



# tema 3 ~ DEW

[Introducción](#)

[Reglas](#)

[Herencia y cascada](#)

[Modelo de cajas](#)

## ▼ Introducción

```
h1{color:blue; font-size:12px;}
```

**h1:** selector → define sobre que elemento se aplica el estilo

**{...}:** declaración → define cuál es el estilo que se aplica sobre los elementos

- **"color":** propiedad → el atributo de estilo que se define
- **"blue":** valor → el valor o valores, de dicho atributo



## ¿Cómo aplico reglas CSS a un documento?

### ▼ Hoja de estilo externa

En un documento independiente .css con el elemento <link> en <head>

```
<head>
  <link rel="stylesheet" type="text/css" href="../../css/stilos.css" />
</head>
```

### ▼ Hoja de estilo interna

El estilo se inserta en la cabecera del documento HTML con el elemento <style> en <head>

```
<head>
  <style type="text/css">
    ...aquí el estilo
  </style>
</head>
```

### ▼ Estilos en línea

Se inserta el estilo DENTRO del elemento HTML del documento con el atributo style

```
<body style="...">
  <p style="...">Titulo de la página</p>
</body>
```

## ▼ Reglas

### 1. SELECTORES

- **De tipo (o tag):** Selecciona todos los elementos del mismo tipo → `elem {}`

```
h1 {
  color: #FFFFFF
}
```

```
p {
  color: black;
}
```

- **De ID:** El selector selecciona el elemento con el identificador especificado → `#nombreID {}`

```
DOCUMENTO HTML
<p id="titulo1">HOLA</p>
```

```
DOCUMENTO CSS
#titulo1 {
```

```
color: black;
}
```

- **De clase:** El selector selecciona todos los elementos que pertenecen a la misma clase (un elemento puede pertenecer a varias clases). → `.nombreClase{}`

DOCUMENTO HTML

```
<p class="rojo">HOLA</p>
<p class="rojo texto">HOLA</p>
```

DOCUMENTO CSS

```
.rojo {color:red;}
.texto {font-family:sans-serif;}
```

- **Pseudo-clases:** Se usan para representar eventos dinámicos, un cambio de estado, o alguna otra situación que no se puede representar de otra manera. → `:nombrePseudo-clase {}`
  - hacer clic, posicionar el ratón sobre un elemento, etc.
  - Permiten aplicar estilos a elementos que se encuentran en cierta situación/estado

```
a:link {color:blue;}
a:visited {color:magenta;}
```

#### ▼ SOBRE LINKS

Nombre	Ejemplo	Qué selecciona
:link	a:link	Todos los links no visitados
:visited	a:visited	Todos los links visitados
:active	a:active	Todos los links activos (cuando se hace clic sobre ellos)
:hover	a:hover	Todos los links que tienen el ratón por encima

```
a:link {color:#0000ff; text-decoration:none;}
a:visited {color:#ff3333; text-decoration:none;}
a:active {color:#33ff33; text-decoration:underline;}
a:hover {background-color:#e9e9e9; text-decoration:underline;}
```

link  
visited  
active  
hover

#### ▼ SOBRE OTROS ELEMENTOS

Nombre	Ejemplo	Qué selecciona
:focus	input:focus	El <input> que tiene el foco
:first-child	p:first-child	El primer hijo de todos los elementos <p>
:last-child	p:last-child	El último hijo de todos los elementos <p>
:lang	html:lang(es)	Un documento HTML con idioma español
:nth-child(n)	p:nth-child(3)	El tercer hijo de todos los elementos <p>
:nth-last-child(n)	p:nth-last-child(3)	El tercer último hijo de todos los elementos <p>

- **Pseudo-elementos:** Similares a las pseudo-clases, pero en este caso se refieren a aspectos físicos de los elementos que seleccionan. → `:nombrePseudo-elemento {}`
  - primera línea, primera letra, etc

Nombre	Ejemplo	Qué selecciona
:first-letter	p:first-letter	La primera letra del texto contenido en cualquier <p>
:first-line	p:first-line	La primera línea del texto contenido en cualquier <p>
:before	h1:before	Antes de cualquier <h1>
:after	p:after	Después de cualquier <p>

- **Combinación de selectores:** Los de antes se pueden combinar de diferentes formas →

#### ▼ De tipo + De clase / ID

DOCUMENTO HTML

```
<p id="aviso"> ... </p>
<h1 class="rojo"> ... </h1>
```

DOCUMENTO CSS

```
p#aviso {color:blue;}
h1.rojo {color:red;}
```

#### ▼ De clase encadenados

DOCUMENTO HTML

DOCUMENTO CSS

```
<p class="important code"> ... </p>
```

```
p.important.code{...}
```

#### ▼ Aplicar la misma regla a varios selectores

```
p, .aviso, p#rojo {...}
```

## 2. VALORES

Son asignados a una propiedad, pueden ser individuales o múltiples, hay distintos tipos dependiendo de la propiedad.

