CSS

CFP Programador full-stack

Herencia, Selectores combinados, Cascada

Repaso 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

Herencia

CFP Programador full-stack

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.

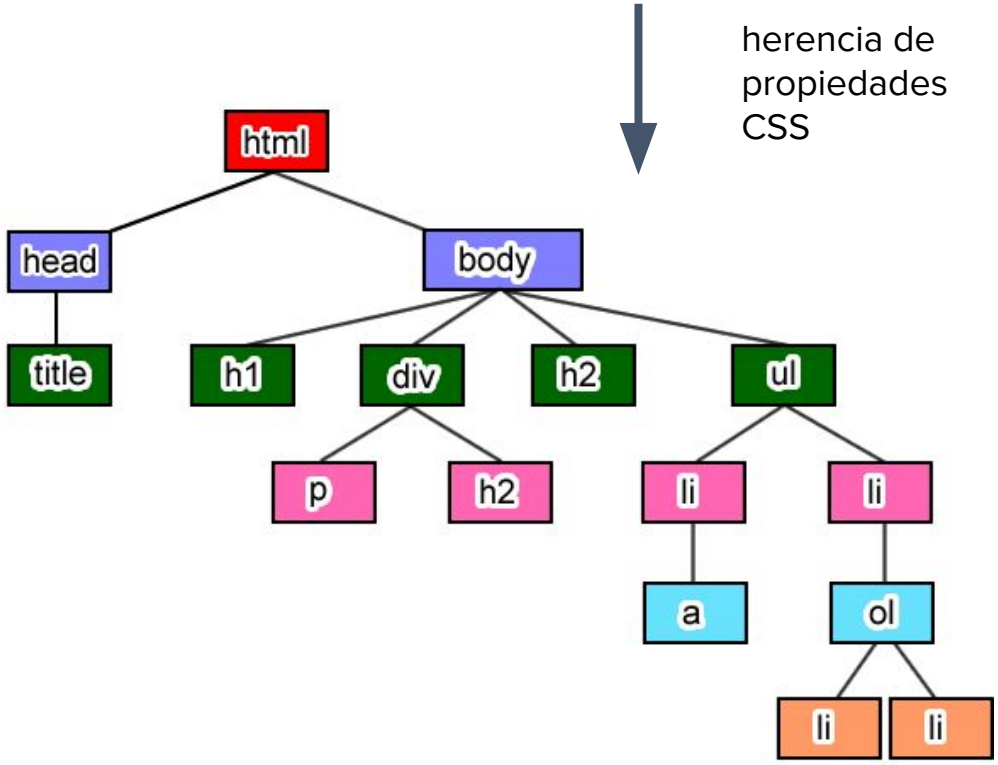
Lista completa de las propiedades de CSS, sí se heredan o no.

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

Herencia

Para qué 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

CFP
Programador
full-stack

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.

Ejemplo

```
<h1>Asi esta el Campeonato</h1>
<p>El campeonato recien 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>
```

