

CSS

Prioridades, Pseudo-clases, Pseudo-elementos y selectores

Módulo: **Desarrollo de Interfaces Web**

CFGS Desarrollo de Aplicaciones Web

Contenido (HTML) y Presentación (CSS)

HTML (contenido) y **CSS** (presentación) pueden unirse de 3 formas:

- **Inline**: las reglas CSS indicadas con el atributo **style** dentro de las etiquetas html.
- **Embebido**: en el `<head>` dentro de la etiqueta `<style> ... </style>`
- **Externo**: las reglas CSS están en un fichero separado
 - Normalmente un fichero con la extensión **.css**
 - Enlazado mediante la etiqueta :
`<link rel="stylesheet" href="...">`
 - Por medio de la directiva **@import** embebido dentro un bloque CSS.

Contenido (HTML) y Presentación (CSS)

@import

- Otra forma de vincular ficheros CSS externos
- Ejemplo:

```
<style type="text/css">  
  @import url("styles.css");  
  /* es lo mismo que lo siguiente */  
  @import "styles.css";  
</style>
```

- Navegadores antiguos no reconocen @import
- Utiliza @import en un archivo CSS externo para solucionar el límite de 32 archivos CSS de IE

Contenido (HTML) y Presentación (CSS)

@import

Para que los navegadores puedan evitar recuperar recursos para tipos de medios no soportados, los autores pueden especificar reglas dependientes del tipo de medio @import.

```
@import url("fineprint.css") print;
```

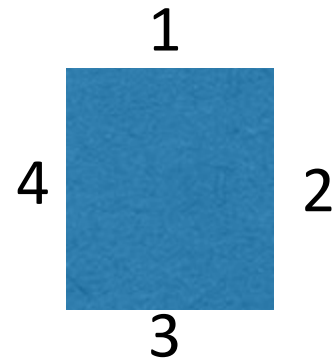
```
@import "common.css" screen;
```

http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Propiedades Shorthand

Permiten establecer simultáneamente el valor de varias propiedades diferentes:

- `margin: 20px;`
Los 4 lados tendrán un margen de 20px
- `padding: 20px 10px;`
Top y bottom 20px, left y right 10px del padding
- `padding: 10px 5px 20px;`
Top 10px, left y right 5px, bottom 20px
- `margin: 5px 10px 20px 0px;`
Top 5, right 10px, bottom 20px, left 0px



La gran ventaja de las propiedades shorthand es que permiten crear hojas de estilos mucho más concisas y por tanto, mucho más fáciles de leer.

Propiedades Shorthand

```
p {  
  font-style: normal;  
  font-variant: small-caps;  
  font-weight: bold;  
  font-size: 1.5em;  
  line-height: 1.5;  
  font-family: Arial, sans-serif;  
}
```

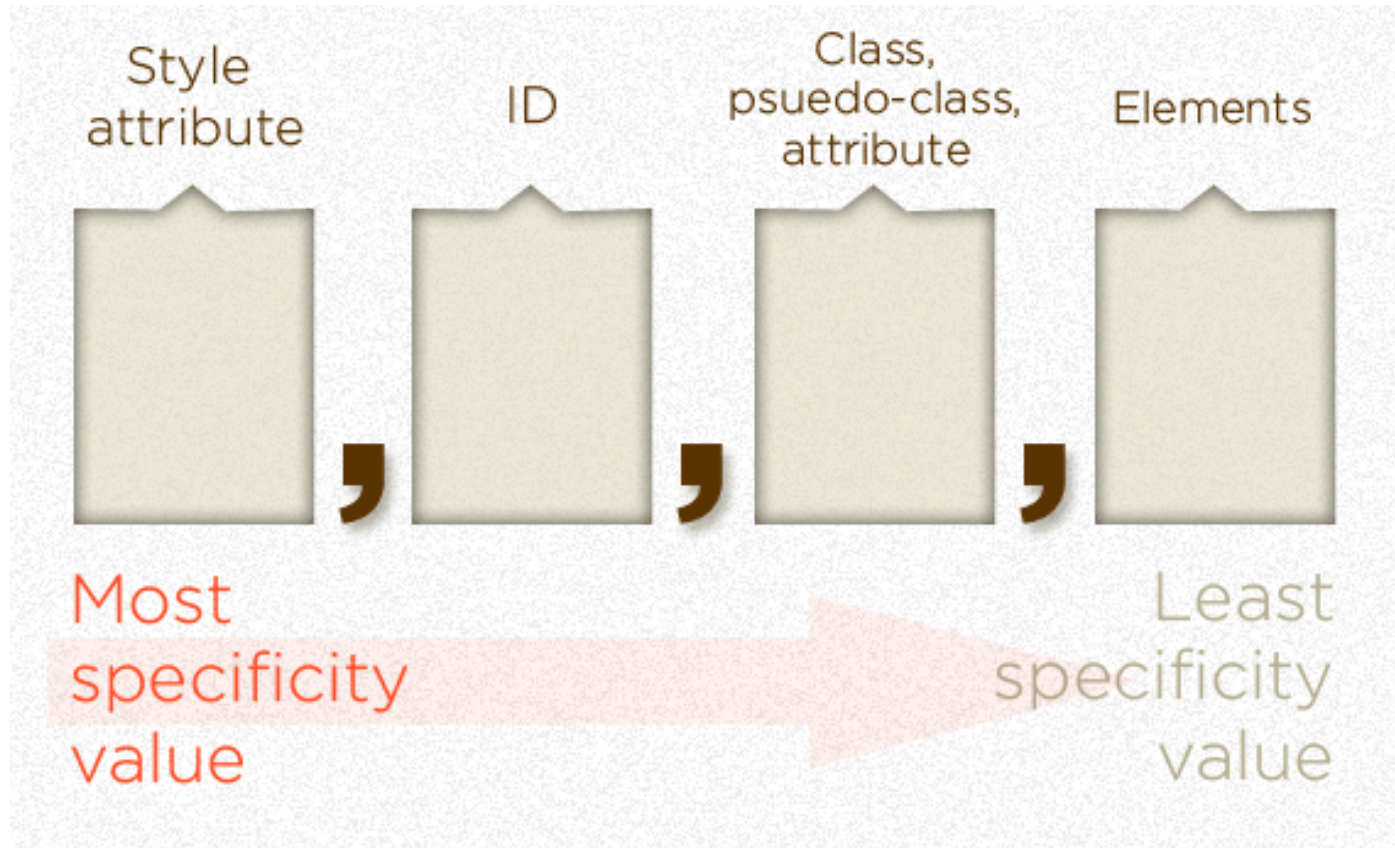
```
div {  
  margin-top: 5px;  
  margin-right: 10px;  
  margin-bottom: 5px;  
  margin-left: 10px;  
  padding-top: 3px;  
  padding-right: 5px;  
  padding-bottom: 10px;  
  padding-left: 7px;  
}
```

```
h1 {  
  background-color: #FFFFFF;  
  background-image: url("imagenes/icono.png");  
  background-repeat: no-repeat;  
  background-position: 10px 5px;  
}
```

```
p {  
  font: normal small-caps bold 1.5em/1.5 Arial, sans-serif;  
}  
div {  
  margin: 5px 10px;  
  padding: 3px 5px 10px 7px;  
}  
h1 {  
  background: #FFF url("imagenes/icono.png") no-repeat 10px 5px;  
}
```

