

01-02 CSS (by bluuweb)

- **HTML**: es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página.
- **CSS**: es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.
- **JS**: es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás. (Está bien, no todo, pero es sorprendente lo que puedes lograr con unas pocas líneas de código JavaScript).



CSS

- CSS (en inglés Cascading Style Sheets) Hojas de estilo en Cascada.
- Es el responsable de todo lo visual de nuestro sitio web.
- CSS describe cómo deben mostrarse los elementos HTML.
- Es un lenguaje de hojas de estilo, es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

3 Formas de incorporar CSS

```
<h1 style="color: peru;">Título del sitio web</h1>
```

```
<head>
  <style>
    h1 {
      color: peru;
    }
  </style>
</head>
```

```
<link rel="stylesheet" href="style.css" />
```

Archivos separados

Como estamos trabajando con un lenguaje diferente a HTML podemos ordenar nuestro código en archivos con la extensión **.css**

Dentro del **<head>** escribe lo siguiente:

```
<link rel="stylesheet" href="style.css" />
```

Crea el archivo en cuestión `style.css` Y ahora escribamos nuestras primeras líneas de código CSS.

```
h1 {  
  font-size: 50px;  
}
```

Ten en cuenta que ahora todos los `<h1>` que crees tendrás los mismos estilos.

Sintaxis

Anatomía de una regla en CSS:



- El selector apunta al elemento HTML que desea diseñar.
- El bloque de declaración contiene una o más declaraciones separadas por punto y coma.
- Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos puntos.
- Múltiples declaraciones CSS están separadas por punto y coma, y los bloques de declaración están rodeados por llaves.

Nota las otras partes importantes de la sintaxis:

- Cada una de las reglas (aparte del selector) deben estar encapsuladas entre llaves `{ }`.
- Dentro de cada declaración, debes usar los dos puntos `:` para separar la propiedad de su valor.
- Dentro de cada regla, debes usar el punto y coma `;` para separar una declaración de la siguiente.

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

Esto pasa porque se pueden sobre-escribir los estilos para un determinado elemento.

Por lo tanto, un estilo en línea tiene la máxima prioridad y anulará los estilos externos e internos y los valores predeterminados del navegador. [Más información](#)

Selectores

- [Todos los selectores](#)

Selector de etiqueta: Estilos generales: Ya que todos nuestros párrafos tendrás el mismo diseño.

```
p {  
  text-align: center;  
  color: red;  
}
```

ID (no se recomienda en la práctica): Cuidado: Los ID son únicos por cada documento HTML por lo tanto no puede haber dos o más elementos HTML con el mismo ID.

```
#mi_id_unico {  
  text-align: center;  
  color: red;  
}
```

```
<p id="mi_id_unico">Lorem ipsum dolor sit amet</p>
```

Clase: Ventaja: Podemos agregar esa clase a más de un elemento HTML.

```
.mi-clase {  
  background-color: #e8505b;  
  color: #f3ecc2;  
}
```

```
<button class="mi-clase">Mi botón</button>
```

Cascada

- [Comprender la Cascada](#)

1. En un primer nivel de simplicidad, la cascada en las hojas de estilo significa que **el orden de las reglas importa en CSS**: cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

```
h1 {  
  color: red;  
}  
h1 {  
  color: blue;  
}
```

2. **Especificidad**: La especificidad es el modo que tiene el navegador de decidir qué regla se aplica si diversas reglas tienen selectores diferentes pero podrían aplicarse a un mismo elemento. Básicamente, la especificidad mide **cuán específica es la selección de un selector**:

```
<h1 class="main-heading">This is my heading.</h1>
```

```
.main-heading {  
  color: red;  
}  
  
h1 {  
  color: blue;  
}
```

3. **Herencia**: La herencia también debe entenderse en este contexto: algunos valores de las propiedades CSS que se han establecido para los elementos padre los heredan los elementos hijo, pero otros no.

