

# CSS #2

Herencia, Selectores combinados, Cascada

# Qué sabemos hasta ahora de CSS?

---

- Sirve para aplicar el estilo a una página web.
- Se escribe en un archivo aparte del HTML con extensión .css
- Para seleccionar elementos de HTML tenemos 3 tipos de selectores:
  - Tipo (p, h1, div, etc.)
  - Clase
  - Id



# Selector Universal

---

- Se utiliza para seleccionar todos los elementos de la página.
- El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML.

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

# Herencia

---

**¿A que les suena herencia?**



# Herencia

---

- La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos
- No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran.

Por ejemplo, los relacionados al tamaño de los elementos.

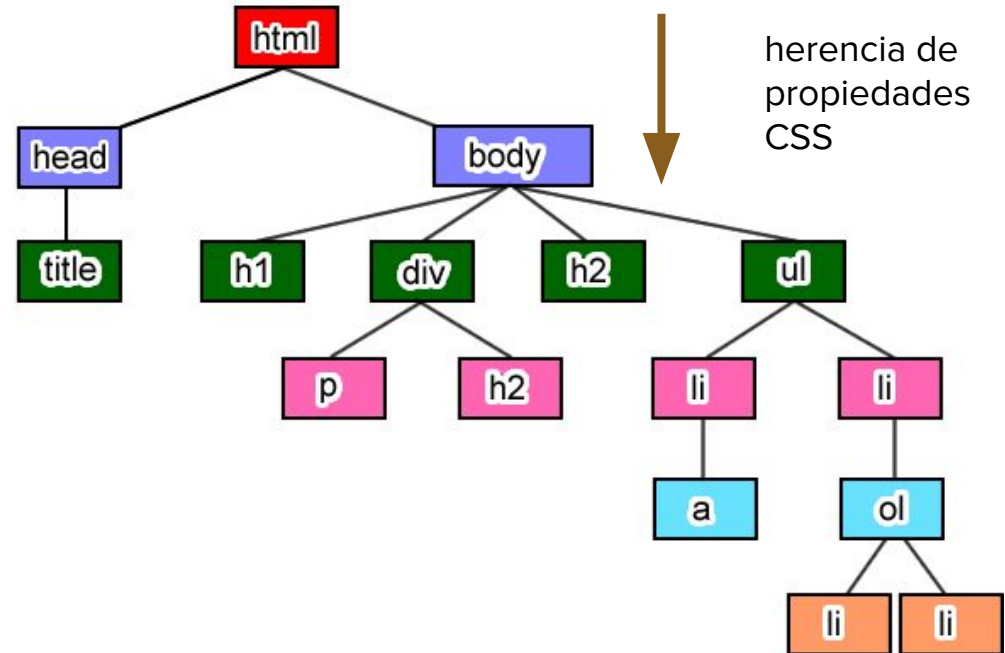
Lista completa de las propiedades de CSS, sí se heredan o no.

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

# Arbol HTML - DOM

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>
```



# 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 del div, que lo hereda del body, que lo hereda del html\*/*

## 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;
}
```



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

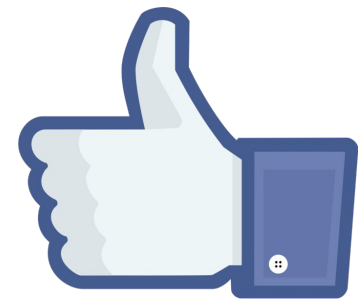


## Para que nos sirve entonces la Herencia?

### **Escribir menos código.**

Si ponemos la font-family al elemento HTML

- todos los elementos lo heredan
- no tengo que reescribirlo para que otros lo tengan
- más mantenible
- menos redundante (menos código repetido)
  - la redundancia lleva a inconsistencia



# Selectores combinados

---

# Pregunta

---

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



# Combinación de Selectores

Se pueden combinar selecciones para hacerlas más específicas.

