

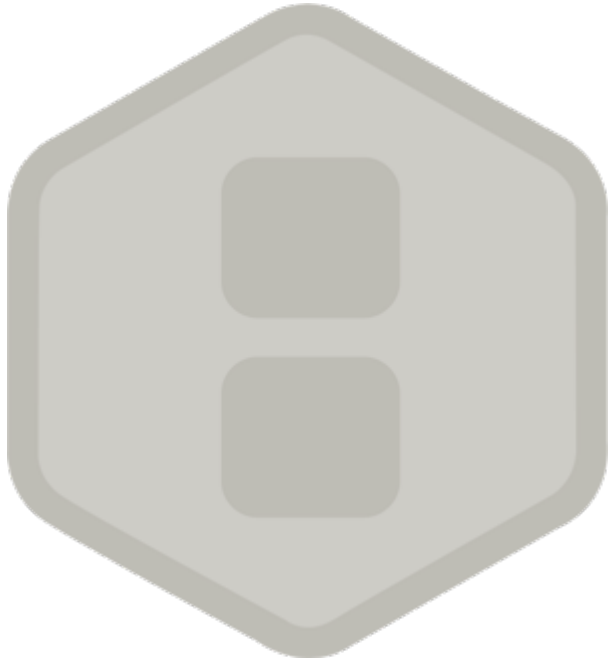
<Uso de estilos. Selectores>

Diseño de Interfaces Web

<CSS3>

Cascading Style Sheets Level 3

José María Torresano
Octubre 2018
daw.cierva@gmail.com



PSEUDO CLASES

Pseudo clases

Un estado o uso predefinido de un elemento al que se le puede aplicar un estilo independientemente de su estado por defecto.

1. Links
2. Dinámicas
3. Estructurales
4. Otras

1. Links

Son los más básicos y se centran en si se ha pulsado un enlace o no (elemento <a>)

:link	Aplica un estilo a un enlace sin visitar
:visited	Aplica un estilo a un enlace visitado

Por defecto, suelen estar en azul y púrpura, respectivamente

```
a:link {  
    color: maroon;  
}  
a:visited {  
    color: gray;  
}
```

Enlaces visitados y privacidad

Durante más de una década, fue posible personalizar los enlaces visitados con cualquier propiedad CSS disponible, al igual que con los enlaces no visitados. Sin embargo, a mediados de la década del 2000, varias personas demostraron que se podía usar un estilo visual y secuencias de sencillos comandos DOM para determinar si un usuario había visitado una página determinada. Por ejemplo, dada la regla: `visited {fontweight: bold;}`, una secuencia de comandos podría encontrar todos los enlaces en negrita e indicar al usuario cuáles de esos sitios había visitado o, lo que es peor, informar de esos sitios a un servidor. Una táctica similar usa imágenes de fondo para lograr el mismo resultado.

Si bien esto puede no parecerle demasiado grave, puede ser completamente devastador para un usuario de la red en un país donde se te puede encarcelar por visitar determinados sitios: partidos de oposición, organizaciones religiosas no autorizadas, sitios "inmorales" o "corruptos", etcétera. También puede ser utilizado por sitios de phishing para determinar qué bancos en línea ha visitado un usuario.

Como consecuencia, se dieron dos pasos. El primer paso es que solo se pueden aplicar propiedades relacionadas con el color a los enlaces visitados: color, color de fondo, color regla de columna, color de contorno, color de borde y propiedades de color de borde de lado individual (por ejemplo, `borde-top-color`). Se ignorarán los intentos de aplicar cualquier otra propiedad a un enlace visitado. Además, cualquier estilo definido para `:link` se aplicará a los enlaces visitados así como a los enlaces no visitados, lo que hace de `:link` el "estilo para cualquier hipervínculo", en lugar de "estilo para cualquier hipervínculo no visitado".

El segundo paso es que si se consultan los estilos a través de DOM de un enlace visitado, el valor resultante será como si el enlace no se hubiera visitado. Por lo tanto, si ha definido que los enlaces visitados sean de color violeta en lugar del azul de los enlaces no visitados, aunque el enlace aparezca en color púrpura en la pantalla, una consulta al DOM devolverá el valor azul, no el violeta.

A finales de 2018, este comportamiento está presente en todos los modos de navegación, no solo en los modos de "navegación privada". Aunque estamos limitados en cómo poder usar CSS para diferenciar los enlaces visitados de los enlaces no visitados, es importante para la usabilidad y la accesibilidad utilizar los estilos limitados compatibles con los enlaces visitados para diferenciarlos de los enlaces no visitados.

2. Dinámicas

Son las relacionadas con acciones del usuario (mover el ratón, pasar sobre un elemento, etc)

:focus	Aplica un estilo cuando se ha seleccionado un elemento y está listo para entradas de teclado, etc
:hover	Aplica un estilo cuando el ratón pasa sobre un elemento
:active	Aplica un estilo cuando el elemento (tal como un enlace o un botón) está listo para recibir una pulsación del ratón, una pulsación del teclado, etc

No existe el estado **hover** en los dispositivos móviles aunque pueden mostrarlo después de pulsar una vez el elemento.

```
input:focus { color: yellow; }  
a:hover { color: marron; background-color: #ffd9d9;}  
a:active { color: red; background-color: #ffd9d9;}
```

```
<h3>Ejemplos de mi trabajo</h3>
```

```
<ul>
```

```
<li><a href=#uno>Ilustraciones a boli</a></li>
```

```
<li><a href=#dos>Cuadros</a></li>
```

