

CSS #2

Selectores Combinados, Herencia y Cascada

WEB 1 TUDAI UNICEN 2024

Qué sabemos hasta ahora de CSS?



- Sirve para aplicar el **estilo visual** a una página web.
- Se escribe en un archivo aparte del HTML con extensión .css

Tipo Selector	Ejemplo HTML	Ejemplo CSS
TIPO (o elemento)	<code><p>...</p></code>	<code>p {...}</code>
CLASE	<code><p class="clase">...</p></code>	<code>.clase {...}</code>
IDENTIFICADOR	<code><p id="identif">...</p></code>	<code>#identif {...}</code>

Tipo de selectores (II)

Tipo Selector	Ejemplo HTML	Ejemplo CSS
TIPO (o elemento)	<code><p>...</p></code>	<code>p {...}</code>
CLASE	<code><p class="clase">...</p></code>	<code>.clase {...}</code>
IDENTIFICADOR	<code><p id="identif">...</p></code>	<code>#identif {...}</code>
ATRIBUTO	<code><input type="text" /></code>	<code>input[type=text] {...}</code>
UNIVERSAL	todos los elementos de la página	<code>* {...}</code>

En sitios web o aplicaciones de gran escala, el código CSS generado puede volverse extenso. Por ello, es crucial **emplear y conocer técnicas avanzadas** para mantener el código limpio, organizado y fácilmente reutilizable.

- ✓ **SELECTORES ESPECÍFICOS**
- ✓ **SELECTORES COMBINADOS**
- ✓ **SELECTORES AGRUPADOS**
- ✓ **HERENCIA**
- ✓ **CASCADA**



Selectores específicos

¿Cómo hacemos para aplicar un estilo solo a los párrafos con clase 'destacado'?

```
<h1>Asi esta el Campeonato</h1>
```

```
<p>La segunda fecha del Turismo Carretera en Neuquén ya es pasado y el campeonato tuvo modificaciones.</p>
```

```
<p class='destacado'>Christian Ledesma sigue como puntero pero el nuevo escolta es Jonatan Castellano.</p>
```

```
<p>Al ser consultado por este medio...</p>
```

Asi esta el Campeonato

La segunda fecha del Turismo Carretera en Neuquén ya es pasado y el campeonato tuvo modificaciones.

Christian Ledesma sigue como puntero pero el nuevo escolta es Jonatan Castellano.

Al ser consultado por este medio...

SELECTORES ESPECÍFICOS

Se pueden concatenar dos o más selectores para hacerlos **más específicos**.

```
p.destacado {  
  color: red;  
}
```

*“selecciona los **párrafos** (p) que contengan la clase **destacado**”*

Lo mismo se aplica a Id's, clases o cualquier tipo de selector:

```
p#principal.resaltado {  
  font-size: 25px;  
}
```

**¿Tiene sentido
usar ID?**

EJEMPLO: Selectores específicos

```
<h1>Asi está el Campeonato</h1>
<p>El campeonato recién comenzo...</p>
<p class='destacado'> Chevrolet ya va primero.</p>
<p>Nunca va a ser superado por una marca menor. <span
class='especial'>Ford</span></p>

<ul>
  <li id="unico">Ortelli, Guillermo - Chevrolet</li>
  <li>Ledesma, Cristian - Chevrolet</li>
  <li class='especial'>Guri Martinez - Ford</li>
  <li>Castellano, Jonatan -<span class='especial'> Ford </span></li>
  <li class='destacado'>Canapino, Agustín - Chevrolet</li>
</ul>
```

```
/* aplica a los parrafos con clase "destacado" */
p.destacado {
  background-color: green;
  color: white;
}

/* aplica a todo elemento con la clase 'especial' */
.especial {
  text-decoration: line-through;
}

/* aplica a los li con id "unico" */
li#unico {
  color: orange;
}
```

Asi esta el Campeonato

La segunda fecha del Turismo Carretera en Neuquén ya es pasado y el campeonato tuvo modificaciones.

Christian Ledesma sigue como puntero pero el nuevo escolta es Jonatan Castellano.

Al ser consultado por este medio...

