

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Selectores de Filtrado II © ADRINFOR S.L.

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

www.adrformacion.com © ADRINFOR S.L.
Héctor García González

Indice

Selectores de Filtrado II	3
Selectores de Filtrado Jerárquico	3
Selector :first-child	3
Selector :last-child	4
Selector :first-of-type	5
Selector :last-of-type	7
Selector :nth-child()	8
Selector :nth-last-child()	8
Selector :only-child	8
Selector :only-of-type	10
Selectores de Filtrado por Contenido	11
Selector :contains()	11
Selector :empty	11
Selector :has(selector)	13
Selector :parent	14
Selectores de Filtrado por Visibilidad	14
Selector :hidden	14
Selector :visible	16
Selectores de Filtrado por Elementos de Formulario	18
Selectores por tipo de elemento de formulario	18
Selector :checked	20
Selector :disabled	20
Selector :enabled	21
Selector :selected	21
Ejercicios	22
Ejercicio 1: Selectores de Filtrado Jerárquico	22
Lo necesario para comenzar	22
Ejercicio 2: Selectores de Filtrado por Contenido y Filtrado por Visibilidad	22
Lo necesario para comenzar	23
Ejercicio 3: Selectores de Filtrado por Elementos de Formulario	23
Lo necesario para comenzar	23

Selectores de Filtrado II

Selectores de Filtrado Jerárquico

Los selectores de filtrado jerárquico permitirán aplicar filtros que modificarán el conjunto de elementos obtenido por el selector previo a partir de diversas condiciones basadas en la estructura jerárquica del DOM.

Selector :first-child

El selector de filtrado jerárquico :first-child filtrará de entre los elementos obtenidos por el selector previo aquellos que son primer elemento hijo, es decir, que dentro del elemento padre que los contiene.



HTML

Vamos a ver el resultado de aplicar al mismo documento HTML este selector:

```
<div id="header">      <div id="site_title">      <h1><a href="#"><strong>GLOSSY</strong> BO
X<span>YOUR TAGLINE GOES HERE</span></a></h1>      </div>      <ul id="lang_links">
  <li><a href="#" lang="es" dir="ltr">ES</a></li>      <li><a href="#" lang="en" dir="ltr">EN</a></li>
    <li><a href="#" lang="fr" dir="rtl">FR</a></li>      </ul>      <ul id="social_box">
  <li><a href="#" title="Publicar en Facebook"></a></li>
    <li><a href="#" title="Publicar en Twitter"></a></li>
  </li>      <li><a href="#" title="Publicar en LinkedIn"></a></li>
  <li>      <li><a href="#" title="En Technorati"></a></li>
    </ul>      <div class="cleaner"></div>      </div>      <div id="menu">      <ul>      <li><a
href="#" class="current" accesskey="h" title="Home">Home</a></li>      <li><a href="#services.html" a
ccesskey="s" title="Services">Services</a></li>      <li><a href="#partners.html" accesskey="p" title="P
artners">Partners</a></li>      <li><a href="#about.html" accesskey="a" title="About us">About us</a></l
i>      <li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>      </ul>      </div>
>
```

Para este código, vamos a disponer del siguiente estilo CSS

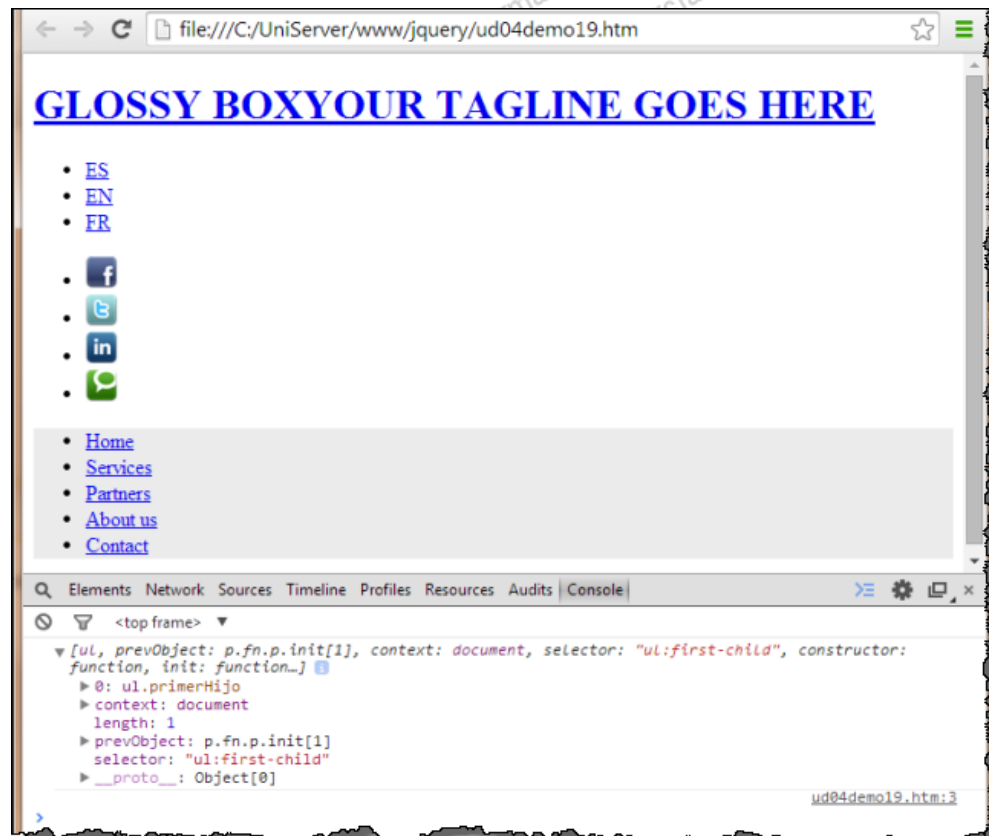
```
.primerHijo{ background-color: #ebebeb; }
```

Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){ //mostrar por consola los elementos que componen la seleccion console.log( $("ul:first-child") ); $("ul:first-child").addClass('primerHijo'); }); </script>
```

Resultado

En este ejemplo, el selector localizará los elementos de tipo ul que sean el primer hijo de sus correspondientes padres. A todos ellos, se les asignará el estilo CSS .primerHijo



Como se observa en la imagen, solamente el tercer ul de nuestra página cumple con esta condición dado que es el primer elemento hijo del div#menu

Selector :last-child

Con el selector :last-child conseguiremos filtrar la selección reduciéndola a únicamente aquellos elementos que son el último hijo de su elemento padre.

\$("selector:last-child")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que son el
último hijo de su elemento padre



Selectores :first-child y :last-child

Selector :first-of-type

El selector de filtrado jerárquico :first-of-type filtrará de entre los elementos obtenidos por el selector previo aquellos que sean el primer elemento hijo de su tipo. No tiene porqué ser el primer elemento hijo, pero sí el primero que sea del tipo indicado.

\$("selector:first-of-type")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que son el
primer hijo del tipo indicado

HTML

Vamos a ver el resultado de aplicar al mismo documento HTML este selector:

```
<div id="header">
X<span>YOUR TAGLINE GOES HERE</span></a></h1>
<li><a href="#" lang="es" dir="ltr">ES</a></li>
<li><a href="#" lang="fr" dir="rtl">FR</a></li>
<li><a href="#" title="Publicar en Facebook"></a></li>
<li><a href="#" title="Publicar en Twitter"></a></li>
<li><a href="#" title="Publicar en LinkedIn"></a></li>
<li><a href="#" title="En Technorati"></a></li>
</ul>
<div class="cleaner"></div>
<div id="menu">
<ul>
<li><a href="#" class="current" accesskey="h" title="Home">Home</a></li>
<li><a href="#services.html" accesskey="s" title="Services">Services</a></li>
<li><a href="#partners.html" accesskey="p" title="Partners">Partners</a></li>
<li><a href="#about.html" accesskey="a" title="About us">About us</a></li>
<li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>
</ul>
</div>
```

Esta página dispondrá del siguiente estilo CSS:

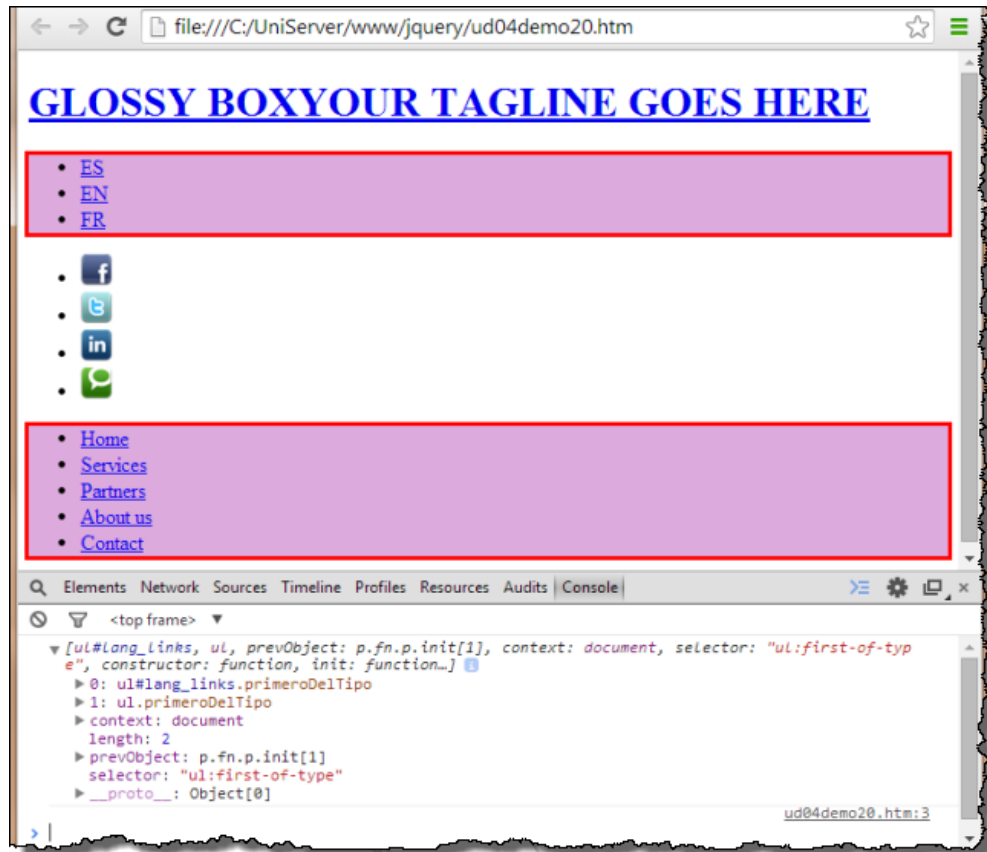
```
.primeroDelTipo{ background-color: #ddaadd; outline: #FF0000 solid; }
```

Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){ //mostrar por consola los elementos que componen la seleccion console.log( $("ul:first-of-type") ); $("ul:first-of-type").addClass('primeroDelTipo'); }); </script>
```

