CSS Grid para maquetar contenido con rejillas

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Ene - Feb

Carmelo José Jaén Díaz

- iPasen - cjaedia071@g.educaand.es

Índice

OBJETIVO

- Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.
- Utilizar marcos y tablas para presentar la información de manera ordenada.
- Identificar nuevos elementos y etiquetas en HTML5.
- Reconocer las posibilidades de modificar etiquetas HTML.
- Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.

GLOSARIO

Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO

HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa ”modo o estilo en que está impreso un texto” o ”clase de tipos de imprenta”.

INTRODUCCIÓN

CSS Grid nos permite maquetar contenido ajustándolo a cuadrículas o rejillas (grids) totalmente configurables mediante estilos CSS.

CSS GRID

Mediante CSS Grid podemos dividir la página en una rejilla a partir de la cual se pueden posicionar los diferentes elementos de manera muy sencilla. Gracias a los sistemas de maquetación de CSS Grid y CSS Flexbox se pueden crear estructuras con menos código y de una forma más fácil que con los métodos tradicionales.

La diferencia básica entre CSS Grid y CSS Flexbox es que Flexbox se creó para diseños de una dimensión, en una fila o una columna. En cambio CSS Grid Layout se pensó para el diseño bidimensional, en varias filas y columnas al mismo tiempo.

CÓMO UTILIZAR CSS GRID

Para utilizar **CSS Grid** definiremos un contenedor padre y en su interior los ítems que se necesiten crear. Además, se debe definir una serie de propiedades, tanto para el contenedor padre como para las rejillas que contiene. Veamos en detalle las propiedades más importantes.

CÓMO UTILIZAR CSS GRID
Propiedad *display*: grid | inline-grid

Para crear la cuadrícula **grid** hay que definir sobre el elemento contenedor la propiedad `display` y especificar el valor **grid** o **inline-grid**.

```
<!DOCTYPE html>
<html>
<head>
<style>
.contenedor{
  display: grid;
}
</style>
</head>
```

```
<body>
<div class="contenedor"> <!--
contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
</body>
</html>
```

CÓMO UTILIZAR CSS GRID
Propiedad *display*: grid | inline-grid

Los valores **inline-grid** y **grid** indican cómo se comporta el contenedor con respecto al contenido exterior. Con el valor **inline-grid** el contenedor aparece en línea con respecto al contenido exterior. Con el valor **grid**, el contenedor aparece en bloque con respecto al contenido exterior.

Valor	Descripción
`inline-grid`	Cuadrícula en línea con respecto al contenido exterior
`grid`	Cuadrícula en bloque con respecto al contenido exterior

CÓMO UTILIZAR CSS GRID
Propiedad *display*: grid | inline-grid

Código HTML:

```
<h2>display: inline-grid;</h2>
<b>Contenido exterior</b>
<div class="contenedor"> <!--
contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
```

```
<h2>display: grid;</h2>
<b>Contenido exterior</b>
<div class="contenedor2"> <!--
contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
```

CÓMO UTILIZAR CSS GRID
Propiedad *display*: grid | inline-grid

Código CSS:

```
.contenedor{
  display: inline-grid;
}
.contenedor2{
  display: grid;
}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/raBYyNw

display: inline-grid;

Contenido exterior Item 1
Item 2
Item 3
Item 4

display: grid;

Contenido exterior
Item 1
Item 2
Item 3
Item 4

CÓMO UTILIZAR CSS GRID
Propiedad *grid-template colums | rows*

Para definir el tamaño de las columnas y las filas usamos las propiedades `grid-template-colums` y `grid-template-rows`.

Propiedad	Valor	Descripción
`grid-template-columns`	[col1] [col2] ...	Tamaño de cada columna
`grid-template-rows`	[fila1] [fila2] ...	Tamaño de cada fila

CÓMO UTILIZAR CSS GRID
Propiedad *grid-template colums | rows*

Como vemos a continuación, si definimos los siguientes valores crearemos una cuadrícula de 2 filas por 3 columnas.