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

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

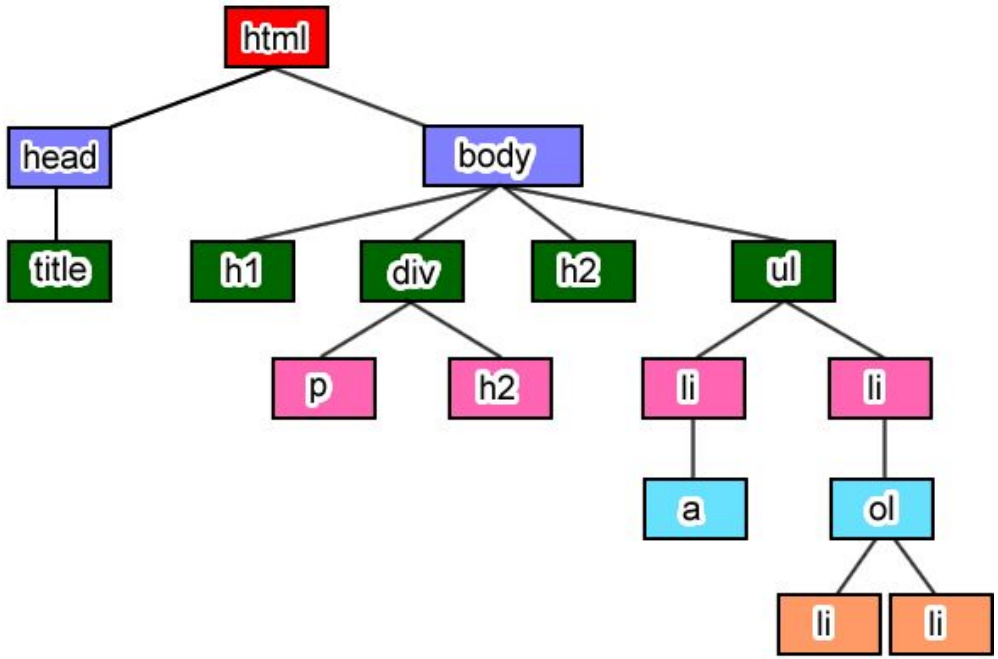
Selectores anidados

CFP Programador full-stack

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; }
```

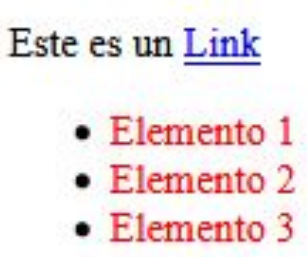
Aplica estilo a los links que se encuentren dentro de un item de lista no-ordenada

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

Ejemplo

```html
<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>
```

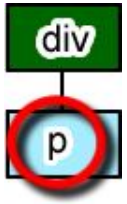
```css
ul li a {
  text-decoration:none;
  color:red;
}
```



DEMO Live: http://codepen.io/webUnicen/pen/gmKwKM

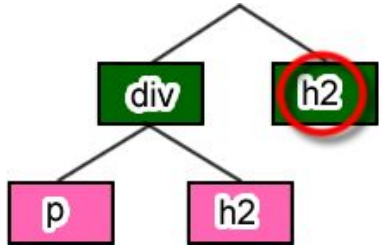
Combinación de Selectores

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



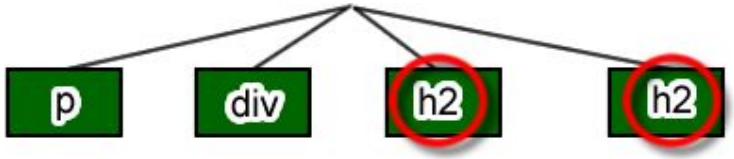
Selecciona todos los elementos <p> que son hijos inmediatos (el 1er hijo) de un elemento .

```
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

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



Selecciona todos los hermanos de <p> que sean <n2>

Ejemplo - **Selectores Combinados**

```html
<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>
```

```css
div > p {
  background-color:yellow;  }
div + h2 {
  background-color:red;  }
p ~ h2{
  background-color: orange;   }
 div {
  border: solid;   }
```

DEMO Live: http://codepen.io/ignaciojonas/pen/JXrqqj

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

- Las pseudo clases cambia el “cuando” se aplica un selector (ej: solo cuando paso el mouse).

Sintaxis:

```
selector:pseudo-clase {
  propiedad:valor;
}
```

```
.boton:hover {
  background-color: red;
}
```

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

Cascada

CFP Programador full-stack

Cascada

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

- 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**

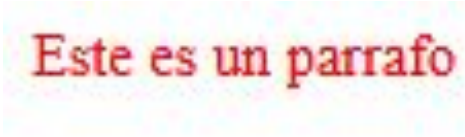
La palabra reservada **!important**, sobrescribe toda cascada.