La tabla queda conformada de la siguiente manera

- Ortelli, Guillermo - Chevrolet
- Ledesma, Cristian - Chevrolet
- Guri Martinez - Ford
- Castellano, Jonatan - Ford
- Canapino, Agustín - Chevrolet

DEMO

<https://codepen.io/webUnicen/pen/JjVegrM>

Selectores combinados

¿Cómo hacemos para aplicar un estilo solo a los links que estén dentro de una lista?

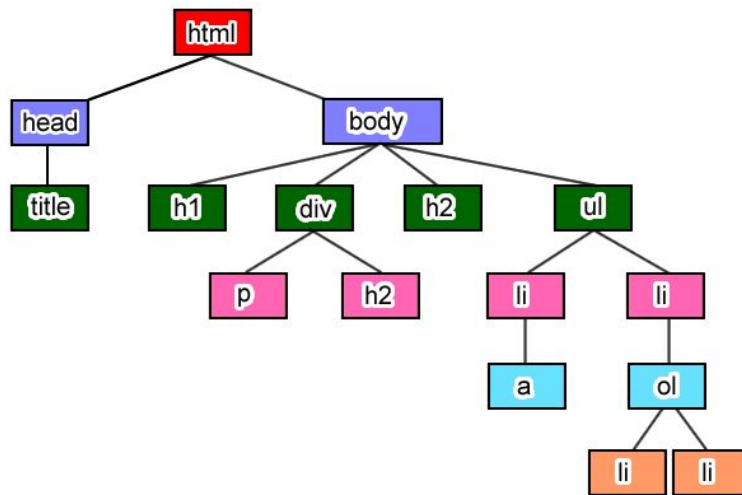
```
<p>Este es un <a href="link.html">Link</a></p>
<ul>
  <li><a href="linkA.html">Elemento 1</a></li>
  <li><a href="linkB.html">Elemento 2</a></li>
  <li><a href="linkC.html">Elemento 3</a></li>
</ul>
```

Este es un [Link](#)

- Elemento 1
- Elemento 2
- Elemento 3

Una manera de comprender las dependencias y relaciones entre elementos es mediante un diagrama de árbol.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo Arbol</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <h1>Titulo 1</h1>
    <div>Div1
      <p>Parrafo dentro de div</p>
      <h2>Titulo 2 en div</h2>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>Elemento 1 <a href="...">Link1</a></li>
      <li>Elemento 2
        <ol>
          <li>Primero</li>
          <li>Segundo</li>
        </ol>
      </li>
    </ul>
  </body>
</html>
```



SELECTORES COMBINADOS

Un selector combinado permite **combinar dos o más selectores CSS, formando uno más complejo y potente.**

Existen varios combinadores en CSS:

Nombre	Símbolo	Ejemplo	Significado
Combinador descendiente	(espacio)	<code>#page div { }</code>	Selecciona elementos dentro de otros (cualquier nivel).
Combinador hijo	<code>></code>	<code>#page > div { }</code>	Selecciona elementos hijos directos (primer nivel).
Combinador hermano adyacente	<code>+</code>	<code>div + div { }</code>	Selecciona elementos contiguos a otros (mismo nivel).
Combinador hermano general	<code>~</code>	<code>div ~ div { }</code>	Selecciona elementos que siguen a otros (mismo nivel).

Selectores combinados (descendiente)

SELECTOR DESCENDIENTE

Permite seleccionar elementos **contenidos dentro** de otros elementos (no importa la profundidad).

padre **hijo** { ... }

ESPACIO

```
p span { color: blue; }
```

Selecciona los que estén dentro de algún <p>

```
ul li a {  
    text-decoration: none;  
    color: red;  
    background-color: yellow;  
}
```

Selecciona los “links” que se encuentren dentro de un “item” de una “lista no-ordenada”



EJEMPLO: Selector combinado descendiente

```
<p>Este es un <a href="link.html">Link</a></p>
<ul>
  <li><a href="linkA.html">Elemento 1</a></li>
  <li><a href="linkB.html">Elemento 2</a></li>
  <li><a href="linkC.html">Elemento 3</a></li>
</ul>
```

