

Introducción

En esta sección veremos los fundamentos de las hojas de estilo en cascada (CSS). Comenzaremos con los conceptos básicos (Parte 01), pasaremos a la resolución de conflictos de reglas, el modelo de caja y las técnicas de diseño CSS (Parte 02). Luego, aprenderemos cómo usar consultas de medios y sitios web sensibles (responsive web sites) al código sin ningún framework (Parte 03).

Conceptualización

HTML define la estructura del contenido de una página. Entonces, si todo lo que necesitamos es el contenido sin procesar, podríamos tomar los archivos HTML, desplegarlos en un servidor y listo. Sin embargo, eso sería un gran error. Existe una frase entre los diseñadores web que dice así: ¡¡¡El contenido es el rey!!!. La forma en que los usuarios absorben y se relacionan con ese contenido depende en gran medida de la experiencia de la presentación del mismo. Cuando se crea un sitio web real, la estructura por sí sola no es suficiente. Es necesario diseñar su contenido de una manera que sea agradable y útil para el usuario. El uso del color, el posicionamiento y el tamaño, entre otras cosas es un factor clave. Las hojas de estilo en cascada, o CSS, es la tecnología que proporciona la capacidad de estilo. CSS es una tecnología increíblemente poderosa. Puede tomar la misma estructura HTML de un documento y presentarla de formas tan drásticamente diferentes que nunca adivinaría que la estructura subyacente del contenido es exactamente la misma. Observe los siguientes ejemplos que ayudan a comprobar esta afirmación:





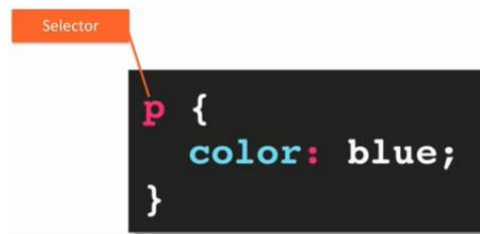
Observe que el contenido de las tres páginas es el mismo, lo único que cambian es el estilo. No cambian la estructura o el contenido del HTML en absoluto. Entonces queda en evidencia lo que el estilo puede hacer en un sitio web, a través del mismo contenido, a través de la misma estructura HTML, pero con CSS personalizados tan diferentes en el resultado final que pareciera que no trataran el mismo contenido.

CSS es una tecnología increíblemente poderosa para diseñar el contenido de una página que permite crear una experiencia de usuario particular que influye notoriamente en el disfrute y entendimiento del contenido. Por lo tanto, la experiencia del usuario del contenido es importante. Entonces, todo lo que queda es descubrir cómo escribir código CSS.

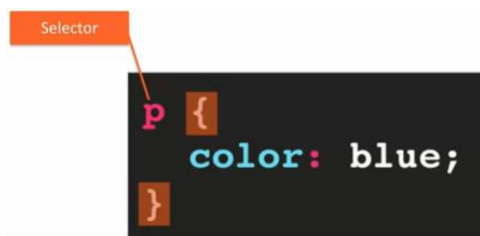
Anatomía de una regla CSS

CSS funciona asociando reglas a los elementos HTML. Estas reglas indican cómo se debe mostrar el contenido de los elementos especificados. Y aunque diseñar una página web completa podría ser un proceso bastante complejo, definir una simple regla CSS es bastante sencillo.

Iremos identificando las partes de la regla en base a un ejemplo sencillo. Observe la siguiente imagen



Una regla CSS consiste en un selector, en este caso es una p, que hace referencia al <p>. Esto significa que el selector indica que la regla de estilo se aplicará al contenido de cualquier etiqueta <p> dentro de la página HTML. El selector es seguido por llaves abiertas y cerradas



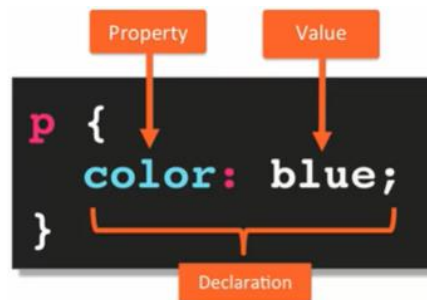
Dentro de esas llaves, se realizan las declaraciones CSS y en este ejemplo en particular solo poseemos una declaración CSS:

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



Una declaración CSS consta de dos partes, la propiedad y el valor

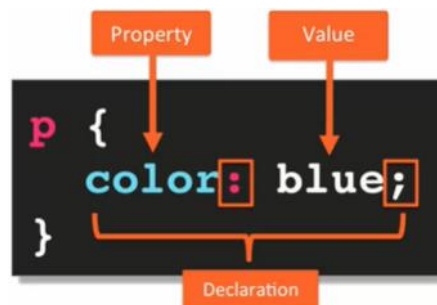
```
p {  
  color: blue;  
}
```



El nombre de la propiedad está predefinido por la especificación CSS, y para cada propiedad hay un número establecido de valores predefinidos (que pueden ser caracteres o valores numéricos).

Observe además que la propiedad está separada de su correspondiente valor mediante dos puntos y además; termina con un punto y coma. Ahora, técnicamente hablando, el punto y coma no es un requisito, pero es una buena práctica, y siempre deberías usarlo

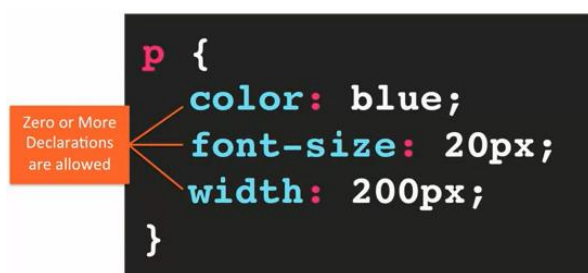
```
p {  
  color: blue;  
}
```



Por último, normalmente será fácil interpretar sobre que aspecto del contenido actúa la propiedad definida; por ejemplo, en este caso la propiedad color indica de qué color será el texto del párrafo mientras que el valor indica que será de color azul.

Como se mencionó anteriormente, una regla CSS puede contener varias declaraciones. Observe el siguiente ejemplo:

```
p {  
  color: blue;  
  font-size: 20px;  
  width: 200px;  
}
```



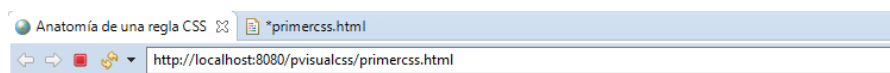
Por ejemplo, en este caso, estamos especificando que el contenido de las etiquetas de párrafo en nuestra página HTML debe tener un color particular, un tamaño de fuente particular y también un ancho particular (lo que significa que el párrafo en sí, el contenido, debe ocupar 200 píxeles). Ahora, el conjunto de reglas CSS se denomina hoja de estilo.

