



GERMINARE

Escola de Negócios

Uma década germinando líderes.

 PicPay



<HTML>CSS: Especificidade, Herança e mais seletores</HTML>

EQUIPE DE TI - 2022

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- Seletores ID;
- Estilo Inline

Considere o seguinte elemento HTML:

```
<div>
  <p>
    Este é um texto referencia para o estudo de especificadade. Ele servirá
    de referência para a verificação das estilizações geradas pelas
    aplicações sucessivas de regras.
  </p>
</div>
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- Seletores ID;
- Estilo Inline

Este texto é uma referência para o estudo de especificidade. Ele servirá de parâmetro para verificação das estilizações geradas pelas aplicações de sucessivas regras.

CSS:

```
div {  
  background-color: blue;  
}
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- Seletores ID;
- Estilo Inline

Considere o seguinte elemento HTML:

```
<div class="espec">
  <p>
    Este texto é uma referência para o estudo de especificidade. Ele servirá
    de parâmetro para verificação das estilizações geradas pelas aplicações
    de sucessivas regras.
  </p>
</div>
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- **Seletor de Classe;**
- Seletores ID;
- Estilo Inline

Resultado:

Este texto é uma referência para o estudo de especificidade. Ele servirá de parâmetro para verificação das estilizações geradas pelas aplicações de sucessivas regras.

CSS

```
div {  
  background-color: blue;  
}  
  
.espec {  
  background-color: green;  
  color: white;  
}
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- Seletores ID;
- Estilo Inline

Considere o seguinte elemento HTML:

```
<div class="espec" id="especificidade">
  <p>
    Este texto é uma referência para o estudo de especificidade. Ele servirá
    de parâmetro para verificação das estilizações geradas pelas aplicações
    de sucessivas regras.
  </p>
</div>
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- **Seletores ID;**
- Estilo Inline

Resultado:

Este texto é uma referência para o estudo de especificidade. Ele servirá de parâmetro para verificação das estilizações geradas pelas aplicações de sucessivas regras.

CSS

```
div {  
  background-color: blue;  
}  
  
.espec {  
  background-color: green;  
  color: white;  
}  
  
#especificidade {  
  background-color: red;  
  color: blue;  
}
```

CSS: regras conflitantes – Especificidade

Existem situações em que se tem uma ou mais regras sendo aplicadas para um mesmo elemento HTML. Nesses casos é importante compreender qual das regras vai prevalecer. Consideramos a seguinte ordem, de acordo com a documentação MDN:

- Seletores de Tipo;
- Seletor de Classe;
- Seletores ID;
- **Estilo Inline**

Considere o seguinte elemento HTML:

```
<div
  class="espec"
  id="especificidade"
  style="background-color: yellow; color: black"
>
  <p>
    Este texto é uma referência para o estudo de especificidade. Ele servirá
    de parâmetro para verificação das estilizações geradas pelas aplicações
    de sucessivas regras.
  </p>
</div>
```

CSS