```
/* aplica solo a los "links" que se
encuentren dentro de un "item" de una "lista
no-ordenada" */
ul li a {
  text-decoration: none;
  color: red;
}
```

Este es un [Link](#)

- Elemento 1
- Elemento 2
- Elemento 3



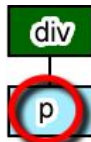
<http://codepen.io/webUnicen/pen/gmKwKM>

Selectores combinados

SELECTOR HIJO >

Selecciona todos los elementos que son **hijos directos** de un elemento.

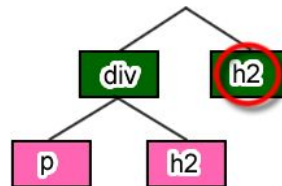
```
div > p {  
    background-color: yellow;  
}
```



SELECTOR HERMANO ADYACENTE +

Selecciona todos los elementos que están **inmediatamente a continuación** de otro elemento.

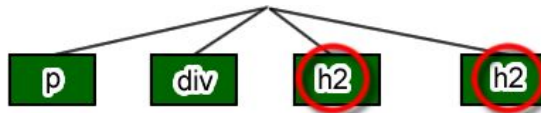
```
div ~ h2 {  
    background-color: orange;  
}
```



SELECTOR HERMANO GENERAL ~

Selecciona todos los **hermanos a continuación** de un elemento

```
div + h2 {  
    background-color: red;  
}
```





EJEMPLO: Selectores combinados (hijo y adyacentes)

```
<h1>Titulo 1</h1>
<div>
  <p>Parrafo dentro de un div</p>
  <h2>Titulo dentro</h2>
  <h2>Otro titulo dentro de un div</h2>
</div>
<h2>Titulo 2</h2>
<h2>Titulo 3</h2>
<p>Otro parrafo</p>
<ul>
  <li>Elemento 1 <a href="link.html">Link 1</a></li>
  <li>Elemento 2
    <ol>
      <li>Primero</li>
      <li>Segundo</li>
    </ol>
  </li>
</ul>
```

```
div > p {
  background-color:yellow;
}

div + h2 {
  background-color:red;
}

p ~ h2{
  background-color: orange;
}

div {
  border: solid;
}
```

Titulo 1

Parrafo dentro de un div

Titulo dentro

Otro titulo dentro de un div

Titulo 2

Titulo 3

Otro parrafo

- Elemento 1 [Link 1](#)
- Elemento 2

DEMO

<https://codepen.io/webUnicen/pen/MWRzMjO>

Qué pasa si
intercambiamos
~ y + ?

Selectores agrupados

¿Cómo hacemos para aplicar las mismas reglas a múltiples selectores?

```
.container-logo {  
  border-color: red;  
  background: white;  
}  
.container-alert {  
  border-color: red;  
  background: white;  
}  
.container-warning {  
  border-color: red;  
  background: white;  
}
```



```
.container-logo, .container-alert, .container-warning {  
  border-color: red;  
  background: white;  
}
```

SELECTORES AGRUPADOS

Permite **crear grupos con diferentes selectores** con el objetivo de escribir menos código, o reutilizar bloques de código CSS en más situaciones.

- Se separan los selectores con una coma (,) creando un grupo de selectores con propiedades en común



The screenshot displays a CodePen editor with two panels: HTML and CSS. The HTML panel contains two headings, `<h1>Titulo Grande</h1>` and `<h2>Titulo No tan grande</h2>`, followed by a paragraph of Lorem Ipsum text. The CSS panel shows a grouped selector `h1, h2, p {` with common properties: `font-family: Arial, Helvetica, sans-serif;` and `color: #8A8E7B;`. A separate rule for `p {` sets `font-size: 40px;`. The right side of the image shows the rendered output: 'Titulo Grande' in a large font, 'Titulo No tan grande' in a smaller font, and the Lorem Ipsum paragraph in a large font.

```
HTML
1 <h1>Titulo Grande</h1>
2 <h2>Titulo No tan grande</h2>
3 <p>
4 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
5 In elementum augue metus, non placerat enim congue ut.
6 Suspendisse ex enim, imperdiet ut iaculis ac
7 </p>
```

```
CSS
1 h1, h2, p {
2   font-family: Arial, Helvetica, sans-serif;
3   color: #8A8E7B;
4 }
5 p {
6   font-size: 40px;
7 }
```

Titulo Grande

Titulo No tan grande

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In elementum augue metus, non placerat enim congue ut. Suspendisse ex enim, imperdiet ut iaculis ac



<http://codepen.io/webUnicen/pen/yMEajv>

Ejemplo - Duplicación de estilos

Esta hoja de estilos se puede mejorar?

