

CSS 3

IES. Martín Rivero Ronda.

Autor: Juan M. Márquez - 2018

Preparación de navegadores antiguos

- Muestran todos los nuevos elementos en línea, por lo que hay que decirles que algunos de ellos funcionan en bloque.
- Añadiremos siempre este código:

```
Article, aside, figure, footer, header, hgroup, nav, section
{
    display:block;
}
```

Nuevos Selectores CSS3

Selector general de elementos Adyacentes

~ Hermanos

h1 ~ p Todos los elementos p que sean hermanos y que se encuentran a continuación de un h1. El elemento, **NO necesita ir inmediatamente a continuación**.

Parecido a selector **+** (selector adyacente). En este caso, el elemento tenía que estar inmediatamente a continuación.

Nuevos Selectores CSS3

Selector general de elementos Adyacentes

~ Hermanos

h1 ~ p ¿A quien afecta?

Y el selector **h1 + p** ?

<p> Bloque 1 </p>

<h1>Titulo </h1>

➡ <p> Párrafo 1 </p>



Solo afectaría este párrafo. <p> que va justo a continuación de <h1>

➡ <p> Párrafo 1 </p>

<div>

<p> Párrafo 1 </p>

</div>



A este P **NO** afectaría. No es hermano

➡ <p>Párrafo 3</p>

<p> hermanos que estén en cualquier posición pero a partir de un <h1>

Selectores CSS3 de atributo:

p[name] : Etiquetas p que contengan el atributo name.

p[name="x"]: Etiquetas p que contengan atributo name con el valor "x"

p[name | ="x"]: Párrafos cuyo name tiene una lista de palabras separados con guiones y que empiezan por "x"

p[name~="x"]: Atributo contiene palabra completa "x" (rodeada de espacio)

p[name^="x"]: Que comiencen por ...

p[name\$="x"]: Que acaben en ... (Vínculo de descarga)

p[name*="x"]: Que contengan(parte de una palabra) ...

Ejemplos:

```
a[href^="http://"] .info[title~=more] *[lang | ="es"]  
/*elementos de la página cuyo atributo "lang" empieza por "es", es decir, "es",  
"es-ES", "es-AR", etc. */
```

Pseudo-Clase de Vínculo

:target Elemento destino del vínculo interno en la página activa.

Cambiará continuamente, según el ancla activa en cada momento.

Pseudo-clases estructurales

Nos permiten seleccionar con mas precisión los elementos.

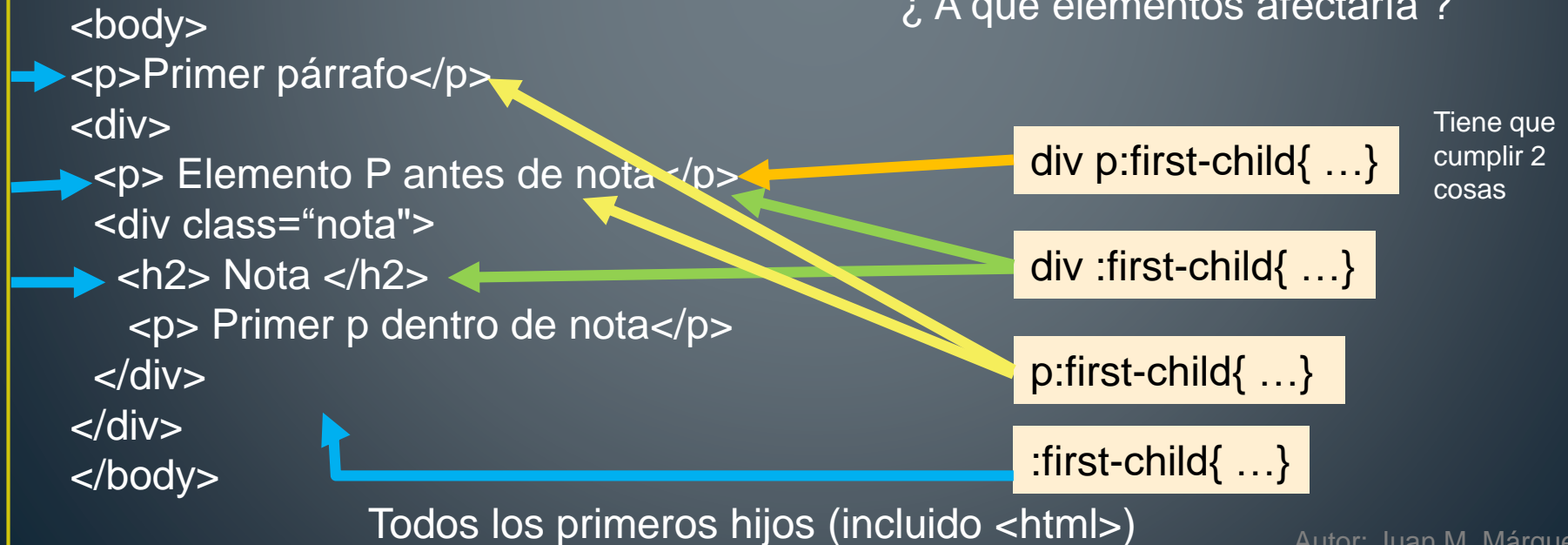
➡ **:first-child** Primer hijo. `<body>` es **last-child** de `<html>`

