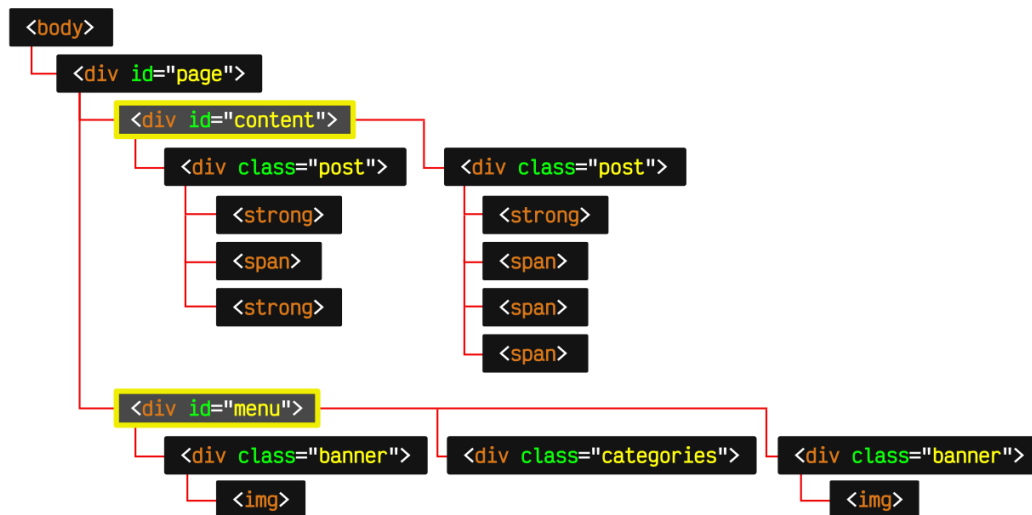


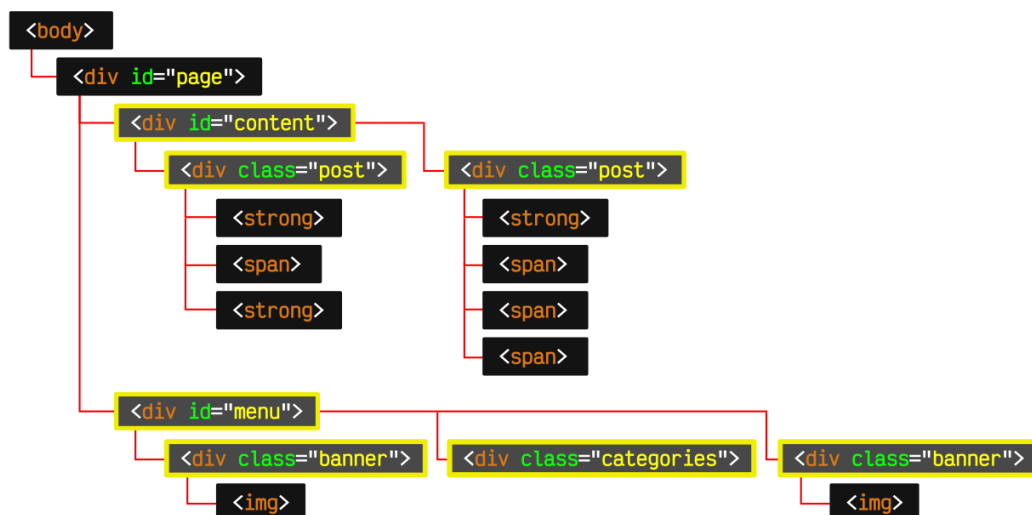
SELECTORES CSS. EJERCICIOS

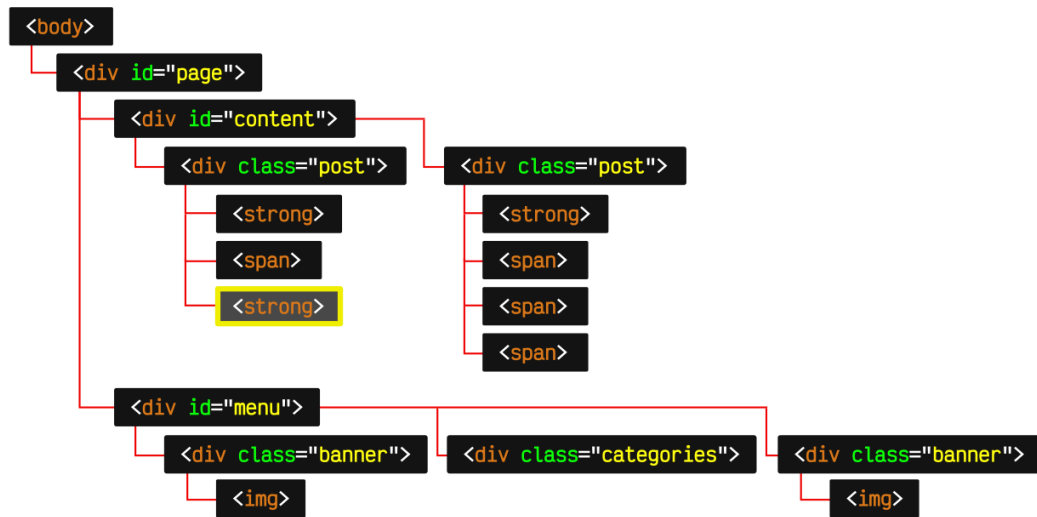
