

53 CSS: Estilos de Formularios

Los formularios son elementos interactivos esenciales en el diseño web, ya que permiten a los usuarios enviar información. Sin embargo, los navegadores suelen aplicar estilos predeterminados poco atractivos a estos elementos. Con CSS, podemos personalizar y mejorar la apariencia de los formularios para hacerlos más funcionales y visualmente atractivos.

1. Estructura Básica de un Formulario

Un formulario típico incluye varios elementos como:

- **Inputs** (`<input>`): Para campos de texto, contraseñas, correos electrónicos, etc.
- **Textareas** (`<textarea>`): Para áreas de texto multilínea.
- **Botones** (`<button>` o `<input type="submit">`): Para enviar el formulario.
- **Selectores** (`<select>`): Para listas desplegables.
- **Checkbox y Radio Buttons**: Para opciones múltiples o únicas.

Ejemplo básico:

```
<form class="contact-form">
  <input type="text" placeholder="Nombre" required>
  <input type="email" placeholder="Correo" required>
  <textarea placeholder="Comentarios"></textarea>
  <button type="submit">Enviar</button>
</form>
```

2. Propiedades Básicas para Estilizar Formularios

CSS permite aplicar estilos a cada elemento del formulario:

- **Ancho y Altura:**
 - Usa `width` y `height` para controlar las dimensiones.
 - Ejemplo:

```
input, textarea {
  width: 100%;
  height: 40px;
}
```

- **Espaciado Interno (Padding):**
 - Mejora la legibilidad y la experiencia del usuario.
 - Ejemplo:

```
input, textarea {
  padding: 10px;
}
```

```
}
```

- **Bordes:**

- Personaliza los bordes para dar un aspecto moderno.
- Ejemplo:

```
input, textarea, button {  
  border: 1px solid #ccc;  
  border-radius: 5px;  
}
```

- **Fondo y Color de Texto:**

- Usa `background-color` y `color` para mejorar la apariencia.
- Ejemplo:

```
input, textarea {  
  background-color: #f9f9f9;  
  color: #333;  
}
```

3. Espaciado entre Elementos

Para evitar que los elementos del formulario se vean demasiado juntos, usa márgenes (`margin`).

- **Ejemplo:**

```
input, textarea, button {  
  margin-bottom: 15px;  
}
```

4. Pseudoclases Especiales

Las pseudoclases permiten aplicar estilos basados en el estado del formulario.

- **`:focus`:**

- Aplica estilos cuando un elemento tiene el foco.
- Ejemplo:

```
input:focus {  
  border-color: blue;  
  outline: none;  
}
```

- **`:valid` y `:invalid`:**

- Aplica estilos según si el contenido es válido o no.

- Ejemplo:

```
input:valid {  
  border-color: green;  
}  
  
input:invalid {  
  border-color: red;  
}
```

- **:placeholder-shown** :

- Aplica estilos cuando el marcador de posición está visible.
- Ejemplo:

```
input:placeholder-shown {  
  background-color: #eaeaea;  
}
```

5. Botones de Envío

Los botones de envío son un punto focal en los formularios. Personalízalos para que sean llamativos y accesibles.

- Ejemplo:

```
button {  
  background-color: #007bff;  
  color: white;  
  border: none;  
  padding: 10px 20px;  
  border-radius: 5px;  
  cursor: pointer;  
}  
  
button:hover {  
  background-color: #0056b3;  
}
```

6. Selectores y Checkbox/Radio Buttons

Estos elementos son más difíciles de personalizar debido a sus características nativas. Sin embargo, puedes usar técnicas avanzadas como ocultar el elemento original y reemplazarlo con un diseño personalizado.

- Ejemplo para Checkbox:

```
input[type="checkbox"] {  
  display: none;
```

```

}

label {
  position: relative;
  padding-left: 25px;
  cursor: pointer;
}

label::before {
  content: '';
  position: absolute;
  left: 0;
  top: 0;
  width: 18px;
  height: 18px;
  border: 1px solid #ccc;
  background-color: white;
}

input[type="checkbox"]:checked + label::before {
  background-color: #007bff;
  border-color: #007bff;
}

```

7. Responsive Design

Asegúrate de que los formularios sean responsivos para adaptarse a diferentes dispositivos.

- Usa porcentajes (`%`) para el ancho de los elementos.
- Ajusta los tamaños de fuente con unidades relativas como `rem`.
- **Ejemplo:**

```

.contact-form {
  max-width: 600px;
  margin: 0 auto;
}

input, textarea {
  font-size: 1rem;
}

```

8. Validaciones Visuales

Usa colores y estilos para proporcionar retroalimentación visual al usuario.

- **Ejemplo:**

```
input:invalid {  
  border: 2px solid red;  
}  
  
input:valid {  
  border: 2px solid green;  
}
```