Por ejemplo, si para un elemento se establece el color `color` y el tipo de letra `font-family`, cada elemento que se encuentre dentro de él también se mostrará de ese color y con ese tipo de letra, a menos que les se haya aplicado un color y un tipo de letra diferentes directamente.

```
<main>  
  <h1>Lorem ipsum dolor sit.</h1>  
  <p>  
    Lorem ipsum dolor, sit amet consectetur adipisicing elit. Sit, pariatur  
    mollitia eveniet, quos excepturi voluptas velit voluptatum iste  
    perferendis laboriosam architecto voluptate, vitae nostrum adipisci?  
    Mollitia excepturi quibusdam asperiores repudiandae.  
  </p>  
</ul>
```

```
<li>Lorem, ipsum dolor.</li>
<li>Lorem, ipsum dolor.</li>
<li>Lorem, ipsum dolor.</li>
</ul>
</main>
```

```
main {
  color: red;
}
```

Algunas propiedades no se heredan. Por ejemplo, si para un elemento se establece un ancho width del 50%, sus descendientes no tendrán un 50% de ancho con respecto al de sus padres. Si este fuera el caso, ¡sería muy frustrante usar CSS!

Listas de selectores

Existen diferentes variantes para combinar selectores: [más info](#)

Nivel de profundidad

```
<body>
  <div class="primary">
    <h2>Lorem, ipsum dolor.</h2>
    <h2>Lorem, ipsum dolor.</h2>
  </div>
  <div class="secondary">
    <h2>Lorem, ipsum dolor.</h2>
    <h2>Lorem, ipsum dolor.</h2>
  </div>
</body>
```

No existe un límite para el nivel de profundidad.

```
.primary h2 {
  color: tomato;
}
.secondary h2 {
  color: peru;
}
```

Seleccionar varios elementos

```
.primary,
.secondary {
```

```
    color: tomato;
}
```

Colores

- **Por palabra clave** [lista](#)
- **Colores en hexadecimal** [Hexadecimal](#), [colorhunt.co](#)
- **RGB y RGBA** [Lista](#)
- **HSL** [HSL #01](#), [HSL #02](#)

```
main {
  color: red;
  background-color: rgb(18, 107, 185);
  border: 10px solid hsl(125, 76%, 27%);
}
```

Fuentes

- [fonts.google](#)

```
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
  href="https://fonts.googleapis.com/css2?family=Kaisei+HarunoUmi&display=swap"
  rel="stylesheet"
/>
```

```
body {
  font-family: "Kaisei HarunoUmi", serif;
}

p {
  font-size: 2rem;
  font-weight: 600;
}
```

Unidades relativas vs Abosultas

En CSS se pueden utilizar diferentes unidades de medida y no existe "la más recomendable". Es necesario conocerlas y saber cómo funcionan en el contexto para evaluar cuál es la más conveniente en cada caso. [más info](#)

Medidas absolutas

Su valor se encuentra definido en términos concretos y de manera medible. Esto quiere decir que no depende de otro valor de referencia, ni del contexto.

- mm: milímetros.
- cm: centímetros.
- in: pulgada ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
- pt: puntos. Un punto equivale a 1 /72 de pulgada, es decir, unos 0.35 milímetros.
- pc: picas. Una pica equivale a 12 puntos, o aproximadamente a 4.23 milímetros.
- px: pixel. Es la unidad mínima de resolución de la pantalla. En realidad suele considerársela una unidad.

Medidas relativas

Las unidades relativas no son valores exactos sino que se calculan a partir de otro valor de referencia. A pesar de parecer más difíciles de calcular son las más utilizadas en el diseño de sitios web responsive por su adaptabilidad a los diferentes dispositivos.

- em: Tamaño relativo al tamaño de texto de su contenedor.
- rem: Funciona igual que el em, con la diferencia que es relativo al valor de la fuente del elemento html

REM

```
html {  
  font-size: 16px;  
}  
  
h1 {  
  font-size: 1rem;  
  background-color: aqua;  
  padding: 5rem;  
}
```

EM

```
html {  
  font-size: 16px;  
}  
  
body {  
  font-size: 32px;  
}  
  
h1 {  
  /* Ahora serán 32px! */  
  font-size: 1em;  
  background-color: aqua;  
  padding: 5rem;  
}
```

```
html {  
  font-size: 16px;  
}  
  
h1 {  
  font-size: 2rem;  
  background-color: aqua;  
  padding: 1em;  
}
```