Prioridad CSS

Prioridad CSS

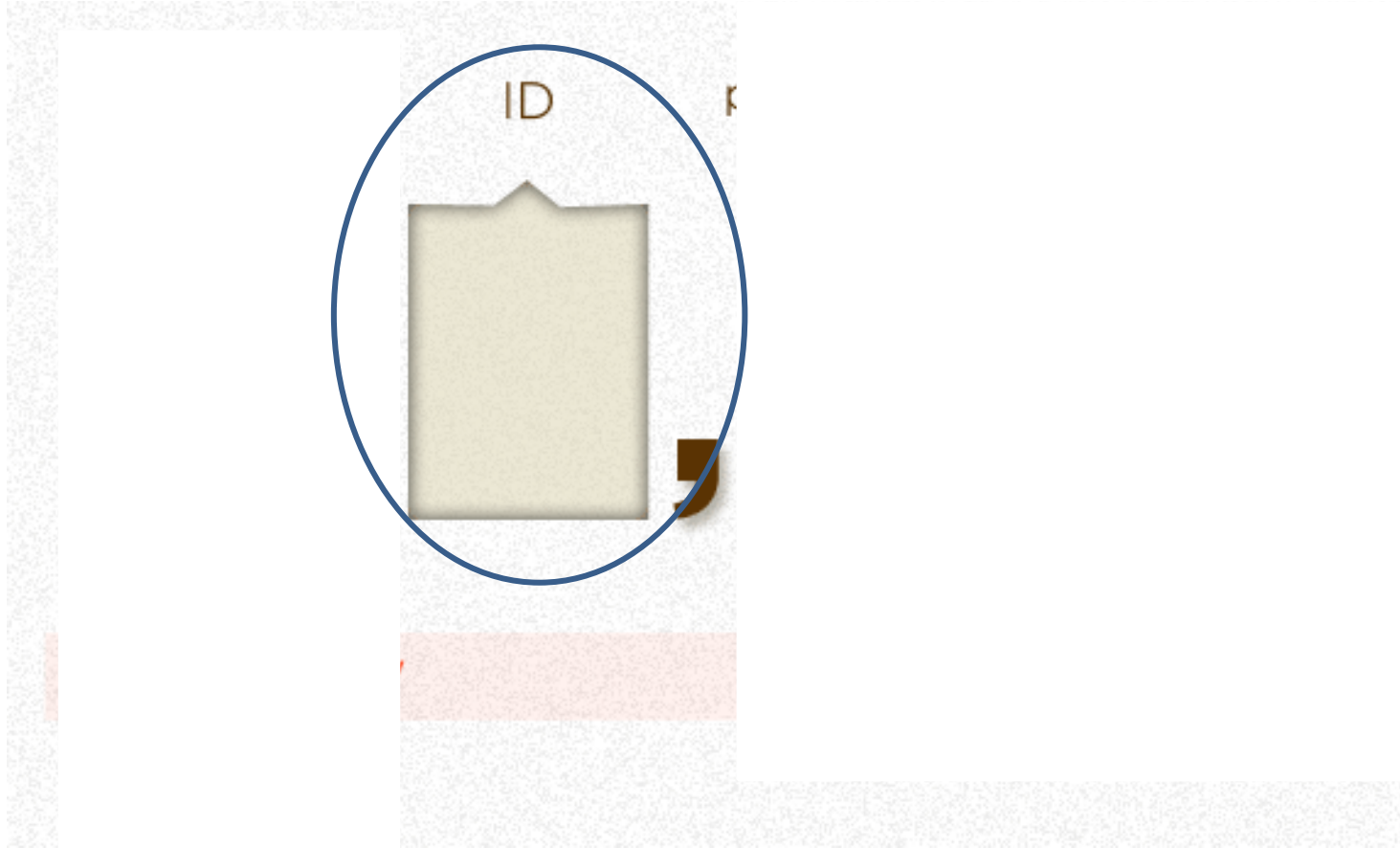


Prioridad CSS



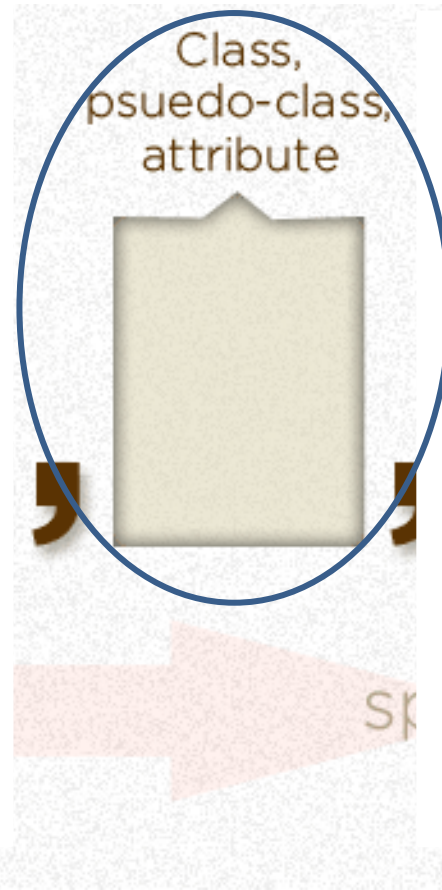
```
<p style="color:red;">Esto es rojo</p>
```

Prioridad CSS



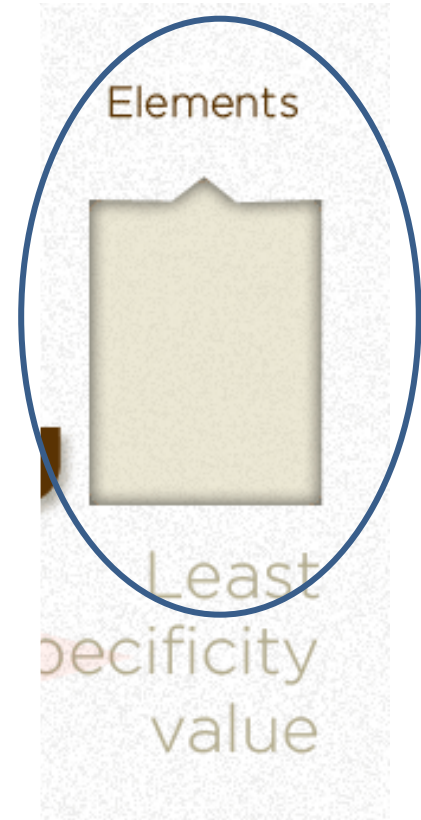
```
#nav { color: red; }
```

Prioridad CSS



```
.warning { color: yellow; }  
p:first-child { padding: 10em; }
```

Prioridad CSS



```
a { margin-top: 5em; }
```

Prioridad CSS

Cuando dos declaraciones afectan a un mismo elemento. **¿cuál de ellas se interpreta en el navegador como más importante?**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```

A = 0 estilos en línea
B = 0 ID
C = 0 clases
D = 1 elemento
Puntuación = 0,0,0,1

Hay que calcular la tupla (A, B, C, D) ganadora de todas las reglas CSS que compiten. A tiene máximo peso y D mínimo. Si hay empate en A, se mira B y así sucesivamente.

A = estilo en línea
B = número de IDs
C = número de clases
D = número de marcas HTML

Prioridad CSS

Cuando dos declaraciones afectan a un mismo elemento. ¿cual de ellas se interpreta en el navegador como más importante?

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```

A = 0 estilos en línea
B = 0 ID
C = 0 clases
D = 2 marcas
Puntuación = 0,0,0,2

Hay que calcular la tupla (A, B, C, D) ganadora de todas las reglas CSS que compiten. A tiene máximo peso y D mínimo. Si hay empate en A, se mira B y así sucesivamente.

A = estilo en línea
B = número de IDs
C = número de clases
D = número de marcas HTML

Prioridad CSS

Cuando dos declaraciones afectan a un mismo elemento. **¿cual de ellas se interpreta en el navegador como más importante?**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```

A = 0 estilos en línea
B = 1 ID
C = 1 clase
D = 1 elemento
Puntuación = 0,1,1,1

Hay que calcular la tupla (A, B, C, D) ganadora de todas las reglas CSS que compiten. A tiene máximo peso y D mínimo. Si hay empate en A, se mira B y así sucesivamente.

A = estilo en línea
B = número de IDs
C = número de clases
D = número de marcas HTML

Prioridad CSS

Cuando dos declaraciones afectan a un mismo elemento. ¿cual de ellas se interpreta en el navegador como más importante?