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

*Elige los párrafos que contengan la clase “destacado”.*

De lo anterior se deduce que el atributo .destacado es equivalente a \*.destacado, por costumbre todos obvian el símbolo \* al escribir un selector de clase normal.

Lo mismo se aplica a Ids.

```
p#uno { font-size: 25px; }
```

**Tiene  
sentido?**

## ¿Cuándo voy a usar esto?

(Ejemplo realizado por un fanático de Chevrolet :)

La segunda fecha del Turismo Carretera en Neuquén ya es pasado y el campeonato tuvo modificaciones pero en el segundo puesto para atrás, ya que Christian Ledesma sigue como puntero, pero el nuevo escolta es Jonatan Castellano.

### CAMPEONATO TC - 2018

1 Ledesma, Christian Chevrolet 79,5

2 Castellano, Jonatan Dodge 78,5

Ejemplo: <https://www.ole.com.ar/autos/autos-campeonato-tc>

# Ejemplo

```
<h1>Asi esta el Campeonato</h1>
<p>El campeonato recién comenzó...</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>
```

```
p.destacado {
  color:green;
}
.especial {
text-decoration:line-through;
}
li#unico {
  color:orange;
}
```

## Asi esta el Campeonato

El campeonato recién comenzó...

Chevrolet ya va primero.

Nunca va a ser superado por una marca menor. Ford



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

# Selectores anidados

---

# Pregunta

---

**¿Cómo hacemos para aplicar un estilo solo a los párrafos dentro de un div?**

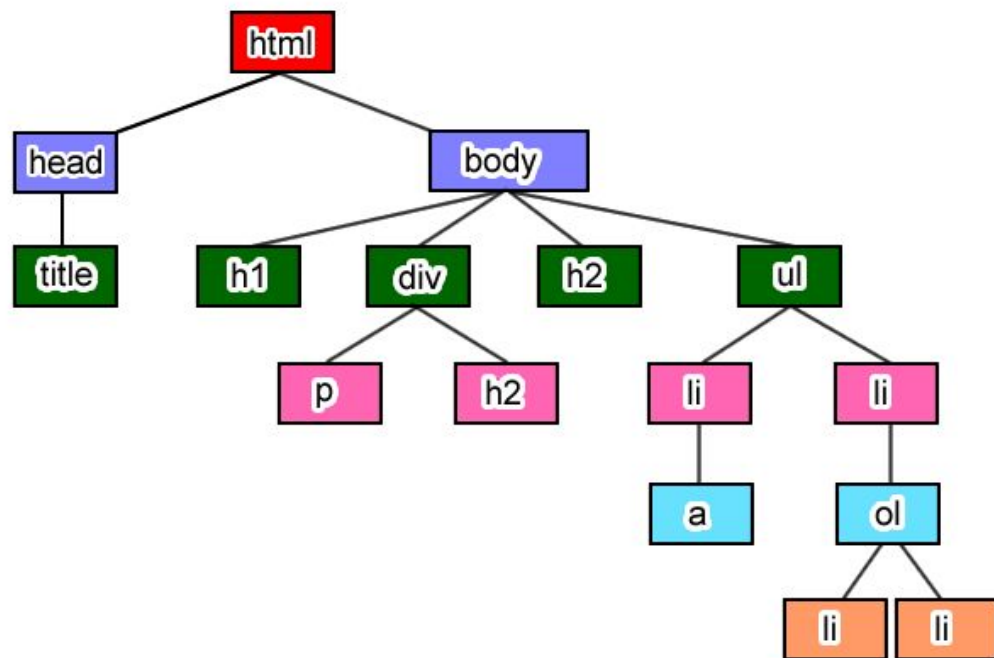




# Arbol HTML - DOM

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 Anidados

---

Permite seleccionar elementos contenidos dentro de otros elementos. Así se puede aumentar el nivel de detalle.

Selecciona los span que estén dentro de algún párrafo (incluye si está contenido indirectamente).