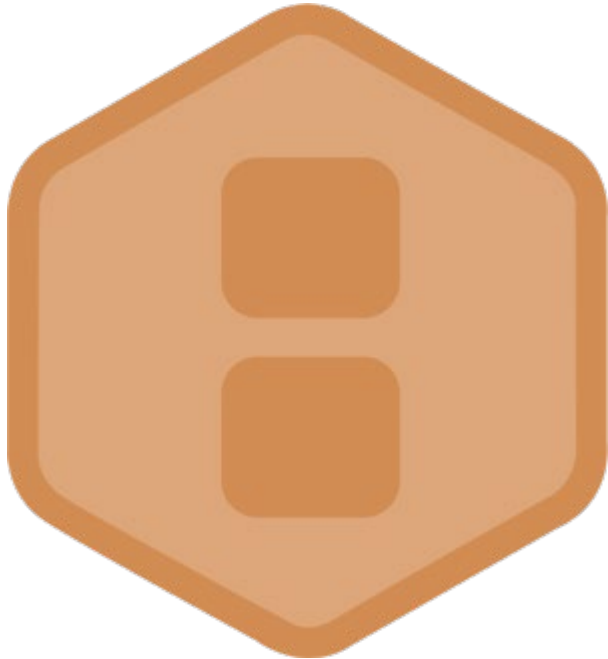
```
<li><a href=#tres>Collages</a></li>
```

```
</ul>
```

EJEMPLOS DE MI TRABAJO

- ILUSTRACIONES A BOLI
- CUADROS
- COLLAGES



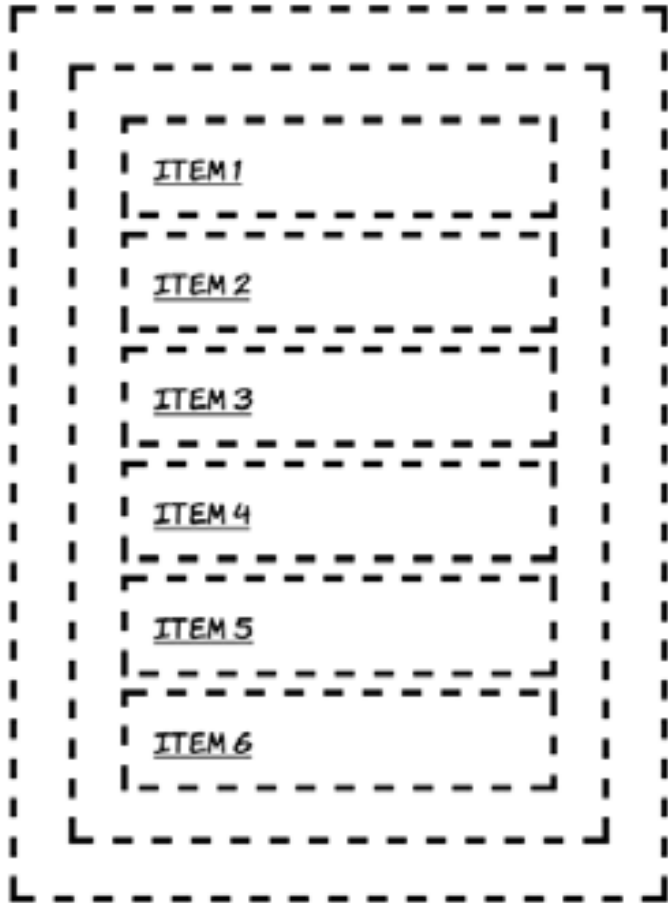


PSEUDO CLASES ESTRUCTURALES

3. Pseudo clases estructurales

Permiten dar estilo a elementos basándonos en una posición numérica exacta del elemento

:root	Raíz	Es el elemento principal de un documento
:only-child	Hijo Único	No tiene nodos hermanos
:only-of-type	Solo del Tipo	Sólo hay uno de su tipo entre sus hermanos
:first-child	Primer Hijo	Es el primer nodo hijo de otro elemento
:last-child	Último Hijo	Es el último nodo hijo de un elemento
:nth-child(n)	Nth hijo	Es el n -enésimo elemento
:nth-last-child(n)	Nth hijo	Es el n -enésimo elemento contando desde el último
:nth-of-type(n)	Nth del Tipo	Es el n -enésimo elemento con ese tipo
:first-of-type	Primero del Tipo	Es el primer elemento de su tipo en el elemento primario
:last-of-type	Último del Tipo	Es el último elemento de su tipo en el elemento primario
:nth-last-of-type(n)	Nth del Tipo	Es el n -enésimo elemento de su tipo desde abajo
:empty()	vacío	Es el elemento que no tiene hijos