```
.contenedor {
  display: grid;
  grid-template-rows: 200px 200px;
  grid-template-columns: 200px 200px 200px;
}
```

CÓMO UTILIZAR CSS GRID

Propiedad *grid-template colums | rows*

Podemos usar las **unidades que ya conocemos como los píxeles o los porcentajes**, o utilizar la **unidad fr (unidad fraccional propia del sistema CSS Grid)** que indica la proporción del espacio que ocupará cada elemento. Por ejemplo, si definimos 3 columnas y le asignamos un valor 1fr a cada una, esto repartirá el espacio a partes iguales entre las 3. En otro caso, si definimos el valor 2fr a una de ellas, esa ocupará el doble de espacio que las otras.

CÓMO UTILIZAR CSS GRID

Propiedad *grid-template colums | rows*

Código HTML:

```
<h2> grid-template-columns: auto auto
auto auto;</h2>
<div class="contenedor"> <!--
contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
```

```
<h2> grid-template-columns: 50px 200px
50px 500px;</h2>
<div class="contenedor2"> <!--
contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
```

CÓMO UTILIZAR CSS GRID

Propiedad *grid-template colums | rows*

Código HTML:

```
<h2> grid-template-columns: 1fr 2fr 1fr 3fr;</h2>
<div class="contenedor3"> <!-- contenedor padre-->
  <div class="rejilla">Item 1</div>
<!-- Ítems del grid -->
  <div class="rejilla">Item 2</div>
  <div class="rejilla">Item 3</div>
  <div class="rejilla">Item 4</div>
</div>
```

Código CSS:

```
.contenedor{
  display: grid;
  grid-template-columns: auto auto auto auto;
}
.contenedor2{
  display: grid;
  grid-template-columns: 50px 200px 300px 500px;
}
.contenedor3{
  display: grid;
  grid-template-columns: 1fr 2fr 1fr 3fr;
}
```

CÓMO UTILIZAR CSS GRID
Propiedad *grid-template colums | rows*

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/XJrzMde

grid-template-columns: auto auto auto auto;

Item 1 Item 2 Item 3 Item 4

grid-template-columns: 50px 200px 50px 500px;

Item 1 Item 2 Item 3 Item 4

grid-template-columns: 1fr 2fr 1fr 3fr;

Item 1 Item 2 Item 3 Item 4

CÓMO UTILIZAR CSS GRID
Propiedades *row-gap* y *column-gap*

Podemos definir el espaciado entre las rejillas mediante las propiedades `row-gap` y `column-gap`. El espaciado se crea entre las columnas/filas, no en los bordes exteriores: laterales, superiores e inferiores. Si quieres añadir espacio en los bordes exteriores de la rejilla, puedes usar propiedades como `margin` en el contenedor de la rejilla.

Propiedad	Descripción
`column-gap`	Espaciado entre columnas
`row-gap`	Espaciado entre filas

CÓMO UTILIZAR CSS GRID
Propiedades *row-gap* y *column-gap*

En el siguiente ejemplo, se ha definido una cuadrícula con tres columnas de igual tamaño utilizando la propiedad `grid-template-columns`. Luego, utilizando la propiedad `column-gap` se establece un espacio de 20 píxeles entre las columnas, a su vez, la propiedad `row-gap` establece un espacio de 10 píxeles entre las filas.

También se ha agregado algunos elementos a la cuadrícula y dado un color de fondo y un relleno para facilitar la visión del efecto del espaciado entre las columnas y las filas.

CÓMO UTILIZAR CSS GRID
Propiedades *row-gap* y *column-gap*

Código HTML:

```html
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
</div>
```

Código CSS:

```css
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  column-gap: 20px;
  row-gap: 10px;
}
.grid-item {
  background-color: lightblue;
  padding: 10px;
}
```

CÓMO UTILIZAR CSS GRID
Propiedad *grid-template colums | rows*

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/GgKOmKZ

CÓMO UTILIZAR CSS GRID

Propiedad *grid-gap*

La propiedad `grid-gap` de CSS Grid es una propiedad abreviada para establecer tanto `grid-column-gap` como `grid-row-gap` en una sola declaración. Esta propiedad acepta uno o dos valores, donde el **primer valor** especifica el tamaño del espacio entre las **filas** y el **segundo valor** especifica el tamaño del espacio entre las **columnas**. Si solo se proporciona un valor, se aplica a ambas propiedades. Ejemplo:

```
.grid-container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-gap: 10px 20px;
}
```

CÓMO UTILIZAR CSS GRID

Propiedad *grid-gap*

En este ejemplo, hemos definido una cuadrícula con tres columnas de igual tamaño utilizando la propiedad `grid-template-columns`. Luego, hemos utilizado la propiedad `grid-gap` para establecer un espacio de 10 píxeles entre las filas y un espacio de 20 píxeles entre las columnas.

En esta lección se han visto los primeros pasos para empezar a usar CSS Grid. Además de las propiedades estudiadas, conviene conocer otras muy interesantes para conseguir alinear los elementos, cambiar su tamaño y modificar diferentes aspectos. Si quieres conocer más a fondo este sistema accede a la siguiente guía completa.