HTML

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

CSS:

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



DEMO 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.
- 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;  }
```

http://codepen.io/webUnicen/pen/peKEqL

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>
```

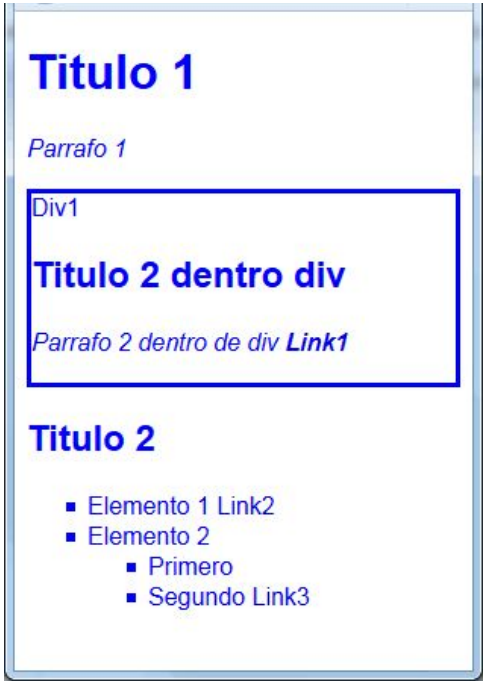
CSS

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

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

Ejemplo

```html
<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>
```




```css
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;
}
```

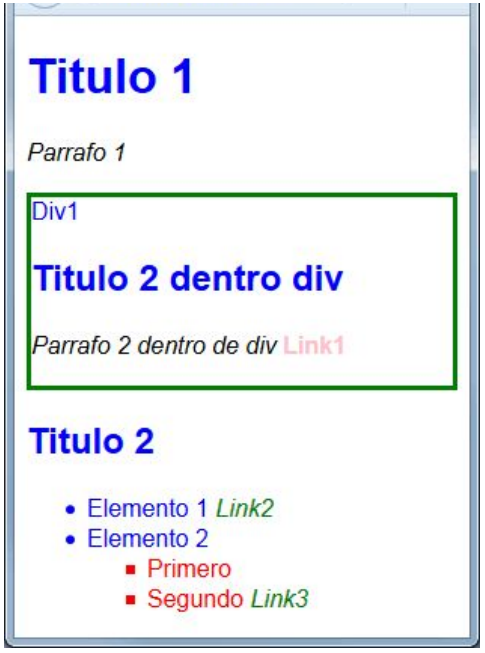
No hay colisiones, porque no coinciden las propiedades en CSS. Hay herencia de estilo.