```
div {
  background-color: blue;
}

.espec {
  background-color: green;
  color: white;
}

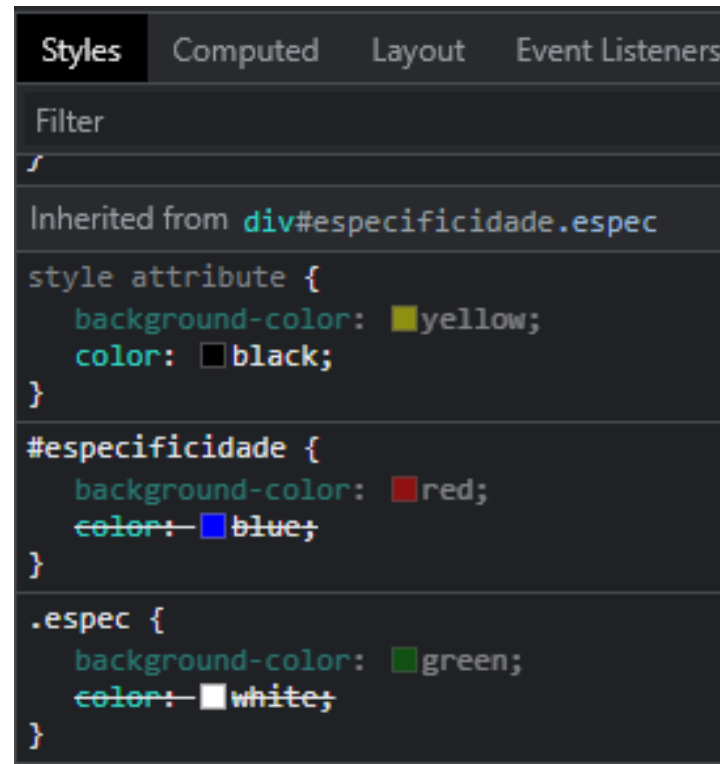
#especificidade {
  background-color: red;
  color: blue;
}
```

Resultado:

Este texto é uma referência para o estudo de especificidade. Ele servirá de parâmetro para verificação das estilizações geradas pelas aplicações de sucessivas regras.

CSS: regras conflitantes – inspect

É possível verificar a sobreposição dos valores das propriedades por meio do navegador, inspecionando o elemento:



O valor da propriedade color é sobrescrito de “white” para “blue” e para “black”.

CSS: Herança

Além da especificidade existe outro mecanismo que merece atenção no que diz respeito ao predomínio das regras de estilo: a **herança**. Essa conceito refere-se ao valor que uma determinada propriedade assume por padrão quando não há nenhum valor especificado pra ela.

Referência de CSS

Use esta **referência de CSS** para navegar por um índice alfabético das propriedades padrão do [CSS](#), [pseudo-classes](#), [pseudo-elementos](#), tipos de dados e [@-rules](#).

Esta referência lista não somente as propriedades do CSS1 e CSS2.1, mas também referências para qualquer propriedade, conceito padronizado ou estabilizado do CSS3.

Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/Reference>

Para isso pode-se consultar sua documentação e verificar o “inherited” da propriedade.

CSS: Herança – Consultando a documentação

Abaixo um exemplo de consulta da propriedade `background-color` na documentação (<https://developer.mozilla.org/pt-BR/docs/Web/CSS/background-color>):

Definição formal

<u>Initial value</u>	transparent
Aplica-se a	all elements. It also applies to <code>::first-letter</code> and <code>::first-line</code> .
<u>Inherited</u>	não
<u>Computed value</u>	computed color
Animation type	a <u>color</u>

O valor inicial da propriedade (*initial value*) é transparente (*transparent*), e ela não é herdada (*Inherited*: não). Portanto, os filhos de um elemento em que se define a propriedade *background-color* não receberão o mesmo valor de seu pai.

CSS: Herança – <div> e <p>

Abaixo um exemplo de consulta de estilização de uma <div> em que seu filho <p> herda os valores das propriedades color e text-aligning:

CSS:

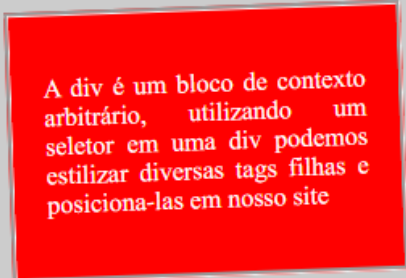
```
#minhaPrimeiraDiv {  
  width: 200px;  
  text-align: justify;  
  background-color: red;  
  padding: 20px;  
  color: white;  
  margin: auto; /*definir a margem como auto irá centralizar o elemento na página*/  
  transform: rotate(  
    -2deg  
  );  
  border-style: groove;  
}
```

HTML:

```
<div id="minhaPrimeiraDiv">  
  <p>  
    A div é um bloco de contexto arbitrário, utilizando um seletor em uma  
    div podemos estilizar diversas tags filhas e posiciona-las em nosso site  
  </p>  
</div>
```

Resultado:

Esse método será muito aplicado a boxes específicas dentro do site, para isso, iremos iniciar a tag div



A div é um bloco de contexto arbitrário, utilizando um seletor em uma div podemos estilizar diversas tags filhas e posiciona-las em nosso site

CSS: Herança – <div> e <p> – inspect

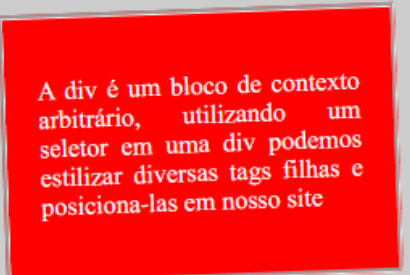
Com o *inspect* é possível verificar como a herança acontece na tag <p>, elas são destacadas:

CSS:

```
#minhaPrimeiraDiv {  
  width: 200px;  
  text-align: justify;  
  background-color: red;  
  padding: 20px;  
  color: white;  
  margin: auto; /*definir a margem como auto irá centralizar o elemento na página*/  
  transform: rotate(  
    -2deg  
  );  
  border-style: groove;  
}
```

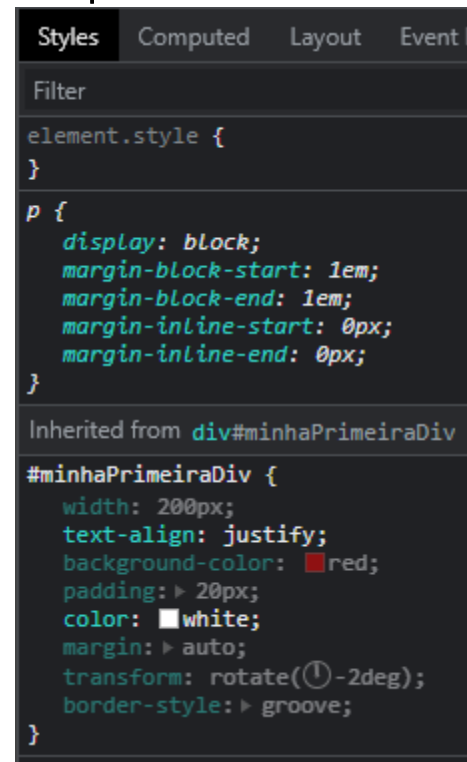
Resultado:

Esse método será muito aplicado a boxes específicas dentro do site, para isso, iremos iniciar a tag div



A div é um bloco de contexto arbitrário, utilizando um seletor em uma div podemos estilizar diversas tags filhas e posiciona-las em nosso site

Inspect:



Styles Computed Layout Event L

Filter

element.style {
}

p {
 display: block;
 margin-block-start: 1em;
 margin-block-end: 1em;
 margin-inline-start: 0px;
 margin-inline-end: 0px;
}

Inherited from div#minhaPrimeiraDiv

#minhaPrimeiraDiv {
 width: 200px;
 text-align: justify;
 background-color: red;
 padding: 20px;
 color: white;
 margin: auto;
 transform: rotate(⌚-2deg);
 border-style: groove;
}

CSS: mais sobre seletores

SELECTOR	MEANING	EXAMPLE
UNIVERSAL SELECTOR	Applies to all elements in the document	<code>* {}</code> Targets all elements on the page
TYPE SELECTOR	Matches element names	<code>h1, h2, h3 {}</code> Targets the <code><h1></code> , <code><h2></code> and <code><h3></code> elements
CLASS SELECTOR	Matches an element whose <code>class</code> attribute has a value that matches the one specified after the period (or full stop) symbol	<code>.note {}</code> Targets any element whose <code>class</code> attribute has a value of <code>note</code> <code>p.note {}</code> Targets only <code><p></code> elements whose <code>class</code> attribute has a value of <code>note</code>
ID SELECTOR	Matches an element whose <code>id</code> attribute has a value that matches the one specified after the pound or hash symbol	<code>#introduction {}</code> Targets the element whose <code>id</code> attribute has a value of <code>introduction</code>
CHILD SELECTOR	Matches an element that is a direct child of another	<code>h1>a {}</code> Targets any <code><a></code> elements that are children of an <code><h1></code> element (but not other <code><a></code> elements in the page)
DESCENDANT SELECTOR	Matches an element that is a descendent of another specified element (not just a direct child of that element)	<code>p a {}</code> Targets any <code><a></code> elements that sit inside a <code><p></code> element, even if there are other elements nested between them
ADJACENT SIBLING SELECTOR	Matches an element that is the next sibling of another	<code>h1+p {}</code> Targets the first <code><p></code> element after any <code><h1></code> element (but not other <code><p></code> elements)
GENERAL SIBLING SELECTOR	Matches an element that is a sibling of another, although it does not have to be the directly preceding element	<code>h1~p {}</code> If you had two <code><p></code> elements that are siblings of an <code><h1></code> element, this rule would apply to both



GERMINARE

Escola de Negócios

Uma década germinando líderes.

 PicPay

CSS: seletor universal

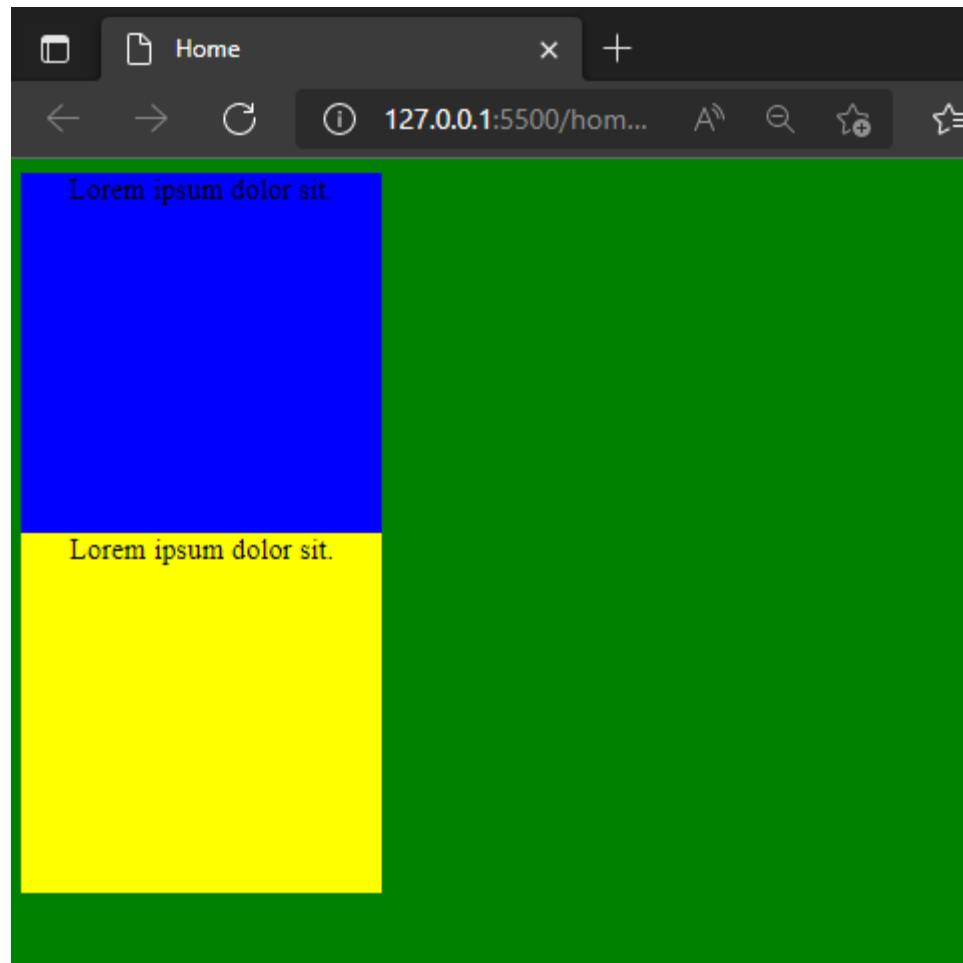
Com o seletor universal é possível aplicar estilos a qualquer elemento HTML. Abaixo um exemplo de sua utilização e escrita.

CSS:

```
* {  
  background-color: green;  
  text-align: center;  
}  
  
.bloco-azul {  
  width: 200px;  
  height: 200px;  
  background-color: blue;  
}  
  
.bloco-amarelo {  
  width: 200px;  
  height: 200px;  
  background-color: yellow;  
}
```

HTML:

```
<body>  
  <div class="bloco-azul">Lorem ipsum dolor sit.</div>  
  <div class="bloco-amarelo">Lorem ipsum dolor sit.</div>  
</body>
```



CSS: seletor de tipo

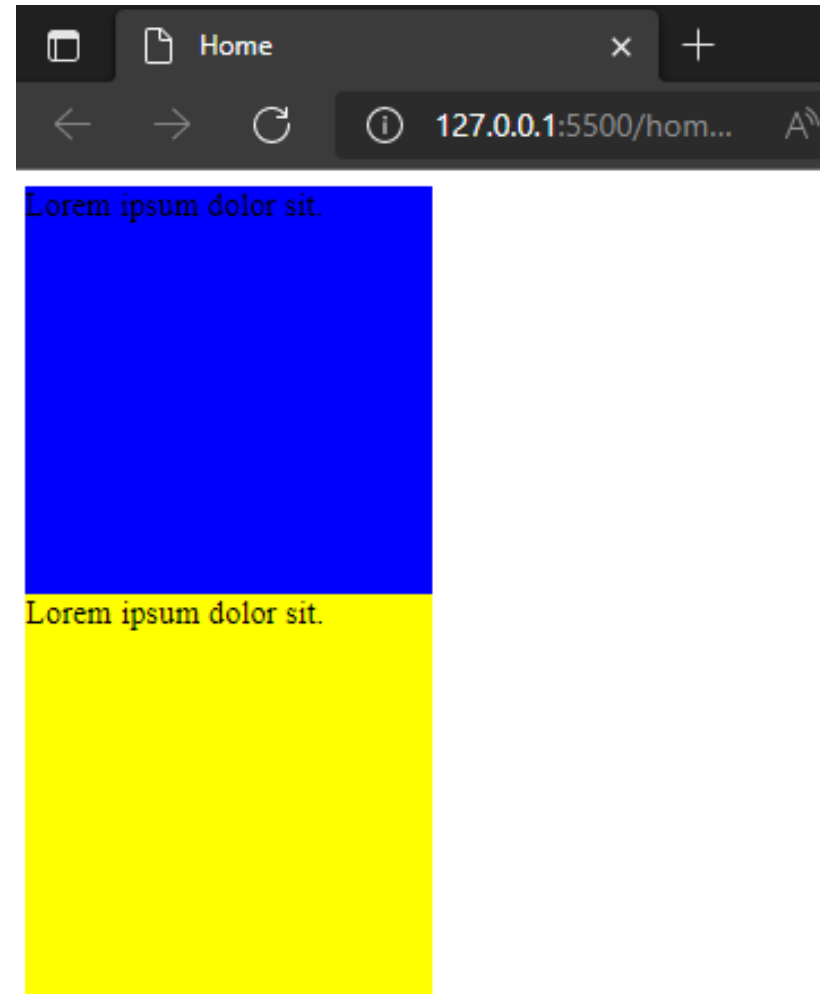
Com o seletor de tipo é possível selecionar um determinado elemento, aplicando uma regra de estilização para ele.

CSS:

```
div {  
  width: 200px;  
  height: 200px;  
  background-color: blue;  
}  
  
section {  
  width: 200px;  
  height: 200px;  
  background-color: yellow;  
}
```

HTML:

```
<body>  
  <div>Lorem ipsum dolor sit.</div>  
  <section>Lorem ipsum dolor sit.</section>  
</body>
```



CSS: seletor de filhos

Os seletor de filhos permite selecionar os filhos de um ou mais determinados elementos HTML. Considere, inicialmente, a implementação abaixo:

HTML:

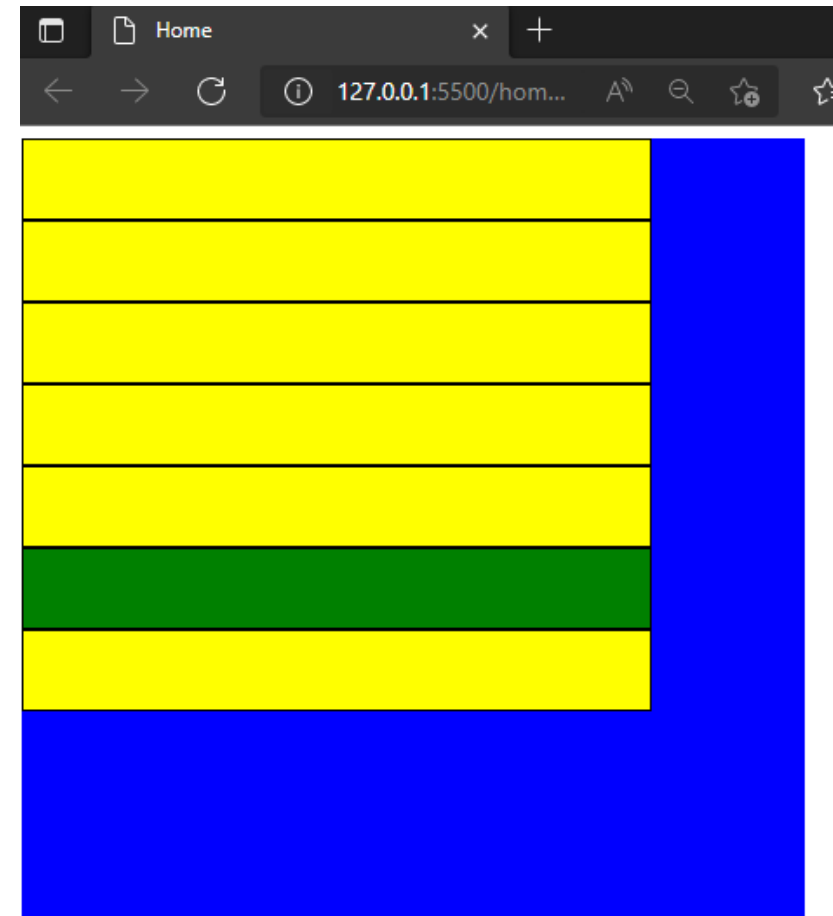
```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho-dif"></div>
    <div class="filho"></div>
  </div>
</body>
```

CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  background-color: yellow;
  border: 1px solid black;
}

.filho-dif {
  width: 400px;
  height: 50px;
  background-color: green;
  border: 1px solid black;
}
```



CSS: seletor de filhos

Considerando, agora, esta outra implementação em CSS, tem-se a seguinte alteração visual:

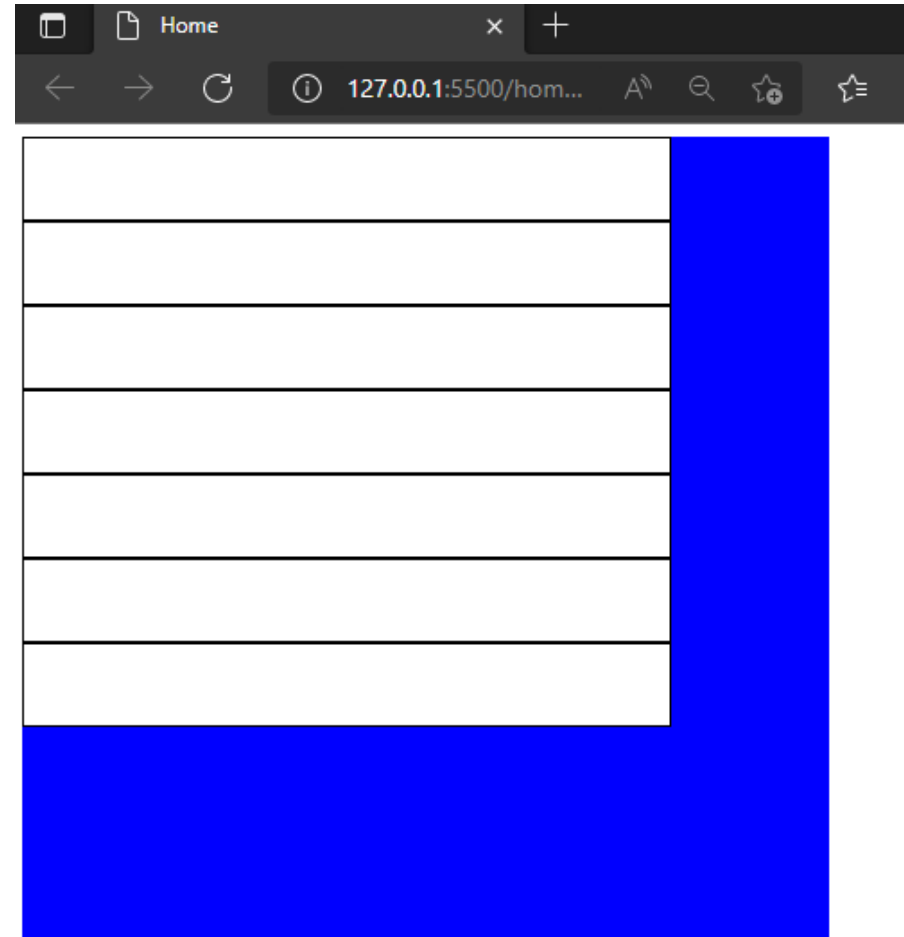
HTML:

```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho-dif"></div>
    <div class="filho"></div>
  </div>
</body>
```

CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

div > div {
  width: 400px;
  height: 50px;
  background-color: white;
  border: 1px solid black;
}
```



CSS: seletor de descendentes

Com o seletor de descendentes é possível selecionar não somente o descendente direto de um elemento HTML mas todos seus descendentes subsequentes. Considere, inicialmente, as implementações abaixo:

HTML:

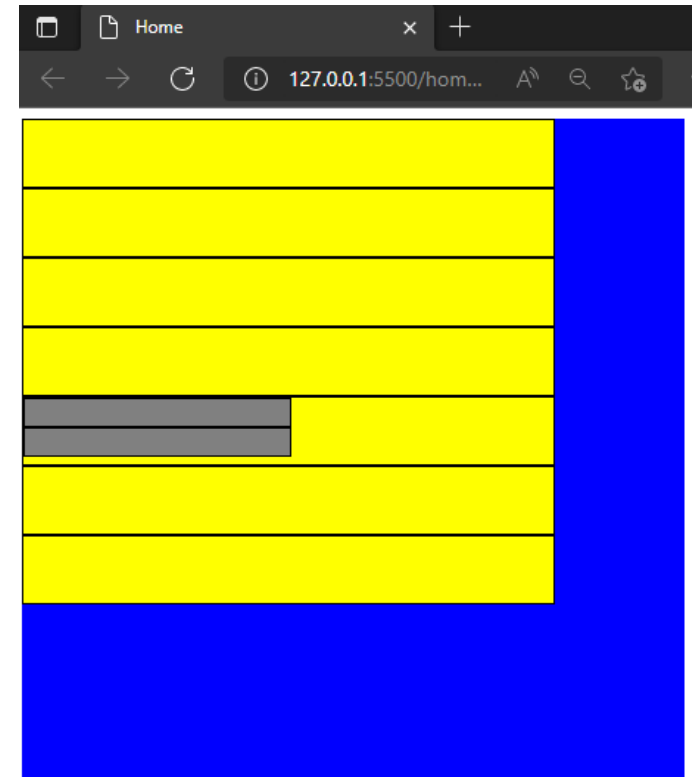
```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho">
      <div class="neto"></div>
      <div class="neto"></div>
    </div>
    <div class="filho"></div>
    <div class="filho"></div>
  </div>
</body>
```

CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  background-color: yellow;
  border: 1px solid black;
}

.neto {
  width: 200px;
  height: 20px;
  background-color: grey;
  border: 1px solid black;
}
```