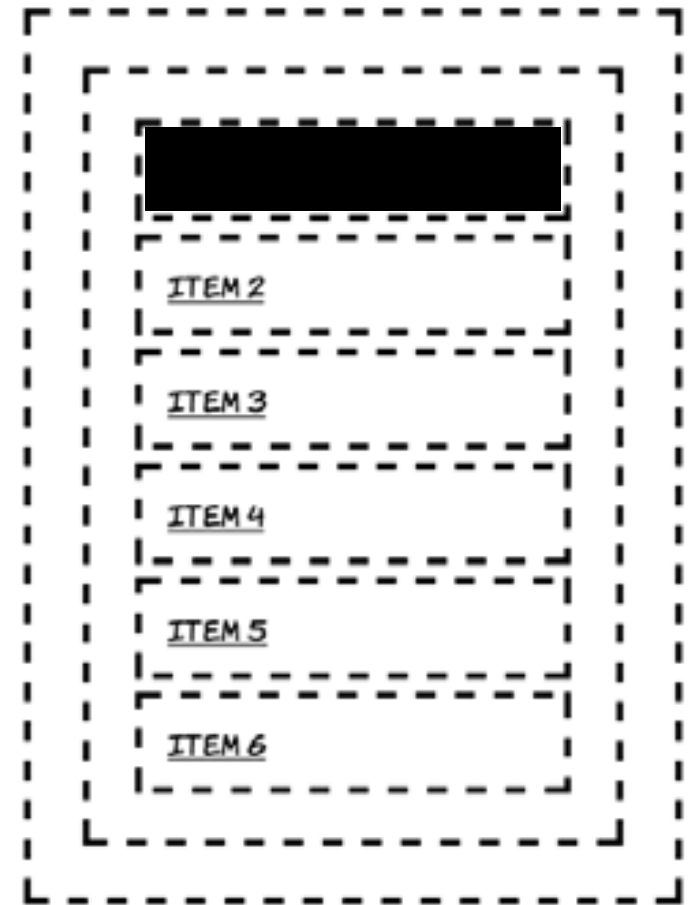
```
<nav>
  <ul>
    <li>
      <a href="#item1">Item 1</a>
    </li>
    <li>
      <a href="#item2">Item 2</a>
    </li>
    <li>
      <a href="#item3">Item 3</a>
    </li>
    <li>
      <a href="#item4">Item 4</a>
    </li>
    <li>
      <a href="#item5">Item 5</a>
    </li>
    <li>
      <a href="#item6">Item 6</a>
    </li>
  </ul>
</nav>
```

E:first-child {}

Selecciona el primer elemento de un tipo de selector (E) si es el primer nodo hijo de su elemento padre.

```
ul li:first-child {  
  background-color: #000;}
```

SELECCIONA LOS ELEMENTOS **** QUE SON
EL PRIMER HIJO DE SUS RESPECTIVOS ****

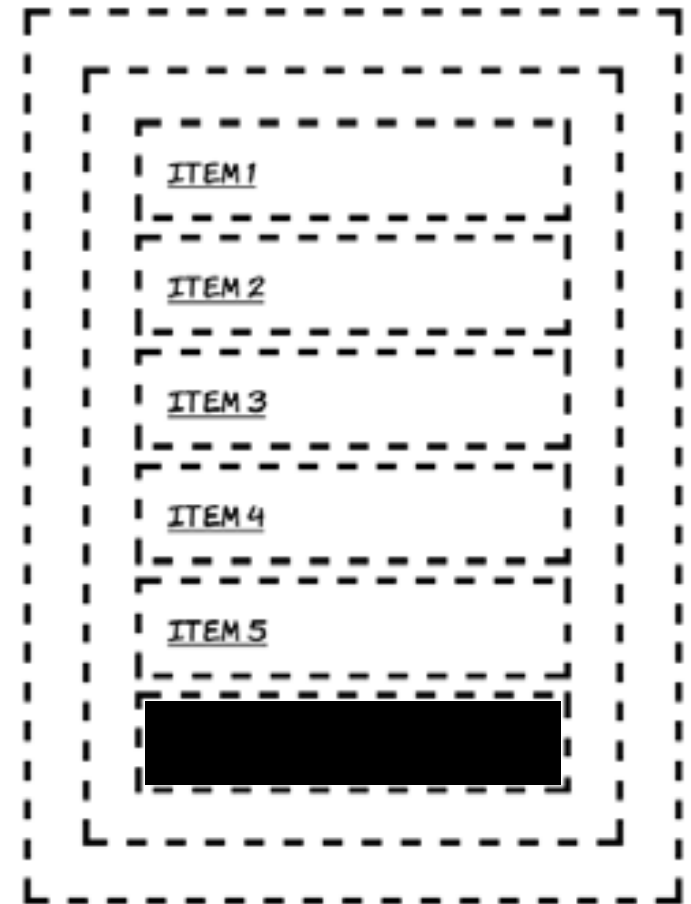


E:last-child {}

Selecciona el último elemento de un tipo de selector (E) si es el último nodo hijo de su elemento padre.

```
ul li:last-child {  
  background-color: #000;}
```