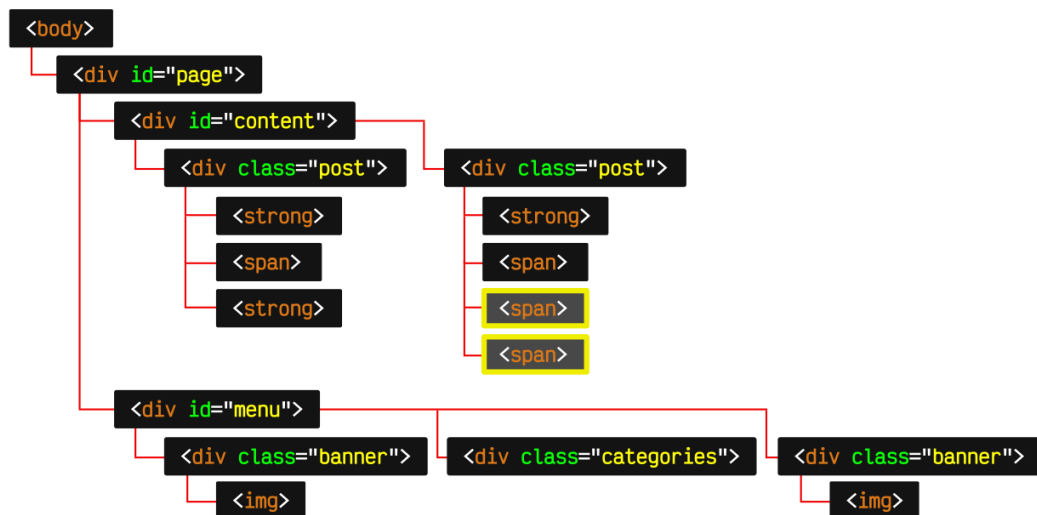
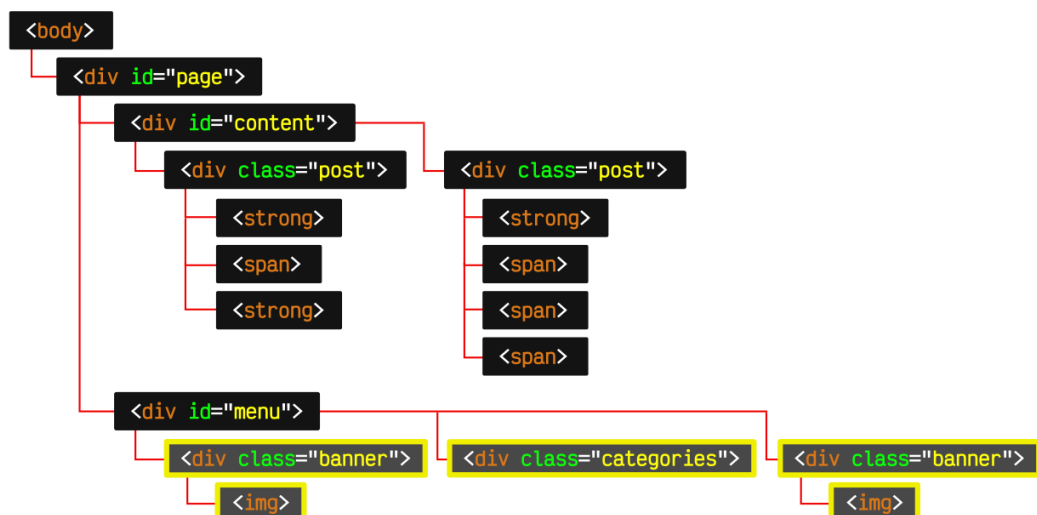
1. A partir de cada imagen, diseña el selector CSS que selecciona los elementos resaltados en amarillo.