Background

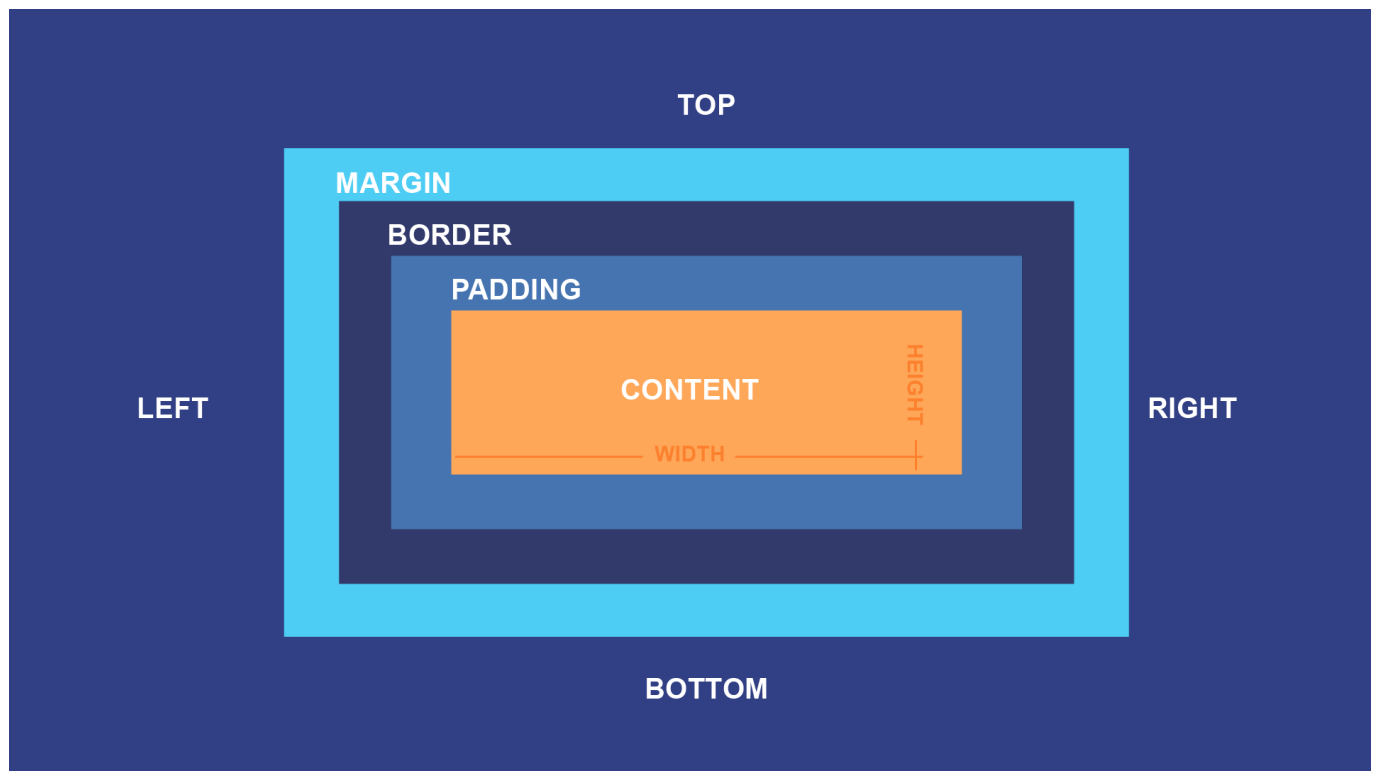
- picsum.photos

```
main {  
  background-image: url("500x500.jpg");  
  background-repeat: no-repeat;  
  background-size: cover;  
  background-position: center;  
}
```

- [repeat](#)
- [size](#)
- [position](#)

Modelo de caja (box model)