```
<h1>Titulo Noticia</h1>
```

```
<h2>Subtitulo</h2>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In  
elementum augue metus, non placerat enim congue ut.  
Suspendisse ex enim, imperdiet ut iaculis ac</p>
```

```
<p class="unico">Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. In elementum augue metus, non placerat enim  
congue ut. Suspendisse ex enim, imperdiet ut iaculis ac</p>
```

```

```

```
<p class="caption-image">Nombre de la imagen</p>
```

 <http://codepen.io/webUnicen/pen/peKEVp>

```
1 * h1 {  
2   font-family: Helvetica;  
3   color: blue;  
4 }  
5 * h2 {  
6   font-family: Helvetica;  
7   color: blue;  
8   font-style: italic;  
9 }  
10 * p {  
11   font-family: Helvetica;  
12   color: blue;  
13 }  
14 * .caption-image {  
15   font-family: Helvetica;  
16   font-style: italic;  
17   color: black;  
18 }
```



Ejemplo - Estilos sin duplicación

- Misma font-family para los elementos
- Mismo color azul para varios elementos

```
<h1>Titulo Noticia</h1>
```

```
<h2>Subtitulo</h2>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing  
elit. In elementum augue metus, non placerat enim congue  
ut. Suspendisse ex enim, imperdiet ut iaculis ac</p>
```

```
<p class="unico">Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. In elementum augue metus, non placerat  
enim congue ut. Suspendisse ex enim, imperdiet ut iaculis  
ac</p>
```

```

```

```
<p class="caption-image">Nombre de la imagen</p>
```

```
1 • h1, h2, p, .caption-image {  
2   font-family: Helvetica;  
3   color: blue;  
4 }  
5 • h2, .caption-image {  
6   font-style: italic;  
7 }  
8 • .caption-image {  
9   color: black;  
10 }  
11 • img {  
12   width: 300px;  
13 }
```



¿Qué ventajas ven de re-usar estilos?



<http://codepen.io/webUnicen/pen/KWegRy>

Cuidado



Un **espacio** o una **coma** puede hacer la diferencia!

HTML

```
<p class='aviso'>Este es un parrafo aviso</p>
```

```
<p> Este es un parrafo con <span  
class='aviso'>un span aviso.</span></p>
```

```
<p>Esto es un parrafo</p>  
<ul class='aviso'>  
  <li>Lista con clase aviso</li>  
</ul>
```

CSS

```
p.aviso {  
  color:blue;  
}
```

```
p .aviso {  
  color:red;  
}
```

```
p, .aviso {  
  color:green;  
}
```



<http://codepen.io/ignaciojonas/pen/vGewMX>

Herencia

¿A qué les suena herencia?



CSS: Herencia

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un **elemento padre se transmiten a sus elementos hijos**.

HTML

```
<div class="container">
  <p>Texto del padre</p>
  <div class="child">
    <p>Texto del hijo</p>
  </div>
</div>
```

CSS

```
.container {
  color: green;
  border: 1px solid red;
}
```



Texto del padre

Texto del hijo



No todas las propiedades CSS son heredables! Algunas de ellas no tendría sentido que lo fueran.

Lista completa de las propiedades de CSS heredables y no heredables

<https://www.w3.org/TR/css-2010/#properties>

Herencia

Todos los elementos de un documento HTML heredan todas las propiedades heredables de su padre (excepto el elemento raíz <html>, que no tiene padre)

El valor por defecto para estas propiedades es `inherit`.

```
p { color: inherit; } /* valor por defecto, hereda el color de su padre */
```

HTML

```
<h1>Título</h1>
<p>Un párrafo de texto.</p>
<div>
  <h1>Título</h1>
  <p>Un párrafo de texto.</p>
</div>
```

CSS

```
html {
  font: 75% Verdana, sans-serif;
}
```



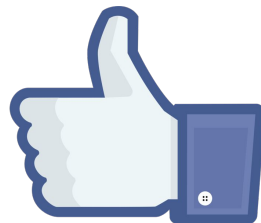
<http://codepen.io/webUnicen/pen/mWKraP>

¿Para qué nos sirve entonces la herencia?

- Entender cómo se definen las reglas visuales de aquellos elementos que no definimos explícitamente en el código.
- Escribir menos código redundante.
- Escribir código limpio y reutilizable.

Más sobre herencia:

[https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#control de la herencia](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#control_de_la_herencia)



Cascada

CSS: Cascada

¿Qué pasa cuando un elemento tiene definidas las **mismas propiedades pero con diferentes valores** desde distintos selectores?

HTML