1 Utiliza #page como referencia.



2 Utiliza #page como referencia.



3 Utiliza un selector de hermanos.**4** Utiliza un selector de hermanos.**5**

Soluciones:

#	Soluciones	Obs.
1	<code>#page > #content,</code> <code>#page > #menu</code>	<code>#page > div</code> es menos específico y, además, tendríamos problemas si se añaden más divs en el futuro.
2	<code>#page div</code>	Se seleccionan todos los descendientes divs de <code>#page</code>
3	<code>div.post > strong ~ strong</code> Sin usar selector de hermanos: <code>div.post:first-of-type > strong:last-of-type</code>	Seleccionamos el hermano siguiente a un <code>strong</code> . No valdría el operador '+' porque el hermano adyacente es un <code>span</code> .
4	<code>div.post > span + span</code> Sin usar selector de hermanos: <code>div.post > span:not(:first-of-type)</code>	Ahora sí usamos el operador Seleccionamos el hermano siguiente a un <code>strong</code> . No valdría el operador '+' porque el hermano adyacente es un <code>span</code> .
5	<code>#menu *</code>	

2. Rellena la siguiente tabla con la declaración CSS o la definición, según corresponda.

Solución (en negrita el enunciado):

#	Selector	Definición
1	<code>a:link</code> ó <code>a:not(:visited)</code>	Selecciona todos los elementos <code><a></code> que son enlaces no visitados.
2	<code>button:hover</code>	Selecciona todos los elementos <code>button</code> cuando el usuario pasa el cursor sobre ellos.
3	<code>[type='submit']:active</code>	Selecciona cualquier elemento <code><input></code> de tipo <code>submit</code> que esté en estado activo, es decir, cuando se encuentra presionado.
4	<code>#mi-id:focus</code>	Selecciona el elemento con identificador <code>mi-id</code> cuando tiene el foco, es decir, cuando ese elemento se encuentra seleccionado por el usuario (por ejemplo, al hacer clic en él o navegar hasta él mediante el teclado).
5	<code>.mi-clase:visited</code>	Selecciona todos los elementos con la clase 'mi-clase' que hayan sido visitados.
6	<code>input:enabled</code>	Selecciona todos los elementos <code>input</code> que están habilitados.
7	<code>label:only-child</code>	Selecciona todos los elementos <code>label</code> que son el único hijo de su padre.
8	<code>tr:nth-child(2)</code>	Selecciona el segundo elemento <code>tr</code> de cada grupo de hermanos. Seleccionaría la segunda fila de una tabla.
9	<code>h1:last-of-type</code>	Selecciona el último elemento <code>h1</code> de cada elemento padre. Equivalente a <code>h1:nth-last-of-type(1)</code>

10	li:nth-last-child(3)	Selecciona el tercer elemento li desde el final de cada elemento padre.
11	div.mi-clase:only-of-type	Selecciona todos los elementos div con la clase 'mi-clase' que son el único elemento de ese tipo entre sus hermanos.
12	input:checked	Selecciona todos los elementos input que están marcados (por ejemplo, en una lista de opciones).
13	[required]:empty	Selecciona todos los elementos con un atributo required que estén vacíos.
14	p:nth-of-type(3)	Selecciona todos los elementos <p> que son el tercer elemento de su tipo en su padre.
15	label:only-child	Selecciona todos los elementos label que son el único hijo de su padre.

