



APRENDERAPROGRAMAR.COM

SELECTORES AVANZADOS
>, +, ~. PSEUDOCASES CSS
FIRST-CHILD, NTH-CHILD,
LAST-CHILD, NOT, FIRST-
LETTER, FIRST-LINE,
AFTER, BEFORE.
EJERCICIOS. (CU01015D)

Sección: Cursos

Categoría: Tutorial básico del programador web: CSS desde cero

Fecha revisión: 2029

Resumen: Entrega nº15 del Tutorial básico: "CSS desde cero".

Autor: César Krall

SELECTORES AVANZADOS CSS

Además de los selectores básicos (por etiqueta, por id o por class) y de los selectores basados en atributos HTML, disponemos de algunas formas más de establecer criterios de selección para la aplicación de estilos CSS: selectores especiales y pseudoclasses CSS.



Hay selectores especiales que usan los símbolos >, + y ~ para especificar la relación entre elementos.

Como selector de atributo tenemos distintas posibilidades. A continuación señalamos algunas:

Selector	Ejemplo CSS	Observaciones
elemPadre > elemHijo	<pre>div > ul { background-color: red;} div > p.destacado {background-color: pink;} div > img[src\$=".png"] {background-color: blue; }</pre>	<p>Selecciona todos los elementos de tipo elemHijo que están contenidos en un elemento elemPadre directamente (sin que existan otros elementos entre ambos). En el ejemplo, todos las listas tipo ul que estén dentro de un div tendrán fondo rojo, todos los párrafos cuya clase sea destacado y estén dentro de un div tendrán fondo rosa y todas las imágenes tipo png dentro de un div tendrán fondo azul .</p>
elemAnterior + elemSiguiente	<pre>p.destacado + p { color: green;}</pre>	<p>Selecciona todos los elementos de tipo elemSiguiente que estén exactamente después de un elemAnterior (sin otros elementos entre ellos) y que tengan al mismo elemento padre que éste. En el ejemplo, aquellos párrafos que estén justo después de otro párrafo cuya clase sea destacado tendrán el texto de color verde.</p>
elemAnterior ~ elemPosterior	<pre>p.destacado ~ p{ color: green;}</pre>	<p>Selecciona todos los elementos de tipo elemAcontinuacion que aparezcan después de un elemAnterior (pudiendo existir elementos intermedios entre ellos) y que tengan al mismo elemento padre que éste. En el ejemplo, aquellos párrafos que estén después de un párrafo cuya clase sea destacado y que tengan el mismo contenedor tendrán color verde.</p>

Es importante tener en cuenta que las reglas que usan > solo se aplican cuando existe una relación directa padre – hijo ó contenedor – contenido sin que existan cajas intermedias entre los elementos indicados. Por ejemplo div ul se aplica a una lista incluso si el elemento ul está dentro de un divisor span intermedio entre la lista y el div. En cambio div > ul no se aplica a una lista si está dentro de un span. Las reglas + sólo se aplican entre “hermanos” (elementos con un mismo padre, por ejemplo dos párrafos dentro de un div serían hermanos cuyo padre es el div) al hermano justo después del indicado. Por último las reglas ~ se aplican entre hermanos a todos los hermanos que se encuentren con posterioridad al indicado incluso aunque estén separados por elementos intermedios.

PSEUDOCLASES CSS

Existen algunos elementos especiales que se pueden seleccionar con lo que en CSS se denominan pseudoclasas, que son identificadores especiales que van precedidos por dos puntos. Entre las pseudoclasas disponibles tenemos las siguientes:

Pseudoclase	Ejemplo CSS	Observaciones
tipoElem:first-child	li:first-child {font-size: 200%;}	Selecciona el primer elemento dentro de una serie de elementos del tipo tipoElem. En el ejemplo, allí donde haya una serie de elementos li, el primero de ellos tendrá el doble de tamaño de fuente normal.
tipoElem:last-child	li:last-child {font-size: 200%;}	Selecciona el último elemento dentro de una serie de elementos del tipo tipoElem. En el ejemplo, allí donde haya una serie de elementos li, el último de ellos tendrá el doble de tamaño de fuente normal.
tipoElem:only-child	div.destacado img:only-child{border-style:solid; border-width:10px;}	Selecciona un elemento si es hijo único, es decir, si es el único elemento dentro del contenedor tipoElem. En el ejemplo se seleccionan las imágenes que estén dentro de contenedores div cuya clase sea destacado y dentro de los cuales exista únicamente una imagen y le aplica un borde de anchura 10 píxeles.
tipoElem:nth-child(number)	li:nth-child(3) {font-size: 200%;} p.destacado:nth-child(2) {font-size: 50%;}	Selecciona el elemento número number dentro de una serie de elementos de tipo tipoElem. No puede haber elementos intermedios de otro tipo en la serie. En el ejemplo, el tercer elemento de una serie de li tendría el doble del tamaño normal de fuente y el segundo elemento dentro de una serie de párrafos con clase destacado tendría el 50 % del tamaño normal.
Elem1:not(tipoElem2)	p:not(.destacado) {color: yellow;}	Selecciona todos los elementos de tipo Elem1 que no sean de tipo tipoElem2. En el ejemplo, se seleccionan todos los párrafos cuya clase no sea destacado.
tipoElem::first-letter	p::first-letter {font-size: 300%;}	Selecciona la primera letra de tipoElem
tipoElem::first-line	p::first-line {color:red}	Selecciona la primera línea de tipoElem
tipoElem::after	h1::after {content:"****"; background-color: yellow;}	Posiciona al final del elemento tipoElem permitiendo añadir contenido usando content:"contenidoDeseado" con las propiedades que se indiquen.
tipoElem::before	h1::before {content:"---"; color: blue;}	Posiciona al principio del elemento tipoElem permitiendo añadir contenido usando content:"contenidoDeseado" con las propiedades que se indiquen.