```
<div class="container">  
  Hola soy un texto!  
</div>
```

CSS

```
.container {  
  color: green;  
}  
div {  
  color: blue;  
}
```



En este ejemplo, el texto podría ser “green” o “blue”.

¿Cuál termina de aplicar el browser?

CASCADA

La cascada es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas (colisiones) al mismo elemento.



CSS

Cascading Style Sheets

(hoja de estilos en cascada)



Cascada

HTML

```
<div class="container">  
  Hola soy un texto!  
</div>
```

CSS

```
.container {  
  color: green;  
}  
div {  
  color: blue;  
}
```

Existe una **colisión**.
CSS lo resuelve por cascada!

Cascada utiliza un **orden de prioridad** analizando tres conceptos a la hora de resolver colisiones:



1. **IMPORTANCIA**
2. **ESPECIFICIDAD**
3. **ORDEN**

Cascada - Importancia

IMPORTANCIA

CSS permite añadir la propiedad `!important` al final del valor de una línea CSS, forzando al navegador a darle prioridad al estilo sobre otros.

`color: blue !important;`

HTML

```
<div class="container">
  Hola soy un texto!
</div>
```

CSS

```
.container {
  color: green;
}
div {
  color: blue !important;
}
```



El abuso y uso indiscriminado debería evitarse si es posible.



<http://codepen.io/webUnicen/pen/NpzRev>

Cascada - Especificidad

ESPECIFICIDAD

Las reglas más específicas tienen mayor importancia que las reglas menos específicas.

HTML

```
<div class="container">  
  Hola soy un texto!  
</div>
```

CSS

```
.container {  
  color: green;  
}  
div {  
  color: blue;  
}
```

En este caso, un **selector de clase** es más específico que un **selector de tipo**



<https://codepen.io/webUnicen/pen/jORXyWy>

Cascada - Especificidad

¿Cómo se calcula la Especificidad?

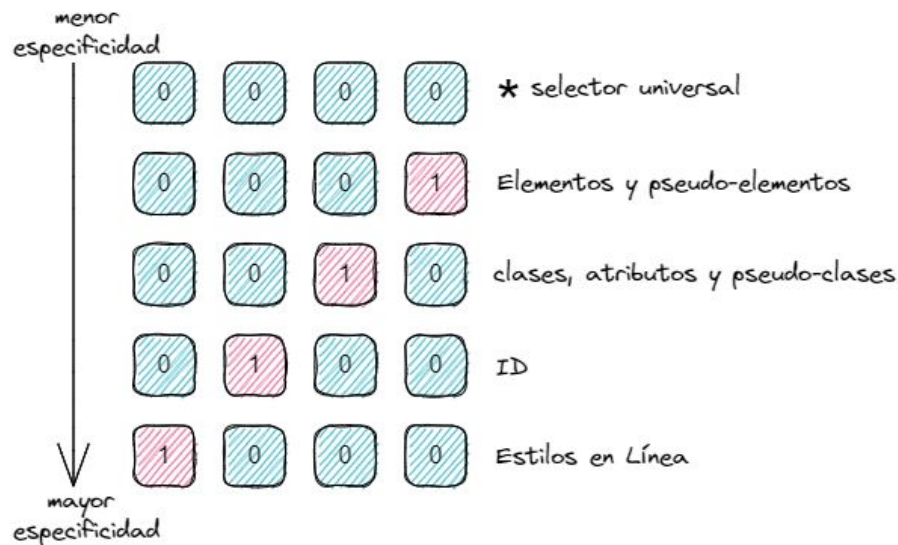
El cálculo de especificidad no es simple, y a veces no es trivial determinar (para nosotros) qué selector tiene más especificidad que otros.

En líneas generales se pueden seguir estas reglas:

- **Especificidad de tipo:** Este es el **nivel más bajo de especificidad** y se aplica a los selectores de tipo de elemento, como div, p, a, etc.
- **Especificidad de clase/atributo/pseudo-clase:** Este nivel de especificidad se aplica a los selectores de clase (.), atributo ([]), y pseudo-clases (:), como .clase, [atributo], :hover, etc.
- **Especificidad de ID:** Este es el **nivel más alto de especificidad** y se aplica a los selectores de ID (#).

Cascada - Especificidad

El navegador tiene un sistema llamado Especificidad CSS para el **cálculo exacto de la especificidad** de los elementos. Siguiendo estas reglas obtiene como resultado un valor numérico que utiliza para comparar en caso de colisiones.



No se toma el calculo exacto en los exámenes. Con la idea anterior general es suficiente.

Explicado con peces
<https://specifishity.com/>



!important gana a todos



<https://codepen.io/webUnicen/pen/peKEqL>

Cascada - Especificidad

<i>selector</i>	<i>id</i>	<i>class</i>	<i>tag</i>	<i>total</i>
<i>p</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>0,0,1</i>
<i>.byline</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0,1,0</i>
<i>p.byline</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>0,1,1</i>
<i>#banner</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1,0,0</i>
<i>#banner p</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>1,0,1</i>
<i>#banner .byline</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>1,1,0</i>
<i>a:link</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>0,1,1</i>
<i>p:first-line</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>0,0,2</i>
<i>h2 strong</i>	<i>0</i>	<i>0</i>	<i>2</i>	<i>0,0,2</i>
<i>#wrapper #content .byline a:hover</i>	<i>2</i>	<i>2</i>	<i>1</i>	<i>2,2,1</i>



EJEMPLO: Especificidad en selectores combinados