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```

A = 0 estilos en línea
B = 1 ID
C = 0 clases
D = 2 marcas
Puntuación = 0,1,0,2

Hay que calcular la tupla (A, B, C, D) ganadora de todas las reglas CSS que compiten. A tiene máximo peso y D mínimo. Si hay empate en A, se mira B y así sucesivamente.

A = estilo en línea
B = número de IDs
C = número de clases
D = número de marcas HTML

¿Cómo determinar al ganador?

1. La puntuación de los selectores se determinan en el orden en que se escriben y comparando columnas
2. A partir de la parte de arriba, bajar por la primera columna en busca de la mayor puntuación. Si no encuentras un ganador o dos columnas son iguales, pasar a la siguiente columna (repite si es necesario).
3. Si encuentras dos selectores con valores iguales, el último selector escrito será el que gane.

Prioridad CSS

La ganadora es, #caja .cabecera h1 = 0,1,1,1

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```

Ganadora:

A = 0 estilos en línea

B = 1 ID

C = 1 clase

D = 1 elemento

Puntuación = 0,1,1,1

¿Cómo determinar al ganador?

(0, 1, 1, 10)

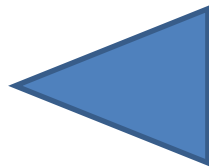
(0, 0, 13, 5)

(0, 0, 7, 3)

(0, 0, 2, 1)

(0, 0, 1, 0)

(0, 1, 2, 2)

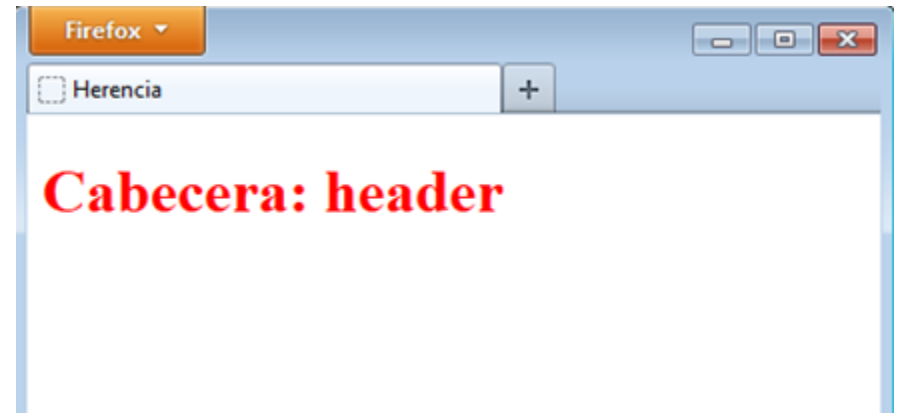


GANADOR

Prioridad CSS

Cuando dos declaraciones tienen el mismo valor:
Será la última especificada

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Herencia</title>
<style type="text/css">
#caja header h1 { color: blue; }
#caja .cabecera h1 { color: red; }
header h1 { color: lime; }
h1 { color: purple; }
h1 { color: grey; }
</style>
</head>
<body>
<div id="caja">
  <header class="cabecera">
    <h1>Cabecera: header</h1>
  </header>
</div>
</body>
</html>
```



Prioridad CSS

- `div #nav li`
- `ol.ing li`
- `ol.ing:hover li`



Prioridad CSS

- `div #nav li` -- 0,1,0,2
- `ol.ing li` -- 0,0,1,2
- `ol.ing:hover li` -- 0,0,2,2



Pseudo-clases

Pseudo-elementos

Pseudo-classes, Pseudo-elementos

Las **Pseudo-classes** nos permiten referirnos a un elemento por su estado

Hover

Active

Focus

Las **Pseudo-elementos** se utilizan para dar estilo a partes concretas de un elemento.

First-line

First-letter

First-child

After

Before

Pseudo-classes

Hover

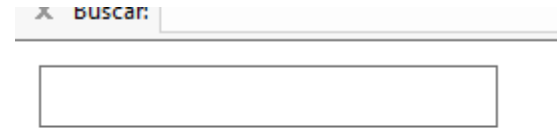
```
p:hover
{
padding:10px;
border:1px gray solid;
background-color:red;
}
```

```
#verde:hover
{
border:1px gray solid;
background-color:green;
}
```

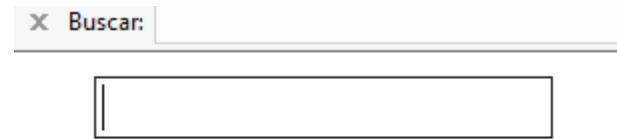
Pseudo-clases

Focus

```
input:focus
{
  color:red;
  position:relative;
  left:10px;
}
```



A search bar with the text "X Buscar:" and a text input field below it.



A search bar with the text "X Buscar:" and a text input field below it.

:focus, se activa cuando el elemento tiene el foco del navegador, es decir, cuando el elemento está seleccionado. Normalmente se aplica a los elementos `<input>` de los formularios cuando están activados y por tanto, se puede escribir directamente en esos campos.

Pseudo-clases

Active

