

46 CSS: Variables y funciones

CSS moderno incluye características avanzadas como **variables** (también conocidas como **custom properties**) y diversas funciones que permiten realizar cálculos dinámicos, optimizar recursos externos y simplificar la escritura de estilos.

1. Custom Properties y Función `var()`

Las **custom properties** son variables en CSS que almacenan valores reutilizables. Estas pueden ser definidas dentro de un selector específico o en el selector `:root` para hacerlas accesibles globalmente.

- **Sintaxis:**

```
--nombre-variable: valor;
```

- **Uso con `var()`:**

La función

`var()` permite acceder al valor de una variable.

```
color: var(--nombre-variable, valor-por-defecto);
```

- **Ejemplo:**

```
:root {  
  --primary-color: #3498db;  
  --font-size: 16px;  
}  
  
body {  
  background-color: var(--primary-color);  
  font-size: var(--font-size);  
}
```

- **Características:**

- **Ámbito:** Las variables tienen alcance limitado al selector donde se definen y a sus descendientes.
- **Valor por defecto:** Si una variable no está definida, se puede especificar un valor predeterminado en `var()`.
- **Herencia:** Las variables definidas en `:root` están disponibles globalmente.

2. Función `url()`

La función `url()` se utiliza para cargar recursos externos, como imágenes, fuentes o archivos CSS.

- **Sintaxis:**

```
url(ruta-del-recurso);
```

- **Uso común:**

- Cargar imágenes de fondo:

```
background-image: url('imagen.jpg');
```

- Importar hojas de estilo adicionales:

```
@import url('estilos.css');
```

- **Consideraciones:**

- La ruta debe ser relativa o absoluta según la estructura del proyecto.
- Es esencial para trabajar con fuentes personalizadas y fondos responsivos.

3. Función `calc()`

La función `calc()` permite realizar cálculos aritméticos directamente en los valores CSS.

- **Sintaxis:**

```
calc(expresión);
```

- **Operaciones admitidas:**

- Suma (+)
- Resta (-)
- Multiplicación (*)
- División (/)

- **Ejemplo:**

```
.box {  
  width: calc(100% - 20px); /* Ancho total menos margen */  
  padding: calc(1rem + 2px); /* Padding dinámico */  
}
```

- **Ventajas:**

- Permite combinaciones de unidades diferentes (por ejemplo, % y px).
- Útil para diseños fluidos y responsivos.

4. Funciones `min()` y `max()`

- **Definición:** Las funciones `min()` y `max()` seleccionan el valor mínimo o máximo entre varios valores proporcionados.

- **Sintaxis:**

```
min(valor1, valor2, ...);
max(valor1, valor2, ...);
```

- **Ejemplo:**

```
.container {
  width: min(500px, 100%); /* El ancho será el menor entre 500px y 100% */
  height: max(200px, 50vh); /* La altura será el mayor entre 200px y 50vh */
}
```

- **Casos de uso:**

- Controlar dimensiones máximas o mínimas en layouts responsivos.
- Evitar desbordamientos en elementos con contenido dinámico.

5. Función `clamp()`

- **Definición:** La función `clamp()` combina tres valores: un valor mínimo, un valor ideal (generalmente relativo) y un valor máximo.

- **Sintaxis:**

```
clamp(valor-min, valor-ideal, valor-max);
```

- **Ejemplo:**

```
h1 {
  font-size: clamp(1rem, 5vw, 3rem);
  /* Tamaño de fuente ajustable según el viewport */
}
```

- **Explicación:**

- **Valor mínimo:** El tamaño nunca será menor que este valor.
- **Valor ideal:** Se ajusta dinámicamente según el contexto (por ejemplo, `vw` o `%`).
- **Valor máximo:** El tamaño nunca excederá este valor.

- **Ventajas:**

- Ideal para textos responsivos y layouts fluidos.
- Combina las ventajas de `min()` y `max()` en una sola función.

Las **custom properties** y las **funciones en CSS** simplifican y optimizan el diseño web. Las variables (`var()`) permiten reutilizar valores y mantener consistencia en el proyecto, mientras que las funciones como `url()`, `calc()`, `min()`, `max()` y `clamp()` ofrecen flexibilidad y control sobre los estilos.

- **Custom Properties:** Perfectas para gestionar colores, tipografías y otros valores globales.
- **Función `url()`:** Indispensable para cargar recursos externos.
- **Función `calc()`:** Útil para cálculos dinámicos y combinaciones de unidades.

- **Funciones** `min()` y `max()` : Ideales para establecer límites en dimensiones.
- **Función** `clamp()` : Excelente para crear diseños responsivos y fluidos.