div p:first-child: Tiene que cumplir 2 cosas:

Que sea primer hijo de div y que sea de tipo p.

ul :first-child : Primeros hijos de ul (da igual qué elemento sea el primer hijo. Similar a **:nth-child(1)**)

¿ A qué elementos afectaría ?



Pseudo-clases estructurales

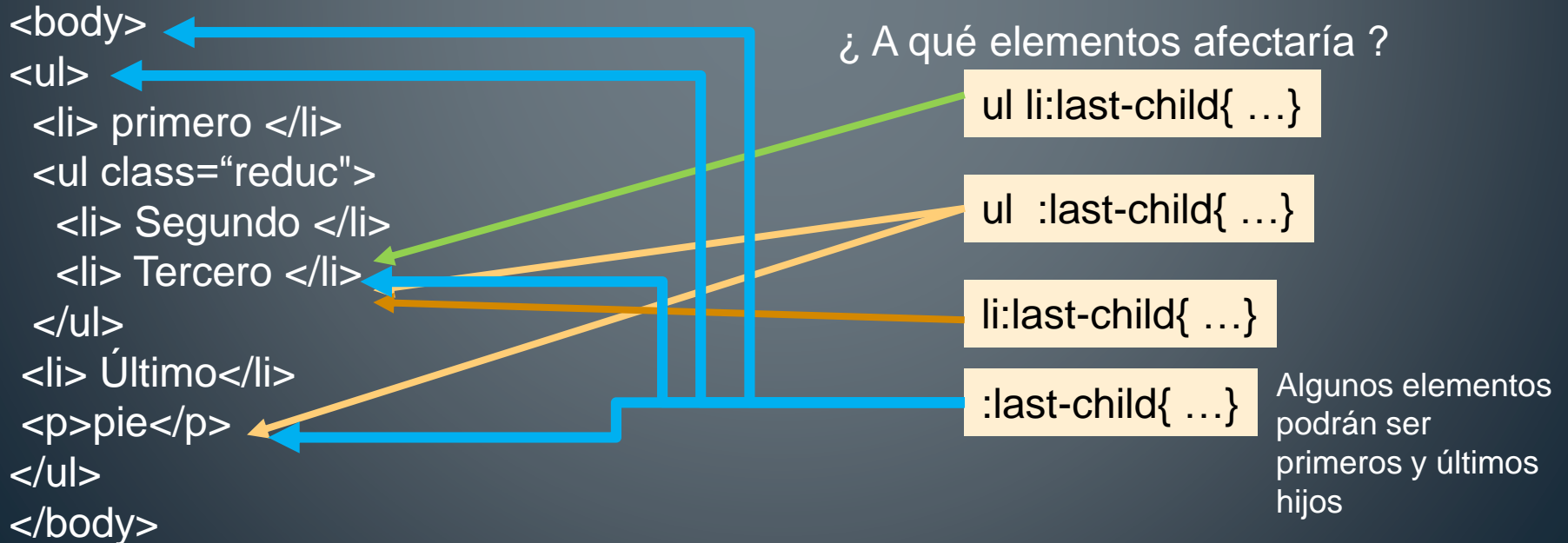
➔ **E:last-child** El elemento E **último** hijo de su padre.

`<html>` es **First-child** del DOM y a su vez **Last-child**

Ejemplo: **footer** **p:last-child**

Tiene que cumplir 2 cosas: Que p sea el último hijo de footer y que sea de tipo p.