3. Usa las pseudoclasas `:is()`, `:has()` y `:not()` para crear la declaración CSS correcta para cada uno de los siguientes enunciados:

3.1. Selecciona los elementos con clase *main* y los elementos con clase *header* que sean hijos directos de un elemento con id *main-section*.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div id="main-section">
  <div class="header">Soy un header</div>
  <div class="main">Soy un main</div>
  <div class="footer">
    Soy un footer
    <div class="main">Soy un main dentro de un footer</div>
  </div>
</div>
```

Soy un header
Soy un main
Soy un footer
Soy un main dentro de un footer

Solución:

```
#main-section > .main,  
#main-section > .header
```

Mejor así:

```
#main-section>:is(.main, .header)
```

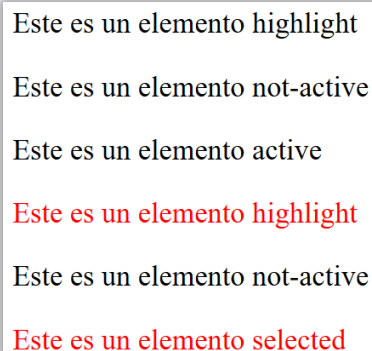
Fíjate con atención al símbolo ‘>’. Si no lo pones y dejas un espacio estarías seleccionando todos los descendientes, no solo los hijos directos, por lo que el div con el texto “Soy un main dentro de un footer” se pondría también en color rojo.

3.2. Selecciona los elementos con clase *highlight* y *selected* que sean hermanos de un elemento con clase *active* y aparezcan después en el código.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div class="highlight">
  <p>Este es un elemento highlight</p>
</div>
<div class="not-active">
  <p>Este es un elemento not-active</p>
</div>
<div class="active">
  <p>Este es un elemento active</p>
</div>
<div class="highlight">
  <p>Este es un elemento highlight</p>
</div>
<div class="not-active">
  <p>Este es un elemento not-active</p>
</div>
<div class="selected">
  <p>Este es un elemento selected</p>
</div>
```



Este es un elemento highlight

Este es un elemento not-active

Este es un elemento active

Este es un elemento highlight

Este es un elemento not-active

Este es un elemento selected

Solución:

Una opción:

```
.active~.highlight,
.active~.selected {
  color: red;
}
```

Pero mejor:

```
.active~:is(.highlight, .selected)
```

3.3. Selecciona los elementos con clase *card* hijos directos de divs con clase *container*, donde este elemento con clase *container* tenga al menos un descendiente con clase *active*.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    background-color: coral;
  }
  .container {display: flex;}
  .card {
    border: 1px solid black;
    padding: 15px;
  }
</style>

<div class="container">
  <div class="card">
    <h2>Tarjeta 1</h2>
    <p>Contenido de la tarjeta 1</p>
  </div>
  <div class="card active">
    <h2>Tarjeta 2</h2>
    <p>Contenido de la tarjeta 2</p>
  </div>
  <div class="card">
    <h2>Tarjeta 3</h2>
    <p>Contenido de la tarjeta 3</p>
  </div>
</div>

<div class="container">
  <div class="card">
    <h2>Tarjeta 1</h2>
    <p>Contenido de la tarjeta 1</p>
  </div>
  <div class="card">
    <h2>Tarjeta 2</h2>
    <p>Contenido de la tarjeta 2</p>
  </div>
  <div class="card">
    <h2>Tarjeta 3</h2>
    <p>Contenido de la tarjeta 3</p>
  </div>
</div>
```

Tarjeta 1 Contenido de la tarjeta 1	Tarjeta 2 Contenido de la tarjeta 2	Tarjeta 3 Contenido de la tarjeta 3
Tarjeta 1 Contenido de la tarjeta 1	Tarjeta 2 Contenido de la tarjeta 2	Tarjeta 3 Contenido de la tarjeta 3

Solución:

```
div.container:has(.active) > .card
```

3.4. Selecciona los elementos `<section>` que contengan al menos un tag `<a>`, siempre y cuando ese link apunte a un documento pdf, en mayúsculas o minúsculas.

Ejemplo:

```
<style>
  AQUÍ_TU_SOLUCION {
    background-color: red;
  }
</style>

<section>
  <a href="calendario.pdf">Calendario PDF</a>
  <a href="notas.txt">Notas TXT</a>
</section>
<section>
  <a href="peliculas.xls">Películas XLS</a>
  <a href="cumple.PDF">Cumpleaños PDF</a>
</section>
<section>
  <a href="peliculas.xls">Películas XLS</a>
</section>
```

[Calendario PDF](#) [Notas TXT](#)
[Películas XLS](#) [Cumpleaños PDF](#)
[Películas XLS](#)

Solución:

```
section:has(a[href$=".pdf" i])
```


3.5. Selecciona los elementos con clase *.main* que tengan al menos un elemento con un atributo *href* dentro de ellos.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    background-color: red;
  }
</style>

<div class="main">
  <p>Contenido principal</p>
  <a href="https://www.example.com">Link 1</a>
</div>
<div class="main">
  <p>Contenido principal</p>
</div>
```

Contenido principal

[Link 1](https://www.example.com)

Contenido principal

Solución:

```
.main:has(*[href])
```

También valdría:

```
.main:has([href])
```

3.6. Selecciona todos los elementos con clase *card* excepto aquellos que tienen una clase *disabled*.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    background-color: red;
  }
</style>

<div class="card">
  <h2>Tarjeta 1</h2>
  <p>Contenido de la tarjeta 1</p>
</div>
<div class="card disabled">
  <h2>Tarjeta 2</h2>
  <p>Contenido de la tarjeta 2</p>
</div>
<div class="card">
  <h2>Tarjeta 3</h2>
  <p>Contenido de la tarjeta 3</p>
</div>
```



Solución:

```
.card:not(.disabled)
```

3.7. Selecciona todos los párrafos con clase *item* excepto aquellos que son descendientes de un elemento con clase *active*.

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<p class="item">Item 1</p>
<div class="active">
  <p class="item">Item 2</p>
</div>
<p class="item">Item 3</p>
<div class="inactive">
  <p class="item">Item 4</p>
</div>
```

Item 1

Item 2

Item 3

Item 4

Solución:

p.item:not(.active p.item)

- **p.item** selecciona todos los elementos **<p>** con la clase **item**.
- **:not(.active p.item)** excluye de la selección a aquellos **<p class="item">** que son descendientes de cualquier elemento con la clase **active**.

3.8. Selecciona los elementos con clase *item* que son hijos directos de un elemento con clase *container* y que no sean elementos ** o *<a>*.

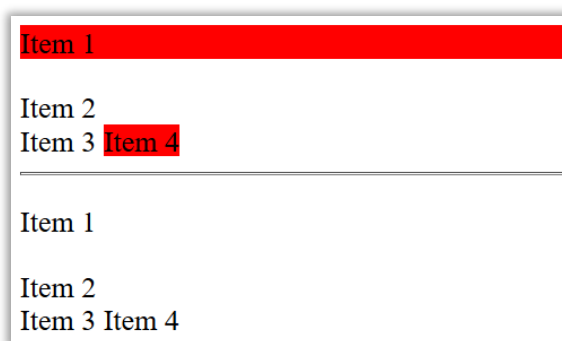
Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    background-color: red;
  }
</style>

<div class="container">
  <p class="item">Item 1</p>
  <li class="item">Item 2</li>
  <a class="item">Item 3</a>
  <span class="item">Item 4</span>
</div>

<hr>

<div class="main">
  <p class="item">Item 1</p>
  <li class="item">Item 2</li>
  <a class="item">Item 3</a>
  <span class="item">Item 4</span>
</div>
```



Solución:

```
.container > .item:not(:is(li, a))
```

4. Usa las pseudoclasas que hemos visto en clase para crear la declaración CSS correcta en los siguientes enunciados:

4.1. Selecciona el primer hijo de la clase *container* si y solo si ese hijo es de clase *item*:

Ejemplo:

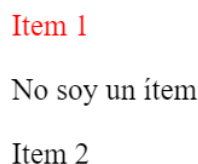
```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div class="container">
  <p class="item">Item 1</p>
  <p class="no-item">No soy un ítem</p>
  <p class="item">Item 2</p>
</div>
```

En este ejemplo, deberíamos ver de color rojo el texto “Item 1”. Sin embargo, si alternamos el orden de los dos primeros hijos:

```
<div class="container">
  <p class="no-item">No soy un ítem</p>
  <p class="item">Item 1</p>
  <p class="item">Item 2</p>
</div>
```

Entonces no veríamos ningún texto de color rojo, ya que el primer hijo de *container* no tiene clase *item*.



Item 1

No soy un ítem

Item 2

Solución:

```
.container > .item:first-child
```

¡Cuidado! No pongas un espacio entre *item* y los dos puntos ‘:’. Tal y como está selecciona exactamente lo que nos piden: el primer hijo de la clase *container* si y sólo si ese hijo es de clase *item*. Sin embargo, ¿qué estaríamos seleccionando si pusiéramos un espacio?