```
<nav>
  <ul class="c1">
    <li class="c2"><a href="link.html">Google</a></li>
    <li><a href="contacto.html">Contacto</a></li>
    <li><a href="#parrafoInteresante">Parrafo</a></li>
  </ul>
</nav>
```

```
li {
  list-style-type: circle;
}
nav ul li {
  list-style-type: rectangle;
}
.c1 li {
  color: red;
}
nav ul li {
  color: green;
}
a {
  color: inherit;
}
```

- Google
- Contacto
- Parrafo



<http://codepen.io/webUnicen/pen/bqmByo>

Cascada - Orden

ORDEN

En caso de que dos reglas tengan la misma importancia y especificidad, la regla que aparece “última” en el código CSS tiene mayor prioridad.

HTML

```
<p>Este es un parrafo</p>
```

CSS

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

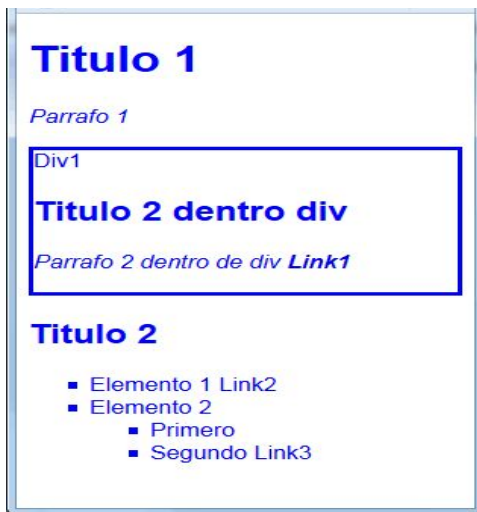
Misma importancia, misma especificidad, gana la última por **orden**.



<http://codepen.io/webUnicen/pen/XMYjoo>

Ejemplo

```
<body>
  <h1>Titulo 1</h1>
  <p>Parrafo 1</p>
  <div>Div1
    <h2>Titulo 2 dentro div</h2>
    <p>Parrafo 2 dentro de div <a href="...">Link1</a></p>
  </div>
  <h2>Titulo 2</h2>
  <ul>
    <li>Elemento 1 <a href="...">Link2</a></li>
    <li>Elemento 2
      <ul>
        <li>Primero</li>
        <li>Segundo <a href="...">Link3</a> </li>
      </ul>
    </li>
  </ul>
</body>
```



```
body {
  font-family: Arial;
  color: blue;
}

div {
  border: solid;
}

a {
  text-decoration: none;
}

p {
  font-style: italic;
}

p a {
  font-weight: bold;
}

ul {
  list-style-type: square;
}
```

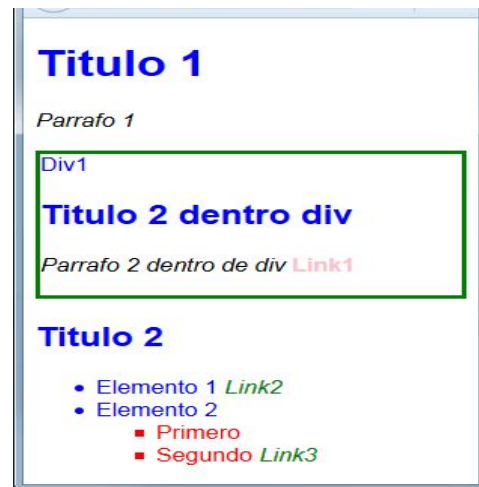
No hay colisiones, porque no coinciden las propiedades en CSS. Hay herencia de estilo.

Ejemplo 2

