

# 11 CSS: Pseudoelementos

Mientras que las pseudoclases ( `:hover` , `:nth-child()` , etc.) seleccionan elementos *completos* cuando están en un estado particular o tienen una cierta posición estructural, los **pseudoelementos** actúan como si añadieran elementos "virtuales" al documento antes, después o dentro de un elemento existente, permitiéndonos estilizar partes específicas del contenido de un elemento que no son seleccionables de otra manera.

## Sintaxis Clave: Doble Dos Puntos `::`

La sintaxis moderna y recomendada para los pseudoelementos utiliza **doble dos puntos** ( `::` ) para diferenciarlos claramente de las pseudoclases (que usan un solo dos puntos `:` ).

```
selector::pseudoelemento {  
  propiedad: valor;  
}
```

*Nota Histórica:* En CSS2, los pseudoelementos también usaban solo dos puntos (ej., `:before` , `:after` ). Para mantener la compatibilidad con navegadores antiguos, la sintaxis de solo dos puntos todavía funciona para los pseudoelementos definidos originalmente en CSS2 ( `:before` , `:after` , `:first-letter` , `:first-line` ). Sin embargo, para nuevos pseudoelementos y por claridad, **siempre deberías usar la sintaxis de doble dos puntos** `::` .

Veamos los pseudoelementos más importantes:

### 1. `::before`

- **Qué hace:** Crea un pseudoelemento que es el **primer hijo** del elemento seleccionado. Se muestra *antes* del contenido real del elemento.
- **Requisito Fundamental:** Debe tener la propiedad `content` definida para que el pseudoelemento sea generado y visible (incluso si `content` está vacío: `content: ""` ).
- **Uso:** Muy versátil. Se usa para:
  - Añadir contenido decorativo (iconos, viñetas personalizadas, comillas).
  - Insertar texto o contadores.
  - Crear efectos visuales sin añadir HTML extra.
- **Ejemplo:** Añadir comillas decorativas a elementos `<blockquote>` .

```
<blockquote>Esta es una cita importante.</blockquote>
```

```
blockquote::before {  
  content: "''"; /* Contenido a insertar */  
  font-size: 3em;  
  color: lightgrey;  
  margin-right: 5px;  
  vertical-align: -0.2em; /* Ajuste vertical */  
}
```

## 2. `::after`

- **Qué hace:** Crea un pseudoelemento que es el **último hijo** del elemento seleccionado. Se muestra *después* del contenido real del elemento.
- **Requisito Fundamental:** También requiere la propiedad `content`.
- **Uso:** Similar a `::before`, pero para añadir contenido al final. Usos comunes:
  - Añadir contenido decorativo final.
  - Insertar texto o contadores.
  - **Clearfix Hack:** Uno de sus usos más famosos es para contener elementos flotantes (veremos `float` más adelante). El "clearfix" moderno utiliza `::after`.
- **Ejemplo:** Añadir un pequeño icono después de enlaces externos.

```
<a href="https://externo.com" target="_blank">Sitio Externo</a>
```

```
a[target="_blank"]::after {  
  content: " ➤"; /* Flecha indicando externo */  
  font-size: 0.8em;  
  display: inline-block; /* Para que no rompa la línea */  
  margin-left: 3px;  
}
```

- **Ejemplo Clearfix (conceptual):**

```
.contenedor-flotante::after {  
  content: "";  
  display: table; /* O block */  
  clear: both; /* Propiedad clave para limpiar floats */  
}
```

## 3. `::first-letter`

- **Selecciona:** La **primera letra** del contenido de texto de un elemento de **bloque** (como `<p>`, `<h1>`, `<div>`, `<li>`).
- **Uso:** Principalmente para crear "Letras Capitales" o "Drop Caps", donde la primera letra de un párrafo es más grande y/o tiene un estilo diferente.
- **Limitaciones:** Solo aplica a elementos de bloque. Ciertas propiedades CSS se pueden aplicar (fuente, color, fondo, margen, padding, borde, float, etc.).
- **Ejemplo:**

```
<p>Érase una vez un párrafo...</p>
```

```
p::first-letter {  
  font-size: 3em; /* Más grande */  
  color: navy;
```

```
float: left; /* Para que el texto fluya alrededor */
margin-right: 0.1em;
line-height: 0.8; /* Ajuste de altura de línea */
}
```