Resultado

Este selector localizará los primeros elementos hijos que sean del tipo indicado. Así, en nuestro código, la selección contendrá a los elementos `ul#lang_links` y `ul#menu` puesto que, además de ser de tipo `ul` (tal cómo indicaba el selector previo) son primeros elementos hijo de su tipo en sus respectivos padres.



Selector `:last-of-type`

Con el filtro de selector `:last-of-type` filtraremos la selección para incluir únicamente aquellos elementos que sean el último hijo de su tipo.





Selectores :first-of-type y :last-of-type

Selector :nth-child()

El selector :nth-child nos permitirá filtrar la selección para solamente obtener los elementos cuya posición dentro de su elemento padre coincida con el índice especificado en el filtro.

\$("selector:nth-child(index)")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos hijo que se encuentran
en el índice indicado

Selector :nth-last-child()

El selector de filtrado jerárquico :nth-last-child filtrará de entre los elementos obtenidos por el selector previo aquellos que, en orden inverso, se encuentren en la posición indicada.

\$("selector:nth-last-child(index)")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos hijo que, en orden
inverso, se encuentran en el índice indicado

Selector :only-child

Seleccionará los elementos que son el único hijo dentro de sus elementos padre

\$("selector:only-child")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que sean el
único hijo de su elemento padre

HTML

Vamos a ver cómo el selector de filtrado `:only-child` limitará la selección a los elementos que son el único hijo de su predecesor:

```
<div id="header">
  <div id="site_title">
    <h1><a href="#"><strong>GLOSSY</strong> BO
X<span>YOUR TAGLINE GOES HERE</span></a></h1>
    </div>
    <ul id="lang_links">
      <li><a href="#" lang="es" dir="ltr">ES</a></li>
      <li><a href="#" lang="en" dir="ltr">EN</a></li>
      <li><a href="#" lang="fr" dir="rtl">FR</a></li>
    </ul>
    <ul id="social_box">
      <li><a href="#" title="Publicar en Facebook"></a></li>
      <li><a href="#" title="Publicar en Twitter"></a></li>
      <li><a href="#" title="Publicar en LinkedIn"></a></li>
      <li><a href="#" title="En Technorati"></a></li>
    </ul>
    <div class="cleaner"></div>
    <div id="menu">
      <ul>
        <li><a href="#" class="current" accesskey="h" title="Home">Home</a></li>
        <li><a href="#services.html" accesskey="s" title="Services">Services</a></li>
        <li><a href="#partners.html" accesskey="p" title="Partners">Partners</a></li>
        <li><a href="#about.html" accesskey="a" title="About us">About us</a></li>
        <li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>
      </ul>
    </div>
```