```
div:active
{
background-color:blue;
cursor:pointer;
}
```

```
a:active
{
cursor:pointer;
color:blue;
}
```

:active, se activa cuando el usuario activa un elemento, por ejemplo cuando pulsa con el ratón sobre un elemento. El estilo se aplica durante un espacio de tiempo prácticamente imperceptible, ya que sólo dura desde que el usuario pulsa el botón del ratón hasta que lo suelta.

Pseudo-elementos

first-line y :first-letter

- **:first-line** y **:first-letter** seleccionan una parte de otro elemento, y esas partes operan como si estuviesen separados en elementos inline;
- Sin embargo, solo ciertas propiedades pueden aplicarse a estos pseudo elementos:

Pseudo-elementos

first-line y :first-letter

Propiedad o Categoría	:first-line	:first-letter
Background	yes	yes
Border		yes
Color	yes	yes
Font	yes	yes
Margin		yes
Padding		yes
clear	yes	yes
float		yes
letter-spacing	yes	yes
line-height	yes	yes
text-decoration	yes	yes
text-shadow	yes	yes
text-transform	yes	yes
vertical-align	yes	yes
word-spacing	yes	

Pseudo-elementos (ejemplo)

```
<html>
<head>
  <title>Fortune of Reversal</title>
  <link type="text/css" rel="stylesheet" href="story-5.6.css">
</head>
<body>
  <h1 id="storytitle">Fortune of Reversal</h1>
  <div class="storybody">
    <p>They dined on heaping platters of Szechuan chicken, of
      spicy beef, of shrimp and vegetables in some exotic dish
      without a name. Bits of food were passed from chopsticks
      to chopsticks, violating all known laws of Chinese
      cuisine etiquette. The tea flowed hot and fast that night,
      until the meal finally concluded itself.</p>
    <p>"Thank you for dining here tonight," said the badgeless,
      anonymous waitress. She placed a small tray containing the
      check and two wrapped fortune cookies on the edge of the
      table, and hefted the empty plates one by one, forming a
      stack on the crook of her elbow.</p>
    <p>"Absolutely delicious," declared Oliver as he pulled a card
      from his wallet and flicked it onto the bill. He picked up
      the two cookies, an afterthought. "Fortune cookie, my
      love?" he asked Amanda.</p>
  </div>
</body>
</html>
```

Pseudo-elementos (ejemplo)

```
#storytitle
```

```
{ font-family: Verdana; }
```

```
.storybody p
```

```
{ font-family: Arial; }
```

```
.storybody p:first-line
```

```
{background-color:silver;}
```

```
.storybody p:first-letter
```

```
{ font-size:xx-large;
```

```
color:white;
```

```
background-color:black;
```

```
font-family:Verdana; }
```

Pseudo-elementos (ejemplo)

Fortune of Reversal

Uhey dined on heaping platters of Szechuan chicken, of spicy beef, of shrimp and vegetables in some exotic dish without a name. Bits of food were passed from chopsticks to chopsticks, violating all known laws of Chinese cuisine etiquette. The tea flowed hot and fast that night, until the meal finally concluded itself.

"Thank you for dining here tonight," said the badgeless, anonymous waitress. She placed a small tray containing the check and two wrapped fortune cookies on the edge of the table, and hefted the empty plates one by one, forming a stack on the crook of her elbow.

"Absolutely delicious," declared Oliver as he pulled a card from his wallet and flicked it onto the bill. He picked up the two cookies, an afterthought. "Fortune cookie, my love?" he asked Amanda.

Pseudo-elementos

First-line

`<div>`

`<p> Este es mi texto en Letras Mayuscula </p>`

`<p> Este texto no estara en Mayuscula </p>`

`</div>`

ELEMENTO

PRIMERA LÍNEA

`div:first-line`

`{`

`text-transform:uppercase;`

`}`

Pseudo-elementos

First-line

```
<div>  
<p> Este es mi texto en Letras Mayuscula </p>  
<p> Este texto no estara en Mayuscula </p>  
</div>
```

X Buscar:

[Anterior](#) [Siguiete](#)




0

ESTE ES MI TEXTO EN LETRAS MAYUSCULA

Este texto no estara en Mayuscula

Pseudo-elementos

First-letter



```
<div>  
<p class="LG" >Lorem ipsum dolor sit  
<p class="LG" >Et lorem nunc montes  
<p class="LG" >Libero sed elit turpi  
<p class="LG" >Hendrerit sociis aliqu  
</div>
```

```
.LG:first-letter  
{  
  text-transform:uppercase;  
  font-size:22px;  
  font-weight:bold;  
  padding:4px;  
}
```

Pseudo-elementos

First-letter