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

Aplica estilo a los links que se encuentren dentro de un item de lista no-ordenada

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

# Ejemplo

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

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

Este es un Link

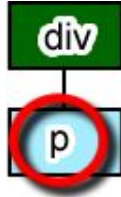
- Elemento 1
- Elemento 2
- Elemento 3



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

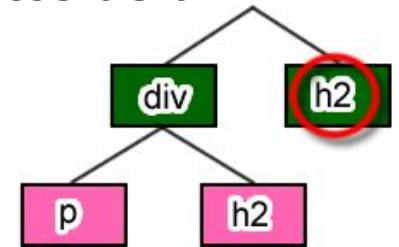
# Combinación de Selectores

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



Selecciona **todos** los elementos <p> que son hijos directos de un elemento <div>.

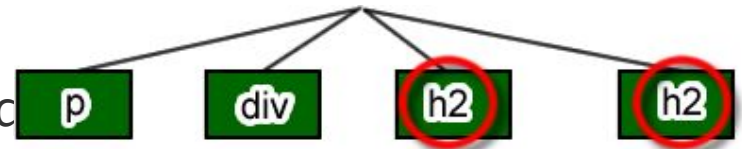
```
div + h2 {  
  background-color: red; }
```



Selecciona todos los elementos <h2> que están inmediatamente a continuación de un div. Sería el hermano <h2> siguiente adyacente a <div>

Selecciona todos los hermanos de <p> que

```
background-color: orange; }
```



# Ejemplo - Selectores Combinados

```
<h1>Titulo 1</h1>
<div>Div 1
  <p>Parrafo dentro de un div</p>
  <h2>Titulo dentro de un Div</h2>
</div>
<h2>Titulo 2</h2>
<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

Div 1

Parrafo dentro de un div

**Titulo dentro de un Div**

## Titulo 2

- Elemento 1 [Link 1](#)
- Elemento 2
  1. Primero
  2. Segundo



Live: <http://codepen.io/ignaciojonas/pen/JXrqqi>


# Mejorando el código CSS

---

**¿Cómo hacemos para aplicar el mismo estilo a múltiples elementos?**



# Grupo de Selectores

- Se pueden usar varios selectores juntos.
- Esto permite **evitar duplicación de estilos**. 
- Además se pueden refinar las diferencias aparte
- Se separan los selectores con ',' creando un grupo de selectores con propiedades en común

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

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

DEMO

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



# Ejemplo - Duplicación de Estilos

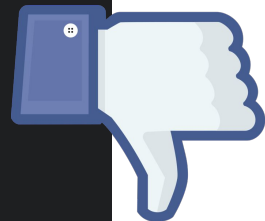
## Esta hoja de estilos se puede mejorar?

```
<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 id='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>
```

```

CSS
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
11 p {
12   font-family: Helvetica;
13   color: blue;
14 }
15 img {
16   width: 300px;
17 }
18 .caption-image{
19   font-family: Helvetica;
20   color:black;
21   font-style: italic;
22 }
23
```



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

# Ejemplo - Estilos sin duplicación

- Misma Font-Family para toda la página
- Color azul unificado para los tipos de selectores
- Font Style del caption heredad

`<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>`

`<img`

`src="http://s2.quickmeme.com/img/0e/0e2d0d53ed6d5ca7ea  
026a8c6592a7c414ba0e9b7e9aa1a02a851b5a058ddeaf.jpg"`

`alt="css-meme" />`

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

```

CSS
1 * h1, h2, p, .caption-image {
2     font-family: Helvetica;
3     color: blue;
4 }
5
6 * h2, .caption-image {
7     font-style: italic;
8 }
9
10 * img {
11     width: 300px;
12 }
13 * .caption-image{
14     color:black;
15 }
16
17 * p.unico {
18     color: green;
19 }
20
```



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

**¿Qué ventajas ven de re-usar estilos?**





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;  
}
```

