

Instrucciones para completar el CSS con Flexbox (Ejercicio 2)

En este ejercicio se te proporciona un archivo CSS vacío con selectores listos pero sin propiedades definidas.

Tu objetivo es completar las propiedades CSS utilizando Flexbox para conseguir la distribución correcta de los elementos.

A continuación, se explica qué debes hacer en cada sección y por qué.

*

Añade la propiedad 'box-sizing: border-box;' para que el cálculo del ancho y alto de los elementos incluya el padding y el borde. Esto ayuda a evitar problemas de diseño.

.container

Centrar el contenedor horizontalmente usando 'margin: 20px auto;' y definir su ancho con 'width: 80%;'.

header

Convierte el encabezado en un contenedor flexible con 'display: flex;'. Añade también algo de espacio interno con 'padding: 1em;' y un color de fondo para identificarlo visualmente.

header div

Establece el color de fondo y el borde de los elementos internos del header. Agrega padding para darles espacio interno.

header :nth-child(1)

Este elemento debe aparecer en la tercera posición (order: 3) y ocupar un 15% del ancho disponible (flex-basis: 15%).

header :nth-child(2)

Este elemento debe aparecer en la primera posición (order: 1) y ocupar un 15% del ancho (flex-basis: 15%).

header :nth-child(3)

Este debe ir en segunda posición (order: 2), ocupar el 15% del ancho inicial, pero puede crecer más si hay espacio (flex-grow: 1).

header :nth-child(4)

También en segunda posición (order: 2), pero con más crecimiento que el anterior (flex-grow: 2), lo que le da más espacio relativo.

footer

Convierte el pie de página en un contenedor flexible con 'display: flex;', dale altura y espacio interno. El margen superior separa el footer del contenido anterior.

footer div

Define color de fondo, bordes y altura para cada bloque del footer.

footer :nth-child(1)

Este elemento tiene un tamaño base del 20% y crece un poco (flex-grow: 1).

footer :nth-child(2)

Aunque su base también es del 20%, este crecerá mucho más (flex-grow: 10), ocupando el mayor espacio disponible.

footer :nth-child(3)

Igual que el primero: base del 20% y crecimiento limitado (flex-grow: 1).

Aquí hay un ejemplo de como puede quedar.

```
* {  
  
  box-sizing: border-box;  
  
}
```

```
.container {  
  
  margin: 20px auto;  
  
  width: 80%;  
  
}
```

```
header {  
    background-color: #bbb;  
    display: flex;  
    padding: 1em;  
}
```

```
header div {  
    background-color: burlywood;  
    border: 1px solid black;  
    padding: 1em;  
}
```

/ Cambiando el orden de los elementos flexibles */*

```
header :nth-child(1) {  
    order: 3;  
    flex-basis: 15%;  
}
```

```
header :nth-child(2) {  
    order: 1;  
    flex-basis: 15%;  
}
```

```
header :nth-child(3) {  
    order: 2;  
    flex-basis: 15%;  
}
```

```
flex-grow: 1;  
}
```

```
header :nth-child(4) {  
  order: 2;  
  flex-basis: 15%;  
  flex-grow: 2;  
}
```

```
footer {  
  background-color: cadetblue;  
  display: flex;  
  height: 200px;  
  padding: 2em;  
  margin-top: 2em;  
}
```

```
footer div {  
  background-color: burlywood;  
  border: 1px solid black;  
  height: 150px;  
}
```

```
footer :nth-child(1) {  
  flex-basis: 20%;  
  flex-grow: 1;
```

```
}
```

```
footer :nth-child(2) {  
  flex-basis: 20%;  
  flex-grow: 10;  
}
```

```
footer :nth-child(3) {  
  flex-basis: 20%;  
  flex-grow: 1;  
}
```