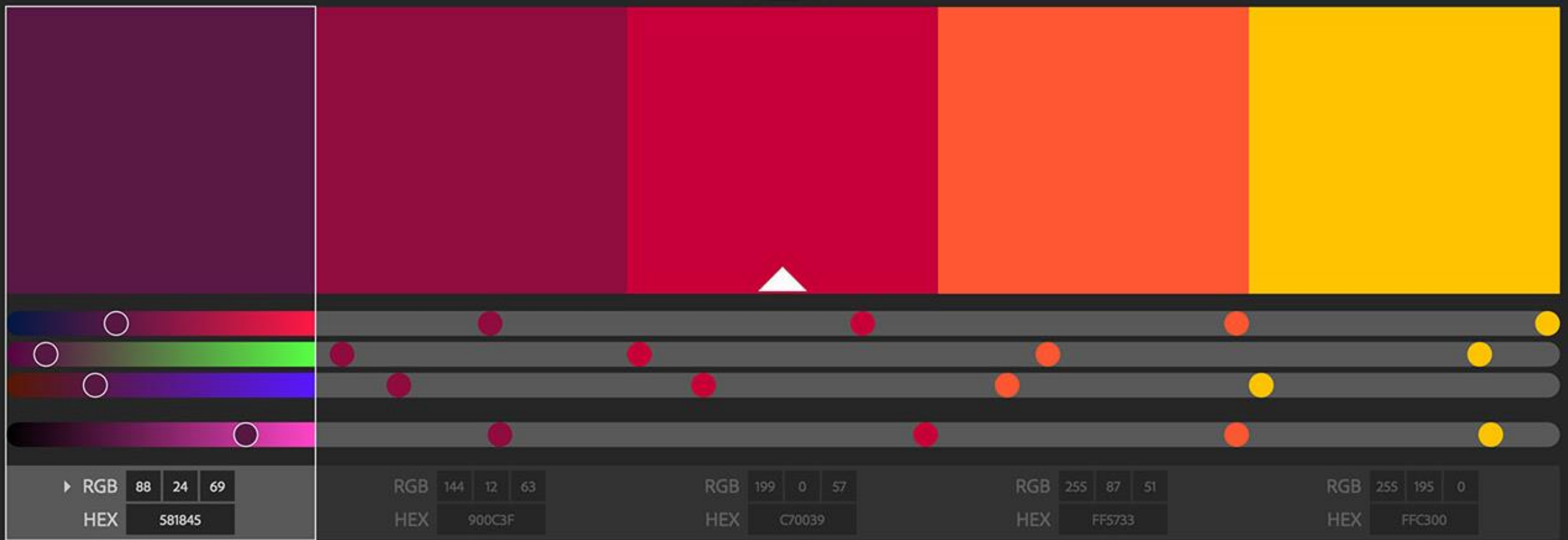
Los selectores de ID son únicos y se definen con un símbolo de almohadilla. Se usan para designar un solo elemento en el documento. Esto garantiza que solo un elemento reciba un estilo particular.



Selectores combinados

Los selectores combinados permiten seleccionar elementos con diferentes relaciones. Estos incluyen selectores de descendencia, adyacentes y de hermanos. Son útiles para aplicar estilos a grupos de elementos relacionados.





Propiedades CSS

Color

El color define el tono visual de un elemento. Puede aplicarse al texto y al fondo. Se puede especificar en diferentes formatos, como RGB, HEX o nombres de colores.

Margin y Padding

Margin controla el espacio exterior de un elemento. Padding, por su parte, define el espacio interior. Ambas propiedades son esenciales para la disposición y el espaciado de los elementos.

Font Size

El tamaño de la fuente afecta la legibilidad del texto. Puede ajustarse en píxeles, em o rem. Elegir un tamaño adecuado es crucial para la experiencia del usuario.

Background

La propiedad de fondo permite personalizar la apariencia de un elemento. Incluye color, imágenes y patrones. Un buen uso de esta propiedad mejora la apariencia general de la página.

Variables en JavaScript



Tipos de datos

JavaScript admite varios tipos de datos. Los más comunes son números, cadenas y booleanos. Comprender los tipos es esencial para el manejo de variables.



Declaración de variables

En JavaScript, puedes declarar variables usando `let`, `const` y `var`. Esto determina su alcance y mutabilidad. Elegir el correcto es crucial para la calidad del código.



Ejemplos de variables

Por ejemplo, puedes definir una variable así: `let nombre = "Juan"`. Esto almacena el nombre como una cadena. Las variables son fundamentales en cualquier programa de JavaScript.





Operadores en JavaScript

Operadores Aritméticos

Los operadores aritméticos son esenciales para realizar cálculos. Incluyen suma, resta, multiplicación y división. Por ejemplo, el operador "+" suma dos números.



Operadores de Comparación

Los operadores de comparación se utilizan para comparar valores. Incluyen "=", "!=" y ">". Estos operadores devuelven valores booleanos, true o false.



