

UNIVERSIDAD SERGIO ARBOLEDA

Programación
Nivel Explorador

PREPARACIÓN SEMANA 3

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ HTML (HyperText Markup Language)

- Estructura Básica:
 - HTML define la estructura y contenido de una página web utilizando etiquetas.
 - La estructura básica incluye elementos como `<html>`, `<head>`, `<body>`, etc.
- Etiquetas y Elementos:
 - Las etiquetas `<tag>` se utilizan para estructurar y dar significado al contenido.

Ejemplos comunes son `<p>` para párrafos, `<h1>` para encabezados, `` y `` para listas, etc.

- Atributos:
 - Los atributos agregan información adicional a las etiquetas para definir su comportamiento o apariencia.

Ejemplo: `href` en `` para enlaces, `src` en `` para imágenes.
CSS (Cascading Style Sheets)

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

- Selección de Elementos:
 - CSS se usa para dar estilo y presentación a los elementos HTML.
 - Selecciona elementos usando selectores como etiquetas, clases o IDs.
- Propiedades y Valores:
 - Las propiedades son características que se aplican a los elementos, como color, font-size, background, etc.
 - Los valores especifican cómo se ve la propiedad, como red, 16px, url('imagen.jpg'), etc.
- Hojas de Estilo:
 - Las hojas de estilo CSS se pueden incluir en línea (<style>...</style>), en un archivo externo (<link>), o directamente en el documento HTML (style attribute)
 - Etiquetas y Elementos:
 - Las etiquetas <tag> se utilizan para estructurar y dar significado al contenido.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ Relación entre HTML y CSS:

HTML define la estructura y el contenido, mientras que CSS controla el diseño y la presentación de esos elementos.

CSS se utiliza para cambiar el color, tamaño, disposición y otros aspectos visuales de los elementos HTML.

- Importancia de la Separación de Responsabilidades:
 - Mantener HTML y CSS separados permite una mejor organización, facilita la edición y el mantenimiento del código.
- Diseño Responsivo y Flexibilidad:
 - CSS permite crear diseños responsivos con técnicas como Flexbox, Grid Layout y media queries, adaptando el contenido a diferentes dispositivos.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ ENLACE DE ESTILOS EXTERNOS

- Uso de la Etiqueta <link> para Hojas de Estilo Externas
La etiqueta <link> es una forma fundamental de enlazar hojas de estilo externas a documentos HTML. Se utiliza para especificar la relación entre el documento actual y un recurso externo, como un archivo CSS. Esto permite aplicar estilos definidos en un archivo separado al contenido HTML.

Sintaxis Básica de la Etiqueta <link>

```
<link rel="stylesheet" href="ruta_del_archivo.css">
```

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ ENLACE DE ESTILOS EXTERNOS

- **rel="stylesheet"**: Define la relación del enlace, indicando que se trata de una hoja de estilo.
- **href="ruta_del_archivo.css"**: Especifica la ruta o URL del archivo CSS que se va a enlazar.

Vamos a ver un ejemplo:

Supongamos que tienes un archivo HTML llamado index.html y un archivo CSS llamado styles.css.

La estructura que deben tener los archivos es:

```
proyecto/  
├── index.html  
└── css/  
    └── styles.css
```

El contenido del estilo llamado styles.css es:

```
/* Estilos CSS */  
body {  
    font-family: Arial, sans-serif;  
    background-color: #f0f0f0;  
}  
  
h1 {  
    color: blue;  
}
```

Por tanto el contenido del archivo index.html donde referenciamos el estilo es:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Mi Página</title>  
    <link rel="stylesheet" href="css/styles.css">  
</head>  
<body>  
    <!-- Contenido HTML -->  
</body>  
</html>
```

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ Tips:

- **Estructura de Carpetas:** Mantener un directorio para archivos CSS (css/) ayuda a organizar y estructurar los archivos.
- **Nombres Descriptivos:** Usar nombres significativos para los archivos CSS facilita su identificación y mantenimiento.
- **Rutas Relativas:** Especificar rutas relativas en el atributo href del <link> para asegurarse de que el archivo se enlace correctamente.
- **Ventajas de usar hojas de estilo externas**
- **Reutilización:** Permite reutilizar los estilos en múltiples páginas HTML.
- **Mantenimiento Simplificado:** Cambiar los estilos es más sencillo al tenerlos en un archivo separado.
- **Optimización de la Carga:** Los estilos externos se pueden cachear, lo que puede mejorar el rendimiento del sitio.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

❖ INTEGRACIÓN DE ESTILOS EN LINEA

- Uso del atributo style para aplicar estilos en línea, el atributo style se agrega a una etiqueta HTML y contiene declaraciones de estilo en línea. Estas declaraciones se aplican exclusivamente al elemento al que están asociadas.

Sintaxis

```
<tagname style="property: value; otra-propiedad: valor;">  
    Contenido del Elemento  
</tagname>
```

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

- **property: value;;** Define una propiedad de estilo y su valor para el elemento.
- **otra-propiedad: valor;;** Puedes agregar múltiples propiedades y valores separados por punto y coma.

Veamos un ejemplo

Supongamos que se quiere aplicar un estilo específico al elemento **<p>** en un documento HTML llamado **index.html**

Contenido de index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Estilos en Línea</title>
5      <link rel="stylesheet" href="styles.css">
6  </head>
7  <body>
8      <p style="color: blue; font-size: 18px;">Este es un párrafo con estilos en línea.</p>
9      <p>Este es otro párrafo sin estilos en línea.</p>
10 </body>
11 </html>
```

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

- **Prioridad de los Estilos en Línea**

Los estilos en línea tienen prioridad sobre los estilos externos e internos (definidos en hojas de estilo externas o en el <style> del documento). Esto significa que si hay conflictos entre los estilos en línea y los estilos externos/internos, los estilos en línea prevalecerán.