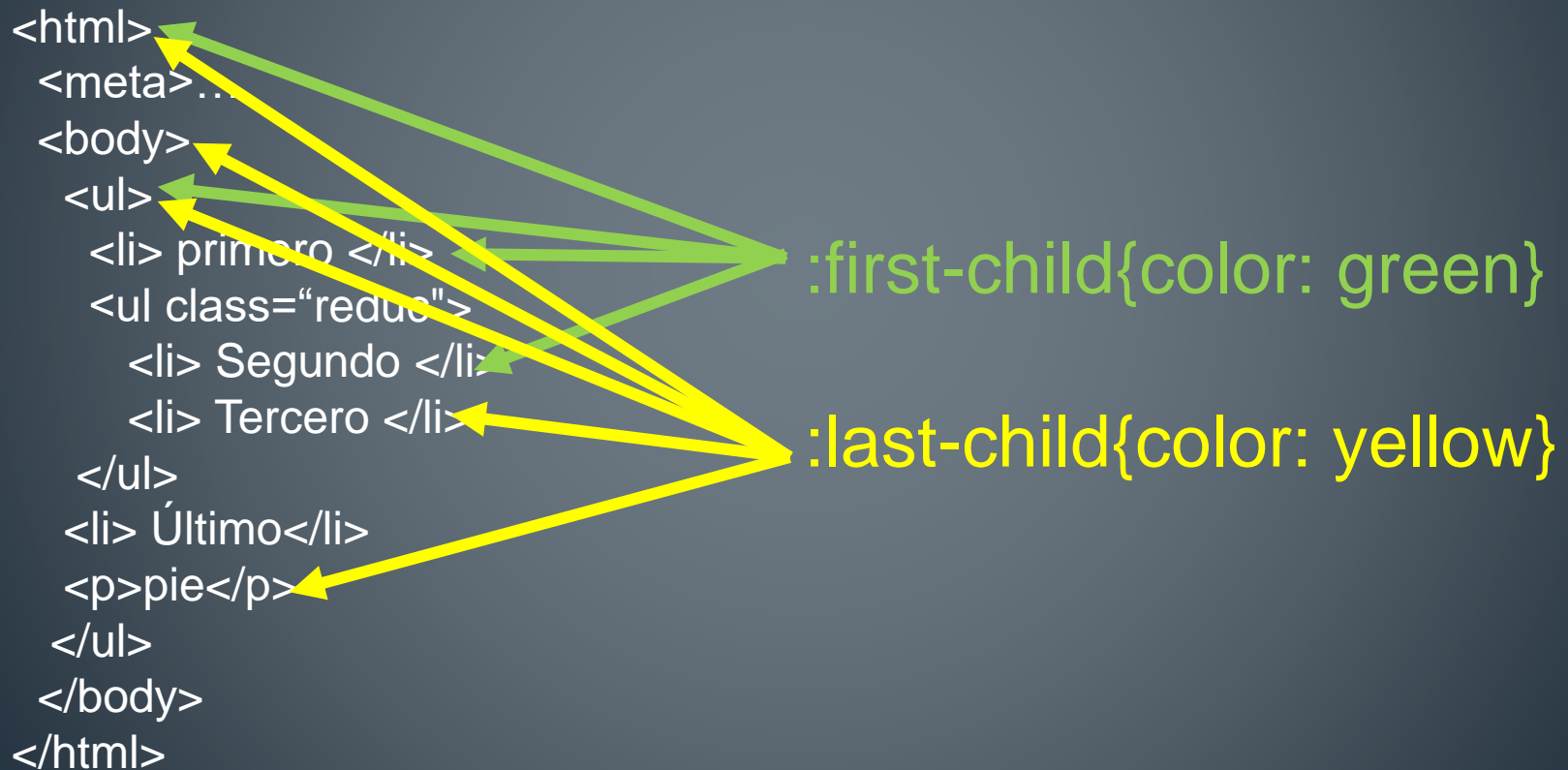
Similar a **:nth-last-child(1)**



Pseudo-clases estructurales

¿ Estas dos reglas, a qué elementos afectaría?

!!!! Recuerda: <html> es **First-child** y **last-child** del DOM
y <body> solo **last-child** de <html> !!!!



Pseudo-classes estructurales

¿ Tiene lógica este color del texto ?

```
<!DOCTYPE html>
<html>
<head>
<style> |
  :first-child {
    color:red;}
  :last-child {
    color:green ;
  }
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li><div>
    <ul> <li>hola</li>
    </ul></div>
</ul>
```

- Coffee
- Tea
- Coca Cola
 - hola

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_firstchild_more4

Pseudo-clases estructurales. Elementos Descendientes

E F:nth-child(n)

El elemento F que es el hijo enésimo de su padre E.

