



# Estilos en formularios CSS, propiedades necesarias

- 
- C.F.G.S. DAW
  - 6 horas semanales
  - Mes aprox. de impartición: Nov - Dic
  - iPasen - [cjaedia071@g.educaand.es](mailto:cjaedia071@g.educaand.es)

Carmelo José Jaén Díaz

# Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

# 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

---



En el Capítulo 1:HTML estudiamos las etiquetas HTML para la creación de formularios.

Vamos a ver ahora cómo podemos dar estilo a esas etiquetas mediante CSS.



# ESTILOS CSS EN FORMULARIOS

---



Gracias a los estilos podemos romper el diseño gris con líneas negras de los formularios y convertirlos en una parte más que se integra perfectamente en nuestro diseño web.

Como sabemos, a la hora de crear nuestros formularios, seguiremos buenas prácticas CSS, no crearemos clases extra innecesarias y utilizaremos, en la medida de lo posible, selectores descendientes y etiquetas predefinidas.

A las etiquetas de los formularios se les puede dar estilo CSS al igual que al resto de elementos de nuestro documento. Sin embargo, debemos conocer una serie de propiedades que nos ayuden a elaborar nuestros diseños de forma óptima.

# PROPIEDAD *box-sizing*

---



Por defecto en el modelo de cajas de CSS, el ancho y alto asignado a un elemento es aplicado solo al contenido de la caja del elemento. Si el elemento tiene algún borde (border) o relleno (padding), este es entonces añadido al ancho y alto del tamaño de la caja o contenedor. Esto significa que cuando se define el ancho y alto, se tiene que ajustar el valor para permitir cualquier borde o relleno que se pueda añadir.

La propiedad `box-sizing` puede ser usada para ajustar el siguiente comportamiento:

- `content-box` es el comportamiento CSS por defecto para el tamaño de la caja (box-sizing). Si se define el ancho de un elemento en 100 pixeles, la caja del contenido del elemento tendrá 100 pixeles de ancho, y el ancho de cualquier borde o relleno será añadido al ancho final desplegado.
- `border-box` tiene en cuenta cualquier valor que se especifique de borde o de relleno para el ancho o alto de un elemento. Es decir, si se define un elemento con un ancho de 100 pixeles. Esos 100 pixeles incluirán cualquier borde o relleno que se añada, y la caja de contenido se encogerá para absorber ese ancho extra. Esta propiedad es especialmente útil para redimensionar cualquier elemento.



# PROPIEDAD *box-sizing*

---



**Ejemplo:** en el caso de los inputs, normalmente les añadimos un padding para darle algo de aire. Sin embargo, si después centramos el contenido y le dotamos de un «width:100%», este ancho no tendrá en cuenta el padding añadido al input y el elemento no se encontrará perfectamente redimensionado.

Realiza la prueba quitando y añadiendo la propiedad «box-sizing: border-box» en el código del formulario básico creado a continuación.

# PROPIEDAD *box-sizing*

---



Código HTML:

```
<form>
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <input type="submit" value="Enviar">
</form>
```

# PROPIEDAD *box-sizing*

---



Código CSS:

```
input {  
  -webkit-box-sizing: border-box;  
  box-sizing: border-box;  
  padding: 10px;  
  width: 100%;  
  margin-bottom: 10px;  
  border: 1px solid #ccc;  
  border-radius: 5px;}
```

```
form{  
  max-width: 500px;  
  margin: 0 auto;  
  background-color: white;  
  padding: 40px;}
```

```
body{background-color: #EEE;}
```

Como puedes ver en el código, se han añadido los prefijos para navegadores necesarios para esta nueva propiedad.

```
-webkit-box-sizing: border-box;  
box-sizing: border-box;
```

# PROPIEDAD *box-sizing*

---



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/OPLJXWm>

- Con «box-sizing: border-box»

Nombre:

Email:

Enviar

- Sin «box-sizing: border-box»

Nombre:

Email:

Enviar

# PROPIEDAD *resize*

---



Por defecto, los elementos `<textarea>` permiten cambiar el tamaño por el usuario. Podemos anular este comportamiento utilizando la propiedad `«resize:none»`.

Valores: none | both | horizontal | vertical

Valores	Descripción
<b>none</b>	No permite cambiar el tamaño del elemento
<b>horizontal</b>	Permite cambiar el tamaño del elemento horizontalmente
<b>vertical</b>	Permite cambiar el tamaño del elemento verticalmente
<b>both</b>	Permite cambiar el tamaño del elemento en horizontal y vertical

# PROPIEDAD *resize*

---



En este ejemplo, el `<textarea>` no permite que el usuario cambie su tamaño. Puedes ajustar la propiedad `resize` según tus necesidades.

Código HTML:

```
<h2>Ejemplo de Textarea con resize</h2>
<textarea placeholder="Escribe aquí...">
</textarea>
```

Código CSS:

```
textarea {
  width: 300px;
  height: 150px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  resize: none;
  /* Evita que el usuario pueda
  redimensionar el textarea */
}
```

# PROPIEDAD *resize*

---



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/QwLWEQx>

## Ejemplo de Textarea con **resize**

Escribe aquí...