Veamos un ejemplo

Supongamos que en el archivo **styles.css** tienes un estilo que afecta a todos los párrafos así:

```
/* styles.css */  
p {  
    color: red;  
    font-size: 16px;  
}
```

A pesar de tener un estilo definido en el archivo CSS, el párrafo con el estilo en línea (<p style="color: blue; font-size: 18px;">) (del archivo index.html visto en el ejemplo anterior) mostrará el texto en azul y con un tamaño de fuente de 18px, ya que los estilos en línea tienen prioridad sobre los estilos externos.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

- **FLEXIBILIDAD Y DISEÑO RESPONSIVO**

Recordemos que para tener flexibilidad y diseño responsivo hablamos de dos temas: Flexbox y media queries.

- Las principales propiedades de Flexbox son:
 - ❑ display: flex
 - ❑ flex-direction
 - ❑ justify-content
 - ❑ align-items
 - ❑ flex-wrap
 - ❑ align-content

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

Veamos un ejemplo

Nuestro objetivo es crear una barra de navegación flexible y adaptable

Nuestro archivo index.html tendría una estructura así:

```
<nav class="navbar">
  <a href="#">Inicio</a>
  <a href="#">Productos</a>
  <a href="#">Servicios</a>
  <a href="#">Contacto</a>
</nav>
```

Nuestro archivo index.html tendría una estructura así:

```
.navbar {
  display: flex;
  justify-content: space-around;
  background-color: #333;
  color: #fff;
  padding: 10px 0;
}

.navbar a {
  text-decoration: none;
  color: #fff;
}
```

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

Ahora recordemos la estructura general de Media Queries

```
@media tipo_de_medio and (característica) {  
    /* Reglas de estilo a aplicar */  
}
```

- Donde **@media**: Define una regla de estilo condicional basada en ciertos criterios.
- **tipo_de_medio**: Opcional. Define el tipo de medio al que se aplicará la media query, como screen (pantallas), print (impresión), all (todos los dispositivos), entre otros.
- **característica**: Especifica las condiciones bajo las cuales se aplicarán los estilos, utilizando operadores lógicos como and, not, only, entre otros, junto con parámetros como min-width, max-width, orientation, resolution, etc.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

Por Ejemplo

```
@media tipo_de_medio and (característica) {  
    /* Reglas de estilo a aplicar */  
}
```

- Donde **@media**: Define una regla de estilo condicional basada en ciertos criterios.
- **tipo_de_medio**: Opcional. Define el tipo de medio al que se aplicará la media query, como screen (pantallas), print (impresión), all (todos los dispositivos), entre otros.
- **característica**: Especifica las condiciones bajo las cuales se aplicarán los estilos, utilizando operadores lógicos como and, not, only, entre otros, junto con parámetros como min-width, max-width, orientation, resolution, etc.

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

Por Ejemplo

```
@media screen and (min-width: 768px) {  
  /* Estilos a aplicar cuando el ancho de la pantalla es igual o mayor a 768px */  
}  
  
@media screen and (max-width: 1024px) {  
  /* Estilos a aplicar cuando el ancho de la pantalla es menor o igual a 1024px */  
}  
  
@media only screen and (min-resolution: 192dpi) {  
  /* Estilos a aplicar para pantallas con alta resolución (pantallas retina) */  
}
```

- Veamos otro ejemplo donde nuestro objetivo es Crear una sección principal con un encabezado, un párrafo y una imagen, y deseamos que esta sección se adapte de manera diferente en dispositivos móviles y pantallas más grandes:

REPASO CONCEPTO FUNDAMENTALES HTML Y CSS

Nuestro código se vería así:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Sección Principal Adaptable</title>
5   <style>
6     /* Estilos generales para la sección principal */
7     .main-section {
8       padding: 20px;
9       text-align: center;
10    }
11
12    /* Media Query para dispositivos móviles */
13    @media screen and (max-width: 767px) {
14      .main-section {
15        background-color: lightblue;
16      }
17    }
18
19    /* Media Query para pantallas más grandes */
20    @media screen and (min-width: 768px) {
21      .main-section {
22        background-color: lightgreen;
23      }
24    }
25  </style>
26 </head>
27 <body>
28   <section class="main-section">
29     <h1>Título Principal</h1>
30     <p>Texto descriptivo sobre la sección principal.</p>
31     
32   </section>
33 </body>
34 </html>
```

• Para este ejemplo se ha creado una sección principal (<section class="main-section">) que contiene un título (<h1>), un párrafo (<p>) y una imagen ().

Se han definido estilos generales para la sección principal en el bloque de <style> en el encabezado HTML.

Se han añadido media queries dentro del bloque de estilo para cambiar el color de fondo de la sección (background-color) dependiendo del ancho de la pantalla.

Esto permitirá que la sección principal tenga un color de fondo diferente en dispositivos móviles (ancho de pantalla menor o igual a 767px) y en pantallas más grandes (ancho de pantalla mayor o igual a 768px).

¿Preguntas?