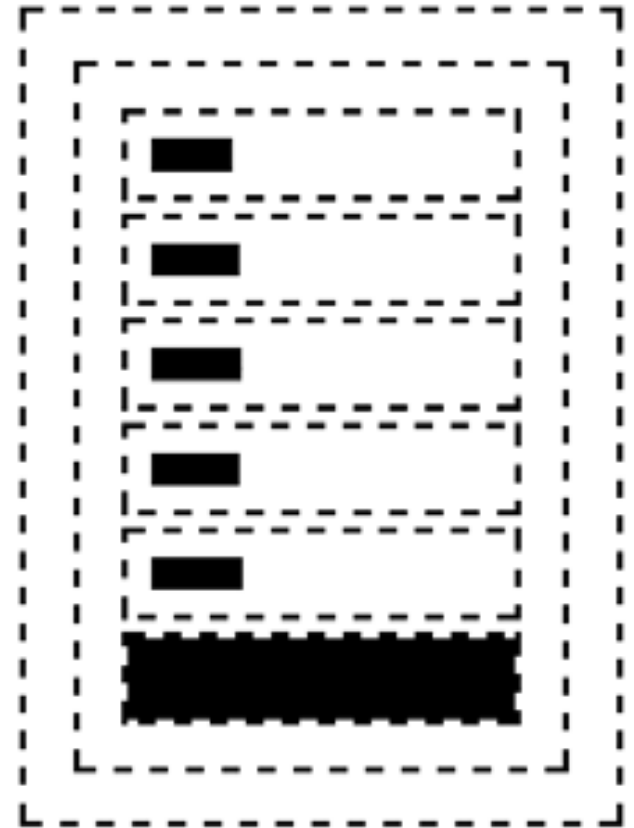
SELECCIONA LOS ELEMENTOS **** QUE SON
EL ÚLTIMO HIJO DE SUS RESPECTIVOS ****





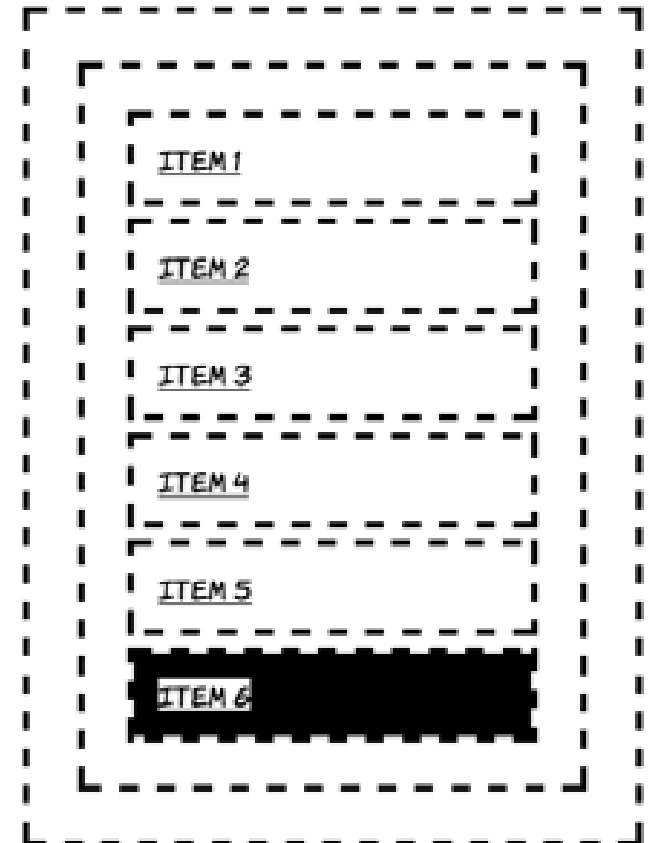
```
ul :last-child {  
  background-color: #000;}  
}
```

SELECCIONA CUALQUIER
ELEMENTO QUE SEA UN
ÚLTIMO HIJO






```
li:last-child {  
    background-color: #000;}  
  
li:last-child a {  
    background-color: #fff;}
```



$E:nth-child(n) \{ \}$
 $E:nth-of-type(n) \{ \}$
 $E:nth-last-child(n) \{ \}$
 $E:nth-last-of-type(n) \{ \}$

Basan la selección del elemento por la posición en el árbol del documento.

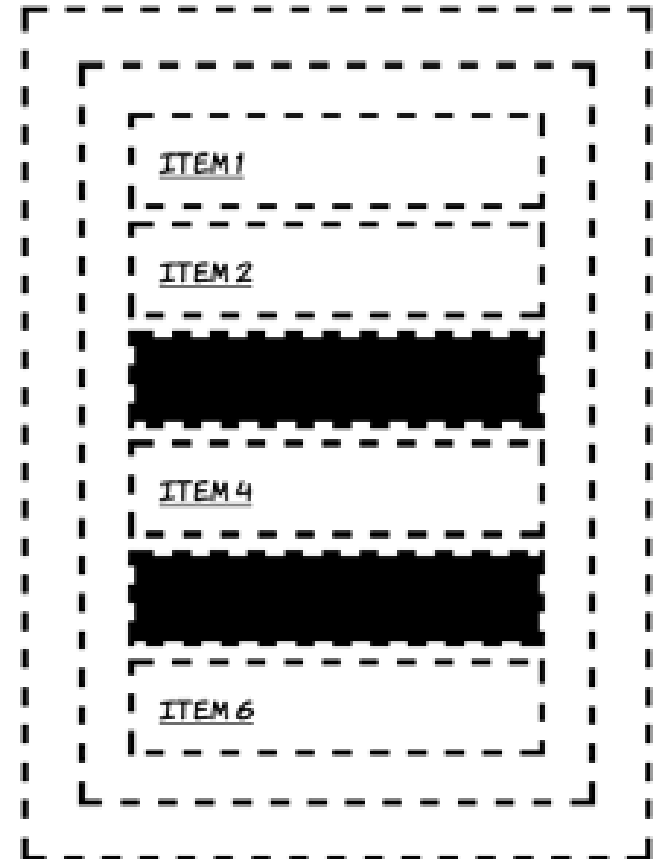
$$n \rightarrow \pm a * N \pm b$$


 CON $N = 0, 1, 2, 3, 4, \dots$

n	N = ODD		N = EVEN		$4n$	$5n-2$	$-n+3$
	$2n+1$	$2n$	$4n+1$	$4n+4$			
0	1	-	1	4	-	-	3
1	3	2	5	8	4	3	2
2	5	4	9	12	8	8	1
3	7	6	13	16	12	13	-
4	9	8	17	20	16	18	-
5	11	10	21	24	20	23	-