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

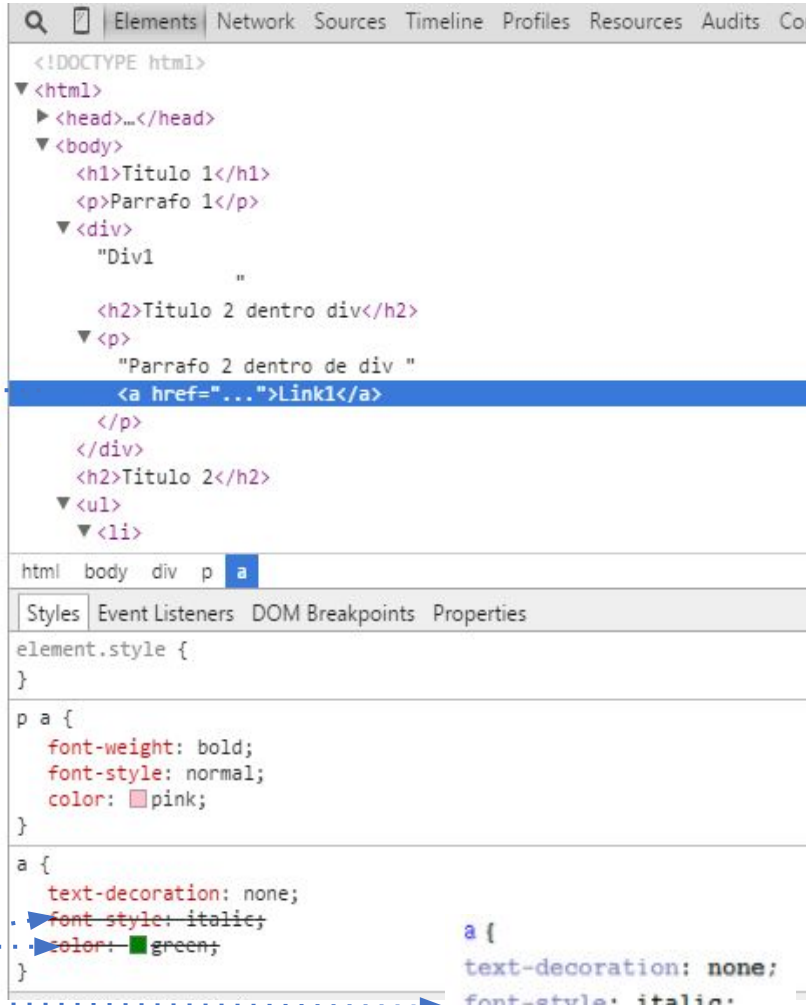
Div1

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

Mejorando el código CSS

CFP Programador full-stack

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

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

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

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

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 inherent styles. The end goal is to write clean and light code, of which is semantic and easily managed.

Ejercicios

CFP Programador full-stack

Ejercicio

Realicen una página:

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

Herencia vs Cascada

Donde se usa Herencia o Cascada?

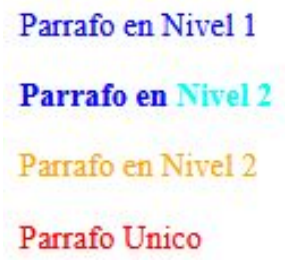
HTML

```
<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

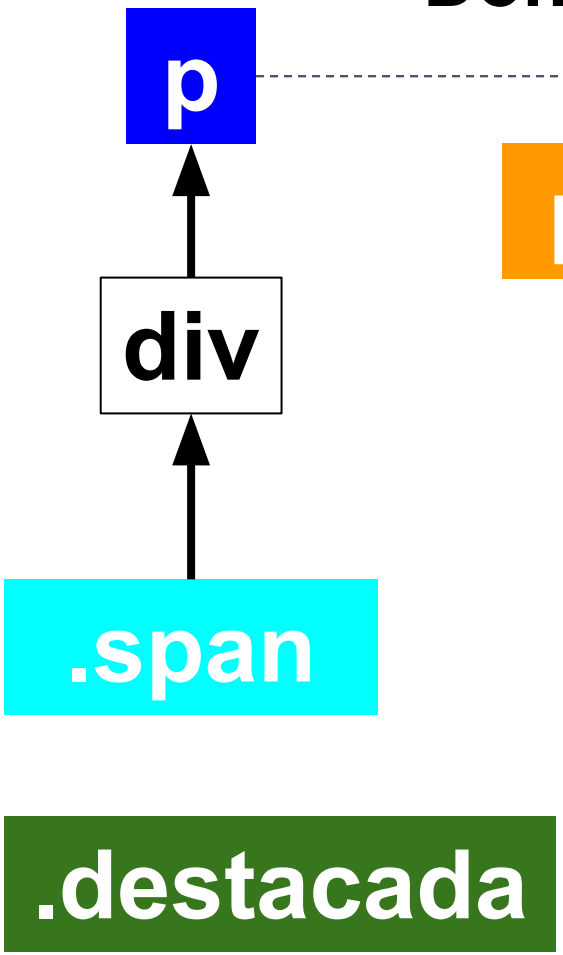
```
p{
  color: blue;  }
div{
   font-weight: bold;  }
span{
  color: cyan;  }
.destacada {
  color:green;  }
p.destacada {
  color:orange;  }
p#unico{
  color:red;  }
```



DEMO Live: https://codepen.io/webUnicen/pen/EEZNWK

Herencia vs Cascada

Donde se usa Herencia o Cascada?



p.destacada

p#unico

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;     }
```

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.