CSS: seletor de descendentes

Considerando, agora, esta outra implementação em CSS, tem-se a seguinte alteração visual:

HTML:

```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho"></div>
    <div class="filho">
      <div class="neto"></div>
      <div class="neto"></div>
    </div>
    <div class="filho"></div>
    <div class="filho"></div>
  </div>
</body>
```

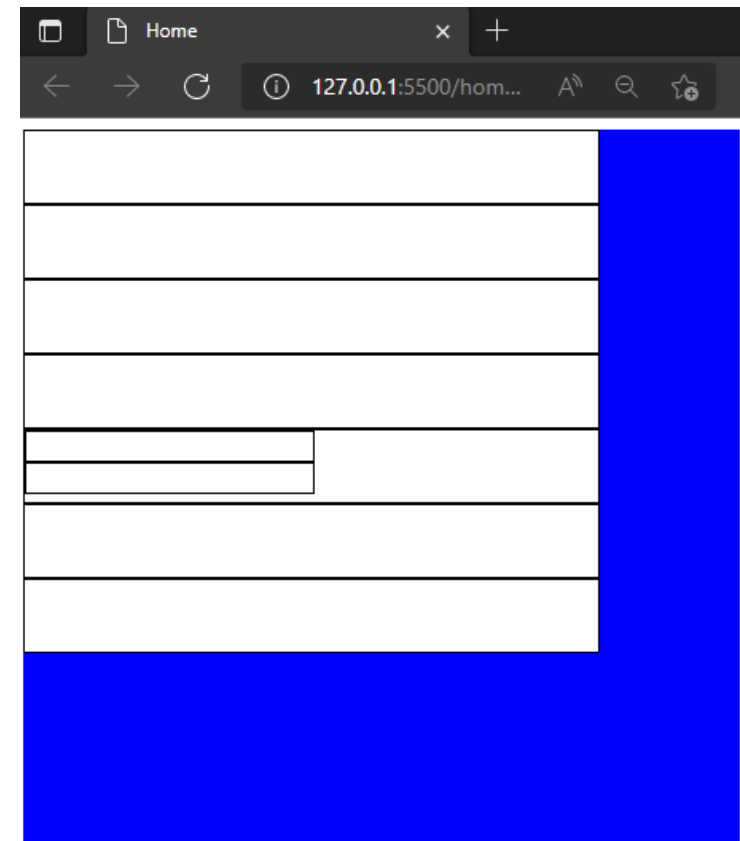
CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.neto {
  width: 200px;
  height: 20px;
  border: 1px solid black;
}

div div {
  background-color: white;
}
```



CSS: seletor de irmão adjacente

Com o seletor de irmão adjacente é possível selecionar um determinado elemento HTML que é um irmão subsequente de um determinado elemento. Considere, inicialmente, as implementações abaixo:

HTML:

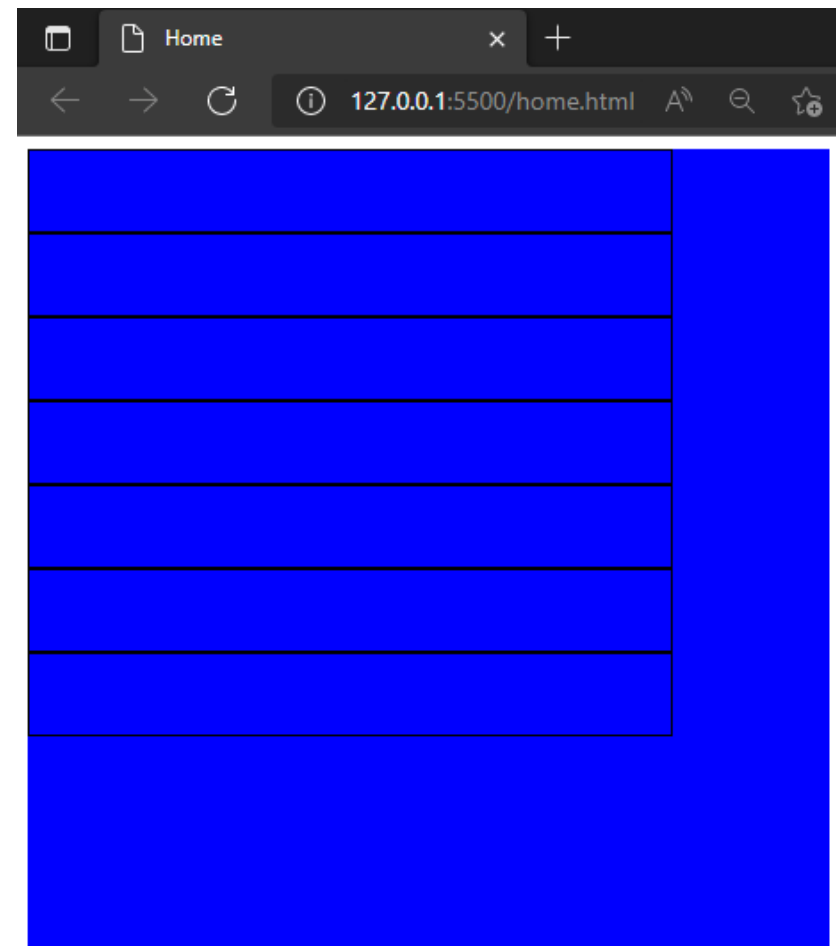
```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
  </div>
</body>
```

CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.irmao {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}
```