E:nth-* {}

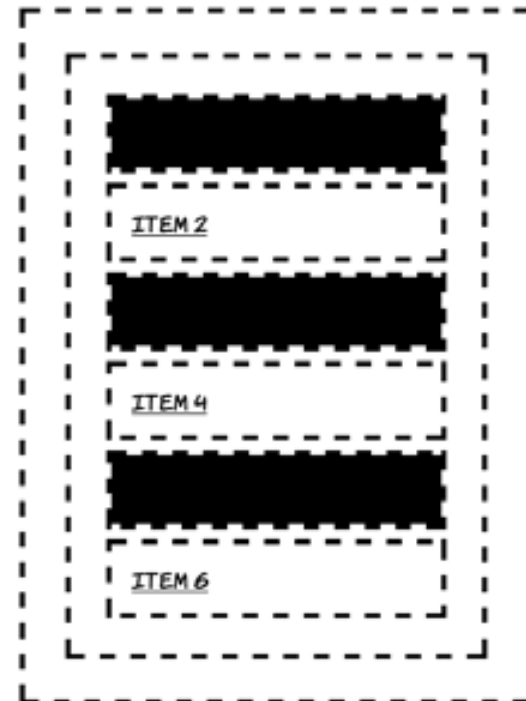
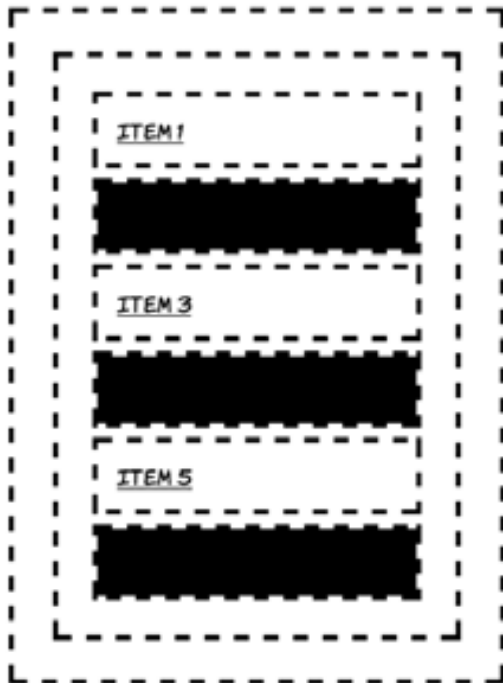
```
li:nth-child(3), li:nth-child(5) {  
  background-color: #000;}  
}
```



E:nth-* {}

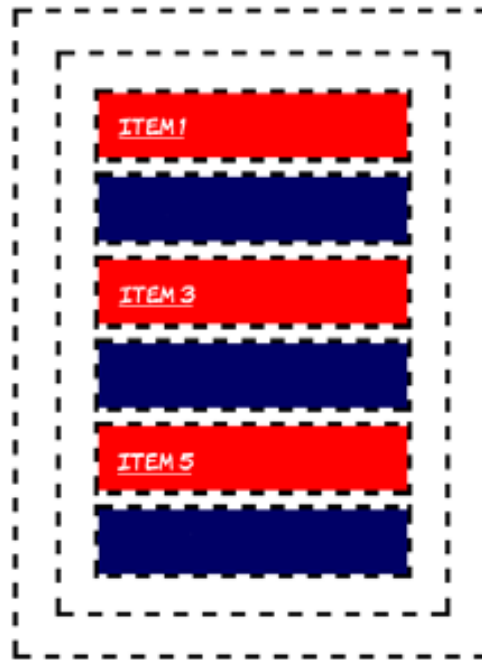
```
li:nth-child(2n) {  
  background-color: #000;}
```

```
li:nth-child(2n+1) {  
  background-color: #000;}
```