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos.



```
main {  
  color: white;  
  text-align: center;  
  background-image: url("500x500.jpg");  
  background-repeat: no-repeat;  
  background-size: cover;  
  background-position: center;  
  width: 500px;  
  height: 500px;  
  padding: 1rem;  
  border: 1rem solid #000;  
}
```

box-sizing

Para mantener el ancho a 500px, sin importar la cantidad de relleno, puede usar la propiedad **box-sizing: border-box**. Esto hace que el elemento mantenga su ancho; Si aumenta el relleno, el espacio de contenido disponible disminuirá.

```
main {  
  box-sizing: border-box;  
  color: white;  
  text-align: center;  
  background-image: url("500x500.jpg");  
  background-repeat: no-repeat;  
  background-size: cover;  
  background-position: center;  
  width: 500px;  
}
```

```
height: 500px;
padding: 1rem;
border: 1rem solid #000;
}
```

- [css-tricks](#)

```
html {
  box-sizing: border-box;
}
*,
*:before,
*:after {
  box-sizing: inherit;
}
```

Display

La propiedad CSS display especifica si un elemento es tratado como block or inline element

```
<body>
  <main>
    <h1>Lorem ipsum dolor sit.</h1>
    <p>
      Lorem ipsum dolor, sit amet consectetur adipisicing elit. Sit,
      pariatur mollitia eveniet, quos excepturi voluptas velit voluptatum
      iste perferendis laboriosam architecto voluptate, vitae nostrum
      adipisci? Mollitia excepturi quibusdam asperiores repudiandae.
    </p>
    <ul>
      <li>Lorem, ipsum dolor.</li>
      <li>Lorem, ipsum dolor.</li>
      <li>Lorem, ipsum dolor.</li>
    </ul>
    <a href="#">Ancla</a>
    <a href="#">Ancla</a>
  </main>
</body>
```

```
a {
  background-color: tomato;
  width: 300px;
  height: 300px;
  padding: 100px;
}
```

Podemos notar que no respeta el ancho, alto ni márgenes verticales, además al aplicar un padding este se desborda y tapa los otros elementos.

- `display`
- ¿De línea o de bloque?

display:inline

- No respeta el ancho ni alto.
- Padding se desborda cubriendo otros elementos.
- No respeta los márgenes verticales (top-bottom)

display:block

Por el contrario, ciertas etiquetas se renderizan en el navegador en líneas independientes, no mezcladas con el resto del texto. Ejemplos de estas etiquetas son los encabezados `<h1>` hasta `<h6>`, las citas en bloque `<blockquote>`, por supuesto los párrafos `<p>`, y quizá la más conocida de todas que es la etiqueta `<div>` usada normalmente para envolver a otros elementos. A estos elementos se les denomina elementos de bloque.

```
a {  
  display: block;  
  background-color: tomato;  
  width: 300px;  
  height: 300px;  
  padding: 100px;  
  margin-bottom: 200px;  
}
```

- Respetan los anchos y altos.
- Respetan los márgenes verticales.

Display: inline-block

```
a {  
  display: inline-block;  
  background-color: tomato;  
  width: 100px;  
  height: 100px;  
  padding: 100px;  
  margin-bottom: 200px;  
}
```

- En comparación con `display: inline`: la principal diferencia es que `display: inline-block` **permite establecer un ancho y alto** en el elemento.
- Además, con `display: inline-block`, se respetan los márgenes / rellenos superiores e inferiores, pero con `display: inline` no.

- En comparación con `display: block`: la principal diferencia es que `display: inline-block` **no agrega un salto de línea después del elemento**, por lo que el elemento puede sentarse junto a otros elementos.

Reset

- [reboot resets reasoning](#)
- [normalize.css/](#)

No recomendado!

```
* {  
  margin: 0;  
  padding: 0;  
}
```

Position

La `position` propiedad especifica el tipo de método de posicionamiento utilizado para un elemento (`static`, `relative`, `fixed`, `absolute` or `sticky`).