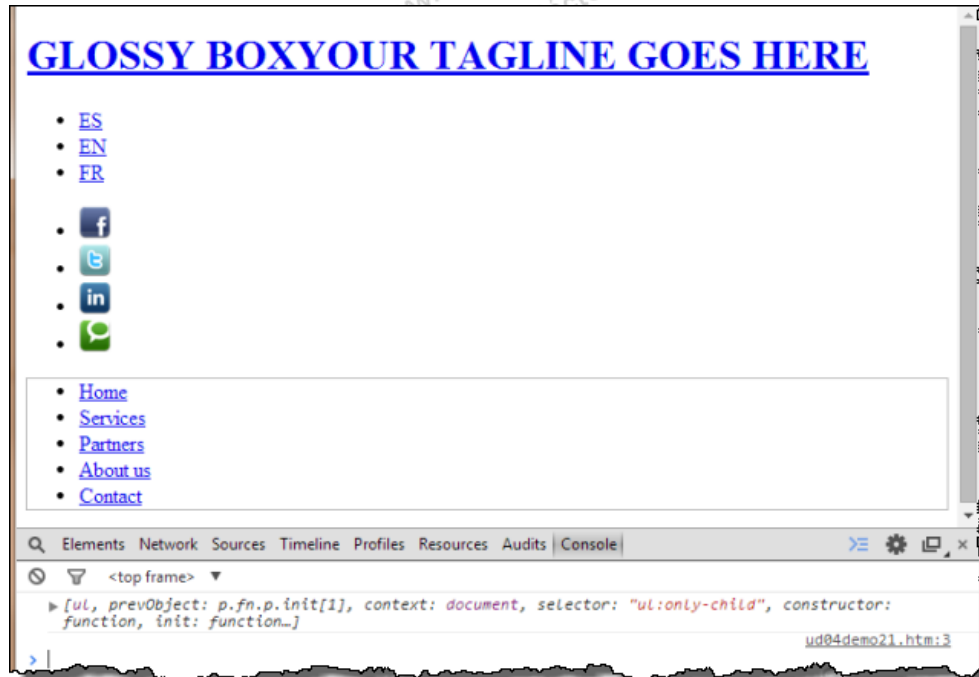
Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){ //mostrar por consola los elementos que componen la seleccion
  console.log( $("ul:only-child") );
  $("ul:only-child").css("outline", "#bababa 1px solid");
});
</script>
```

Resultado

La consola del navegador mostrará la selección, que únicamente contiene un elemento, el ul hijo del div#menu

Posteriormente, le aplicaremos la propiedad CSS outline.



Selector :only-of-type

Seleccionará los elementos que son el único hijo del tipo indicado dentro de su elemento padre.





Selectores :only-child y :only-of-type

Selectores de Filtrado por Contenido

Con los selectores de filtrado por contenido podremos establecer, dentro de nuestros selectores, condiciones relacionadas con el contenido de los elementos a filtrar.

Selector :contains()

Con el selector :contains() podremos obtener los elementos que contienen el texto indicado en el selector.



Para que un elemento cumpla con la condición indicada por el selector, debe cumplir alguna de las siguientes condiciones:

- El texto se encuentra directamente en el elemento seleccionado
- El texto se encuentra en cualquiera de sus elementos descendientes
- Una combinación de las dos condiciones indicadas anteriormente

El texto pasado como argumento al selector debe encontrarse entre comillas.

La comparación del texto es case-sensitive: deben coincidir mayúsculas y minúsculas.

Selector :empty

El filtro :empty (en inglés, "vacío") filtrará de entre los elementos localizados por el selector previo, aquellos que se encuentran vacíos. Es decir, no incluyen ningún elemento hijo ni un nodo de texto.



HTML

Vamos a obtener todos los elementos que se encuentren vacíos en este fragmento de HTML:

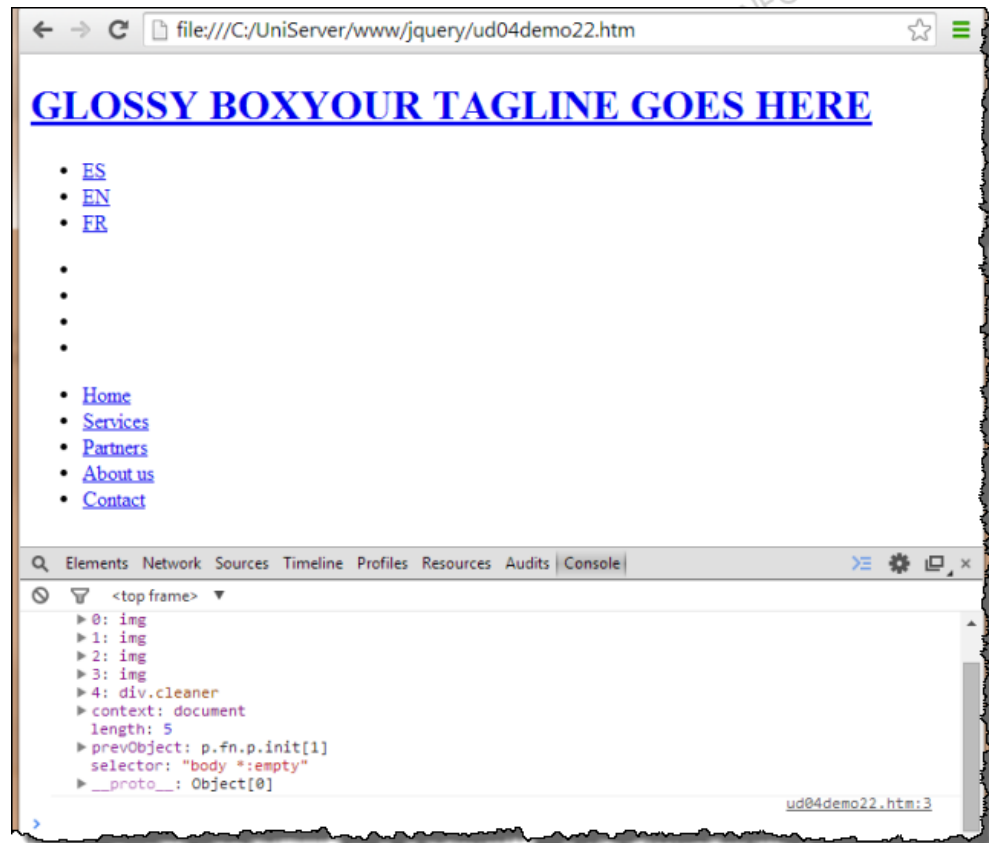
```
<div id="header">
  <div id="site_title">
    <h1><a href="#"><strong>GLOSSY</strong> BO
X<span>YOUR TAGLINE GOES HERE</span></a></h1>
    <li><a href="#" lang="es" dir="ltr">ES</a></li>
    <li><a href="#" lang="fr" dir="rtl">FR</a></li>
    <li><a href="#" title="Publicar en Facebook"></a></li>
    <li><a href="#" title="Publicar en Twitter"></a></li>
    <li><a href="#" title="Publicar en LinkedIn"></a></li>
    <li><a href="#" title="En Technorati"></a></li>
  </ul>
  <div class="cleaner"></div>
  <div id="menu">
    <ul>
      <li><a href="#" class="current" accesskey="h" title="Home">Home</a></li>
      <li><a href="#services.html" accesskey="s" title="Services">Services</a></li>
      <li><a href="#partners.html" accesskey="p" title="Partners">Partners</a></li>
      <li><a href="#about.html" accesskey="a" title="About us">About us</a></li>
      <li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>
    </ul>
  </div>
</div>
```

Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){ //mostrar por consola los elementos que componen la seleccion
  console.log( $("body *:empty") );
  $("body *:empty").remove();
}); </script>
```