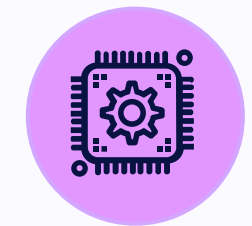
Operadores Lógicos

Los operadores lógicos son importantes para tomar decisiones. Incluyen "&&" (y) y "||" (o). Se utilizan a menudo en estructuras condicionales.



Operadores de Asignación

Los operadores de asignación asignan valores a variables. El operador "=" es el más sencillo. También existen operadores combinados, como "+=" y "-=".





Desarrollo de una página web



El archivo HTML

Define la estructura de tu página web. Incluye un título, un encabezado, una sección principal con un botón, y un pie de página.

```
3  ### 1. index.html
4  ```html
5  <!DOCTYPE html>
6  <html lang="es">
7  <head>
8      <meta charset="UTF-8">
9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10     <title>Mi Primera Página Web</title>
11     <link rel="stylesheet" href="styles.css">
12 </head>
13 <body>
14     <header>
15         <h1>Bienvenido a Mi Primera Página Web</h1>
16     </header>
17     <main>
18         <section id="intro">
19             <p>Esta es mi primera página web utilizando HTML, CSS y JavaScript.</p>
20             <button id="changeTextButton">Haz clic para cambiar el texto</button>
21         </section>
22     </main>
23     <footer>
24         <p>&copy; 2024 Mi Nombre</p>
25     </footer>
26     <script src="scripts.js"></script>
27 </body>
28 </html>
29 ```
```



Desarrollo de una página web

El archivo CSS

Aplica estilos a tu página. Define estilos para el cuerpo, encabezado, botón y pie de página.

```
1  ### 2. styles.css
2  ```css
3  body {
4      font-family: Arial, sans-serif;
5      margin: 0;
6      padding: 0;
7      background-color: #f0f0f0;
8  }
9
10 header {
11     background-color: #4CAF50;
12     color: white;
13     padding: 1em;
14     text-align: center;
15 }
16
17 main {
18     padding: 2em;
19 }
20
21 button {
22     padding: 0.5em 1em;
23     background-color: #008CBA;
24     color: white;
25     border: none;
26     cursor: pointer;
27 }
28
29 button:hover {
30     background-color: #005f6b;
31 }
32
33 footer {
34     background-color: #333;
35     color: white;
36     text-align: center;
37     padding: 1em 0;
38     position: fixed;
39     width: 100%;
40     bottom: 0;
41 }
42 ```
```




Desarrollo de una página web

El archivo JavaScript

Añade interactividad a tu página. Escucha el evento de clic en el botón y cambia el contenido de la sección de introducción cuando se hace clic.

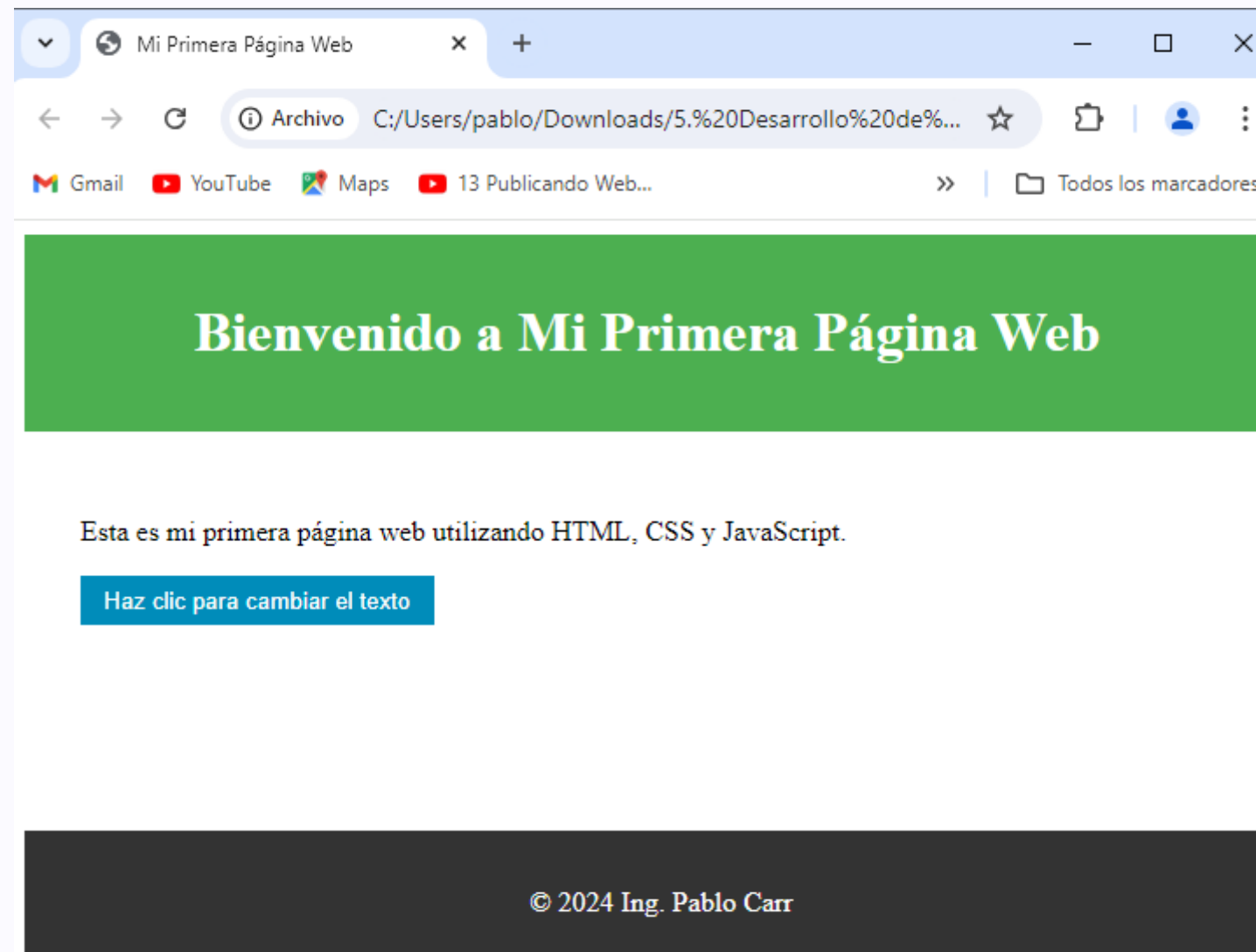
```
1  ### 3. scripts.js
2  ```javascript
3  document.addEventListener('DOMContentLoaded', function() {
4      const button = document.getElementById('changeTextButton');
5      button.addEventListener('click', function() {
6          const introSection = document.getElementById('intro');
7          introSection.innerHTML = '<p>El texto ha cambiado después de hacer clic en el botón!</p>';
8      });
9  });
10 ```
```