Veamos un ejemplo de contenido en una página HTML y su hoja de estilo:

```

Anatomía de una regla CSS  *primercss.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Anatomía de una regla CSS</title>
6     <style>
7       p {
8         color: blue;
9         font-size: 20px;
10        width: 400px;
11      }
12
13      h1 {
14        color: green;
15        font-size: 36px;
16        text-align: center;
17      }
18    </style>
19  </head>
20  <body>
21    <h1>Anatomía de un Regla CSS</h1>
22    <h2>Sub encabezado 1</h2>
23    <p>CSS trabaja asociando reglas a los elementos HTML. Estas reglas
24    indican como será visualizado el contenido de esos elementos.
25    Una regla CSS contiene dos partes: un selector y una declaración.</p>
26    <h2>Su encabezado 2</h2>
27    <p>Las declaraciones constan de una propiedad y el correspondiente valor asignado
28    a esa propiedad. Tanto la propiedad como su valor están predefenidos
29    en CSS. Pueden existir varias declaraciones en una regla CSS.
30    </p>
31  </body>
32 </html>
    
```

Como puede observar mediante el tag `<style>` definido entre la línea 6 y 18 se pueden declarar las reglas de estilo en la misma página. Además, esta etiqueta se define dentro del tag `<head>`. ¿Puede identificar que se están definiendo dos reglas CSS? Una para los elementos `p` y otra para los elementos `h1`. Si ejecuta esta página puede observar también claramente el efecto de cada una de las declaraciones dentro de cada regla



Anatomía de un Regla CSS

Sub encabezado 1

CSS trabaja asociando reglas a los elementos HTML. Estas reglas indican como será visualizado el contenido de esos elementos. Una regla CSS contiene dos partes: un selector y una declaración.

Su encabezado 2

Las declaraciones constan de una propiedad y el correspondiente valor asignado a esa propiedad. Tanto la propiedad como su valor están predefenidos en CSS. Pueden existir varias declaraciones en una regla CSS.

Como puede ver, nuestro encabezado de nivel 1 está ubicado en medio de la pantalla y el texto es de color verde. Y los dos párrafos definidos son azules y solo ocupan alrededor de 400 píxeles. Se puede confirmar que el estilo aplicado al párrafo afecta a todos los párrafos de la página. Por lo tanto, queda evidente la consistencia y también la eficiencia del código CSS. Ahora, quiero señalar una cosa rápida pero importante. Observe que los encabezados de nivel 2 poseen su propio estilo predefinido por el browser. Cada browser cuenta con algunos estilos predeterminados que se aplican a diferentes elementos HTML. Por lo tanto, muchas veces será tarea del diseñador web restablecer o anular estos estilos, así como poner los estilos que desea que tenga su página web.

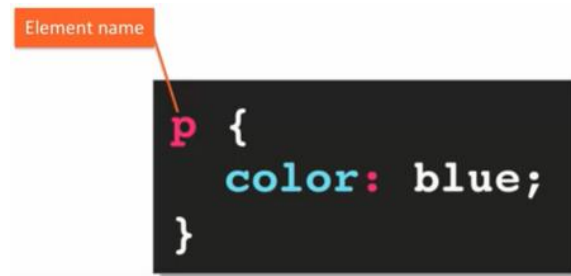
Trabajando con los selectores CSS

Los selectores CSS se utilizan para determinar a qué elemento HTML, o conjunto de elementos, se aplicarán las declaraciones CSS.

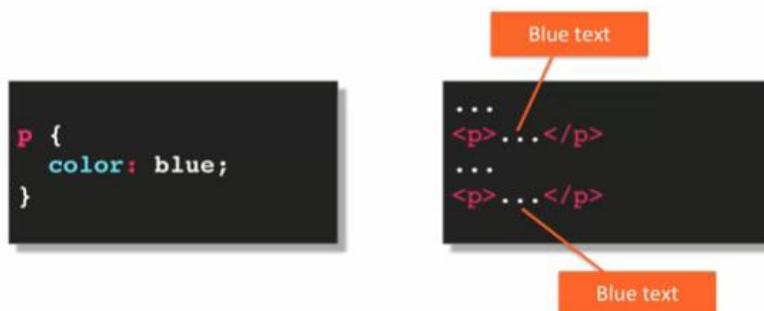
El browser usa su API de selectores a través del Modelo de Objeto de Documento (DOM) para seleccionar los elementos que coinciden con el selector. Vamos a trabajar con tres tipos diferentes de selectores, elemento, clase y selector id.

El selector de elementos

El selector de elementos puede ser utilizado para indicar que la declaración se aplica a un elemento, para ello simplemente se especifica el nombre del elemento. Por ejemplo:



En este caso, al indicar el selector usando como nombre el elemento `p`, entonces todos los párrafos del documento HTML serán de color azul. El siguiente esquema indica lo que recién hemos comentado



Cada párrafo en el documento HTML tendrá el texto de su contenido en azul. Obviamente, esto realmente no afecta a ningún otro elemento que contenga texto en ellos. Por ejemplo, podemos tener un elemento `div` que contenga texto, pero ese texto no se verá afectado por nuestra regla CSS.

El selector de clase

Se especifica con un punto y el nombre de la clase. Observemos como se especifica en un ejemplo:

```
Class name
.blue {
  color: blue;
}
```

En este caso, estamos creando una clase denominada `blue`, una clase CSS, que coloreará con azul aquello que sea referenciado con esta clase.

El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo `class`, cuyo valor será el nombre del selector de clase. Observe el siguiente esquema para experimentar como funciona lo comentado:

```
Defined with .
.blue {
  color: blue;
}

Used without
...
<p class="blue">...</p>
<p>...</p>
<div class="blue">...</div>
...
```

Tenga en cuenta las reglas para la definición y uso del selector de clase. El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo `class` debe recibir como valor el nombre del selector de clase SIN el punto.

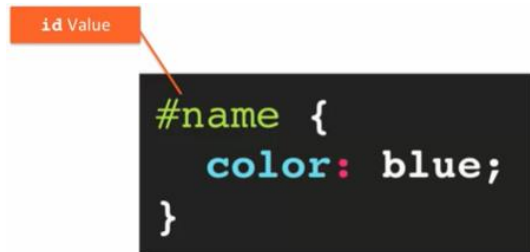
El primer tag `<p>` y el tag `<div>` poseen el atributo `class = "blue"`, por lo tanto, su contenido será de color azul. Observe que hay un tag `<p>` que no tiene especificado el atributo `class`, por lo que no se ve afectado por el estilo.

```
Blue text
...
<p class="blue">...</p>
<p>...</p>
<div class="blue">...</div>
...

Unaffected
Blue text
```

El selector id

La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo `id` de una etiqueta), precedido por un signo numeral (`#`). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML.



```
#name {  
  color: blue;  
}
```

Ahora observemos un esquema de la forma en que se utiliza este tipo de selector:

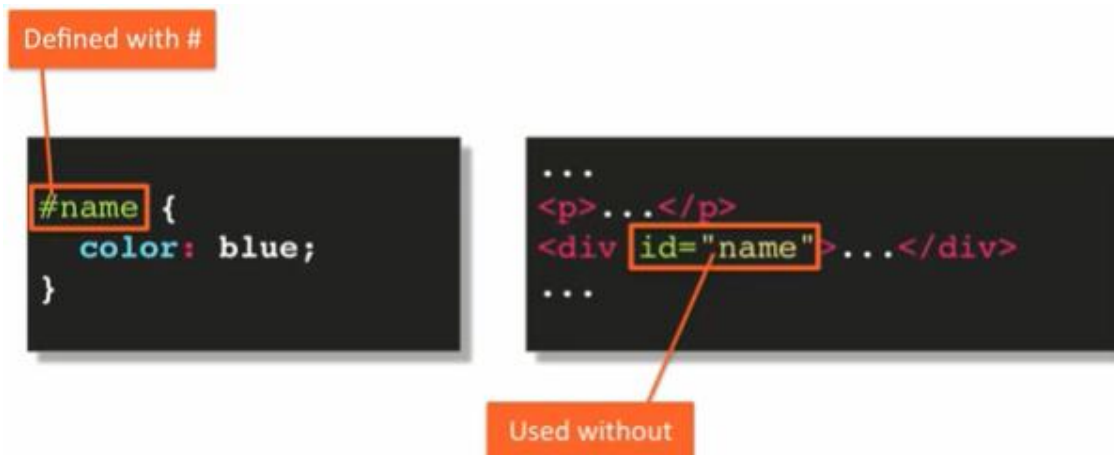


```
#name {  
  color: blue;  
}
```

```
...  
<p>...</p>  
<div id="name">...</div>  
...
```

Como el tag `<div>` posee un atributo `id="name"` se aplicará la regla CSS cuyo selector se ha definido de la siguiente manera: `#name`. Las declaraciones CSS para la regla `#name` se aplicarán al contenido del elemento `div` por el valor de su `id`. Tenga en cuenta que obviamente todo lo demás en el documento HTML no se ve afectado por esto.