E:nth-* {}

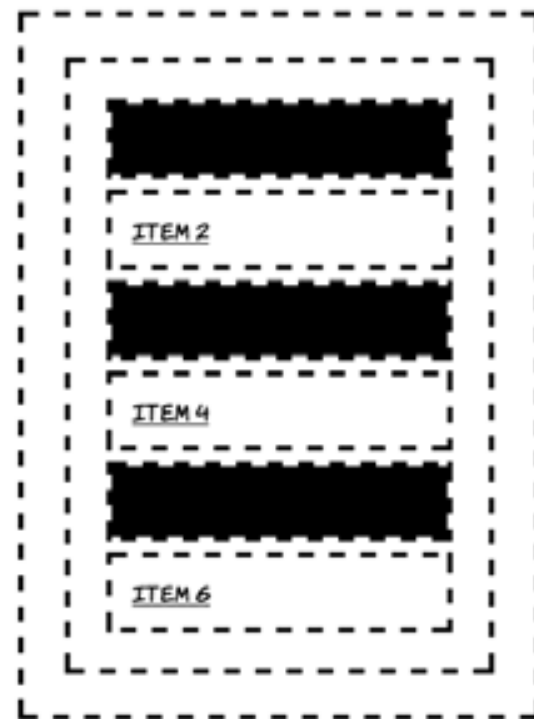
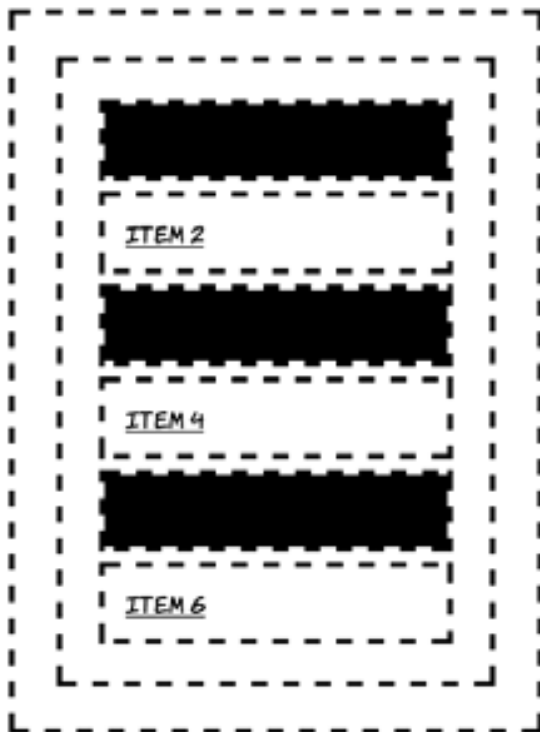
```
li:nth-child(even) {  
  background-color: #006;}  
li:nth-child(odd) {  
  background-color: #f00;}
```



E:nth-* {}

```
li:nth-last-child(2n) {  
  background-color: #000;}
```

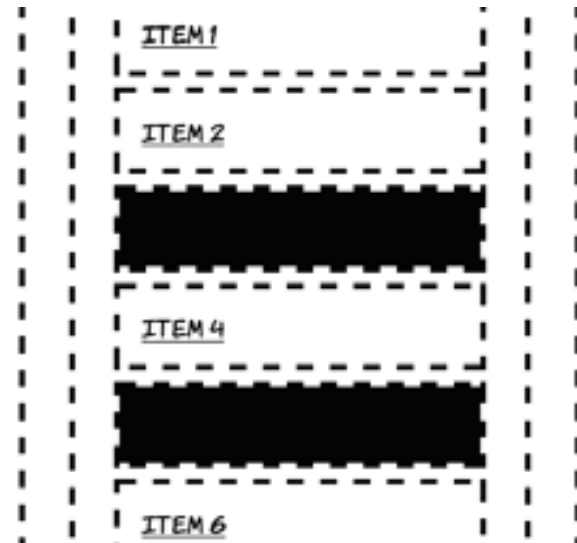
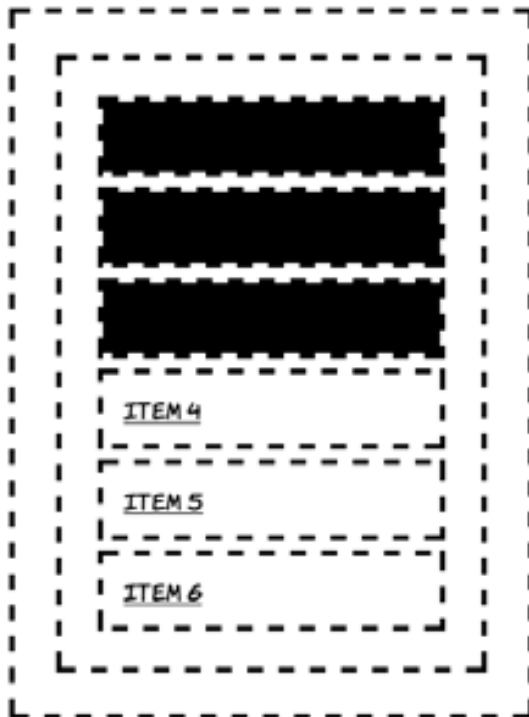
```
li:nth-last-child(2n+2) {  
  background-color: #000;}
```



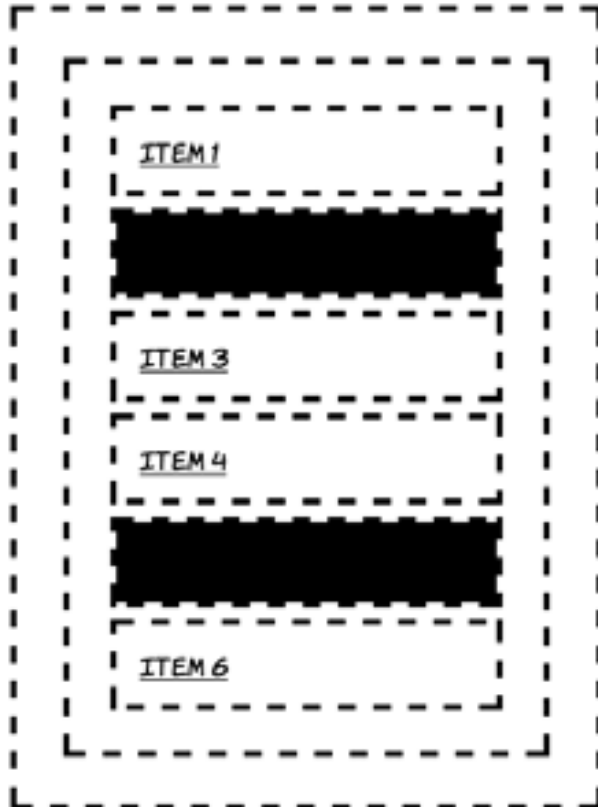
E:nth-* {} $n \rightarrow \pm a * N \pm b$ 

```
li:nth-child(-n+3) {
  background-color: #000;}
```

```
li:nth-last-child(-2n+4) {
  background-color: #000;}
```