# Cascada

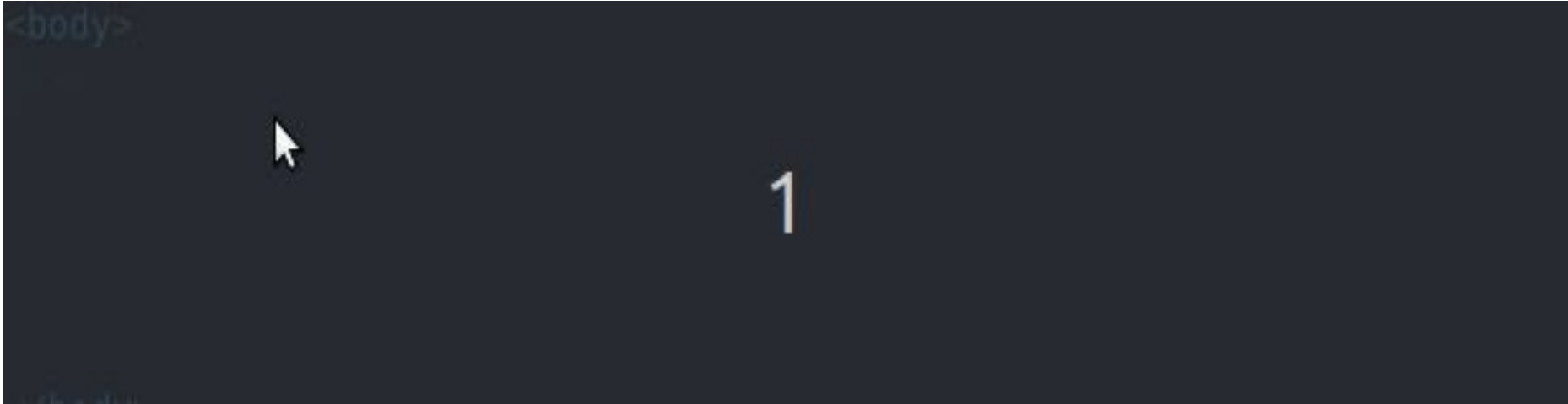
---

# Asignar dos clases a un mismo elemento

```
<div class="resaltado redondeado">
```

```
</div>
```

Podemos usar el autocompletar con el selector CSS

A screenshot of a dark-themed code editor. The text "<body>" is visible in the top left corner in a light blue font. A white mouse cursor is positioned in the middle of the editor. A large white number "1" is centered in the lower half of the editor. The editor background is dark gray, and the text is light gray.

```
<body>
```

1

# Pregunta

---

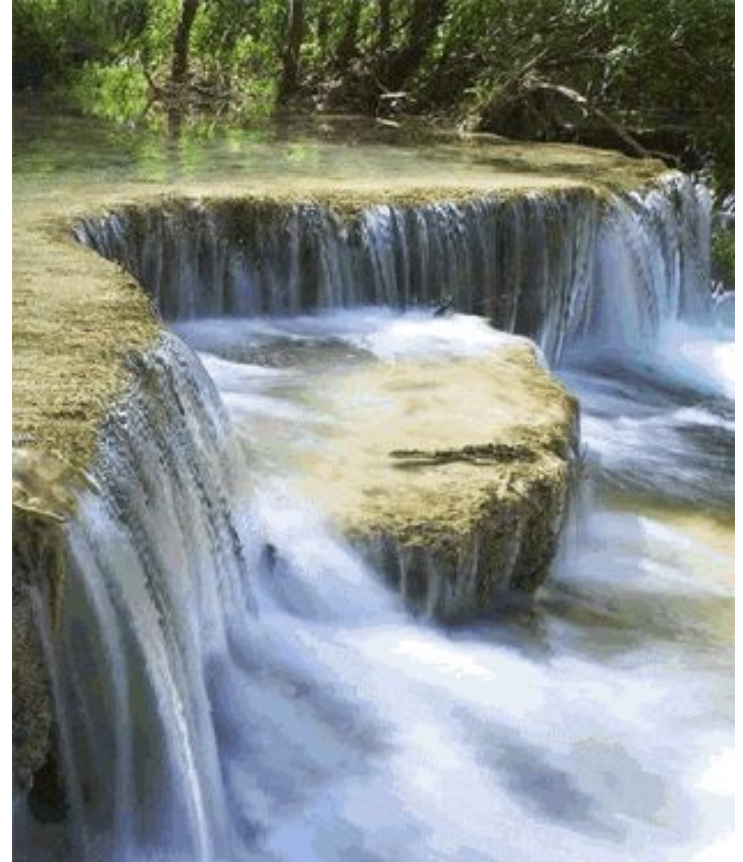
**¿Que pasa cuando a un elemento tiene 2 estilos definidos pero diferentes?**