Resultado

Al utilizar el método de filtrado `:empty` limitamos la selección de todos los elementos descendientes del elemento `body` de la página y, con la posterior llamada al método `.remove()`, los eliminamos del DOM.



Selectores `:contains` y `:empty`

Selector `:has(selector)`

El filtro `:has()` reducirá el conjunto de elementos obtenidos por el selector previo, incluyendo como resultado final únicamente aquellos que contengan al menos un elemento que cumpla con las especificaciones del selector indicado como parámetro.

\$("selector:has(selector)")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que contienen
algún elemento que cumpla con el
selector indicado entre paréntesis

Selector :parent

Con el filtro :parent podremos reducir el conjunto de elementos devuelto por el selector a aquellos que, bien contienen un elemento hijo o contienen un texto.

\$("selector:parent")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que contienen
un elemento hijo o un fragmento de texto

Selectores de Filtrado por Visibilidad

Los selectores de filtrado por visibilidad nos ayudarán a filtrar los resultados de una selección dependiendo de si éstos se encuentran visibles o no dentro de la página.

Dentro de cada uno de ellos indicaremos las condiciones que evalúa jQuery para considerar los elementos del DOM visibles o invisibles.

Selector :hidden

El selector :hidden filtrará de la selección los elementos del documento que no sean visibles.

\$("selector:hidden")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que
no sean visibles

jQuery considera que un elemento no es visible cuando éste no ocupa espacio en el documento. Para ello, debe cumplir con alguna de estas condiciones:

- Que la anchura y altura del elemento se encuentren establecidos a 0.
- Que dispongan del atributo de CSS "display" con el valor "none" (display: none)

- Que sea un elemento de un formulario de tipo hidden (type="hidden")
- Que alguno de sus elementos padres no sea visible.

HTML

Vamos a ver cómo funciona el selector :hidden de jQuery. Una vez más, vamos a utilizar nuestro habitual fragmento de HTML sobre el que aplicaremos el selector:

```
<div id="header">
  <div id="site_title">
    <h1><a href="#"><strong>GLOSSY</strong> BO
X<span>YOUR TAGLINE GOES HERE</span></a></h1>
    <div id="lang_links">
      <li><a href="#" lang="es" dir="ltr">ES</a></li>
      <li><a href="#" lang="en" dir="ltr">EN</a></li>
      <li><a href="#" lang="fr" dir="rtl">FR</a></li>
    </ul>
    <div id="social_box">
      <li><a href="#" title="Publicar en Facebook"></a></li>
      <li><a href="#" title="Publicar en Twitter"></a></li>
      <li><a href="#" title="Publicar en LinkedIn"></a></li>
      <li><a href="#" title="En Technorati"></a></li>
    </ul>
    <div class="cleaner"></div>
    <div id="menu">
      <ul>
        <li><a href="#" class="current" accesskey="h" title="Home">Home</a></li>
        <li><a href="#services.html" accesskey="s" title="Services">Services</a></li>
        <li><a href="#partners.html" accesskey="p" title="Partners">Partners</a></li>
        <li><a href="#about.html" accesskey="a" title="About us">About us</a></li>
        <li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>
      </ul>
    </div>
  </div>
</div>
```