E:nth-* {}

$$n \rightarrow \pm a * N \pm b$$


¿Un elemento de cada tres?

```
li:nth-child(3n+2) {  
  background-color: #000;}
```

$E:nth-* \{ \}$

$n \rightarrow \pm a^* N \pm b$



¿Patrón ABC?

```
ul li:nth-child(3n+1) {
  background-color: #006;}
ul li:nth-child(3n+2){
  background-color: #f00;}
ul li:nth-child(3n+3) {
  background-color: #0f0;}
```

E:only-child {}

Selecciona el único hijo de ese elemento (si existe)

E:only-of-type {}

Selecciona el único hijo de su tipo de ese elemento (si existe)

E:empty {}

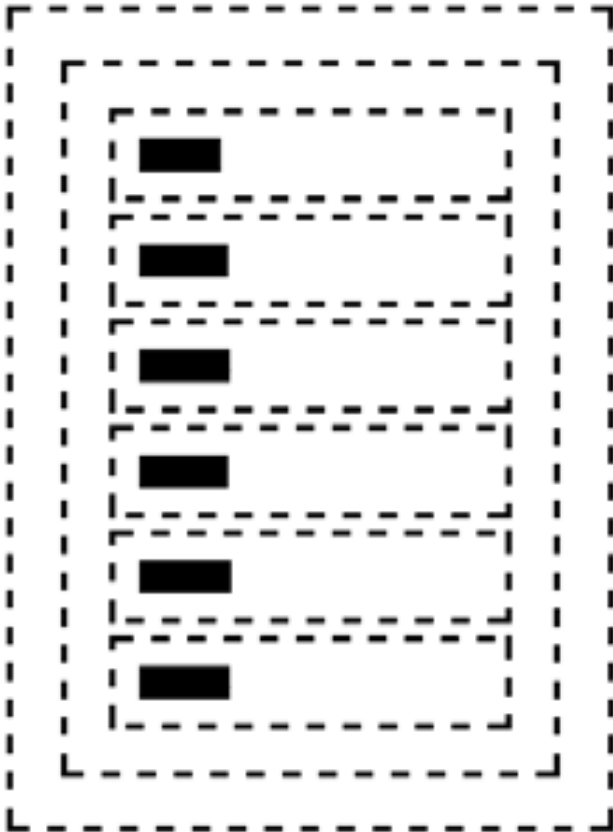
Selecciona el elemento que no tiene hijos (el texto se considera hijo)

E:root {}

Selecciona el elemento HTML que es el raíz.

E:only-child {}

```
ul :only-child {  
  background-color: #000;}
```



```
ul :first-child:last-child {  
  background-color: #000;}
```


E:first-of-type {}

Primer elemento hijo de un determinado tipo (E).

E:last-of-type {}

Último elemento hijo de un determinado tipo (E).

E:nth-of-type {}

El elemento hijo enésimo de un determinado tipo (E).

E:nth-last-of-type {}

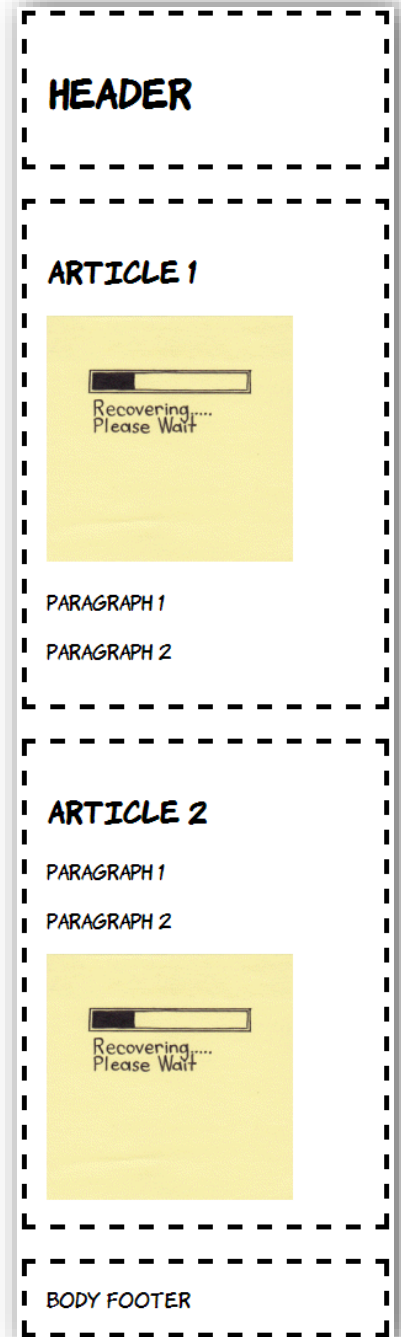
El elemento hijo enésimo, desde el último, de un determinado tipo (E).