```
<div>
<p class="LG" >Lorem ipsum dolor sit
<p class="LG" >Et lorem nunc montes
<p class="LG" >Libero sed elit turpi
<p class="LG" >Hendrerit sociis aliq
</div>
```

Lorem ipsum dolor sit amet consectetur nibh dui tellus sit convallis. Ut dolor In wisi Donec Pellentesque Sed tellus Sed tempus elit. Malesuada consectetur Curabitur fringilla vel sapien interdum Sed tincidunt et laoreet. At porttitor sit non Vivamus metus nonummy adipiscing tellus at convallis. Volutpat Nullam eget orci cursus mauris Duis Sed amet vel semper. Vivamus semper ultrices ornare interdum mauris.

Et lorem nunc montes elit vel tortor congue sodales dictum auctor. Elit condimentum arcu amet Aliquam consectetur vestibulum vel urna sit et. Ipsum id Curabitur faucibus mauris quis odio ornare auctor Duis consequat. Nulla id Aenean consequat feugiat tempus orci Sed tincidunt vitae vel. Hac sapien risus Nam adipiscing elit sed feugiat rhoncus.

Libero sed elit turpis dignissim interdum neque diam eget Phasellus ligula. Non sit In quis consectetur Integer hendrerit In In Sed Cum. At a eleifend wisi eros interdum semper Vestibulum Nulla Donec venenatis. Nulla orci Phasellus Cras Pellentesque accumsan ante non facilisis vel convallis. Morbi Fusce at platea Curabitur ornare pede pretium.

Hendrerit sociis aliquam nulla convallis feugiat nunc Nam magnis Curabitur interdum. Congue malesuada eget Curabitur justo magnis lacinia id pretium lacinia nisl. Sed id elit in magnis Suspendisse sagittis tortor habitant aliquam eu. Et condimentum Aenean eu accumsan hac et semper ante id In. Dui platea libero tellus semper Cras sagittis leo Cum metus congue. Amet pretium sit morbi turpis porttitor orci lacus turpis Sed.

Pseudo-elementos

First-child

```
div:first-child  
{  
  background-color:black;  
  color:white;  
}
```

PRIMERA ETIQUETA DIV
QUE CONSIGA
APLICARLE LOS
ATRIBUTOS.

```
<div>
```

```
Este no sera tomado (Debido a que es Segundo)
```

```
</div>
```

Pseudo-elementos

First-child

PRIMERA ETIQUETA DIV
QUE CONSIGA
APLICARLE LOS
ATRIBUTOS.

DIV

Este sera mi texto tomado.

/DIV

DIV

Este no sera tomado (Debido a que es Segundo)

/DIV

Pseudo-elementos

Before

Nos permitirá Insertar contenido antes de un Elemento.

```
<div>  
<p id="elemento" > Contenido </p>  
</div>
```

```
#elemento:before  
{  
  content:" Previo Elemento: ";  
  color:red;  
}
```

Pseudo-elementos

Before

Previo Elemento: Contenido

Pseudo-elementos

After