# Cascada

---

- CSS significa cascading style sheets (hojas de estilo en cascada)
- La cascada es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.

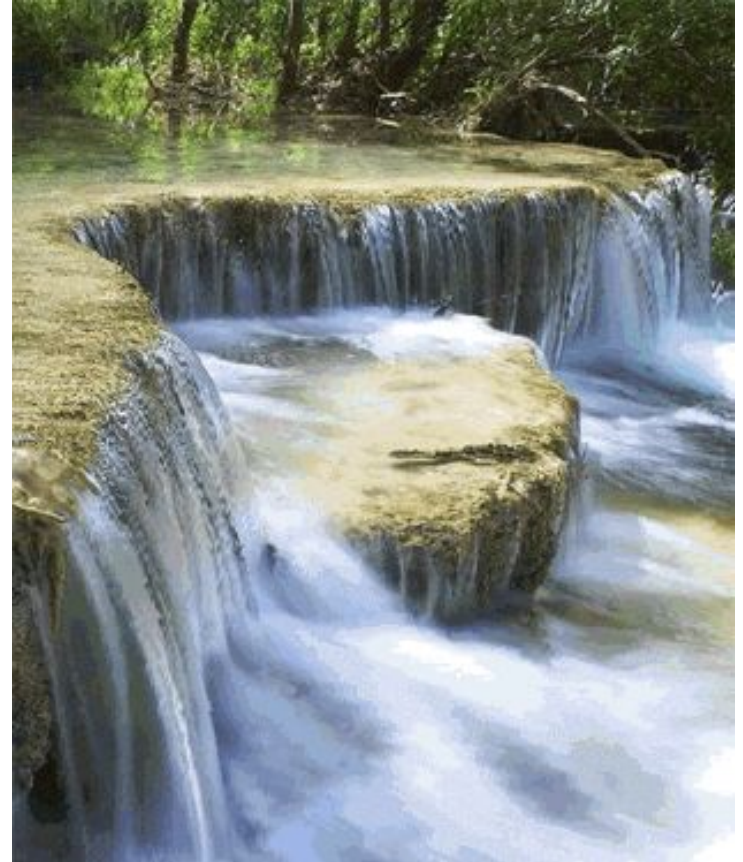




# Cascada

---

- Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:
  - Importancia.
  - Especificidad.
  - Orden en el código fuente.



# Cascada - Importancia

---

El orden en que se aplica la cascada es:

1. Hoja de estilos default del navegador.
2. Declaraciones normales en hojas de estilo de **usuario**.
3. Declaraciones normales en hojas de estilo.
4. Declaraciones importantes en hojas de estilo.
5. Declaraciones importantes en hojas de estilo de **usuario**.

*Notas:*

***Hojas de estilo de usuario:*** Son estilos que se aplican en los navegadores por los usuarios, por ejemplo, una persona disléxica puede tener una hoja de estilo de usuario que especifique determinadas fuentes y colores que le faciliten la lectura.

# Cascada - Importancia

La palabra reservada **!important**, sobrescribe toda cascada.

## HTML

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

## CSS:

```
* {  
    color: red !important;  
}  
  
p {  
    color: blue;  
}
```

Este es un parrafo

DEMO

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

# Cascada - Especificidad

Más específico es el selector, entonces ese es el estilo que se aplica.

