

Guía para Ordenar una Hoja de Estilos CSS

Ordenar tu hoja de estilos mejora la legibilidad, el mantenimiento y la colaboración en proyectos. Aquí tienes una estructura básica con descripciones y ejemplos para cada sección, además de buenas prácticas y recursos.

1. Reset o Normalize

Descripción:

Comienza con un "reset" o "normalize" para establecer una base uniforme de estilos en todos los navegadores. Esto ayuda a evitar inconsistencias.

Ejemplo:

```
/* Reset básico */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Normalize */
body {
  line-height: 1.5;
  font-family: Arial, sans-serif;
}
```

2. Estilos Globales

Descripción:

Aplica estilos generales para elementos comunes como `body`, encabezados, párrafos y listas.

Ejemplo:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f5f5f5;
  color: #333;
  margin: 0;
  padding: 16px;
}

h1, h2, h3 {
  color: #00509e;
}

ul {
  list-style: none;
  padding: 0;
}
```

3. Estilos de Estructura

Descripción:

Define los estilos relacionados con el diseño estructural del sitio: contenedores, `header`, `main`, `footer`.

Ejemplo:

```
header, footer {
  background-color: #00509e;
  color: white;
  text-align: center;
  padding: 16px;
}

main {
  max-width: 1200px;
  margin: 0 auto;
  padding: 16px;
}
```

4. Componentes

Descripción:

Estiliza componentes reutilizables como botones, tarjetas (`cards`), y formularios.

Ejemplo:

```
button {
  background-color: #fcd34d;
  color: white;
  border: none;
  padding: 8px 16px;
  cursor: pointer;
}

button:hover {
  background-color: #e63946;
}
```

5. Modificadores y Utilidades

Descripción:

Incluye clases utilitarias que aplican estilos específicos y modificadores para componentes.

Ejemplo:

```
/* Clases utilitarias */
.text-center {
  text-align: center;
}

.margin-bottom {
  margin-bottom: 16px;
}
```

Tips de Buenas Prácticas

1. **Mantén un orden lógico:** Desde estilos generales a específicos.

2. **Usa nombres descriptivos:** Nombres de clases deben ser claros y relacionados con su propósito.
3. **Evita IDs en CSS:** Los IDs tienen alta especificidad, lo que complica sobrescribirlos.
4. **Comentarios:** Documenta tu código con comentarios para explicar secciones o estilos complejos.
5. **Reutiliza estilos:** Aplica clases globales para mantener consistencia.