Escribe código CSS utilizando estos selectores y pruébalos sobre el documento HTML base que estamos usando para el curso.

No deben existir espacios al escribir estas declaraciones. Por ejemplo `li:nth-child(3) {font-size: 200%;}` es válido pero `li : nth-child (3) {font-size: 200%;}` no es válido por existir espacios.

Si el número referenciado con `nth-child` no existe, la regla no tendrá efectos. En lugar de `first-child` se puede usar `nth-child(1)` alternativamente.

Existen más pseudoclases CSS que iremos estudiando a lo largo del curso.

COSAS QUE PUEDEN OCURRIR...

Quizás con el código que hemos ido probando en lo que llevamos de curso hayas encontrado algún selector o fragmento de código que no te ha funcionado como cabía esperar. Quizás hayas pensado que has hecho algo mal, o que hay algún error en los contenidos del curso. Muchas veces no será ni una cosa ni otra. El motivo de que muchas veces no se obtenga el resultado esperado es la distinta respuesta de los distintos navegadores (Internet Explorer, Mozilla Firefox, Chrome, Opera, Safari, etc.) ante determinado código. A lo largo del curso iremos estudiando cómo resolver, en la medida de lo posible, estos problemas. Pero como ya hemos indicado en alguna ocasión, en el mundo de los desarrollos web los cambios y novedades son constantes y no queda otra que tener paciencia, tratar de adquirir experiencia y mantenerse actualizado a través de cursos, internet, libros, revistas, etc.

Pondremos un ejemplo relacionado con lo anterior. Hay una pseudoclase que es `::selection` cuya inclusión en el estándar CSS se ha discutido y es permitida por algunos navegadores pero por otros no. La sintaxis sería del tipo `p::selection { color: gold; background-color: red; }` y daría lugar a que cuando el usuario hace una selección dentro de un elemento `p`, aquello seleccionado aparezca de color oro y con fondo rojo. Al no haberse estandarizado, es posible que no se reconozca la sintaxis por el navegador y que no se pueda ver este efecto, o bien que los distintos navegadores tengan distintas sintaxis para aplicar estos efectos y haya que escribir una línea específica de distinta manera para cada navegador.

Al igual que con esta pseudoclase puede ocurrir algo similar con otros elementos CSS. Sería ideal que todos los navegadores funcionaran igual, pero en la práctica nos encontraremos pequeñas o grandes divergencias entre ellos.

Otra cosa que puede ocurrir con cierta frecuencia es que estemos analizando el código CSS de una web o aplicación web y encontremos código que no hayamos estudiado. En ese caso tenemos que acudir a nuestro conocimiento general sobre CSS para interpretar que por ejemplo `div:empty` seleccionará los elementos `div` que no tengan hijos o contenedores internos, mientras que `input:checked {background-color: green;}` aplicará color de fondo verde a aquellos elementos `input` con el atributo `checked="checked"`. Sin embargo, buena parte de este código puede que no sea soportado por muchos navegadores.

Algunas personas dicen: ¡Esto de los desarrollos web es caótico! Quizás no sea para tanto, en realidad hay un grado de estandarización bastante amplio, pero sí es verdad que en algunos momentos puede parecer un tanto caótico.

EJERCICIO

Dado este fragmento de código HTML que estamos usando como base para el curso:

```
<ul>
<li><a href="#">Inicio</a></li>
<li> <a href="libros.html">Libros de programación</a> </li>
<li> <a href="cursos.html">Cursos de programación</a> </li>
<li> <a href="humor.html">Humor informático</a> </li>
</ul>
<!-- fin menu -->
```

Definir que esta lista es de clase (class) especial. La clase especial no tendrá estilos especificados. En cambio, deberá definirse que los elementos li dentro de la lista ul de clase especial tengan color de fondo gris para los elementos impares (primero, tercero, quinto, séptimo...) y color de fondo rosa para los elementos pares (segundo, cuarto, sexto, octavo...).

Indicar los cambios en el documento HTML y el código a incluir en el archivo css.

SOLUCIÓN

En el archivo HTML escribiremos: `<ul class="especial">`

El contenido del archivo css puede ser el siguiente:

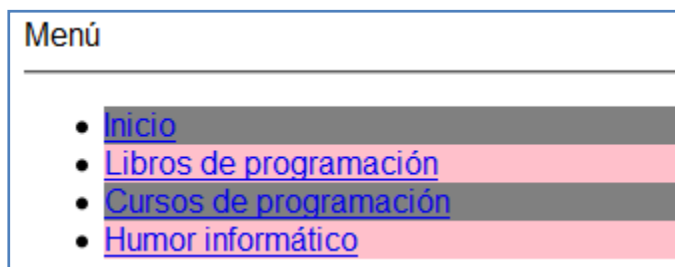
```
/* Curso CSS estilos aprenderaprogramar.com */
ul.especial li:nth-child(1) {background-color: grey;}
ul.especial li:nth-child(2) {background-color: pink;}
ul.especial li:nth-child(3) {background-color: grey;}
ul.especial li:nth-child(4) {background-color: pink;}
```

Dado que solo tenemos 4 elementos en la lista solo definimos los cuatro primeros hijos.

Pero como queremos indicar que se apliquen los estilos a todos los elementos impares y pares sería más correcto usar esta definición:

```
/* Curso CSS estilos aprenderaprogramar.com */
ul.especial li:nth-child(odd) {background-color: grey;}
ul.especial li:nth-child(even) {background-color: pink;}
```

El resultado será similar a este:



Próxima entrega: CU01016D

Acceso al curso completo en aprenderaprogramar.com --> Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=75&Itemid=203