En esta ocasión, vamos a aplicar los siguientes estilos CSS a la página:

```
.cleaner{ height: 0; clear:both; } #menu{ display: none; }
```

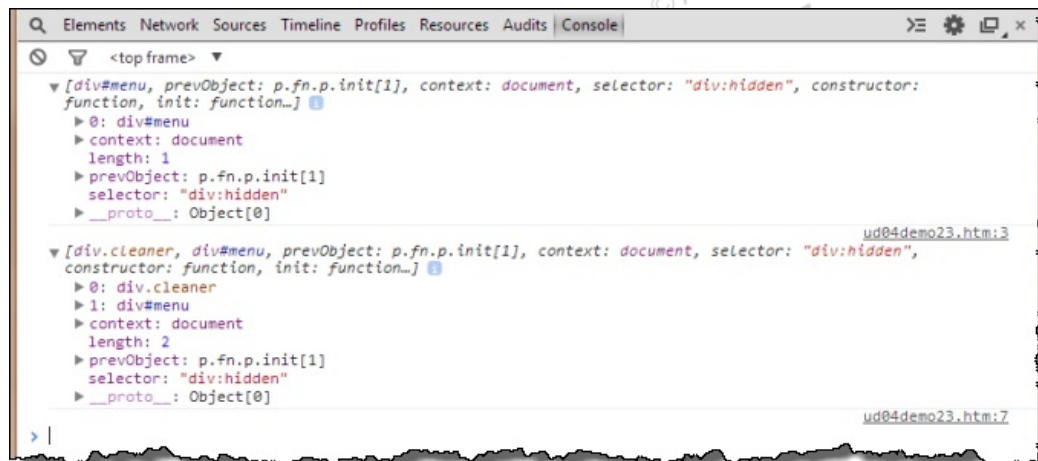
Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){ console.log( $("div:hidden") ); $("div:hidden").css("width","0"); console.log( $("div:hidden") ); }); </script>
```

Resultado

En la primera llamada en la que utilizamos el selector de filtrado `:hidden` solamente el `div#menu` cumple con las características necesarias para cumplir con el filtro.

Posteriormente, aplicamos el estilo CSS `width` a los elementos con la clase `"cleaner"`, consiguiendo así que sus propiedades `"height"` y `"width"` se establezcan a cero. Así, la segunda vez que utilizamos el selector de filtrado `:hidden` el objeto resultante contiene, además del `div#menu`, el `div.cleaner`



Selector `:visible`

Al contrario que el selector `:hidden`, el selector `:visible` filtrará los elementos del documento que sí sean visibles.



Así, el funcionamiento de este selector es contrapuesto al del selector `:hidden`, considerando elementos visibles aquellos que ocupan espacio en el documento y que dispongan de altura y anchura superior a cero.

HTML

Veamos el mismo ejemplo utilizado con el selector `:hidden`.

```
<div id="header">          <div id="site_title">          <h1><a href="#"><strong>GLOSSY</strong> BO
X<span>YOUR TAGLINE GOES HERE</span></a></h1>          </div>          <ul id="lang_links">
  <li><a href="#" lang="es" dir="ltr">ES</a></li>          <li><a href="#" lang="en" dir="ltr">EN</a></li>
    <li><a href="#" lang="fr" dir="rtl">FR</a></li>          </ul>          <ul id="social_box">
  <li><a href="#" title="Publicar en Facebook"></a></li>
  <li><a href="#" title="Publicar en Twitter"></a></li>
  <li><a href="#" title="Publicar en LinkedIn"></a></li>
  <li><a href="#" title="En Technorati"></a></li>
</ul>          <div class="cleaner"></div>          </div>          <div id="menu">          <ul>          <li><a
href="#" class="current" accesskey="h" title="Home">Home</a></li>          <li><a href="#services.html" a
ccesskey="s" title="Services">Services</a></li>          <li><a href="#partners.html" accesskey="p" title="P
artners">Partners</a></li>          <li><a href="#about.html" accesskey="a" title="About us">About us</a></li>
  <li><a href="#contact.html" accesskey="c" title="Contact">Contact</a></li>          </ul>          </div>
>
```

Una vez más, la página tiene los mismos estilos CSS que en el caso anterior:

```
.cleaner{ height: 0; clear:both; } #menu{ display: none; }
```

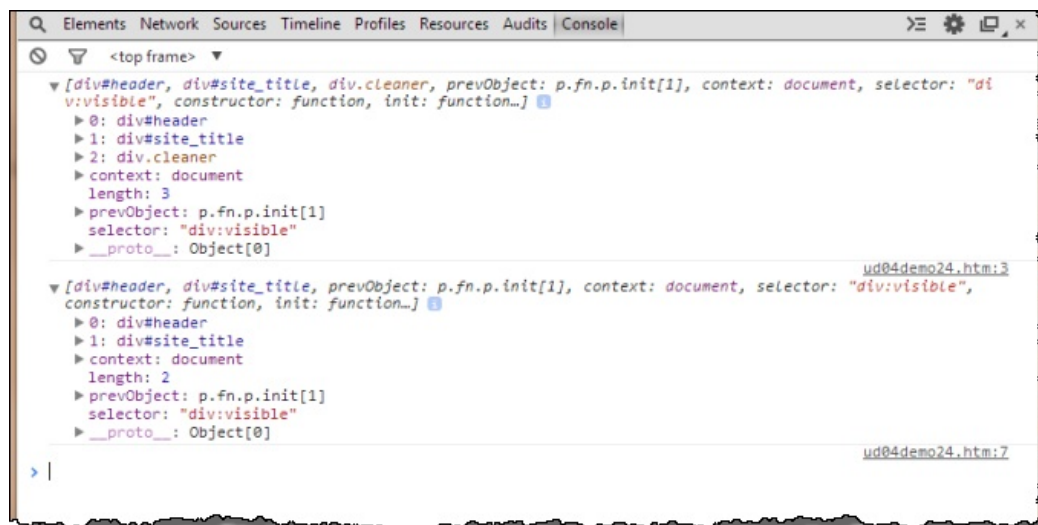
Código de Ejemplo

