

# PSEUDO ELEMENTOS

## after::

Este pseudo elemento hace referencia a que sucede con un elemento luego del mismo

## before::

Hace referencia a un elemento con lo que sucede antes del mismo.

```
<style>
p::after { content: 'mi otro texto';}

</style>
```

Hemos incluido la propiedad content. Esta propiedad es obligatoria para trabajar ya que es la que me permite justamente luego incorporar elementos antes o después del elemento al cual le haya aplicado ::AFTER o ::BEFORE

Por otro lado, es interesante saber que también puedo aplicar imágenes. Por caso si quisiera que apareciera una imagen por ejemplo luego de un párrafo lo haría de la siguiente manera:

```
<style>
p::after { content: url(imagen.jpg);}

</style>
```

Recordemos que siempre que traemos una imagen como lo hicimos en la propiedad **background-image** como así también cuando hemos trabajado con import en el caso de traer un archivo , lo hacemos con el protocolo url.

En realidad, sería simple hablar sólo de estos ejemplos de ::AFTER y ::BEFORE pero la realidad es que tenemos mucho más por hacer con estos pseudo elementos.

# Contadores

No sería interesante, por caso que tengamos varios enunciados, que contienen un texto y una imagen repetitiva, por ejemplo **CAPÍTULO** y una imagen de una estrella y que aparte quisiéramos ponerles números?. Para que lo podamos entender de una manera mucho más simple, en realidad, contamos con tres propiedades básicas para poder construir una especie de contador con estos pseudo elementos

## counter-reset

Esta propiedad me indica donde se inicializa, o donde arranca nuevamente el contador si por ejemplo se ubica en el body, eso claramente representa que el contador, será inicializado una sola vez porque el body está una sólo vez en nuestro documento, si por el contrario lo aplicara a un h1 , cada vez que apareciera un h1 el contador volvería a inicializarse. Esto resulta útil para títulos y subtítulos por ejemplo

## counter-increment

Es donde voy a incrementar el contador, es decir donde va a ir aumentado su número

## counter

Esto me permite llamar el contador según el nombre que le haya aplicado para poder identificarlo

```
<head>

<style>

body { counter-reset: miTitulo;}
h1 { counter-increment: miTitulo;}
h1::after { content: 'Titulo' counter(miTitulo) url(imagen.jpg);}

</style>
</head>

<body>

<h1></h1>
<h1></h1>

</body>
</html>
```

Dependiendo el orden con el cual quiera mostrar los elementos será si primero pongo el contenido

de texto, luego la imagen o el contador en el medio como en este caso, el anterior ejemplo en el navegador se verá de la siguiente forma:

**Titulo1**★

**Titulo2**★

También se puede profundizar sobre estos pseudo elementos y hacer referencia al valor de un atributo colocando como contenido también, veamos un ejemplo:

```
<head>
<style>
a::after {
  content: " (" attr(title) ")";
}
</style>
</head>
<body>

<p><a href="http://www.educacionit.com" title="vinculo">Educacion IT</a>.</p>

</body>
```

De esta forma vemos cómo podemos generar estructuras repetitivas con CSS que son sumamente interesantes cuando no sabemos en principio cuántos elementos tendrá nuestro Sitio más adelante entonces preparamos el entorno para múltiples escenarios.

Muchas veces incluso `::AFTER` y `::BEFORE` terminan siendo herramientas para generar cortes en la estructura por ejemplo es común ver como a un contenido a su vez se le agrega la propiedad `display:block` o mismo `:inline` para cortar líneas por ejemplo de imágenes en una galería.

Es decir en conclusión `::AFTER` y `::BEFORE` son mucho más que simples elementos para agregar imágenes o contenidos.

Veamos un ejemplo con un vínculo

```
#encabezado a::before { content: '-'; }
```