- **Palabras clave (keywords):** Identificadores predefinidos en CSS (Pueden aparecer varios, separados por espacios).

```
body {  
  font-weight: bold;  
  text-align: left;  
  border: thin solid black;  
}  
ul {list-style-type: square;}  
ol {list-style-type: decimal-leading-zero;}
```

- **Strings:** Con comillas simples (') o dobles ("). Para definir contenido HTML, rutas de ficheros, fuentes (con espacios), etc.

```
body {  
  font-family: 'Times new Roman', Times, serif;  
}
```

- **Medidas absolutas:** No dependen de otras medidas o del entorno (resolución de la pantalla, tamaño de otro elemento, etc.)
  - Basadas en unidades del mundo real → in (pulgadas), cm, mm, pt (puntos).
- **Medidas relativas:** Sí dependen del entorno, en cada plataforma pueden visualizarse de manera diferente.
  - px, %, em (relativo al tamaño regular de fuente), ex (relativo al tamaño de la letra x)

- **Números:**

```
body {font-size: 12px;}  
h1 {margin-left: -20px;}
```

- **Colores:**
  - Palabras clave: black, red, orange, etc.
  - RGB (red-green-blue): rgb(128,128,128)
  - Otros: rgb(50%,50%,50%), #FFA500 (255,165,0)

- **URLs:** Referencia un recurso a través de una URL url(path)

```
body {  
  background-image: url(laimagen.jpg);  
}
```

## ▼ Herencia y cascada

### 1. HERENCIA

Las propiedades de estilo de un elemento padre **se propagan a los elementos hijos** (en ausencia de reglas más específicas)

Algunas propiedades se heredan automáticamente (ej: estilos de texto, etc.), pero otras no (ej: bordes)

DOCUMENTO HTML

```
<div>
  <p>
    This <em>sentence</em> Will have a 20px <a href="#">font-size</a>.
  </p>
</div>
```

DOCUMENTO CSS

```
div {
  font-size: 20px;
}
```

## 2. CASCADA

Se refiere a cómo se determina el valor de una propiedad específica cuando varias declaraciones entran en conflicto.

### Algoritmo de cascada:

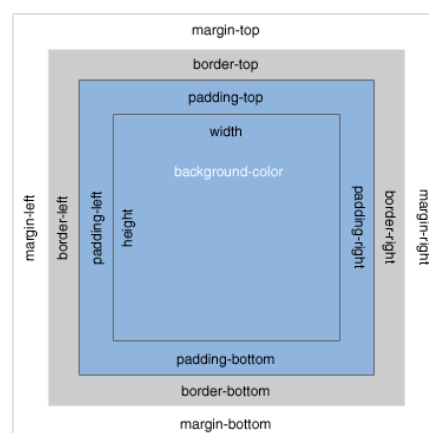
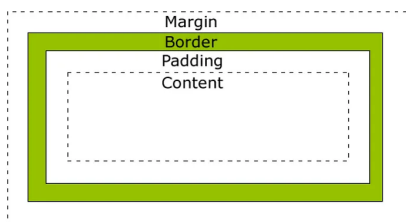
1. **Búsqueda de Declaraciones:** Para una propiedad en un elemento dado, se buscan todas las declaraciones que se aplican a ese elemento.
2. **Orden por Origen e Importancia:** Las declaraciones se ordenan según su origen y su importancia. Por ejemplo, las reglas definidas en una hoja de estilo externa tienen más peso que las reglas en una hoja de estilo interna o en línea.
3. **Especificidad del Selector:** Si aún no se resuelve, se ordenan las declaraciones según la especificidad del selector. Las reglas más específicas (por ejemplo, las que apuntan a un elemento con una clase específica) tienen prioridad sobre las reglas más generales.
4. **Orden de Definición:** Si aún no se resuelve, se ordenan las declaraciones según el orden en que aparecen en el código fuente. La última regla definida prevalece.

## ▼ Modelo de cajas

Determina cómo se posicionan los elementos en la ventana del navegador

💡 Todos los elementos HTML son cajas. Cada caja tiene: margen, borde, relleno y contenido

Cada lado se puede controlar individualmente



📁 [tema 4 ~ DEW](#)