```
<div>  
<p id="elemento" >Elemento Contenido</p>  
</div>
```

```
#elemento:after  
{  
content:url("carpeta/imagen.jpg");  
}
```

Pseudo-elementos

After



Elemento Contenido

Selectores

¿Cómo trabaja `:nth-child()` ?

- `:nth-child()` selecciona un elemento en base a su posición dentro de la lista de hijos de su padre; en otras palabras, se selecciona un elemento basado en cuanto hermanos hay antes de él.
- Escribes la posición del elemento que quieres seleccionar dentro de los paréntesis del elemento.
- Por ejemplo `li:nth-child(5)` podría emparejará el quinto elemento `li` de una lista.
- Además del número dentro de los paréntesis (el argumento del selector), podrían usarse las palabras **odd** o **even**.

¿Cómo trabaja :nth-child() ?

- `:nth-child()` donde realmente es poderoso es cuando se utiliza una fórmula en su argumento.
- Esto permite crear patrones de alternancia más complejos.
- La fórmula tiene la sintáxis $an+b$, donde a es el tamaño del incremento que elijas, n es un contador que comienza en cero, y b es el valor de desplazamiento.

¿Cómo trabaja `:nth-child()` ?

- Por ejemplo: `li:nth-child(3n+1)`
 - $(3 \times 0) + 1 = 1$ = primer elemento de la lista
 - $(3 \times 1) + 1 = 4$ = cuarto elemento de la lista
 - $(3 \times 2) + 1 = 7$ = séptimo elemento de la lista
 - $(3 \times 3) + 1 = 10$ = décimo elemento de la lista

También disponen otros selectores `:nth-last-child()`, `:nth-of-type()`, `:nth-last-of-type()`, `:empty()`, `:first-child()` y `:last-child()`.

Pseudo-clases

`:nth-last-child(an+b)`

- Todos los elementos DESDE EL FINAL en la lista de todos los elementos con el mismo padre puede contarse con $an+b$

- ejemplos:

`:nth-last-child(-n+2)` – últimos 2 elementos de cada padre

Pseudo-classes - Herramientas on-line

- NTH-TEST (<http://nth-test.com/>)
- CSS-TRICKS (<http://css-tricks.com/examples/nth-child-tester/>)
- CSS nth-child Tester
(<http://coveloping.com/tools/css-nth-child-tester>)

Pseudo-clases

¿Trata de conseguir el siguiente aspecto?

Text

Text

Text

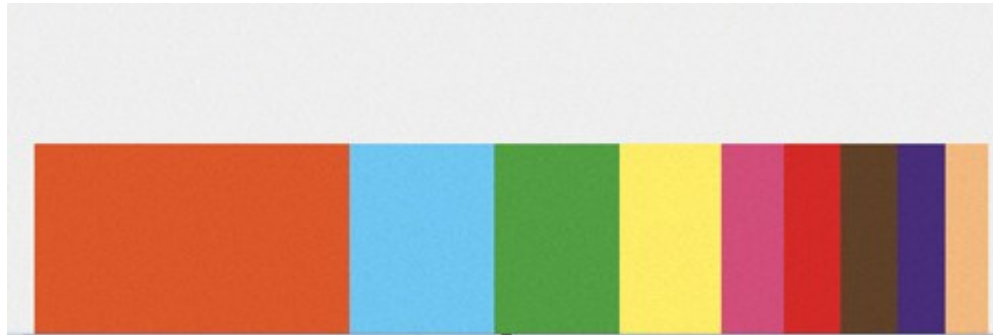
Text

Text

Text

Pseudo-classes

¿Trata de conseguir el siguiente aspecto?



Pseudo-clases

¿Trata de conseguir el siguiente aspecto?

Rango positivo de childs