Observe que existe una regla para la forma de definir y usar este tipo de selector:



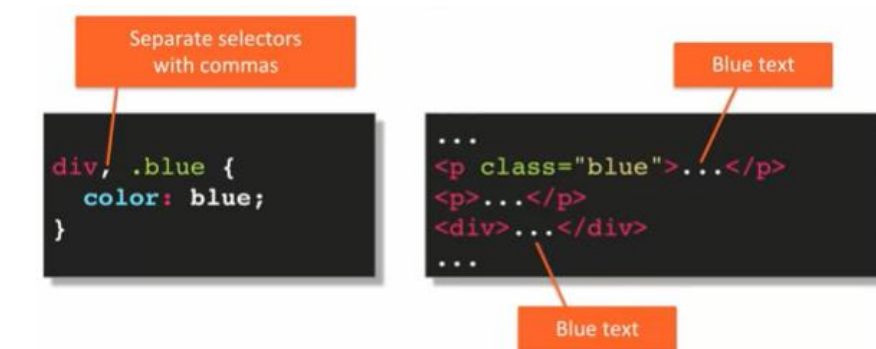
```
#name {  
  color: blue;  
}
```

```
...  
<p>...</p>  
<div id="name">...</div>  
...
```

El selector id se define con un signo `#` seguido del valor de una identificación dentro de su documento HTML y no puede haber espacio entre los dos. Pero cuando lo usa, el `#` desaparece, solo se utiliza el valor real de la identificación.

Agrupamiento de selectores

Para escribir reglas más eficientes, CSS nos permite agrupar varios selectores en una sola regla CSS. Observe el siguiente esquema de ejemplo



Aquí tenemos dos selectores agrupados, el selector de elemento div y el selector de clase .blue, ya que ambas comparten la misma declaración CSS. Y puede ver, en nuestro HTML, el texto de color azul aparecerá en la etiqueta de párrafo que tiene el atributo class = "blue" y en cualquier etiqueta div en el documento HTML.

Ejemplo Integrador

Observe las reglas CSS del archivo ejemploselectores.html:

```

ejemploselectores.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Selectores simples</title>
6     <style>
7
8       /* todos los elementos h2 */
9       xxx {
10        color: red;
11        text-align: center;
12      }
13
14      /* todos con class="highlight" */
15      xxx {
16        font-size: 20px;
17        font-weight: bold;
18        font-style: italic;
19        background-color: green;
20        opacity: .6;
21      }
22
23      /* todos los elementos con id="mainPoint" */
24      xxx {
25        font-size: 24px;
26        font-weight: bold;
27        background-color: red;
28        opacity: .7;
29      }
30
31      /* todos los elementos p y h1 */
32      xxx {
33        color: blue;
34        text-align: center;
35      }
36
37    </style>
38  </head>
39
  
```

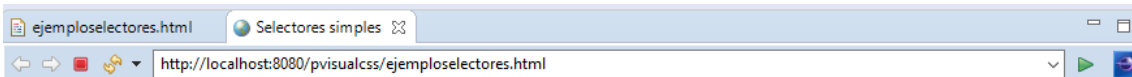



Como puede notar la especificación de los selectores no se ha realizado, sin embargo, previo a cada selector se utiliza una línea de comentario para indicar cual debería ser el valor que toma el selector. La idea es ir viendo el efecto de cada selector a medida que se completa el nombre del selector.

Antes de ello observe el resto del contenido del documento:

```
42 <h1>Selectores simples (h1)</h1>
43 <h2>Selector de clase (h2)</h2>
44 <p class="highlight">Se especifica con un punto y el nombre de la clase.El selector de clase requiere
45 un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee
46 aplicar este selector de clase deberá especificar el atributo class, cuyo valor
47 será el nombre del selector de clase.</p>
48 <p class="highlight">Este párrafo tiene el atributo class="highlight". El selector de clase siempre se
49 define con un punto delante de su nombre, no puede haber ningún espacio entre el
50 punto y el nombre de la clase. Sin embargo, para usar el selector de clase,
51 el atributo class debe recibir como valor el nombre del selector de clase SIN el
52 punto</p>
53 <h2>Selector id (h2)</h2>
54 <p>La forma de especificar un selector id es definiendo el nombre de un identificador de elemento
55 (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor
56 de alguna identificación dentro de nuestro documento HTML</p>
57 <div>En este div se define un span que invoca al selector id mediante|
58 <span id="mainPoint">id="mainPoint".</span>
59 </div>
60
61 </body>
62 </html>
```

Si ejecutamos tal como está la página obtendremos lo siguiente



Selectores simples (h1)

Selector de clase (h2)

Se especifica con un punto y el nombre de la clase.El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo class, cuyo valor será el nombre del selector de clase.

Este párrafo tiene el atributo class="highlight". El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo class debe recibir como valor el nombre del selector de clase SIN el punto

Selector id (h2)

La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML

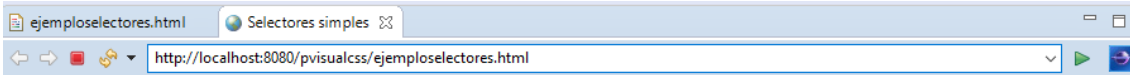
En este div se define un span que invoca al selector id mediante| id="mainPoint".

Esto significa que no se ha aplicado ningún estilo.

Observe que sucede si activamos el primer estilo, esto es:

```
8      /* todos los elementos h2 */
9      h2 {
10         color: red;
11         text-align: center;
12     }
```

Es un selector de elemento, todos los encabezados de nivel 2 se renderizarán en color rojo y se centrarán en la pantalla:



Selectores simples (h1)

Selector de clase (h2)

Se especifica con un punto y el nombre de la clase. El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo class, cuyo valor será el nombre del selector de clase.

Este párrafo tiene el atributo class="highlight". El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo class debe recibir como valor el nombre del selector de clase SIN el punto

Selector id (h2)

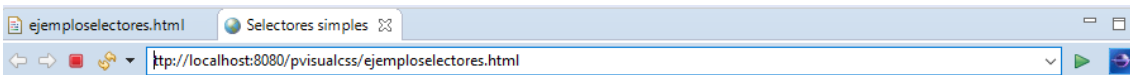
La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML

En este div se define un span que invoca al selector id mediante `id="mainPoint"`.

Si ahora activamos el segundo estilo, esto es:

```
15      /* todos con class="highlight" */
16      .highlight {
17          font-size: 20px;
18          font-weight: bold;
19          font-style: italic;
20          background-color: green;
21          opacity: .6;
22      }
```

Es un selector de clase, lo identificamos como tal porque inicia con ".". Entonces todos aquellos elementos que incorporen el atributo class="highlight" tendrán un tamaño de fuente de 20 pixeles, con formato en negrita y cursiva. Además, el color de fondo de todo el elemento será verde y se bajará opacidad al 60% (el valor 1 representa el 100%):



Selectores simples (h1)

Selector de clase (h2)

Se especifica con un punto y el nombre de la clase. El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo class, cuyo valor será el nombre del selector de clase.