#### 4. `::first-line`

- **Selecciona:** La **primera línea** formateada del contenido de texto de un elemento de **bloque**. La definición de "primera línea" depende del ancho del contenedor, el tamaño de fuente, etc.
- **Uso:** Para aplicar un estilo diferente a la primera línea de un párrafo o encabezado (p. ej., en versalitas, negrita).
- **Limitaciones:** Solo aplica a elementos de bloque. El conjunto de propiedades aplicables es similar al de `::first-letter`. Es dinámico: si el ancho cambia, el contenido de la primera línea cambia y el estilo se reajusta.
- **Ejemplo:**

```
p::first-line {
  font-weight: bold;
  color: #555;
  letter-spacing: 0.5px; /* Espaciado entre letras */
}
```

#### 5. `::marker`

- **Selecciona:** El **marcador** (viñeta o número) de un elemento de lista (`<li>`) o cualquier elemento con `display: list-item`.
- **Uso:** Permite personalizar directamente la apariencia de las viñetas y números de las listas, algo que antes requería trucos con `::before`.
- **Propiedades Aplicables:** Principalmente `color`, `content`, `font`, `white-space`, y propiedades de animación/transición.
- **Ejemplo:** Cambiar el color y el tipo de viñeta.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

```
li::marker {
  color: green;
  font-size: 1.2em;
  /* content: '►'; También podrías cambiar el símbolo */
}
```

- **Soporte:** Relativamente nuevo, verifica la compatibilidad en [caniuse.com](http://caniuse.com).

#### 6. `::selection`

- **Selecciona:** La parte del documento que el usuario ha **resaltado** (seleccionado con el ratón o teclado).
- **Uso:** Permite personalizar el color de fondo y el color del texto de la selección del usuario.
- **Limitaciones:** Solo unas pocas propiedades CSS tienen efecto: `color`, `background-color`, `text-shadow`, `text-decoration` y propiedades relacionadas.
- **Ejemplo:**

```
/* Nota: A menudo necesita prefijos para navegadores más antiguos, aunque ya no tanto */
::selection {
  background-color: yellow;
  color: black;
}
/* Ejemplo con prefijo (menos necesario hoy) */
/*
::-moz-selection {
  background-color: yellow;
  color: black;
}
*/
```

## 7. `::placeholder`

- **Selecciona:** El texto de marcador de posición (placeholder) en elementos de formulario como `<input>` y `<textarea>` (el texto que aparece como pista cuando el campo está vacío).
- **Uso:** Para cambiar el estilo (color, fuente, opacidad) del texto del placeholder.
- **Ejemplo:**

```
<input type="text" placeholder="Escribe tu nombre aquí...">
```

```
input::placeholder {
  color: lightgrey;
  font-style: italic;
  opacity: 0.8; /* A veces útil para diferenciarlo más */
}
/* También puede necesitar prefijos en navegadores muy antiguos, pero ::placeholder es bastante estándar */
```

## Características Importantes de los Pseudoelementos:

- **No están en el DOM:** Los pseudoelementos (`::before`, `::after`, etc.) son abstracciones del CSS. No son nodos reales en el Document Object Model (DOM), por lo que no pueden ser seleccionados directamente con JavaScript de la misma manera que los elementos normales.
- **Herencia:** Heredan propiedades de su elemento "padre" (el elemento al que están adjuntos), a menos que se especifiquen valores diferentes.

- **Modelo de Caja:** Los pseudoelementos `::before` y `::after` generan cajas que pueden ser estilizadas con `width`, `height`, `padding`, `border`, `margin`, `background`, etc., siempre que su `display` no sea `inline` (el valor por defecto). A menudo se les da `display: block` o `display: inline-block`.
-