 $An + B$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.



Rango genérico de childs:
 $An+B$ $-An+B$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.



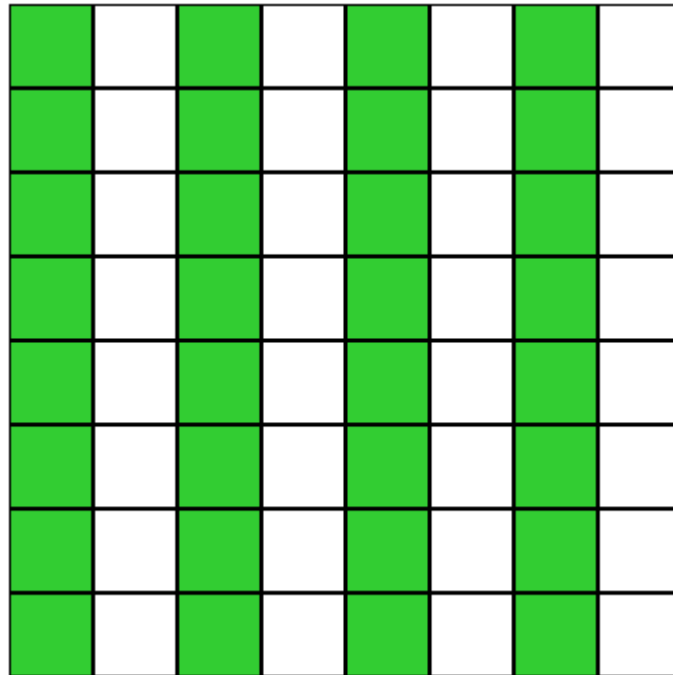
Rango negativo de childs
 $-An + B$

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.



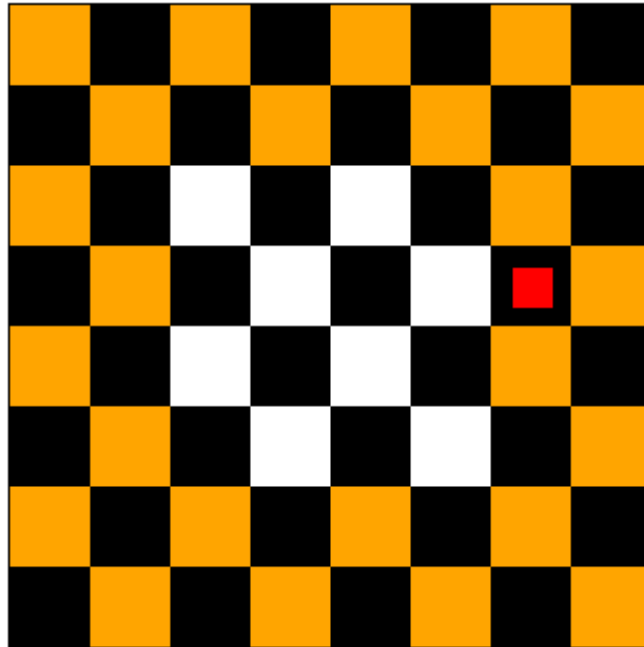
Pseudo-classes

¿Trata de conseguir el siguiente aspecto?



Pseudo-classes

¿Podrías hacer el siguiente tablero con
`:nth-child`, `:nth-last-child`?



Pseudo-classes

`:first-child = :nth-child(1)`

`:last-child = :nth-last-child(1)`

`:first-of-type = :nth-of-type(1)`

`:last-of-type = :nth-last-of-type(1)`

`:only-child = :first-child:last-child`

`:only-of-type = :first-of-type:last-of-type`

`:empty`

Otras Pseudo-clases

`:not (PS)`

- Todos los elementos que no se correspondan a la pseudo-clase *PS*

`:checked`

Muy útil para seleccionar las casillas marcadas

Selector e hijos

Para seleccionar un elemento que es hijo directo de otro elemento.

- Se indica mediante el signo >

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

- Se aplica el selector

```
<p><span>Texto1</span></p>
```

- No se aplica

```
<p><a href="#"><span>Texto2</span></a></p>
```

El selector descendente (`p span {color:blue;}`) se aplicaría en los dos casos previos.

Selector adyacentes

- Para seleccionar elementos que se encuentran seguidos se indica mediante el signo +

```
p + p { text-indent: 1.5em; }
```

- Selecciona todos los párrafos de una página que estén precedidos por otro, esto es, todos menos el primero

```
h1 + p { text-indent: 1.5em; }
```

- Selecciona todos los párrafos que estén precedidos por un h1

Sirve para seleccionar un único elemento que es consecutivo de otro. Ni descendiente directo ni más abajo en el anidamiento, sino a la misma "altura" y justo detrás, sin otros elementos en medio. Para ello se utiliza el símbolo + entre dos selectores.

Nuevos Selectores

Con los selectores de CSS3 podemos seleccionar un elemento basándonos en los contenidos del atributo

Nuevos selectores

Empieza con el prefijo indicado

- El “empieza con” tiene la siguientes sintaxis:
`elemento[atributo^="valor"]`
- Esta opción selecciona al elemento que tiene un atributo cuyo nombre empieza por valor.

Veamos un ejemplo:

```
img[alt^="foto"] {  
    border:2px solid #666;  
}
```

Nuevos selectores

Termina con el sufijo indicado

- Como bien te puedes imaginar, ahora es lo mismo, con la diferencia de que termina con la palabra indicada. La estructura sería:

`elemento[atributo$="valor"]`

- Y en el ejemplo anterior, el símbolo ahora es el dólar \$:

```
img[alt$="foto"] {  
    border:2px solid #666;  
}
```

Nuevos selectores

HTML

```

```

```

```

CSS

```
img[alt*="2031"] {  
  // estilos de las películas del 2031 }  
  
img[alt$="terror"] {  
  // estilos de las películas de terror }  
  
img[alt$="infantil"] {  
  // estilos de las películas infantiles }
```

Combinadores CSS

- El único tipo de **combinador** nuevo introducido en CSS3 es el selector hermano o *sibling*. Está dirigido a todos los hermanos de un elemento que tienen el mismo padre.
- Por ejemplo, para añadir un borde gris en todas las imágenes que sean hermanas de un div en particular hacemos lo siguiente:

```
div~img {  
    border: 1px solid #ccc;  
}
```


Selector atributo

[atributo] elementos con un atributo específico.

[atributo = valor] elementos con un atributo específico y un valor

[atributo ~= valor] elementos con un atributo específico que contiene una palabra

[atributo |= valor] elementos con un atributo específico que comience por una palabra

```
a[target] {  
    background-color: yellow;  
}
```

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

```
[title~="flower"] {  
    border: 5px solid yellow;  
}
```

```
[class|="top"] {  
    background: yellow;  
}
```

Selector atributo

[atributo ^= valor] elementos cuyo valor de atributo comienza por valor

```
[class^="top"] {  
    background:  
yellow;  
}
```

[atributo \$= valor] elementos cuyo valor de atributo terminen por valor

```
[class$="test"] {  
    background:  
yellow;  
}
```

[atributo * = valor] elementos cuyo valor de atributo contiene por valor

```
[class*="te"] {  
    background:  
yellow;  
}
```

Counter-increment

Sirven para llevar un **contador** en CSS utilizado en los atributos **content** para incluirlos en el documento.

```
ul
{
counter-reset:counter;
list-style-type:none;
}
li:before
{
content: counter(contador);
counter-increment: contador;
}
```

```
<ul>
<li> Capitulo </a></li>
<li> Capitulo </a></li>
</ul>
```

Counter-increment

```
ul
{
  counter-reset: counter;
  list-style-type: none;
}
li:before
{
  content: counter(contador);
  counter-increment: contador;
}
```

 ``` Capitulo `` Capitulo ```

Counter-increment

```
ul
{
  counter-reset: counter;
  list-style-type: none;
}
li:before
{
  content: counter(contador);
  counter-increment: contador;
}
```


■ Capitulo

 Capitulo

Counter-increment

```
ul
{
  counter-reset: counter;
  list-style-type: none;
}
li:before
{
  content: counter(contador);
  counter-increment: contador;
}
```


  Capitulo

 Capitulo

Counter-increment

```
ul
{
  counter-reset: counter;
  list-style-type: none;
}
li:before
{
  content: counter(contador);
  counter-increment: contador;
}
```


1 Capitulo

2 Capitulo

Counter-increment

```
ul
{
  counter-reset: contador;
  list-style-type: none;
}
li:before
{
  content: counter(contador);
  counter-increment: contador 2;
}
```

Puede ser incrementado de forma positiva o Negativa

+2

-2

Counter-increment

Sirven para llevar un **contador** en CSS utilizado en los atributos **content** para incluir un conteo automático de selector en el documento.

```
ul
{
  counter-reset:contador +5;
  list-style-type:none;
}
li:before
{
  content: counters(contador, '.');
  counter-increment: contador;
  /*background-color:blue; */
}
```

Counter-increment

decimal
decimal-leading-zero
lower-roman
upper-roman
lower-greek
lower-alpha
upper-alpha

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola

- a. Coffee
- b. Tea
- c. Coca Cola

Selector de parámetros

Es posible que a través de CSS podamos incluir características en elementos que tengan atributos en específico.

```
input[type=text]
{
  color:blue;
  width:120px;
  display:block;
}
[title]
{
  color:blue;
}
```

Selector de parámetros

```
input[type=text]
```

```
{  
color:blue;  
width:120px;  
display:block;  
}
```

```
[title]
```

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

```
<h1 title="TituloP" > Titulo de Pagina </h1>
```

```
<div>
```

```
Nombre:
```

```
<input type="text" value="Nombre" />
```

```
Apellido
```

```
<input type="text" value="Apellido" />
```

```
<input type="button" value="Botoncito" />
```

```
</div>
```

Selector de parámetros

```
input[type=text]
{
color:blue;
width:120px;
display:block;
}
[title]
{
color:blue;
}
```

Ejemplo

```
<h1 title="TituloP" > Titulo de Pagina </h1>
<div>
Nombre:
<input type="text" value="Nombre" />
Apellido
<input type="text" value="Apellido" />
<input type="button" value="Botoncito" />
</div>
```