Este párrafo tiene el atributo class="highlight". El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo class debe recibir como valor el nombre del selector de clase SIN el punto

Selector id (h2)

La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML

En este div se define un span que invoca al selector id mediante `id="mainPoint"`.

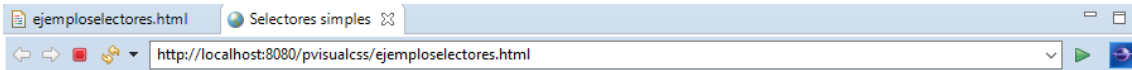


Así que tenemos dos párrafos (las etiquetas `<p>`) que fueron afectados por este estilo, esto porque ambas etiquetas poseen el atributo `class="highlight"`.

Ahora vamos a activar la tercera regla, esto es:

```
24      /* todos los elementos con id="mainPoint" */
25      #mainPoint {
26          font-size: 24px;
27          font-weight: bold;
28          background-color: red;
29          opacity: .7;
30      }
```

Es un selector id, lo identificamos como tal porque la forma en que lo especificamos es mediante "#", y el nombre de este selector es `mainPoint`. Todas las etiquetas que tengan por identificador `mainPoint` tendrán un tamaño de fuente de 24 pixeles, con formato en negrita. Además, el color de fondo de todo el elemento será rojo y se bajará opacidad al 70% (el valor 1 representa el 100%):



Selectores simples (h1)

Selector de clase (h2)

Se especifica con un punto y el nombre de la clase. El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo class, cuyo valor será el nombre del selector de clase.

Este párrafo tiene el atributo class="highlight". El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo class debe recibir como valor el nombre del selector de clase SIN el punto

Selector id (h2)

La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML

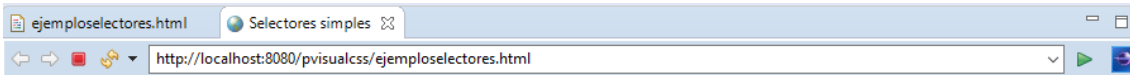
En este div se define un span que invoca al selector id mediante `id="mainPoint"`.

De manera evidente solamente una etiqueta tuvo se identificó con el nombre del selector. Puede comprobar que se trata de la etiqueta `` ubicada en la línea 58.

Finalmente se activa el agrupamiento de selectores, esto es:

```
32      /* todos los elementos p y h1 */
33      p, h1 {
34          color: blue;
35          text-align: center;
36      }
```

El cual aplicará las mismas declaraciones a las etiquetas `<p>` y `<h1>`, es decir que los párrafos y los encabezados de primer nivel se renderizarán de color azul y su texto será centrado:



Selectores simples (h1)

Selector de clase (h2)

Se especifica con un punto y el nombre de la clase. El selector de clase requiere un pequeño cambio en su documento HTML para ser utilizado. Cada tag que desee aplicar este selector de clase deberá especificar el atributo class, cuyo valor será el nombre del selector de clase.

Este párrafo tiene el atributo class="highlight". El selector de clase siempre se define con un punto delante de su nombre, no puede haber ningún espacio entre el punto y el nombre de la clase. Sin embargo, para usar el selector de clase, el atributo class debe recibir como valor el nombre del selector de clase SIN el punto

Selector id (h2)

La forma de especificar un selector id es definiendo el nombre de un identificador de elemento (atributo id de una etiqueta), precedido por un signo numeral (#). Entonces el nombre es un valor de alguna identificación dentro de nuestro documento HTML

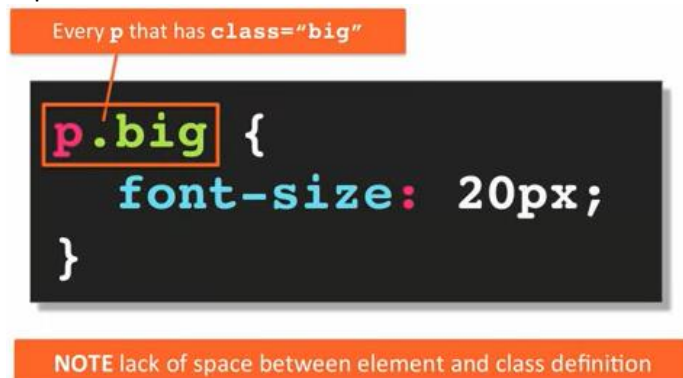
En este div se define un span que invoca al selector id mediante `id="mainPoint"`.

Entonces, una cosa a tener en cuenta aquí es que, de todos los selectores utilizados, el selector id es el menos reutilizable. Eso tiene sentido, ¿verdad? Dado que, según las reglas HTML, un valor de un atributo id solo puede aparecer una vez en un documento. Entonces, al planificar sus estilos, es bueno tener esto en cuenta y no basarse básicamente en asignar identificadores a cada elemento en todo el documento HTML. Porque de esa manera tu CSS no será muy eficiente. Tendrás que escribir reglas específicas para cada elemento, y realmente no podrás generalizar y reutilizar tus estilos CSS muy fácilmente.

Combinar selectores

Combinar selectores es una técnica muy poderosa que le permite otorgar estilos con mayor precisión dentro de los elementos. Veremos algunas de las formas más comunes de combinar selectores CSS.

- La primera forma de combinar selectores es “elemento con selector de clase”. Veamos el siguiente ejemplo:



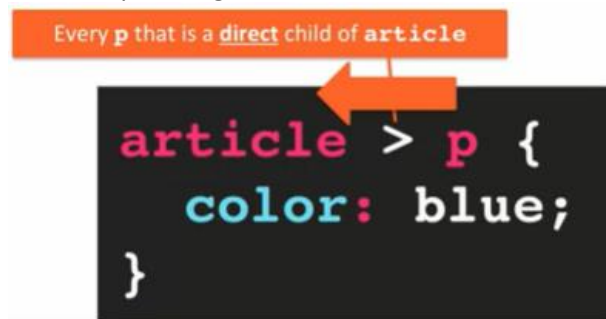
Como puede ver aquí, tenemos un elemento `p` seguido inmediatamente y sin espacios por el selector de clase `.big`. Esto expresa que se aplicará el estilo a los selectores de clase `.big` de cada elemento `p`. Tenga en cuenta la falta de espacio entre el elemento y la clase. Si pone un espacio, significará una combinación completamente diferente, así

que asegúrese de que no haya espacios entre el primer selector y el segundo selector. El siguiente esquema muestra cómo funciona esta combinación



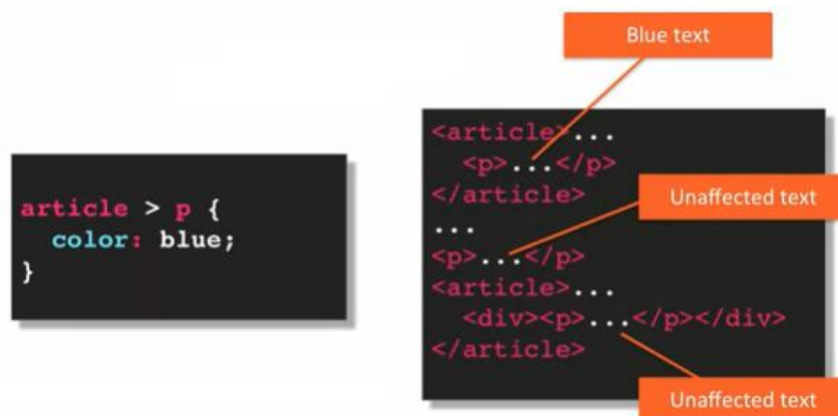
Como puede ver, si bien existen dos etiquetas con atributo `class="big"` únicamente el primero aplicará un tamaño de fuente de 20px, debido a la forma en que se ha definido el selector. Esta técnica se usa con bastante frecuencia cuando tiene una selector de clase CSS que se aplica a diferentes elementos, y se desea que el estilo cambie ligeramente a un elemento en particular.

