

Exercicio 35: Deseño responsivo con Grid

Obxectivo: O obxectivo desta práctica é practicar o deseño web responsivo utilizando Grid. Deberás crear unha páxina web sinxela que se adapte de forma axeitada a diferentes tamaños de pantalla, utilizando Grid para organizar os elementos.

Propiedades de grid: CSS Grid é un sistema de deseño bidimensional que permite crear *layouts* complexos de xeito sinxelo. Algunhas das propiedades principais de Grid que podemos aplicar nunha web sinxela son:

1. **display: grid** - Converte un elemento nun contedor grid.
2. **grid-template-columns** - Define o número e tamaño das columnas no grid.
3. **grid-template-rows** - Define o número e tamaño das filas no grid.
4. **gap** - Establece o espazo entre filas e columnas (tamén se pode usar **column-gap** e **row-gap** por separado).
5. **grid-template-areas** - Permite definir áreas nomeadas no grid para organizar os elementos.
6. **grid-auto-rows** - Define o tamaño predeterminado das filas creadas automaticamente.
7. **justify-items** e **align-items** - Controlan a aliñación dos elementos dentro das súas celas.

Pasos a seguir:

1. Estrutura HTML:

- Crea unha carpeta chamada *Exercicio35* e dentro un arquivo HTML chamado `index.html`.
- Deseña a estrutura básica dunha páxina web que conteña un encabezado (`<header>`), unha barra de navegación (`<nav>`), un contido principal (`<main>`), e un pé de páxina (`<footer>`).

2. Estilos Básicos con CSS:

- Crea unha carpeta chamada `styles` e un arquivo CSS chamado `styles.css` e vincúlao co teu arquivo HTML.
- Establece un reset do CSS por defecto dos navegadores e un estilo básico para o corpo (`body`) da páxina, como un fondo, cores e tipografía.

3. Grid para a barra de navegación:

- Usa Grid para alinear os elementos da barra de navegación. Os elementos deben ter espazo uniforme.

4. Contido principal con Grid:

- Na área do contido principal, crea varios elementos (por exemplo, tarxetas ou seccións) que conteñan información ficticia (por exemplo, imaxes e texto).
- Usa Grid para organizar estes elementos en filas ou columnas, segundo o prefiras.
- As tarxetas deben reorganizarse automaticamente nunha única columna cando o tamaño da pantalla sexa pequeno.

5. Pé de páxina:

- Crea un pé de páxina que estea fixo na parte inferior da páxina, independentemente do contido na área principal.
- Estiliza o pé de páxina de maneira que sexa claramente visible e contrastante co fondo do resto da páxina.

Recursos adicionais:

- Podes utilizar imaxes de mostra de sitios web de imaxes gratuítas como [Unsplash](#) ou [Pexels](#) para engadir contido visual á túa páxina.

Entrega: Sube a carpeta *Exercicio35* ao espazo de subida de exercicios en OneDrive de Microsoft Teams.

Ejercicio 35: Diseño responsivo con Grid

Objetivo: El objetivo de esta práctica es practicar el diseño web responsivo utilizando Grid. Deberás crear una página web sencilla que se adapte de forma adecuada a diferentes tamaños de pantalla, utilizando Grid para organizar los elementos.

Propiedades de grid: CSS Grid es un sistema de diseño bidimensional que permite crear *layouts* complejos de manera sencilla. Algunas de las propiedades principales de Grid que podemos aplicar en una web sencilla son:

1. **display: grid** - Convierte un elemento en un contenedor grid.
2. **grid-template-columns** - Define el número y tamaño de las columnas en el grid.
3. **grid-template-rows** - Define el número y tamaño de las filas en el grid.
4. **gap** - Establece el espacio entre filas y columnas (también se puede usar **column-gap** y **row-gap** por separado).
5. **grid-template-areas** - Permite definir áreas nombradas en el grid para organizar los elementos.
6. **grid-auto-rows** - Define el tamaño predeterminado de las filas creadas automáticamente.
7. **justify-items** y **align-items** - Controlan la alineación de los elementos dentro de sus celdas.

Pasos a seguir:

1. **Estructura HTML:**
 - Crea una carpeta llamada Ejercicio35 y dentro un archivo HTML llamado index.html.
 - Diseña la estructura básica de una página web que contenga un encabezado (<header>), una barra de navegación (<nav>), un contenido principal (<main>), y un pie de página (<footer>).
2. **Estilos básicos con CSS:**
 - Crea una carpeta llamada styles y un archivo CSS llamado styles.css y vincúlalo con tu archivo HTML.
 - Establece un reset del CSS por defecto de los navegadores y un estilo básico para el cuerpo (body) de la página, como un fondo, colores y tipografía.
3. **Grid para la barra de navegación:**
 - Usa Grid para alinear los elementos de la barra de navegación. Los elementos deben tener espacio uniforme.
4. **Contenido principal con Grid:**
 - En el área del contenido principal, crea varios elementos (por ejemplo, tarjetas o secciones) que contengan información ficticia (por ejemplo, imágenes y texto).
 - Usa Grid para organizar estos elementos en filas o columnas, según prefieras.
 - Las tarjetas deben reorganizarse automáticamente en una única columna cuando el tamaño de la pantalla sea pequeño.
5. **Pie de página:**
 - Crea un pie de página que esté fijo en la parte inferior de la página, independientemente del contenido en el área principal.
 - Estiliza el pie de página de manera que sea claramente visible y contrastante con el fondo del resto de la página.

Recursos adicionales:

- Puedes utilizar imágenes de muestra de sitios web gratuitos como Unsplash o Pexels para añadir contenido visual a tu página.

Entrega: Sube la carpeta Ejercicio35 al espacio destinado para ejercicios en OneDrive de Microsoft Teams.