

Exercicio 26: Creación dun deseño flexible con Flexbox

Obxectivo: Practicas as propiedades flexbox para o posicionamento de elementos na páxina web.

Pasos a seguir:

Paso 1: Configuración do HTML

1. Crea unha nova carpeta chamada *Exercicio26* e dentro un novo arquivo e gárdao co nome *index.html*.
2. Engade a estrutura básica dun documento HTML e logo engade o seguinte contido:

```
<h1>Practicando con Flexbox</h1>

<div class="contedor">

    <div class="caixa caixa1">1</div>

    <div class="caixa caixa2">2</div>

    <div class="caixa caixa3">3</div>

    <div class="caixa caixa4">4</div>

</div>
```

Paso 2: Estilos CSS básicos

Crea unha carpeta dentro de *Exercicio26* chamada *styles* e un arquivo chamado `styles.css` e engade os seguintes estilos:

```
.contedor {

    border: 3px solid #333;

    padding: 10px;

}

.caixa {

    background-color: #f0f0f0;

    border: 1px solid #999;

    padding: 20px;

    margin: 10px;

    text-align: center;

}
```

Paso 3: Aplicar Flexbox

Agora, imos converter o noso contedor nun contedor flex:

```
.contedor {  
    border: 3px solid #333;  
    padding: 10px;  
    display: flex;  
}
```

Paso 4: Experimentar con flex-direction

Proba diferentes valores para `flex-direction`:

```
.contedor {  
    /* ... */  
    flex-direction: row; /* Proba tamén con: column, row-reverse, column-  
reverse */  
}
```

Paso 5: Usar justify-content

Engade `justify-content` para alinear os elementos no eixe principal:

```
.contedor {  
    /* ... */  
    justify-content: space-between; /* Proba con: flex-start, flex-end, center,  
space-around */  
}
```

Paso 6: Aplicar align-items

Usa `align-items` para alinear os elementos no eixe cruzado:

```
.contedor {  
    /* ... */  
    align-items: center; /* Proba con: flex-start, flex-end, stretch, baseline */  
}
```

Paso 7: Experimentar con flex-grow

Fai que algunhas caixas crezan máis que outras:

```
.caixa2, .caixa4 {  
    flex-grow: 1;  
}
```

```
.caixa3 {  
    flex-grow: 2;  
}
```

Paso 8: Usar flex-basis

Define un tamaño base para os elementos:

```
.caixa {  
    /* ... */  
    flex-basis: 100px;  
}
```

Paso 9: Combinar propiedades con flex

Utiliza a propiedade abreviada `flex` para simplificar:

```
.caixa {  
    /* ... */  
    flex: 1 1 100px; /* flex-grow, flex-shrink, flex-basis */  
}
```

Experimenta con estes pasos e observa como cambia o deseño. Isto axudarache a comprender mellor como funcionan as propiedades flex en CSS.

Entrega: Sube a carpeta *Exercicio26* ao espazo de subida de exercicios en OneDrive de Microsoft Teams.

Ejercicio 25: Creación de un diseño flexible con Flexbox

Objetivo: Practicar las propiedades flexbox para el posicionamiento de elementos en la página web.

Pasos a seguir:

Paso 1: Configuración del HTML

1. Crea una nueva carpeta llamada *Ejercicio25* y dentro un nuevo archivo y guárdalo con el nombre *index.html*.
2. Añade la estructura básica de un documento HTML y luego agrega el siguiente contenido:

```
<h1>Practicando con Flexbox</h1>

<div class="contenedor">

  <div class="caja caja1">1</div>

  <div class="caja caja2">2</div>

  <div class="caja caja3">3</div>

  <div class="caja caja4">4</div>

</div>
```

Paso 2: Estilos CSS básicos

Crea una carpeta dentro de *Ejercicio25* llamada *styles* y un archivo llamado *styles.css* y añade los siguientes estilos:

```
.contenedor {

  border: 3px solid #333;

  padding: 10px;

}

.caja {

  background-color: #f0f0f0;

  border: 1px solid #999;

  padding: 20px;

  margin: 10px;

  text-align: center;

}
```

Paso 3: Aplicar Flexbox

Ahora, vamos a convertir nuestro contenedor en un contenedor flex:

```
.contenedor {  
    border: 3px solid #333;  
    padding: 10px;  
    display: flex;  
}
```

Paso 4: Experimentar con flex-direction

Prueba diferentes valores para flex-direction:

```
.contenedor {  
    /* ... */  
    flex-direction: row; /* Prueba también con: column, row-reverse, column-reverse */  
}
```

Paso 5: Usar justify-content

Añade justify-content para alinear los elementos en el eje principal:

```
.contenedor {  
    /* ... */  
    justify-content: space-between; /* Prueba con: flex-start, flex-end, center, space-around */  
}
```

Paso 6: Aplicar align-items

Usa align-items para alinear los elementos en el eje cruzado:

```
.contenedor {  
    /* ... */  
    align-items: center; /* Prueba con: flex-start, flex-end, stretch, baseline */  
}
```

Paso 7: Experimentar con flex-grow

Haz que algunas cajas crezcan más que otras:

```
.caja2, .caja4 {  
    flex-grow: 1;  
}
```

```
.caja3 {  
    flex-grow: 2;  
}
```

Paso 8: Usar flex-basis

Define un tamaño base para los elementos:

```
.caja {  
    /* ... */  
    flex-basis: 100px;  
}
```

Paso 9: Combinar propiedades con flex

Utiliza la propiedad abreviada flex para simplificar:

```
.caja {  
    /* ... */  
    flex: 1 1 100px; /* flex-grow, flex-shrink, flex-basis */  
}
```

Experimenta con estos pasos y observa cómo cambia el diseño. Esto te ayudará a comprender mejor cómo funcionan las propiedades flex en CSS.

Entrega: Sube la carpeta Ejercicio25 al espacio de subida de ejercicios en OneDrive de Microsoft Teams.