- Otro tipo de combinación de selectores es el denominado Selector heredado. La sintaxis de la misma es: selector-espacio-signo >- otro selector:



La combinación se lee de derecha a izquierda. El signo > indica "hijo directo". Entonces, en este ejemplo lo que se hace es indicar que a cada elemento `p` que esté dentro de `article` y que además se cumpla que `p` está ubicada inmediatamente después de `article` en la jerarquía de selectores, se aplicará el estilo. Siempre que el padre inmediato de `p` sea un `article`, se aplicará esta regla.

Observe el siguiente esquema para evaluar como afecta esta combinación de selectores a un documento HTML:



Entonces, en este ejemplo, puede ver que sobre el elemento `p` que es un hijo directo de un elemento `article` se aplicará el estilo de color azul, a diferencia del segundo

elemento `p` que obviamente no se verá afectado por el estilo. Por último, observe que el tercer elemento `p` está dentro de un elemento `article`, pero al no ser un hijo directo del mismo (debido a que hay un elemento `div` entre ambos en la jerarquía de elementos) su estilo de fuente no se ve afectado.

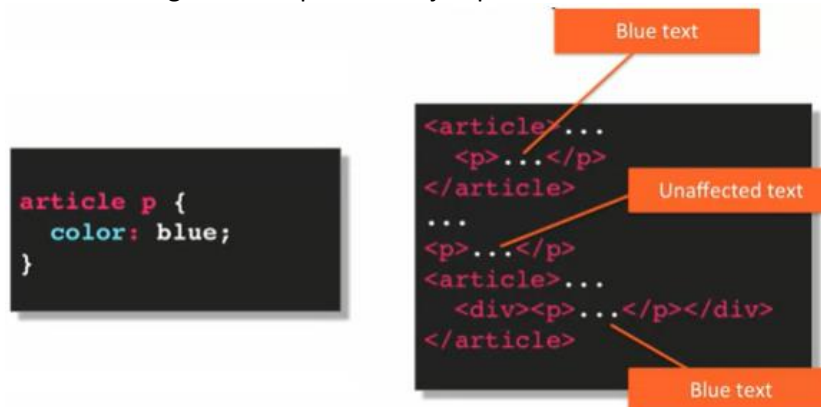
- Se puede mencionar el Selector Descendente. La sintaxis de esta es: selector-espacio-otro selector y también se lee de derecha a izquierda. Entonces, en el siguiente ejemplo:

Every `p` that is inside (at any level) of `article`

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

En este caso, lo que estamos tratando de hacer es referenciar a cada elemento `p` que esté dentro de un `article` en cualquier nivel. Lo que significa que incluso si `p` no es un hijo directo, incluso si está en el fondo, siempre que uno de sus padres en su ascendencia sea un `article`, se aplicará esta regla.

Veamos entonces el siguiente esquema de ejemplo:



En este caso, el primer elemento `p` es un hijo directo de un `article`, entonces el texto se volverá azul. Para el segundo elemento `p` no se aplicará el estilo porque no está dentro de un elemento `article`. En el caso del tercer `p`, el contenido se volverá azul, aunque no sea hijo directo de `article`, ya que es un hijo dentro de la descendencia del selector `article`.

Ahora, quiero señalar que estas combinaciones de selectores no se limitan a los selectores de elementos. Entonces, por ejemplo:



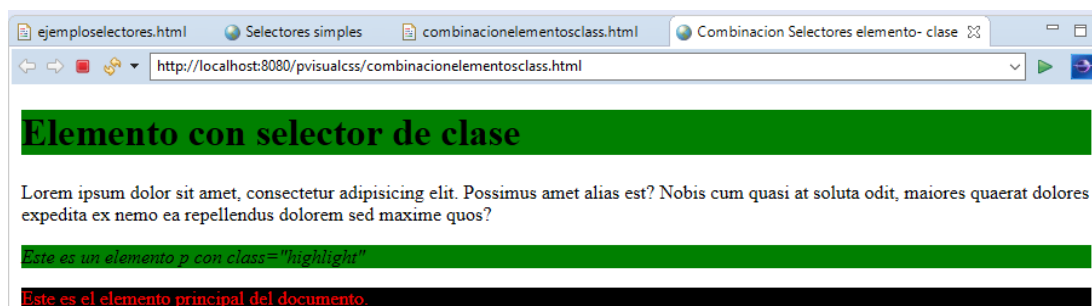
- 1) `.colored p` significaría (leyendo de derecha a izquierda) que cada elemento `p` que esté dentro y en cualquier nivel del elemento con `class="colored"` tendrá su color de texto convertido en azul.
- 2) `article > .colored`: estamos utilizando un selector secundario directo para decir que cada elemento con `class="colored"` que sea un hijo directo de un elemento `article` tendrá su texto en azul.

Echemos un vistazo a algunos ejemplos de codificación para ver estas combinaciones en acción. En primer lugar, observe el contenido del archivo *combinacionelementosclass.html*

```

ejemploselectores.html | Selectores simples | combinacionelementosclass.html | Combinacion Selectores elemento- clase
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Combinacion Selectores elemento- clase
6   </title>
7   <style>
8
9     /* todos los elementos con atributo class="highlight" */
10    .highlight {
11      background-color: green;
12    }
13
14    /* todos los elementos p con atributo class="highlight" */
15    p.highlight {
16      font-style: italic;
17    }
18
19    /* todos los elementos con atributo class='highlight' así como aquellos con atributo class='mainpoint'. */
20    .mainpoint.highlight {
21      color: red;
22      background-color: black;
23    }
24
25  </style>
26 </head>
27 <body>
28
29   <h1 class="highlight">Elemento con selector de clase</h1>
30   <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Possimus amet alias est? Nobis cum quasi at soluta c
31     maiores queraat dolores expedita ex nemo ea repellendus dolorem sed maxime quos?</p>
32   <p class="highlight">Este es un elemento p con class="highlight"</p>
33   <div class="mainpoint highlight">Este es el elemento principal del documento.
34   </div>
35
36 </body>
37 </html>
  
```

Bien, entonces puedes ver que tenemos un documento HTML bastante simple. El tag `<h1>` tiene un atributo `class="highlight"` que aplicará un color de fondo verde sobre el encabezado. Sobre el primer tag `<p>` no aplica ningún estilo, pero si sobre el segundo. Esto se debe a que tiene un atributo `class="highlight"` por lo tanto se combinan dos reglas de estilo: al igual que el encabezado el color de fondo es verde, pero además el texto se visualiza en cursiva por la regla denominada `p.highlight`. El tag `<div>` posee un atributo `class="mainpoint highlight"` correspondiente a la regla cuyo selector se denomina `.mainpoint.highlight`, por lo que el texto es de color rojo y fondo negro. Entonces ahora veamos el resultado obtenido:



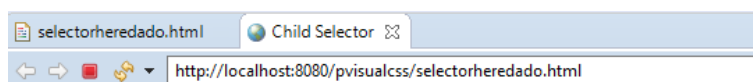
A continuación, echemos un vistazo al archivo llamado selectorheredado.html:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Child Selector</title>
6     <style>
7
8     /* cualquier elemento article que sea hijo directo de un elemento section. */
9     section > article {
10      color: green;
11      font-size: 24px;
12    }
13
14    /* cualquier elemento h2 el cual sea hijo directo de un elemento section . */
15    section > h2 {
16      color: green;
17    }
18
19  </style>
20 </head>
21 <body>
22   <h1>Selector heredado</h1>
23   <section>
24     <h2>Sub encabezado 1</h2>
25     <div>
26       <article>ARTICLE dentro de un DIV, el cual está dentro de un SECTION</article>
27     </div>
28   </section>
29   <section>
30     <article>ARTICLE es un hijo directo del elemento SECTION.</article>
31   </section>
32   <h2>Sub encabezado 2</h2>
33
34 </body>
35 </html>
  
```

El encabezado de primer nivel no posee ningún estilo definido así que adopta el predefinido por el browser.

Observe que el tag <h2> de la línea 24 es un hijo directo del tag <h2> por lo cual se aplica la regla de estilo definida entre las líneas 15 y 17; es decir el color del encabezado será verde. El tag <article> de la línea 26 no es un hijo directo del tag <div> inicializado en la línea 23 por lo tanto no se aplica el estilo definido entre las líneas 9 y 12. En cambio, el elemento article de la línea 30 si es hijo directo del elemento section de la línea 29 por lo cual se aplicará el estilo que consiste en un color de texto verde y un tamaño de texto de 24 px. Por último, el encabezado definido en la línea 32 es hijo del elemento body, entonces no se aplica ningún estilo. Observe el resultado obtenido:



Selector heredado

Sub encabezado 1

ARTICLE dentro de un DIV, el cual está dentro de un SECTION
 ARTICLE es un hijo directo del elemento SECTION.

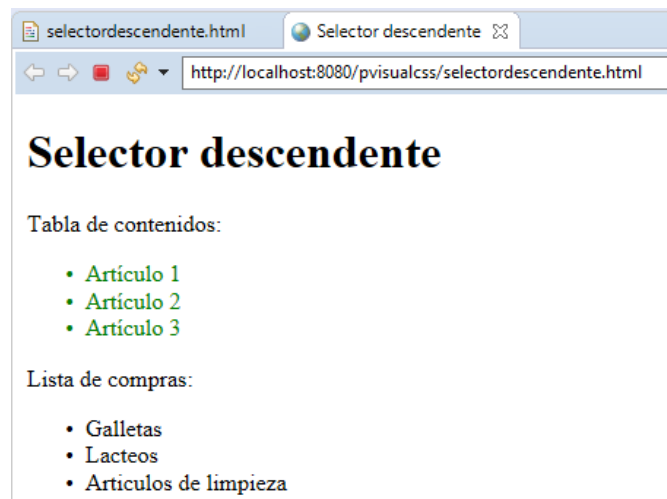
Sub encabezado 2

A continuación, echemos un vistazo al archivo *selectordescendente.html*:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Selector descendente</title>
6     <style>
7
8       /* todos los elementos li elements que están dentro de un elemento section. */
9       section li {
10        color: green;
11      }
12
13    </style>
14  </head>
15  <body>
16    <h1>Selector descendente</h1>
17    <section>
18      Tabla de contenidos:
19      <ul>
20        <li>Artículo 1</li>
21        <li>Artículo 2</li>
22        <li>Artículo 3</li>
23      </ul>
24    </section>
25    <article>
26      Lista de compras:
27      <ul>
28        <li>Galletas</li>
29        <li>Lacteos</li>
30        <li>Articulos de limpieza</li>
31      </ul>
32    </article>
33  </body>
34 </html>
  
```

El cual genera la siguiente salida



Como se observa solamente el primer listado posee un estilo. Esto se debe a que la regla indica que cualquier elemento de lista (*li*) que se halle dentro de un elemento *section* debe ser renderizado de color verde. Como los elementos de la lista de compras están ubicados dentro de un tag *article* no reciben el estilo definido.



Hay un par de selectores combinados que no cubrimos. Por ejemplo, uno es el selector JSON y otro es el selector general sibling. Ahora, estas combinaciones de selectores son un poco menos comunes que los mencionados hasta ahora.

¿Entendió los conceptos vertidos en esta sección? Entonces resuelva los siguientes ejercicios

1) Dado el siguiente código

<pre>1 <div> 2 <p>I am feeling blue</p> 3 </div> 4 <section> 5 <p>I just want to be left alone!</p> 6 </section></pre>	<pre>1 div > p { 2 color: blue; 3 }</pre>
	<pre>1 div.p { 2 color: blue; 3 }</pre>
	<pre>1 div p { 2 color: blue; 3 }</pre>
	<pre>1 p { 2 color: blue; 3 }</pre>

¿Cuál de las reglas logrará que el primer <p> tenga el color de texto azul, pero que no se aplique en el segundo <p>? (pueden ser varias opciones correctas)

2) Dado el siguiente código

<pre>1 <div> 2 <div> 3 <div class="makeMeBlue"> 4 <p>I am feeling blue</p> 5 </div> 6 </div> 7 <section class="makeMeBlue"> 8 <p>I just want to be left alone!</p> 9 </section> 10 </div></pre>	<pre>1 .makeMeBlue > p { 2 color: blue; 3 }</pre>
	<pre>1 div.makeMeBlue p { 2 color: blue; 3 }</pre>
	<pre>1 div > div > p { 2 color: blue; 3 }</pre>
	<pre>1 div p { 2 color: blue; 3 }</pre>

¿Cuál de las reglas logrará que el primer <p> tenga el color de texto azul, pero que no se aplique en el segundo <p>? (pueden ser varias opciones correctas)

Selectores Pseudo-Clase

Los selectores de pseudo-clase abordan su enfoque solo a las estructuras específicas mediante combinaciones simples de selectores regulares. También brindan la capacidad de estilo sobre

estructuras basada en la interacción del usuario con la página. Por ejemplo, nos gustaría cambiar el estilo de un elemento si el usuario pasa el mouse sobre ese elemento o lo mueve.

La forma de especificar el selector de pseudo-clase es indicando algún selector que todos sepamos seguido de dos puntos y finalmente colocamos el nombre de pseudo-clase predefinido:

```
selector:pseudo-class {  
  . . .  
}
```

Existen muchos selectores de pseudo-clase. En este apunte cubriremos cinco de ellos:

- `:link`
- `:visited`
- `:hover`
- `:active`
- `:nth-child(...)`

Los primeros cuatro son bastante básicos y se usan muy, muy a menudo. La última pseudo-clase (`nth-child`) es bastante poderosa y puede volverse bastante complicada, pero sus usos básicos son bastante directos.