- Lo que tiene clases es más específico que lo que no.
- En un selector anidado, si tienen la misma cantidad de clases, gana el que tiene más elementos para cumplir.

## HTML

```
<p>Este es un parrafo</p>  
<p class='destacada' id='unico'> Este es un parrafo</p>
```

## CSS

```
p {  
  color: blue; }  
#unico {  
  color:red; }  
.destacada {  
  color:green; }  
p.destacada {  
  color:orange; }  
p#unico{  
  color:pink; }
```



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

# Especificidad en selectores anidados

## HTML

```
<nav>
  <ul class="c1">
    <li class="c2"><a
href="https://www.google.com"
target="_blank">Google</a></li>
    <li><a
href="contacto.html">Contacto</a></li>
    <li><a
href="#parrafoInteresante">Parrafo
Interesante</a></li>
  </ul>
</nav>
```

## CSS

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

nav ul li {
  list-style-type: circle;
}

.c1 li {
  color: red;
}

nav ul li {
  color: green;
}

a {
  color: inherit;
}
```



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

# Cascada - Orden

Si dos declaraciones afectan al mismo elemento, tienen la misma importancia y la misma especificidad, la selección final es el orden en las fuentes.

La declaración que se ve después en las hojas de estilo "ganará" a las anteriores.

HTML

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

CSS

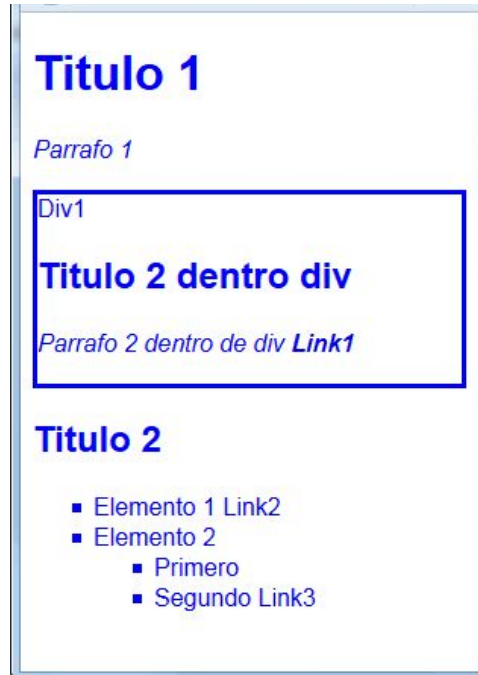
```
p {  
  color: red;  
}  
  
p {  
  color: blue;  
}
```



Live: <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;
}
```

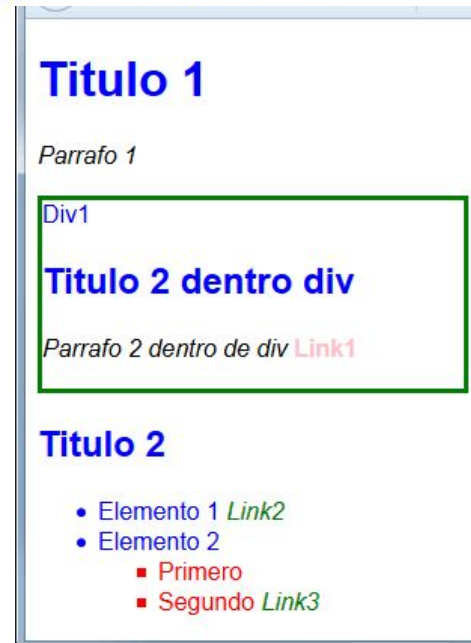
**No hay colisiones**, porque no coinciden las propiedades en CSS. Hay herencia de estilo.