Ejemplo:

#contenido p:nth-child(2) : Segundo hijo p de #contenido

E F:nth-last-child(n)

Elemento F que es el hijo enésimo de su padre E, **contando hacia atrás**.

Ejemplo:

#contenido p:nth-last-child(2) : Penúltimo p hijo de #contenido

Pseudo-clases estructurales

Parámetro **n**:

Impares

Pares

Puede ser un entero, una palabra (**odd** ó **even**) o una expresión numérica ($3n+1$). (**n siempre comenzará en 0 por defecto**).

$3n$ = cada tercer elemento (frecuencia)

1 = Incremento

Ejemplos:

P F:nth-child($3n$)

Coincidirá en el elemento F tercero, sexto, noveno,... hijos de P

P F:nth-child($3n+1$)

Coincidirá en el elemento F primero, cuarto, séptimo,... hijos de P

:nth-child(...) Recuerda orden de los elementos:

`<html>` = 1

`<body>` = 2

Pseudo-classes estructurales

Recuerda orden de los elementos:

`<html>` = 1

`<head><style><meta>..` sin orden

`<body>` = 2

¿ A qué elementos afectaría ?

```
1 <html>
  <head>...
  <body>
    3 <ul>
      1 <p>Prueba</p>
      <li> primero </li>
      3 <ul class="reduc">
        1 <li> Segundo </li>
        <li> Tercero </li>
      </ul>
      <li> Último</li>
    5 <p>pie</p>
  </ul>
</body>
```

`:nth-child{ 2n+1 }`

	$2n+1$
$n=0$	$\Rightarrow 1$
$n=1$	$\Rightarrow 3$
$n=2$	$\Rightarrow 5$

Pseudo-classes estructurales

Selecciona el primer o último elemento de un **tipo** específico.

E:first-of-type

div p:first-of-type Selecciona primer párrafo de todos los div.

Similar a :nth-of-type(1)

E:last-of-type

Similar a :nth-last-of-type(1)

```
<body>
```

```
<ul>
```

```
<p>Prueba</p>
```

```
<li> primero </li>
```

```
<ul class="reduc">
```

```
<li> Segundo </li>
```

```
<li> Tercero </li>
```

```
</ul>
```

```
<li> Último</li>
```

```
<p>pie</p>
```

```
</ul>
```

```
</body>
```

¿ A qué elementos afectaría ?

ul li:first-of-type{ ... }

ul :first-of-type{ ... }

li:first-of-type{ ... }

Podría recibir herencia

Pseudo-clases estructurales

E:nth-of-type(n)

Selecciona el enésimo elemento de un tipo E específico.

Ejemplo:

form#contacto input:nth-of-type(**1**) :

Primer elemento input de un form con id contacto.

tr:nth-of-type(odd) : Filas impares de tabla.

tr:nth-of-type(even) : Filas pares de tabla.

} Antes con clases

div :nth-of-type(1) : Todos los primeros elementos de cada tipo dentro del div.

E:nth-last-of-type(n)

Igual que anterior pero contando hacia atrás.

Ejemplo:

ol li:nth-last-of-type(**2**) → **Penúltimo** elemento li de una lista OL

Pseudo-clases estructurales

E:only-child

Un elemento que es el único hijo de su padre. Que no tengan hermanos de ningún tipo.

E:only-of-type

Elementos de un tipo determinado que no tengan hermanos del tipo **E** especificado. Podrán tener otros tipos de hermanos.

p:empty

Elemento que no tiene hijos. Incluye nodos de texto.

Ejemplo: `<p> hola</p>` .(Este p **no** coincidirá).

Pseudo-clases estructurales

E:lang(en)

Elemento en el idioma indicado. Abreviatura de 2 letras.

E:not(excepción)

Elementos que no coincidan con el selector en el paréntesis.

Ejemplos: `p:not(.alerta)`

Todos los p del documento que no tengan la clase alerta.

`h2:not(header>h2):not(.logo)`

Todos los h2 menos lo que están en un header y los que tienen una clase de logo.

Pseudo-clases estructurales

Webs de ayuda:

<http://www.w3.org/TR/css3-selectors/>

<http://lea.verou.me/demos/nth.html>

<http://css-tricks.com/examples/nth-child-tester/>

<http://tools.css3.info/selectors-test>