

**PROGRAMA DE ESTUDIOS**

**DESARROLLO DE  
SISTEMAS DE  
INFORMACIÓN  
FRONTEND ORIENTADO  
A SERVICIOS**

**Tema**

**DISEÑO AVANZADO CON FLEXBOX  
PARTE I**

## DISEÑO AVANZADO CON FLEXBOX PARTE I

### 1. flex-direction

La propiedad `flex-direction` define la dirección en la que los elementos hijos de un contenedor flexible (`flex container`) se distribuyen. Los elementos pueden organizarse horizontal o verticalmente en función del valor establecido.

- **Valores:**

- `row`: los elementos se distribuyen de izquierda a derecha, en el sentido de la fila (horizontal). Este es el valor predeterminado.
- `row-reverse`: los elementos se distribuyen de derecha a izquierda.
- `column`: los elementos se organizan de arriba hacia abajo, en el sentido de la columna (vertical).
- `column-reverse`: los elementos se distribuyen de abajo hacia arriba.

**Ejemplo de uso:**

### CSS

```
.flex-container {  
    display: flex;  
    flex-direction: row;  
}
```

## html

```
<div class="flex-container">
  <div>Elemento 1</div>
  <div>Elemento 2</div>
  <div>Elemento 3</div>
</div>
```

### 2. justify-content

La propiedad `justify-content` controla la alineación de los elementos hijos en el eje principal, es decir, la distribución del espacio libre entre los elementos. Su comportamiento varía en función de la dirección definida por `flex-direction`.

- **Valores:**

- `flex-start`: los elementos se alinean al inicio del contenedor.
- `flex-end`: los elementos se alinean al final del contenedor.
- `center`: los elementos se alinean en el centro del contenedor.
- `space-between`: el espacio se distribuye uniformemente entre los elementos, con el primer y último elemento pegados a los bordes.
- `space-around`: el espacio se distribuye uniformemente, dejando un espacio alrededor de cada elemento.

- space-evenly: distribuye el espacio entre los elementos de manera uniforme, incluyendo espacio en los extremos del contenedor.

**Ejemplo de uso:**

**css**

```
.flex-container {  
    display: flex;  
    justify-content: space-between;  
}
```

**html**

```
<div class="flex-container">  
    <div>Elemento 1</div>  
    <div>Elemento 2</div>  
    <div>Elemento 3</div>  
</div>
```

### 3. align-items

La propiedad align-items define cómo se alinean los elementos hijos en el eje transversal del contenedor (es decir, en el eje perpendicular al eje principal definido por flex-direction).

- **Valores:**

- flex-start: los elementos se alinean al inicio del eje transversal.
- flex-end: los elementos se alinean al final del eje transversal.
- center: los elementos se alinean en el centro del eje transversal.

- baseline: los elementos se alinean en la línea base de su contenido (ideal para alineación de textos).
- stretch: los elementos se estiran para llenar el contenedor (es el valor predeterminado).

**Ejemplo de uso:**

**css**

```
.flex-container {  
    display: flex;  
    align-items: center;  
}
```

**html**

```
<div class="flex-container">  
    <div>Elemento 1</div>  
    <div>Elemento 2</div>  
    <div>Elemento 3</div>  
</div>
```

**Ejercicios Prácticos**

**Ejercicio 1: Creación de una Barra de Navegación Centrada**

**Objetivo:** Crear una barra de navegación horizontal donde los elementos se centren horizontal y verticalmente utilizando Flexbox.

**Instrucciones:**

Crea un contenedor nav con elementos de lista ul > li.

Establece display: flex; en el contenedor nav.

Aplica justify-content: center; para centrar los elementos horizontalmente y align-items: center; para centrar verticalmente.

**Ejercicio 2: Galería de Imágenes con Espacio Entre Elementos**

**Objetivo:** Crear una galería donde las imágenes se distribuyen uniformemente con espacio entre ellas y centradas verticalmente.

**Instrucciones:**

Crea un contenedor .gallery con varias imágenes.

Usa display: flex y justify-content: space-between en .gallery.

Configura align-items: center para centrar las imágenes verticalmente.