# 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 with the 'Elements' panel on the left and the 'Styles' panel on the right. The 'Elements' panel 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 shows the CSS rules for the selected element (a link inside a paragraph inside 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;
}
```

The 'a' rule is highlighted in blue. Red dotted arrows point from the text 'Se indican tachadas, las propiedades que colisionaron y no fueron aplicadas' to the 'font-style: italic;' and 'color: green;' properties in the 'a' rule, which are crossed out with red lines. This indicates that these properties are overridden by the 'p a' rule's 'font-style: normal;' and 'color: pink;'.

# Buena Práctica

---

## **D.R.Y.** – Don't Repeat Yourself

One principle of development is D.R.Y., also known as don't repeat yourself. Within CSS this principle can speak volumes as it is easy to continually write the same styles over and over again. Don't. CSS was designed in a way to allow you to cascade styles and use classes so that you easily apply and inherit styles. The end goal is to write clean and light code, of which is semantic and easily managed.



# Ejercicio

---

Realicen una página:

- Elementos anidados.
- Utilizar selectores anidados para dar estilos.
- Probar colisiones y ver qué estilo se impone.



# Herencia vs Cascada

---

# 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>`

Parrafo en Nivel 1

Parrafo en Nivel 2

Parrafo en Nivel 2

Parrafo Unico

### CSS

`p{`

`color: blue; }`

`div{`

`font-weight: bold; }`

`span{`

`color: cyan; }`

`.destacada {`

`color:green; }`

`p.destacada {`

`color:orange; }`

`p#unico{`

`color:red; }`

DEMO

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

# Herencia vs Cascada

Donde se usa Herencia o Cascada?



# Contenedores y CSS

---

## Fuentes - Tamaño

---

- La propiedad **font-size** setea el tamaño de la fuente.
- Puede ser, **absoluto** o **relativo**.



# Fuentes - Tamaño Absoluto

---

- Ajusta el texto con un tamaño específico.
- No se puede cambiar el tamaño de letra en todos los navegadores.
- Es útil cuando se sabe dónde se va a mostrar.

## Posibles Valores

- xx-small
- x-small
- small
- Medium
- Large
- x-large
- xx-large

# Fuentes - Tamaño Relativo

---

- Ajusta un tamaño relativo alrededor de los elementos.
- Permite al usuario cambiar la letra de los browsers.
- Si no se especifica el tamaño de la fuente, es por defecto 16px.

## Posibles Valores

- larger
- smaller

# Fuentes - Tamaño

---

Otra opción es usar valores numéricos:

**Pixeles (px):** Es **absoluto**, indica al browser los pixeles que tiene que medir la fuente. **No es accessible**

**Em (em):** Es **relativo** al parent. Por default 1 em = 16 px , 2em = 32px.

**Rem (rem):** Es **relativo** al root (html).

**Porcentaje:** Es **relativo**, un porcentaje del tamaño del elemento padre.

```
p { font-size: xx-large }
```

```
h1 { font-size: 250% }
```

```
span { font-size: 16px; }
```

# Fuentes - Tamaño Relativo

- Ejemplo: Cada DIV usa fuente del doble de su contenedor.

## HTML

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

## CSS

```
div {
  font-size: 2em;
}
```

Título suelto

Un párrafo de texto.

**Título div**

**Título div div**

Un párrafo de texto.



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

# Fuentes - Tamaño Relativo al Root

- Ejemplo: Con rem, los elementos anidados permanecen de igual tamaño respecto al root

## CSS

```
div {  
  font-size: 2rem;  
}
```

Título suelto

Un párrafo de texto.

Título div

Título div div

Un párrafo de texto.

## HTML

```
<h1>Título suelto</h1>
```

```
<p>Un párrafo de texto.</p>
```

```
<div>
```

```
<h1>Título div</h1>
```

```
<div>
```

```
<h1>Título div div</h1>
```

```
<p>Un párrafo de texto.</p>
```

```
</div>
```

```
</div>
```

Igual tamaño



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

# Pseudo Clases y PseudoElementos

---

# Pseudo-clases

---

Las pseudo-clases 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;  
}
```

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



# 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>
```

Este es un link

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 M

## 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; }
```

DEMO

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

# Ejemplo 2

## HTML

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

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

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

## CSS

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

Parrafo en Nivel 1

Parrafo en Nivel 2

Ultimo parrafo



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*)