```
<script type="text/javascript"> $(document).ready(function(){
  console.log ($("div:visible" ));  $(".cleaner").css("width","0");  console.log ($("div:visible" )); }); </script>
```

Resultado

En esta ocasión, la selección inicial localiza tres elementos, los divs `#header`, `#site_title` y `.cleaner`

Una vez más, establecemos la propiedad CSS `"width"` a cero y, al volver a utilizar el selector `:visible` la nueva selección ya no incluirá el `div.cleaner`

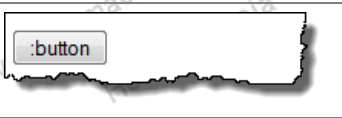
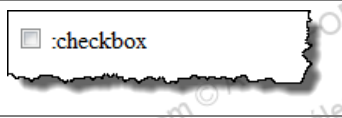
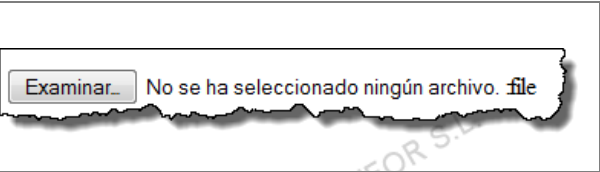
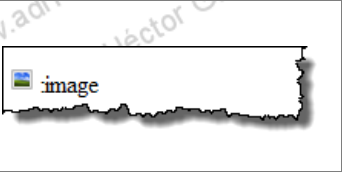




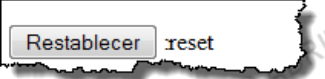


Selectores de Filtrado por Elementos de Formulario

Los selectores de filtrado por elementos de formulario nos permitirán poder establecer en nuestros selectores condiciones relacionadas con elementos de formularios web.

Selectores por tipo de elemento de formulario

Con los selectores por tipo de elemento de fomulario, filtraremos la selección sobre la que se aplique para obtener únicamente los elementos cuyo atributo "type" coincida con el filtro indicado.

Selector	Elementos que seleccionará
:button	<p>Filtrará los elementos que tengan asignado como tipo el valor "button":</p> <div> <div><input type="button" /></div> <div></div> </div>
:checkbox	<p>Filtrará los elementos de tipo "checkbox"</p> <div> <div><input type="checkbox" /></div> <div></div> </div>
:file	<p>Filtrará los elementos de selección de fichero, cuyo atributo "type" contendrá el valor "file"</p> <div> <div><input type="file" /></div> <div></div> </div>
:image	<p>Filtrará los elementos de tipo "image"</p> <div> <div><input type="image" /></div> <div></div> </div>

Selector	Elementos que seleccionará
:password	<p>Filtrará los elementos de tipo "password"</p> <div data-bbox="383 403 1147 573"> <div data-bbox="383 403 691 573"><input type="password" /></div> <div data-bbox="691 403 1147 573">  </div> </div>
:radio	<p>Filtrará los elementos cuyo atributo "type" contenga el valor "radio"</p> <div data-bbox="383 728 991 898"> <div data-bbox="383 728 639 898"><input type="radio" /></div> <div data-bbox="639 728 991 898">  </div> </div>
:reset	<p>Filtrará los elementos de tipo "reset"</p> <div data-bbox="383 1075 991 1245"> <div data-bbox="383 1075 639 1245"><input type="reset" /></div> <div data-bbox="639 1075 991 1245">  </div> </div>
:submit	<p>Filtrará los elementos "submit"</p> <div data-bbox="383 1417 1010 1588"> <div data-bbox="383 1417 659 1588"><input type="submit" /></div> <div data-bbox="659 1417 1010 1588">  </div> </div>
:text	<p>Filtrará los elementos que tengan asignado como tipo el valor "text"</p> <div data-bbox="383 1765 978 1935"> <div data-bbox="383 1765 627 1935"><input type="text" /></div> <div data-bbox="627 1765 978 1935">  </div> </div>



Los selectores por tipo de elemento de formulario equivaldrán al filtrado por atributo "type":

`$(":text")`

equivaldrá al selector

`$("[type='text']")`



Estos filtros no se limitan únicamente a elementos de formulario. Cualquier elemento cuyo atributo "type" contenga uno de los valores anteriormente indicados podría ser incluido bajo estos filtros. Para asegurar que los elementos seleccionados sean el elementos input deberemos aplicar previamente al filtro el selector por tipo:

`$("input:text")`

Selector :checked

El selector :checked seleccionará los elementos que se encuentren activados o seleccionados.

`$("selector:checked")`

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que se
encuentran activados o seleccionados

Este selector escanea los siguientes elementos:

- Elementos input de tipo "checkbox"
- Elementos input de tipo "radio"
- Listas de seleccion (select)

Selector :disabled

El selector :disabled seleccionará los elementos de la página que se encuentren deshabilitados.

\$("selector:disabled")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que se
encuentran deshabilitados



Los elementos deshabilitados tendrán establecido el atributo "disabled". Este atributo es válido en los elementos de tipo input, button, select, textarea, option y optgroup.

Selector :enabled

El selector :enabled seleccionará los elementos habilitados.

\$("selector:enabled")

Selector sobre el que
aplicar el filtro

Filtrado de los elementos que se
encuentran habilitados



Al igual que el selector :disabled, este selector funcionará contra elementos de tipo input, button, select, textarea, option y optgroup.

Selector :selected

Con el selector de filtrado :selected , podremos obtener los elementos option que se encuentren seleccionados en listas de selección (elementos de tipo select).

\$("selector:selected")


Selector sobre el que
aplicar el filtro

Filtrado de los elementos
seleccionados en listas de selección
(elementos select)

Ejercicios

Ejercicio 1: Selectores de Filtrado Jerárquico

Duración estimada del ejercicio



20
minutos

Utiliza el documento "ejercicio-selectores-filtrado-jerarquico.htm" y, por medio del interface de la página, construye los selectores utilizando selectores de filtrado jerárquico, para seleccionar:

1. Modifica el selector "ol.mensajesBox" para que la selección solamente incluya el primero de sus elementos hijos de tipo "li"
2. Selecciona los elementos de tipo "ol" que no tienen ningún elemento hermano
3. Modifica el selector "div.friendsList" para obtener como selección únicamente el primero de sus elementos hijos de tipo "ul"
4. Modifica el selector "div#userMensajes" para que la selección solamente incluya el último de sus elementos hijos de tipo "div"
5. Selecciona los elementos hijos del "div.friendsList" que sean únicos de su tipo dentro del div

Lo necesario para comenzar

Descarga el archivo adjunto y extráelo en tu directorio de trabajo.




ejercicio-selectores-filtrado-jerarquico.zip

Archivo comprimido con la plantilla para realizar el ejercicio

Ejercicio 2: Selectores de Filtrado por Contenido y Filtrado por Visibilidad

Duración estimada del ejercicio



15
minutos

Utiliza el documento "ejercicio-selectores-filtrado-conte-vis.htm" y, por medio del interface de la página, construye los selectores utilizando selectores de filtrado jerárquico, para seleccionar:

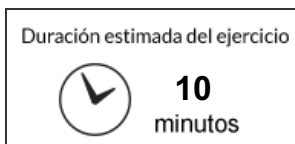
1. Selecciona los elementos del DOM que no contengan ningún elemento hijo ni ningún texto
2. Selecciona los elementos del DOM correspondientes a mensajes con el texto "resumen"
3. Selecciona los elementos del DOM que contienen un elemento de tipo "a" cuyo atributo "alt" tenga el prefijo "message"
4. Selecciona los elementos del DOM de tipo "li" que contienen un elemento de tipo "a" cuyo atributo "alt" tenga el prefijo "message"
5. Selecciona los elementos del DOM que no son visibles (solamente aparecerán por consola)

Lo necesario para comenzar

Descarga el archivo adjunto y extráelo en tu directorio de trabajo.



Ejercicio 3: Selectores de Filtrado por Elementos de Formulario



Utiliza el documento "ejercicio-selectores-filtrado-formulario.htm" y, por medio del interface de la página, construye los selectores utilizando selectores de filtrado jerárquico, para seleccionar:

1. Selecciona los elementos de formulario descendientes del "div" con identificador "orderDetails", de tipo "checkbox" y que estén marcados.
2. Selecciona todos los elementos deshabilitados
3. Selecciona el elemento indicativo del modo de pago seleccionado
4. Selecciona los elementos de formulario de tipo "text" del div con identificador "addrInfo"
5. Selecciona los elementos de formulario de tipo "button"

Lo necesario para comenzar

Descarga el archivo adjunto y extráelo en tu directorio de trabajo.



ejercicio-selectores-filtrado-formulario.zip

Archivo comprimido con la plantilla para realizar el ejercicio

www.adrformacion.com © ADRINFOR S.L.
Héctor García González