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

---



Cuando un usuario interactúa con un *input*, ya sea para ingresar texto o seleccionar una opción, es importante proporcionar retroalimentación visual para mejorar la experiencia del usuario. Una forma de lograr esto es aplicando estilos específicos cuando el *input* está en estado *:focus*.

La pseudo-clase *:focus* se utiliza para aplicar estilos a un elemento cuando recibe el foco de atención. Por ejemplo, cuando un usuario hace clic en un *input*, este entra en estado *:focus*. Veamos algunas técnicas comunes para estilizar *inputs* en estado *:focus*



# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Eliminar el contorno predeterminado

---



Por defecto, los navegadores suelen agregar un contorno alrededor del *input* cuando está en estado *:focus*. Este contorno puede no ajustarse al diseño de tu sitio web o aplicación. Para eliminarlo, puedes usar la propiedad *outline*

```
input:focus {  
    outline: none;  
}
```

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Estilizar el fondo y el borde



Puedes personalizar el fondo y el borde del input cuando está en estado `:focus` para resaltar su estado activo. Por ejemplo:

```
input:focus {  
    background-color: #f0f0f0;  
    border: 1px solid #ccc;  
}
```

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Animaciones y transiciones



Agregar animaciones o transiciones sutiles puede hacer que la interacción con los inputs sea más atractiva. Por ejemplo, puedes animar el cambio de color del borde cuando el input está en estado *:focus*

```
input {  
    transition: border-color 0.3s ease;  
}  
  
input:focus {  
    border-color: #007bff;  
}
```

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Ejemplo



En el siguiente ejemplo se muestra cómo aplicar estilos a un *input* cuando está en estado *:focus* y cómo incluir una transición para el color del borde.

Código HTML:

```
<h2>Estilos en el input: pseudo-clase :focus</h2>

<div class="input-container">
  <label for="example">Ejemplo input:</label>
  <input type="text" id="example" name="example">
</div>
```

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Ejemplo



### Código CSS:

```
/* Estilo base para el input */
.input-container {
    margin-bottom: 20px;
}

input {
    padding: 10px;
    width: 100%;
    border: 1px solid #ccc;
    border-radius: 5px;
    transition: border-color 0.3s ease; /*
Transición para el color del borde */
    box-sizing: border-box;
}
```

```
/* Estilos para el input cuando está en
estado :focus */
input:focus {
    background-color: #f0f0f0;
    border-color: #007bff; /* Cambio de
color del borde */
    outline: none; /* Eliminar contorno
predeterminado */
}
```

# ESTILOS EN EL INPUT PSEUDO-CLASE *:focus*

## Ejemplo



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/qbYOKJa>

**Estilos en el input: pseudo-clase  
:focus**

Ejemplo input:

# ESTILOS PARA INPUTS

---



Los estilos en los inputs pueden mejorar significativamente la estética y la experiencia de usuario de tus formularios. Veamos algunos estilos creativos que puedes aplicar para personalizar tus inputs:

# ESTILOS PARA INPUTS

## Estilo básico

---



El estilo básico es simple y funcional, ideal para una apariencia clásica y limpia.

Código HTML:

```
<input type="text" class="basic-input"
placeholder="Nombre">
```

Código CSS:

```
.basic-input {
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 10px;
    width: 250px;
}
```



# ESTILOS PARA INPUTS

## Bordes redondeados

---



Los bordes redondeados pueden hacer que tus inputs se vean más suaves y modernos.

Código HTML:

```
<input type="text" class="rounded-input"
placeholder="Correo electrónico">
```

Código CSS:

```
.rounded-input {
    border: 2px solid #3498db;
    border-radius: 20px;
    padding: 12px;
    width: 300px;
}
```

# ESTILOS PARA INPUTS

## Sombras

---



Aplicar sombras al fondo del input puede agregar profundidad y estilo.

Código HTML:

```
<input type="text" class="shaded-input"
placeholder="Contraseña">
```

Código CSS:

```
.shaded-input {
    background-color: #f2f2f2;
    border: none;
    padding: 10px;
    box-shadow: 0 2px 5px rgba(0, 0, 0,
0.1);
    width: 280px;
}
```



Contraseña

# ESTILOS PARA INPUTS

## Hover y focus

---



Agregar efectos de hover y focus puede mejorar la interactividad y la retroalimentación para el usuario.

Código HTML:

```
<input type="text"  
class="hover-focus-input"  
placeholder="Teléfono">
```

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/NPKWBPj>

Código CSS:

```
.hover-focus-input {  
  border: 1px solid #ccc;  
  padding: 10px;  
  transition: border-color 0.3s;  
  width: 250px;  
}  
  
.hover-focus-input:hover,  
.hover-focus-input:focus {  
  border-color: #66afe9;  
}
```

# ESTILOS PARA INPUTS

## Iconos en inputs

---



Incorporar iconos o etiquetas dentro del input puede mejorar la usabilidad y el diseño general.