```
<body>
  <h1>Titulo 1</h1>
  <p>Parrafo 1</p>
  <div>Div1
    <h2>Titulo 2 dentro div</h2>
    <p>Parrafo 2 dentro de div <a href="#">Link1</a></p>
  </div>
  <h2>Titulo 2</h2>
  <ul>
    <li>Elemento 1 <a href="#">Link2</a></li>
    <li>Elemento 2
      <ul>
        <li>Primero</li>
        <li>Segundo <a href="#">Link3</a> </li>
      </ul>
    </li>
  </ul>
</body>
```

Hay colisiones:

¿Qué sucede?



```
body {
  font-family: Arial;
  color: blue;
}

div {
  border: solid;
  border-color: red;
}

div {
  border: solid;
  border-color: green;
}

a {
  text-decoration: none;
  font-style: italic;
  color: green;
}

p {
  font-style: italic;
  color: black;
}

p a {
  font-weight: bold;
  font-style: normal;
  color: pink;
}

ul li ul {
  list-style-type: square;
  color: red;
}
```


Como ver las colisiones en Chrome

Titulo 1

Parrafo 1

Div1

Titulo 2 dentro div

Parrafo 2 dentro de div [Link1](#)

Titulo 2

- Elemento 1 [Link2](#)
- Elemento 2
 - Primero
 - Segundo [Link3](#)

Se indican tachadas,
las propiedades que
colisionaron y no
fueron aplicadas

The screenshot shows the Chrome DevTools interface. The top bar includes tabs for Elements, Network, Sources, Timeline, Profiles, Resources, and Audits. The Elements panel on the right shows the DOM tree with the following structure:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Titulo 1</h1>
    <p>Parrafo 1</p>
    <div>
      "Div1"
      <h2>Titulo 2 dentro div</h2>
      <p>
        "Parrafo 2 dentro de div "
        <a href="...">Link1</a>
      </p>
    </div>
    <h2>Titulo 2</h2>
    <ul>
      <li>
        Elemento 1 Link2
        Elemento 2
          Primero
          Segundo Link3
      </li>
    </ul>
  </body>
</html>
```

The Styles panel on the right shows the CSS rules for the selected element (a link inside a paragraph within a div). The rules are:

```
element.style {
}

p a {
  font-weight: bold;
  font-style: normal;
  color: pink;
}

a {
  text-decoration: none;
  font-style: italic;
  color: green;
}
```

Green dashed lines indicate collisions between the rules. The 'font-style: italic;' property in the 'a' rule collides with the 'font-style: normal;' property in the 'p a' rule. The 'color: green;' property in the 'a' rule collides with the 'color: pink;' property in the 'p a' rule. The 'text-decoration: none;' property in the 'a' rule does not collide with any other rule.

Ejercicio: Herencia vs Cascada

Donde se usa Herencia o Cascada?

HTML

```
<p>Parrafo en Nivel 1</p>
```

```
<div>
```

```
  <p>Parrafo en <span> Nivel 2
```

```
  </span></p>
```

```
</div>
```

```
<p class='destacada' > Parrafo en Nivel
```

```
2</p>
```

```
<p id="unico">Parrafo Unico</p>
```

CSS

```
p {
```

```
  color: blue;
```

```
}
```

```
div {
```

```
  font-weight: bold;
```

```
}
```

```
span {
```

```
  color: cyan;
```

```
}
```

```
.destacada {
```

```
  color:green;
```

```
}
```

```
p.destacada {
```

```
  color:orange;
```

```
}
```

```
p#unico {
```

```
  color:red;
```

```
}
```

Parrafo en Nivel 1

Parrafo en Nivel 2

Parrafo en Nivel 2

Parrafo Unico



<https://codepen.io/webUnicen/pen/EEZNWK>

Buena Práctica

D.R.Y. – Don't Repeat Yourself

Un principio de desarrollo es D.R.Y., también conocido como “**no te repitas**”.