E:*-of-type {}

```

<header>
  <h1>Header</h1>
</header>
<article>
  <h1>Article 1</h1>
  
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</article>
<article>
  <h1>Article 2</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  
</article>
<footer>
  Body footer
</footer>

```



E:*-of-type {}

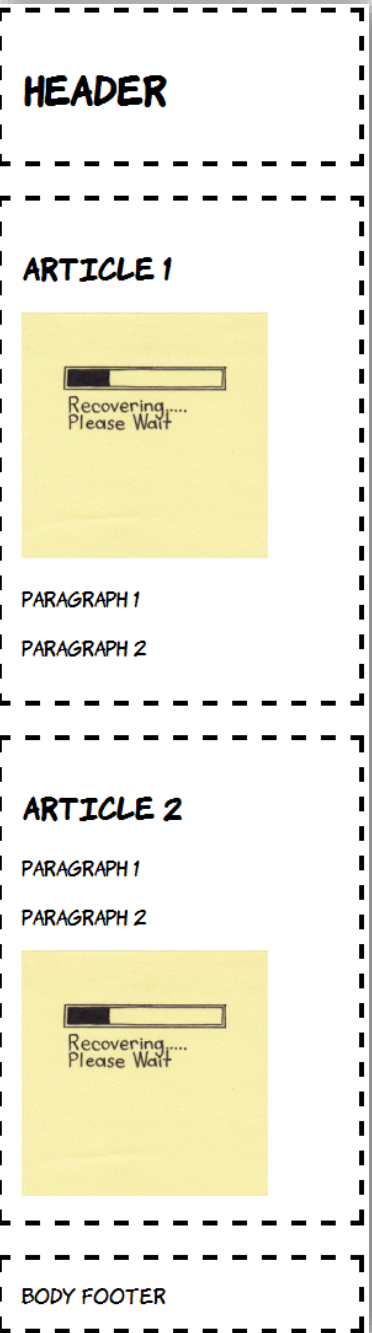
```
article:first-child {  
  background-color: #000;}  
↑
```

SELECCIONA EL PRIMER <ARTICLE>

```
p:last-child {  
  background-color: #000;}  
↑
```

SELECCIONA EL
ÚLTIMO <P> DE CADA
<ARTICLE>

Ni <article> es el primer hijo de ningún contenedor ni <p>, el último.



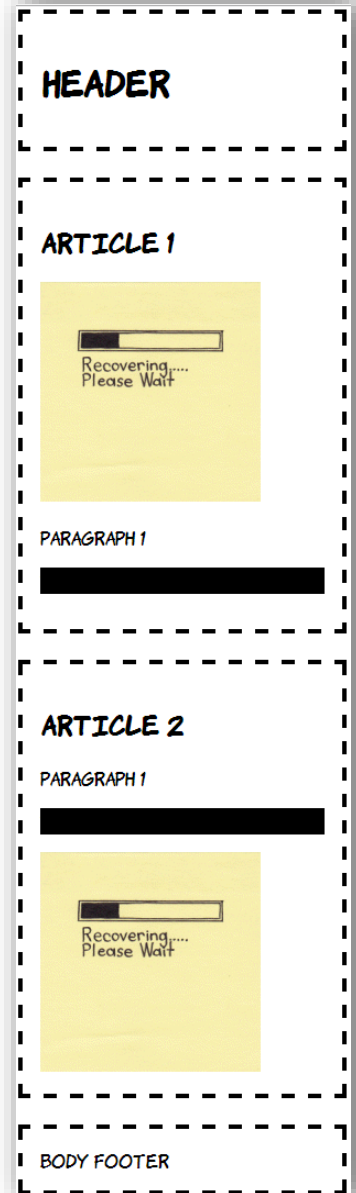
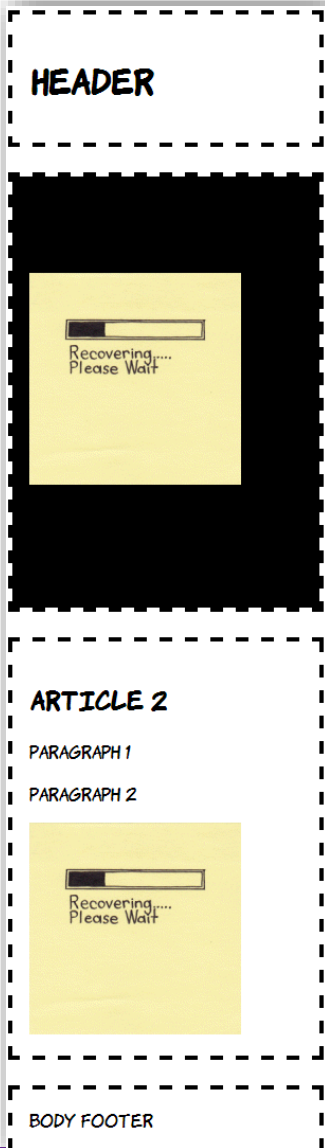
E:*-of-type {}

```
article:first-of-type {
  background-color: #000;}

```

```
p:last-of-type {
  background-color: #000;}

```





</Uso de estilos. Selectores>

Diseño de Interfaces Web

<CSS3>

Cascading Style Sheets Level 3

José María Torresano
Octubre 2018
daw.cierva@gmail.com