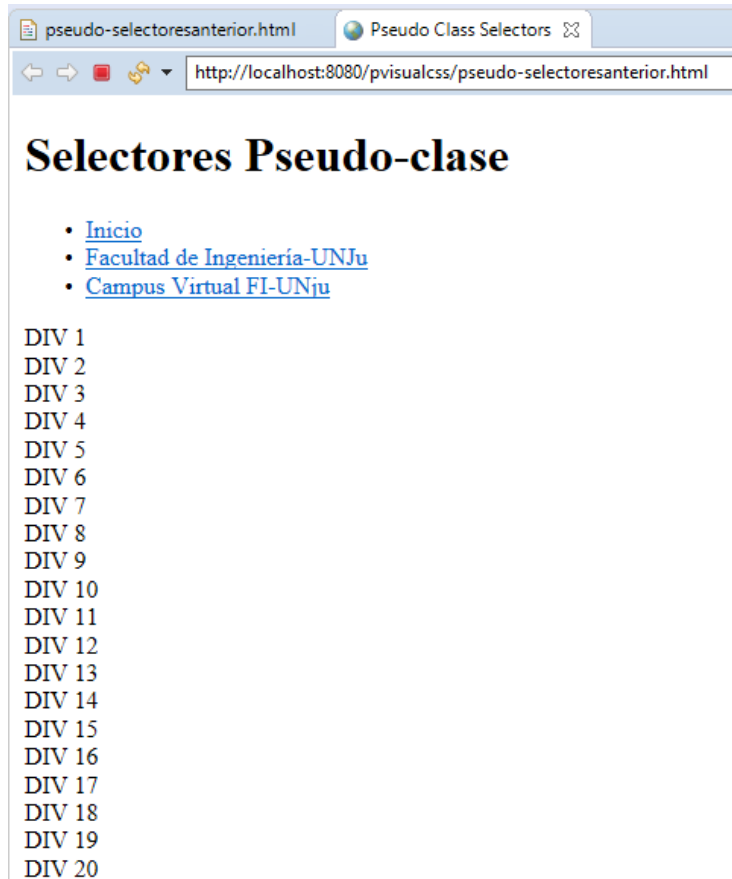
Así que vamos directamente al código y veamos cómo funcionan estos selectores. Abra el archivo *pseudo-selectoresanterior.html*:

```
pseudo-selectoresanterior.html Pseudo Class Selectors  
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8">  
5 <title>Pseudo Class Selectors</title>  
6 <style>  
7 /* Los estilos van aquí. */  
8  
9  
10 </style>  
11 </head>  
12 <body>  
13 <h1>Selectores Pseudo-clase</h1>  
14  
15 <header>  
16 <ul>  
17 <li><a href="/">Inicio</a></li>  
18 <li><a href="https://www.fi.unju.edu.ar/" target="_blank">Facultad de Ingeniería-UNJu</a></li>  
19 <li><a href="https://campus.fi.unju.edu.ar/" target="_blank">Campus Virtual FI-UNJu</a></li>  
20 </ul>  
21 </header>  
22  
23 <section>  
24 <div>DIV 1</div>  
25 <div>DIV 2</div>  
26 <div>DIV 3</div>  
27 <div>DIV 4</div>  
28 <div>DIV 5</div>  
29 <div>DIV 6</div>  
30 <div>DIV 7</div>  
31 <div>DIV 8</div>  
32 <div>DIV 9</div>  
33 <div>DIV 10</div>  
34 <div>DIV 11</div>  
35 <div>DIV 12</div>
```



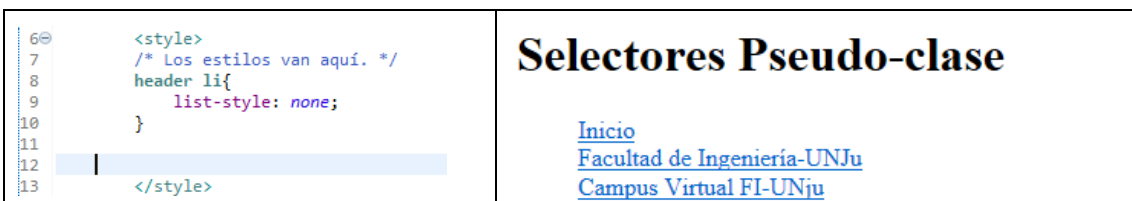
```
36 <div>DIV 13</div>
37 <div>DIV 14</div>
38 <div>DIV 15</div>
39 <div>DIV 16</div>
40 <div>DIV 17</div>
41 <div>DIV 18</div>
42 <div>DIV 19</div>
43 <div>DIV 20</div>
44 </section>
45
46 </body>
47 </html>
```

y ejecútelo en el browser. Se verá de esta manera:



Entonces, esta página tiene un encabezado, y en el encabezado tenemos una lista desordenada de un conjunto de enlaces. Esta lista se renderiza de forma predeterminada con puntos como viñetas. A continuación tenemos una etiqueta `<section>` con 20 etiquetas `<div>`.

Lo primero que diseñaremos será un menú basado en la lista desordenada. Esto es una práctica bastante común. Para estilizarla como un conjunto de botones de menú usaremos la siguiente regla que quita el estilo predefinido a los elementos de lista que estén dentro de un elemento header, sin importar la profundidad en la jerarquía



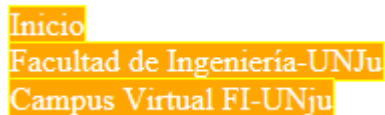
La siguiente tarea es diseñar los enlaces que se encuentran dentro de los elementos `li` para hacer que se vean como botones. Sin embargo, el diseño de enlaces no es tan sencillo como diseñar un elemento regular, y eso se debe a que los enlaces tienen estados. Y estos estados se pueden expresar utilizando nuestras pseudo-clases. Así que sigamos adelante y hagamos eso:

```


12     a:link,a:visited {
13         text-decoration: none;
14         background-color: orange;
15         border: 1px solid yellow;
16         color: white;
17     }

```

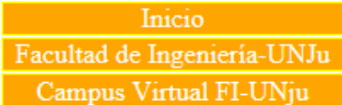
Para los enlaces dentro de la página establecemos el comportamiento para dos estados: `link` y `visited`. El primero hace referencia al estilo que presenta el enlace por defecto, mientras que el segundo indica como queda el estilo del enlace luego de que ha sido usado. Es una práctica común agruparlas para darles el mismo estilo, pero podría aplicarle estilos diferentes a cada estado. Las declaraciones indicadas establecen en orden lo siguiente: se elimina el estilo por defecto de los enlaces (subrayado), luego se establece como color de fondo el naranja. Posteriormente se establece un borde bien marcado (estilo `solid`) de color amarillo, pero de grosor fino. Finalmente, el texto de los enlaces se visualizará de color blanco. Por tanto, el resultado quedará de la siguiente manera:



Ahora, los enlaces son elementos tanto elementos de nivel de línea como elementos de nivel de bloque. Sin embargo, eso solo se refiere a la inclusión de otros elementos dentro de ellos. Por ese motivo el estilo solo se aplica sobre el ancho del texto del enlace de forma predeterminada para cada nueva línea. Bueno, no queremos eso para nuestros botones, queremos que el estilo se aplique sobre todos los elementos como si fueran un solo bloque. Para lograr esto se procede de la siguiente manera:

<pre> 12 a:link,a:visited { 13 text-decoration: none; 14 background-color: orange; 15 border: 1px solid yellow; 16 color: white; 17 display: block;; </pre>	<h3 style="text-align: center;">Selectores Pseudo-clase</h3> 
---	---


Como se puede observar simplemente se estableció la declaración `display: block`. Sin embargo, ahora tenemos el inconveniente de que el estilo se aplica a todo el ancho de la pantalla del browser. Así que vamos a tener que restringirlo un poco. Para ello agregamos la siguiente declaración:

<pre> 12 a:link,a:visited { 13 text-decoration: none; 14 background-color: orange; 15 border: 1px solid yellow; 16 color: white; 17 display: block; 18 width: 200px; 19 text-align: center; 20 margin-bottom: 1px; 21 } </pre>	<h3 style="text-align: center;">Selectores Pseudo-clase</h3> 
--	---



Se ha establecido un ancho de 200 píxeles para los enlaces. Además, se estableció que el contenido de los enlaces esté centrado. Y también hemos separado los diferentes botones de menú entre sí dándoles un margen inferior de un píxel.