- En CSS es común escribir los mismos estilos una y otra vez, generando un código difícil de mantener y modificar.
- Un principio básico en CSS es escribir primero estilos comunes a la identidad visual del sitio, y luego usar técnicas avanzadas para sobreescribir solo aquellos que difieran del resto.

MÁS SELECTORES

Selectores de pseudo-clases y pseudo-elementos

Pseudo-classes

Las pseudo-classes se utilizan para definir un estado o comportamiento especial de un elemento.

Por ejemplo: se puede asignar un estilo cuando se pasa con el mouse encima de un texto

Sintaxis: **selector:pseudo-clase** {
 propiedad:valor;
 }

Ref: <https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

Pseudo-elemento

Un pseudo-elemento de CSS es usado para dar un estilo a una **parte** de un elemento. Pueden usarse combinados.

Por ejemplo: se puede asignar un estilo a la primer letra de un elemento, o a la primer línea.

Sintaxis: **selector::pseudo-elemento** {
 propiedad:valor;
 }

Comparación

En resumen:

- El pseudo selector cambia el “que elegimos” (ej: solo el primer elemento)
- Las pseudo clases cambia el “cuando” (ej: solo cuando paso el mouse).

Más usados:

- [::after](#)
- [::before](#)
- [:first-child](#)
- [:last-child](#)
- [::selection](#)
- [:link](#)
- [:focus](#)
- [:hover](#)
- [::first-letter](#)
- [::first-line](#)

Ejemplo

HTML

```
<a href="link.html">Este es un link</a>
```

```
<p>Etiam eu tristique enim, in molestie enim.
```

```
Phasellus dictum lobortis urna, vel placerat erat  
hendrerit luctus. Suspendisse rutrum, orci ut tincidunt  
consectetur, eros tellus suscipit leo, quis rutrum  
mauris lacus in magna.</p>
```

CSS

```
/* link */  
a:link {  
  color: #FF0000; }  
/* link visitado */  
a:visited {  
  color: #00FF00; }  
/* mouse sobre link */  
a:hover {  
  color: #FF00FF; }  
/* link seleccionado */  
a:active {  
  color: #0000FF; }  
p::first-letter {  
  color: #ff0000;  
  font-size: 40px; }  
p::first-line {  
  color: #0000ff;  
  font-variant: small-caps; }
```

Este es un link

Live: <http://codepen.io/webUnicen/pen/NpzRvi>

ETIAM EU TRISTIQUE ENIM, IN MOLESTIE ENIM. PHASELLUS DICTUM LOBORTIS URNA, VEL PLACERAT ERAT HENDRERIT LUCTUS. SUSPENDISSE RUTRUM, ORCI UT TINCIDUNT CONSECTETUR, EROS TELLUS SUSCIPIT LEO, QUIS RUTRUM MAURIS LACUS IN MAGNA.

Ejemplo 2

HTML

```
<p>Parrafo en Nivel 1</p>
```

```
<p class='destacada'>Parrafo en Nivel 2</p>
```

```
<p id="unico">Ultimo parrafo</p>
```

Parrafo en Nivel 1

Parrafo en Nivel 2

Ultimo parrafo

CSS

```
p{  
  color: blue;  
}  
p:last-child {  
  color: lime;  
}
```



Live: <https://codepen.io/webUnicen/pen/PRWbrL>

Ejercicio - Pseudo-Clases & Pseudo-Elements

Diseñe un botón para un link que debe:

- Cambiar levemente su color al pasar por arriba
- Cambiar de color al hacer click.

pseudo-elements

- Escriba un párrafo que
 - Tenga la primera letra siempre más grande y de color naranja.
 - Cuando seleccionas texto en ese párrafo, el color de selección tiene que ser rojo y las letras blancas.



AHORA LES TOCA PRACTICAR :D



Bibliografía

CSS

Standard: <https://www.w3.org/Style/CSS/>

MDN: <https://developer.mozilla.org/en-US/docs/Web/CSS>

W3 School: <http://www.w3schools.com/css/>

Libros Web: <http://librosweb.es/libro/css/>

Libros

HTML and CSS: Design and Build Websites (*Jon Duckett*)