Desarrollo de una página web

El Resultado

Si se abre el archivo index.html, en el explorador, se verá tu página así:



Casos de uso

Desarrollo de sitios web



Los casos de uso en el desarrollo de sitios web muestran cómo los usuarios interactúan. Ejemplos incluyen la navegación, búsqueda de información, y compras online. Todos estos se deben considerar en el diseño.

Aplicaciones de software



Las aplicaciones de software también tienen casos de uso claros. Estos ayudan a definir las funciones y características necesarias. Los diagramas de casos de uso pueden ilustrar mejor estas interacciones.

Comercio electrónico



En el comercio electrónico, los casos de uso guían el proceso de compra. Desde el registro hasta el pago, cada paso cuenta. Esto asegura una experiencia de usuario fluida y eficiente.





Recursos adicionales

Libros Recomendados

Existen varios libros que cubren HTML, CSS y JavaScript. Estos textos son ideales para principiantes y ofrecen ejemplos prácticos. Algunas opciones incluyen "Eloquent JavaScript" y "HTML & CSS: Design and Build Websites".



Blogs y Tutoriales

Hay numerosos blogs que ofrecen tutoriales sobre desarrollo web. Sitios como FreeCodeCamp y W3Schools son excelentes para aprender. También puedes encontrar recursos interactivos en Codecademy.



Comunidades en Línea

Participar en comunidades puede mejorar tu aprendizaje. Foros como Stack Overflow permiten hacer preguntas y compartir conocimientos. Grupos en redes sociales como Facebook y Discord también son útiles.



Canales de YouTube

Muchos creadores de contenido ofrecen tutoriales visuales. Canales como Traversy Media y Academind son altamente recomendados. Estos videos son excelentes para aprender conceptos de manera dinámica.





Conclusiones

Resumen de Aprendizajes

Al finalizar este curso, se destaca la integración de HTML, CSS y JavaScript. Cada lenguaje cumple un rol esencial en el desarrollo web. Juntos, crean páginas interactivas y visualmente atractivas.

Importancia de la Práctica

La práctica constante es crucial para consolidar los conocimientos adquiridos. Realizar ejercicios y proyectos ayudará a dominar los conceptos. Así se logra crear páginas web efectivas y funcionales.

Perspectivas Futuras

Con los conceptos básicos dominados, los estudiantes pueden explorar temas avanzados. Aprender sobre frameworks y herramientas modernas expandirá sus habilidades. Esto los preparará para el ámbito profesional en el desarrollo web.



Próximos Pasos

1

Investigación y Práctica

Investiga más recursos sobre HTML, CSS y JavaScript. La práctica continua ayudará a consolidar los conocimientos adquiridos. Realiza ejercicios y proyectos personales para mejorar tus habilidades.

2

Desarrollo de Proyectos

Comienza a desarrollar proyectos sencillos para aplicar lo aprendido. Proyectos como un portafolio personal son ideales para demostrar tus habilidades. Usa plataformas como GitHub para alojar tu código.

3

Participación en la Comunidad

Únete a foros y grupos en línea para compartir conocimientos. La colaboración con otros te ayudará a resolver dudas y aprender rápidamente. Participar en hackathons o talleres puede ser muy enriquecedor.



Ejercicios de Práctica



¡Gracias

Por ser parte de esta
Experiencia de aprendizaje!