

Práctica U2 – CSS Avanzado

Práctica Rules CSS



2º DAW – Diseño de Interfaces Web

Rosa Medina / Arturo Bernal
IES San Vicente
Curso 2025 – 2026

Objetivo

Crear una galería de imágenes responsive que utilice una maqueta de grid asimétrica (estilo "masonry" denso). El enfoque principal será el uso de "Progressive Enhancement": la galería debe funcionar en todos los navegadores, pero debe "mejorar" y usar las funciones experimentales `if()` y `@property` solo si el navegador las soporta, comprobándolo con `@supports`.

¡**Importante!** Es obligatorio usar nesting. Sino usáis nesting, la práctica será **muy penalizada** → 0

El resultado final deberá verse como la captura que se adjunta al final del enunciado.

Requisitos de la práctica

Requisitos:

1. **HTML:** No se puede modificar nada de este archivo.
2. Vamos a organizar nuestros archivos css en un directorio, en él habrá el `main.css` que será el que está enlazado en el html y tendrá los estilos a las etiquetas principales (`*`, `header`, `footer`, `body` y `main`). Tendremos un segundo archivo llamado `gallery.css` que tendrá todos los estilos que le aplicaremos a nuestra galería. En el `main` por tanto deberemos importar nuestro `gallery.css` en una capa/layer teniendo en cuenta que las propiedades que primero se aplicarán serán las de nuestro fichero actual. Si consideras necesario, podrás crearte un layer con los estilos a los elementos principales:
 - Recuerda usar el selector `*` para quitar los valores por defecto del navegador
 - **Body** será un grid donde el `header` esté arriba del todo y el `footer` abajo, sin posiciones absolutas. Al `main` ponle un `margin` para que no esté tan pegado a los lados.
 - **header** y **footer** tienen el mismo color de fondo y de texto, esos valores créalos como variables para reutilizarlos.
 - Crea las variables que consideres necesarias, para el desarrollo de la web.
 - Modifica el `footer` desde el `css` para que por defecto el texto sea que tu navegador no soporta esta funcionalidad
3. **Layout Principal (grid-auto-flow), en gallery.css:**
 - El `.gallery-grid` debe usar `display: grid`.
 - Las columnas deben ser flexibles, para ello usa el método `repeat` con la función `minmax`, donde el tamaño mínimo sea `150px`.
 - Define una altura base para las filas automáticas de `150px`; (Esto ayuda a que los `span 2` sean proporcionales).
 - **Importante** Usa `grid-auto-flow` para que la galería rellene los huecos vacíos.
 - Haz que `.item-wide` ocupe `grid-column: span 2`; y `.item-tall` ocupe `grid-row: span 2`;

4. Imágenes Adaptables:

- **Importante:** Las imágenes deben adaptarse perfectamente al tamaño de su item (.gallery-item) sin deformarse. Para ello haz uso de estas propiedades que veremos en las siguientes sesiones:

```
object-fit: cover;  
display: block;
```

- El .gallery-item debe "recortar" la imagen si no cabe. Usa la propiedad overflow, de tal manera que el sobrante se oculte.

5. Media Query (@media) para pantallas cuya anchura sea menor a 400px: Tu CSS pasará a ser un flex de una sola columna. Para que las imágenes de item-tall no se vean muy diferentes a la sotras ponle la propiedad: `aspect-ratio: 16 / 10;`

6. Implementación experimental:

- Crea la propiedad `--is-featured`, donde sea un integer con un valor inicial a 0 y que se herede.
- Creemos los valores por defecto para aquellos navegadores que no soportan las funciones: dale un borde a cada imagen, una sombra y haz que esa imagen por defecto esté escalada a 1.03. `transform: scale(1.03);`
- Comprueba que tu navegador soporta la sintaxis if, y en el caso de ser cierto:
 - Quita los valores dados en el punto de arriba (pon las tres propiedades a none)
 - Crea una variable para los elementos de la clase gallery-item y dale el valor 0. Modifica esa variable para los elementos de la clase item-featured con el valor 1.
 - Para aquellas imágenes de la gallería dale un borde si tienen la `--is-featured` a 1 y dale otro borde distinto si lo tienen a 0. haz lo mismo con la propiedad transform, si lo tienen ponle un `scale(1,03)`, en el caso de no tenerlo `scale(1)`.
 - Cuando hagas hover sobre las imágenes modifica el scale anterior asignándole un mayor zoom en cada caso, así como modifica la sobra de la imagen.
 - Modifica el footer desde el css para que el texto sea que tu navegador soporta esta funcionalidad experimental
- Importante, verifica tu resultando usando chrome y firefox. Recuerda revisar en [caniuse](#) o en la [documentación oficial](#) desde qué versión está implementado estas funcionalidades que no estés peleándote y sea porque debes actualizar chrome.



Figura 1: Web lanzada con el chrome



Figura 2: Web lanzada con el Firefox