```
.container > .item :first-child
```

Como ya sabemos, el carácter espacio selecciona *descendientes de*, por tanto, estamos seleccionando a todos los últimos hijos que sean descendientes de un elemento con clase *item* que a su vez sea hijo de un elemento con clase *container*.

Por ejemplo:

```
<style>
  .container > .item :last-child {
    color: red;
  }
</style>

<div class="container">
  <div class="no-item">No soy un ítem</div>
  <div class="item">
    <div>
      <div>Soy nieto de ítem</div>
    </div>
    <div>Soy hijo de ítem</div>
  </div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

No soy un ítem
Soy el penúltimo nieto de ítem
Soy el último nieto de ítem
Soy el penúltimo hijo de ítem
Soy el último hijo de ítem
Item 2
Item 3

4.2. Selecciona el último elemento del mismo tipo dentro de un elemento con clase "container":

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div class="container">
  <p>Primer párrafo</p>
  <div>Primer div</div>
  <p>Segundo párrafo</p>
  <div>Segundo div</div>
  <p>Tercer párrafo</p>
  <div>Tercer div</div>
</div>
```

Primer párrafo
Primer div
Segundo párrafo
Segundo div
Tercer párrafo
Tercer div

Solución:

```
.container :last-of-type
```

O, también:

```
.container *:last-of-type
```

De nuevo, cuidado con el espacio entre container y el carácter dos puntos. Si no ponemos ese espacio estaríamos seleccionando el último elemento de la clase *container*, y como solo hay uno dentro de **body**, lo estaríamos seleccionando a él. Por tanto, esta declaración:

```
.container:last-of-type
```

En este caso es equivalente a:

```
.container
```

Y nos pondría de color rojo todo el texto del interior de *container*, dado que el valor por defecto de la propiedad **color** para divs y en párrafos es **inherit**.

4.3. Selecciona el único elemento hijo de un elemento con clase "container":

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div class="container">
  <p>Soy único hijo</p>
</div>
<div class="container">
  <p>Tengo un hermano</p>
  <p>Yo soy el hermano</p>
</div>
```

Soy único hijo

Tengo un hermano

Yo soy el hermano

Solución:

```
.container > :only-child
```

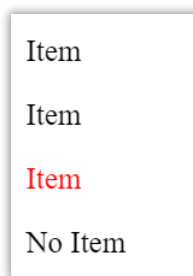
Sólo se nos pone de rojo el primer párrafo porque es el único que es hijo único.

4.4. Selecciona el tercer elemento con clase *item* dentro de un elemento con clase "container":

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    color: red;
  }
</style>

<div class="container">
  <p class="item">Item</p>
  <p class="item">Item</p>
  <p class="item">Item</p>
  <p class="no-item">No Item</p>
</div>
```



Item

Item

Item

No Item

Solución:

Valdría cualquiera de las dos siguientes:

```
.container .item:nth-child(3)
.container .item:nth-of-type(3)
```

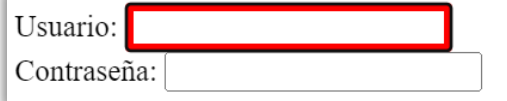
En este ejemplo dependemos del HTML, si modificas la posición del elemento con clase **no-item**, el selector falla. CSS no tiene una forma directa de seleccionar el "tercer elemento de una clase específica" independientemente de los elementos en el DOM. Este tipo de selección avanzada requeriría usar JavaScript para contar los elementos que cumplen con la clase y seleccionar el tercero, sin importar su posición en el HTML.

4.5. Aplica un estilo de borde rojo a cualquier input de tipo *text* que reciba el foco:

Ejemplo:

```
<style>
  AQUI_TU_SOLUCION {
    border: 5px solid red;
  }
</style>

<label for="usu">Usuario:</label>
<input type="text" id="usu" name="usu">
<br>
<label for="pass">Contraseña:</label>
<input type="password" id="pass" name="pass">
```

A screenshot of a web form with two input fields. The first field is labeled 'Usuario:' and the second is labeled 'Contraseña:'. The 'Usuario' input field is currently focused, indicated by a thick red border. The 'Contraseña' input field is a standard password field with a grey border.

Solución:

```
input[type="text"]:focus
```

No valdría:

```
[type="text"]:focus
```

Porque hay otros elementos HTML que pueden tener atributo *type*.