- [position](#)

Iconos

Material Icons

```
<link  
  href="https://fonts.googleapis.com/icon?family=Material+Icons"  
  rel="stylesheet"  
>
```

```
<span class="material-icons"> thumb_up </span>
```

Todos los estilos: **pero no utilizar todos a la vez**

```
<link  
  href="https://fonts.googleapis.com/css?  
family=Material+Icons|Material+Icons+Outlined|Material+Icons+Two+Tone|Material+Icons+Round|Material+Icons+Sharp"  
  rel="stylesheet"  
>
```

Ejemplo para Outline:

```
<link
  href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined"
  rel="stylesheet"
/>
```

```
<span class="material-icons-outlined icono-fuente"> thumb_up </span>
```

```
.icono-fuente {
  font-size: 20rem;
  color: tomato;
}
```

Si vas a utilizar pocos iconos la mejor alternativa es descargarlos en SVG. **La desventaja es que no podrás modificar directamente como si fuera una fuente**

```

```

```
.icono {
  width: 300px;
  /* color: tomato; */
  /* fill: tomato */
}
```

Cambiar color directamente en SVG:

```
<svg
  xmlns="http://www.w3.org/2000/svg"
  height="24px"
  viewBox="0 0 24 24"
  width="24px"
  fill="tomato"
>
  <path d="M0 0h24v24H0V0zm0 0h24v24H0V0z" fill="none" />
  <path
    d="M9 21h9c.83 0 1.54-.5 1.84-1.22l3.02-7.05c.09-.23.14-.47.14-.73v-2c0-1.1-.9-2-2-2h-6.31l.95-4.57.03-.32c0-.41-.17-.79-.44-1.06L14.17 1 7.58 7.59C7.22 7.95 7 8.45 7 9v10c0 1.1.9 2 2 2zM9 14.34-4.34L12 10h9v2l-3 7H9V9zM1 9h4v12H1z"
  />
</svg>
```

Font-Awesome

- fontawesome.com
- [font-awesome CDN](#)

3 columnas centradas

Nota #01: Esta práctica no es recomendada pero a futuro conocerán otros métodos más efectivos. Nota #02: El ideal es hacer un archivo CSS aparte pero por temas de documentación está en el mismo archivo HTML.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>

    <style>
      h1 {
        color: white;
        font-size: 40px;
        padding: 40px 0;
        margin: 0;
        text-transform: uppercase;
      }

      .bg-tomato {
        background-color: tomato;
      }

      .container {
        width: 800px;
        margin-left: auto;
        margin-right: auto;
        text-align: center;
      }

      .display-inline-block {
        text-align: left;
        display: inline-block;
        margin-left: auto;
        margin-right: auto;
        width: 200px;
        padding: 30px 30px;
        vertical-align: middle;
        color: white;
      }
    </style>
  </head>

  <body>
```

```
<main class="bg-tomato">
  <section class="container">
    <h1>Lorem, ipsum dolor.</h1>
    <div class="display-inline-block">
      <h2>Lorem ipsum dolor sit.</h2>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Magnam delectus architecto libero possimus vitae rem
        error quaerat, dicta molestiae deserunt corporis
        repellendus reprehenderit ea dolor suscipit molestias
        aspernatur, illo quam.
      </p>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Modi debitis voluptatibus rem numquam repudiandae
        perferendis delectus sed qui? Cum deserunt corrupti
        nihil sit in sequi vel aliquid dolorum accusamus ea.
      </p>
    </div>
    <div class="display-inline-block">
      <h2>Lorem ipsum dolor sit.</h2>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Magnam delectus architecto libero possimus vitae rem
        error quaerat, dicta molestiae deserunt corporis
        repellendus reprehenderit ea dolor suscipit molestias
        aspernatur, illo quam.
      </p>
    </div>
    <div class="display-inline-block">
      <h2>Lorem ipsum dolor sit.</h2>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Magnam delectus architecto libero possimus vitae rem
        error quaerat, dicta molestiae deserunt corporis
        repellendus reprehenderit ea dolor suscipit molestias
        aspernatur, illo quam.
      </p>
      <p>
        Lorem ipsum dolor, sit amet consectetur adipisicing
        elit. Necessitatibus enim quibusdam natus vel a iure
        corporis ea iusto? Id autem veniam quasi eos eveniet
        quibusdam impedit error saepe placeat voluptatem!
      </p>
    </div>
  </section>
</main>
</body>
</html>
```