En este ejemplo, estamos utilizando los iconos de Google Material Icons. Asegúrate de que el enlace a los iconos de Google Material Icons esté correctamente incluido en la sección `<head>` de tu documento HTML.

Código HTML:

```
<h2>Ejemplo de iconos en inputs (Google
Material Icons)</h2>
<div class="input-container">
  <i class="material-icons
input-icon">person</i>
  <input type="text" class="icon-input"
placeholder="Nombre">
</div>

<div class="input-container">
  <i class="material-icons
input-icon">email</i>
```

```
  <input type="text" class="icon-input"
placeholder="Correo electrónico">
</div>

<div class="input-container">
  <i class="material-icons
input-icon">lock</i>
  <input type="password"
class="icon-input"
placeholder="Contraseña">
</div>
```

# ESTILOS PARA INPUTS

## Iconos en inputs

---



### Código CSS:

```
.input-container {  
  position: relative;  
  margin-bottom: 20px;  
}  
.icon-input {  
  padding: 10px 15px 10px 45px;  
  border: 2px solid #2196F3;  
  border-radius: 25px;  
  width: 300px;  
  font-size: 16px;  
  color: #333;  
  background-color: #f7f7f7;  
  outline: none;  
}
```

```
.input-icon {  
  position: absolute;  
  top: 50%;  
  left: 15px;  
  transform: translateY(-50%);  
  color: #777;  
  font-size: 20px;  
}
```

### Ejemplo de iconos en inputs (Google Material Icons)

Nombre

Correo electrónico

Contraseña

# ESTILOS PARA INPUTS

## Animaciones en inputs

---



La inclusión de animaciones sutiles puede elevar la interactividad y el atractivo visual de tus inputs. Un ejemplo claro es cuando aplicamos una transición al hacer foco en el input, expandiéndolo ligeramente para enfatizar su estado activo.

Código HTML:

```
<input type="text" class="animated-input"
placeholder="Buscar">
```

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/EaYxpyr>

```
.animated-input {
  padding: 10px;
  width: 250px;
  border: 1px solid #ccc;
  transition: width 0.3s;
}
.animated-input:focus {
  width: 300px;
}
```

# ESTILOS PARA INPUTS REQUERIDOS

## Asterisco en label



Veamos ahora cómo aplicar estilos específicos a los *inputs* que son marcados como requeridos en el formulario. Esto se hace mediante la adición de un indicativo, por ejemplo un asterisco, al final de las etiquetas `<label>` correspondientes a los campos obligatorios. Estos estilos ayudan a resaltar visualmente los campos que deben ser completados por el usuario.

En el siguiente ejemplo se usa el pseudo-elemento `:after` y la propiedad `content` para insertar un asterisco al final de los `label` que tienen la clase `required`.

# ESTILOS PARA INPUTS REQUERIDOS

## Asterisco en label



Código HTML:

```
<form>
  <div>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre">
  </div>
  <div>
    <label for="email"
class="required">Correo
Electrónico:</label>
    <input type="email" id="email">
  </div>
```

```
<div>
  <label for="mensaje"
class="inline">Mensaje:</label>
  <textarea id="mensaje" rows="4"
cols="50"></textarea>
</div>
<button type="submit">Enviar</button>
</form>
```



# ESTILOS PARA INPUTS REQUERIDOS

## Asterisco en label

---



### Código CSS:

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f2f2f2;  
  margin: 0;  
  padding: 20px;  
}  
  
form {  
  background-color: #fff;  
  border-radius: 10px;  
  padding: 20px;  
  box-shadow: 0 0 10px rgba(0, 0, 0,  
0.1);  
}
```

```
label {  
  display: block;  
  font-weight: bold;  
  margin-bottom: 5px;  
  color: #333;  
  font-size: 16px;  
  padding-bottom: 5px;  
  font-family: 'Arial', sans-serif;  
}  
  
label.required::after {  
  content: "*";  
  color: red;  
  margin-left: 5px;  
}
```

# ESTILOS PARA INPUTS REQUERIDOS

## Asterisco en label

---



### Código CSS:

```
label.inline {
  display: inline-block;
  margin-right: 10px;
}
input[type="text"],
input[type="email"],
textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}
```

```
input[type="text"]:focus,
input[type="email"]:focus,
textarea:focus {
  border-color: #007bff;
  outline: none;
}
button {
  padding: 10px 20px;
  background-color: #007bff;
  border: none;
  color: #fff;
  border-radius: 5px;
  cursor: pointer;
  font-size: 16px;
}
button:hover {
  background-color: #0056b3;
}
```

# ESTILOS PARA INPUTS REQUERIDOS

## Asterisco en label



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/KwPKxKJ>

Nombre:

Correo Electrónico: \*

Mensaje:

Enviar