Sin embargo, no hemos terminado porque lo que nos gustaría hacer es que cuando el usuario mueva el mouse sobre estos botones, además de cambiar el cursor a un puntero, tenga tener algún tipo de interacción mediante un estilo. También es deseable un estilo que se aplique cuando en el momento exacto en el que se presiona el botón de enlace. Para lograr esto hay que definir dos estados más del enlace. Uno es hover y el otro es active:

<pre>23 a:hover,a:active { 24 background-color: yellow; 25 color:red; 26 } 27</pre>	<h2>Selectores Pseudo-clase</h2> 
---	---

Al pasar el mouse sobre los botones se aplica el estilo definido para el estado hover. Observe que cambia el color del fondo por amarillo y el texto se renderiza en color rojo. Al presionar el botón de enlace se aplica el estilo definido para el estado active y se mantendrá ese estilo mientras no se suelte el botón del mouse presionado.

Observe que en este ejemplo en particular no se ha diferenciado entre los estilos de los estados hover y active, por lo que se los ha definido agrupados.

En su forma más simple, un selector puede usar el estado nth-child() para establecer un estilo a un elemento particular del selector. Por ejemplo, suponga que desea que de todos los botones de enlace desea destacar el del aula virtual, podría hacer de la siguiente manera:

```
28     header li:nth-child(3) {
29         font-size: 24px;
30     }
```

Con nth-child (3) se está haciendo referencia al tercer enlace ubicado dentro de un header (sin importar si es hijo directo o no), el cual se renderizará con un tamaño de 24 píxeles. Entonces, si actualizo la página:

Selectores Pseudo-clase



Como se puede observar solo el tercer enlace, el tercer botón de menú, se hizo más grande. El resto de los botones de enlace se quedaron exactamente igual. Pero el pseudo selector nthchild() puede hacer cosas mucho más divertidas e interesantes. Por ejemplo, suponga que desea que la lista de DIVS se renderice de manera intercalada con diferentes colores de fondo,

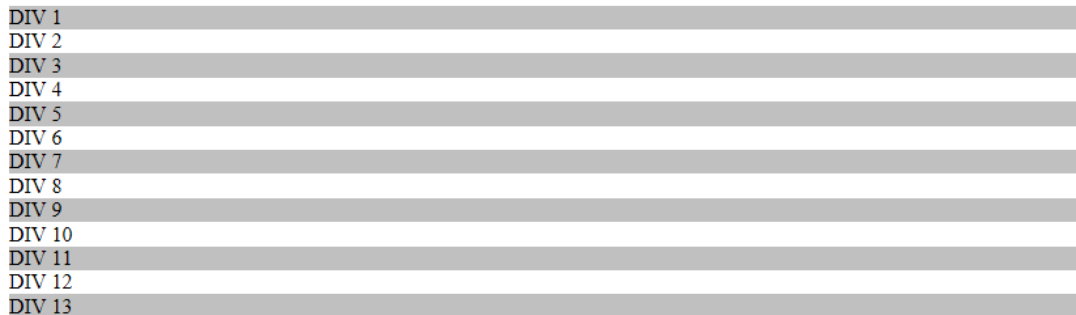


por ejemplo, que los miembros impares de esta lista tengan un color de fondo plateado. Entonces, la forma en que podría lograr eso es:

```
32     section div:nth-child(odd) {  
33         background-color: silver;  
34     }
```

Donde en lugar de especificar un elemento en particular con *odd* (impar en inglés) asignamos a los `div` impares que estén ubicados dentro de un elemento `section` (sin importar si son hijos directos) el color de fondo plateado. Si ejecutamos la página se obtendrá el siguiente resultado:

Selectores Pseudo-clase



Por cierto, al igual que los selectores regulares se pueden combinar, los pseudo-selectores también se pueden combinar. Por ejemplo, si quisiera que el cuarto elemento de esta lista de `div`s, tuviera un comportamiento diferente cuando se pase el puntero del mouse sobre él podríamos realizarlo de la siguiente manera:

```
36     section div:nth-child(4):hover {  
37         background-color: green;  
38         cursor: pointer;  
39     }
```

Entonces, para el cuarto `div` de todas las secciones, cuando se pose el puntero del mouse se cambiará el color de fondo a verde y el puntero del mouse adquirirá la forma de puntero:

Selectores Pseudo-clase





Estoy seguro de que ahora tiene una idea de cuán poderosos pueden ser los selectores, especialmente usando los selectores de pseudo-clase. Sin embargo, una rápida advertencia, asegúrese de que su selector aún sea legible. Estas cosas pueden complicarse muy rápidamente. Y si lo hace de tal manera que otros desarrolladores, y usted mismo en algún tiempo futuro no pueda entender con bastante facilidad que ha diseñado, esto puede convertirse en una especie de pesadilla a la hora de realizar el mantenimiento. Definitivamente debe prevalecer lo simple y legible sobre lo complicado. Es posible que si aplica estilos complicados pueda impresionar con el resultado final, pero ciertamente no es una buena práctica de codificación hacer que esto sea muy complicado. A continuación, vamos a hablar sobre dónde ubicamos los estilos CSS. ¿Cuál es el mejor lugar para colocarlo y cuáles son las diferencias entre esos lugares? Pero antes, algunos ejercicios para analizar cuan fijados están los conceptos:

- 1) Por defecto, un elemento de nivel de bloque intenta ocupar tanto espacio horizontal como lo permita su elemento contenedor (Verdadero o falso): _____
- 2) Dado el siguiente código HTML y su correspondiente regla de estilo

<pre>... <body> <div>Hello</div> World! </body> ...</pre>	<pre>div, span { background-color: blue; }</pre>
--	--

¿Cuáles de las siguientes afirmaciones es verdadera?

El color de fondo de las palabras “Hello” y “World” es azul: _____

Ambos colores de fondo abarcarán todo el ancho de la pantalla del browser: _____

El color de fondo de todos los elementos de la etiqueta <div> abarcará todo el ancho de la pantalla del browser: _____

El color de fondo de todos los elementos de la etiqueta abarcará todo el ancho de la pantalla del browser: _____

El texto “World” aparecerá en una línea separada, debajo del texto “Hello”: _____

- 3) Dado el siguiente código HTML

```
<a>  
  <div>Div 1</div>  
  <div>Div 2</div>  
  <div>Div 3</div>  
  <div>Div 4</div>  
  <div>Div 5</div>  
  <div>Div 6</div>  
</a>
```

¿Cuál de las siguientes reglas configurará el tamaño de fuente del texto “DIV 5” en 24 pixeles cuando el usuario pose el mouse sobre **cualquier área dentro del elemento a**?

```
1 a:hover:nth-child(5) {  
2   font-size: 24px;  
3 }
```

```
1 a:hover div:nth-child(5) {  
2   font-size: 24px;  
3 }
```

```
1 a:hover > div > div > div > div > div {  
2   font-size: 24px;  
3 }
```

```
1 div:hover {  
2   font-size: 24px;  
3 }
```


Referencia teórica: Curso MOOC Html, CSS and Javascript for web developers de Yaakov Chaikin, profesor de la Johns Hopkins University y Coursera.

Traducción: Mg Ariel Alejandro Vega

Ejemplos usados en el entorno STS: Mg Ariel Alejandro Vega