CSS: seletor de irmão adjacente

Considerando, agora, esta outra implementação em CSS, tem-se a seguinte alteração visual:

HTML:

```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
  </div>
</body>
```

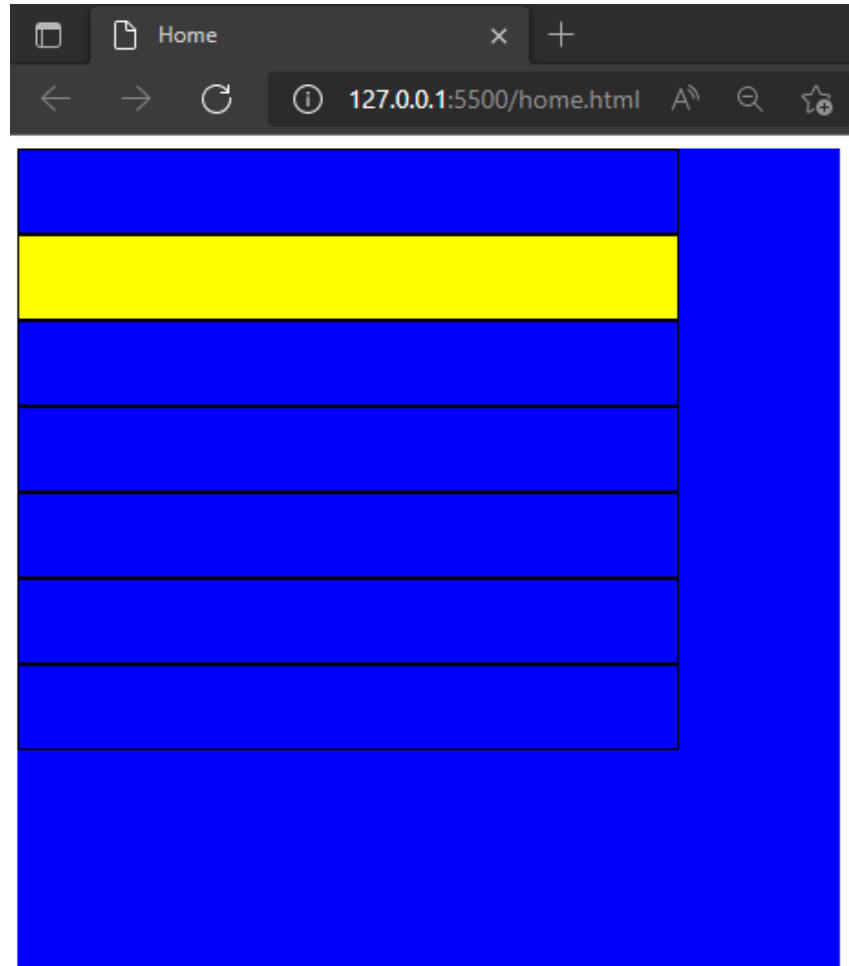
CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.irmao {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.filho + div {
  background-color: yellow;
}
```



CSS: seletor geral de irmãos

Com o seletor geral de irmãos é possível selecionar todos os determinados elementos HTML que são irmãos subsequentes de um determinado elemento. Considere, inicialmente, as implementações abaixo:

HTML:

```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
  </div>
</body>
```

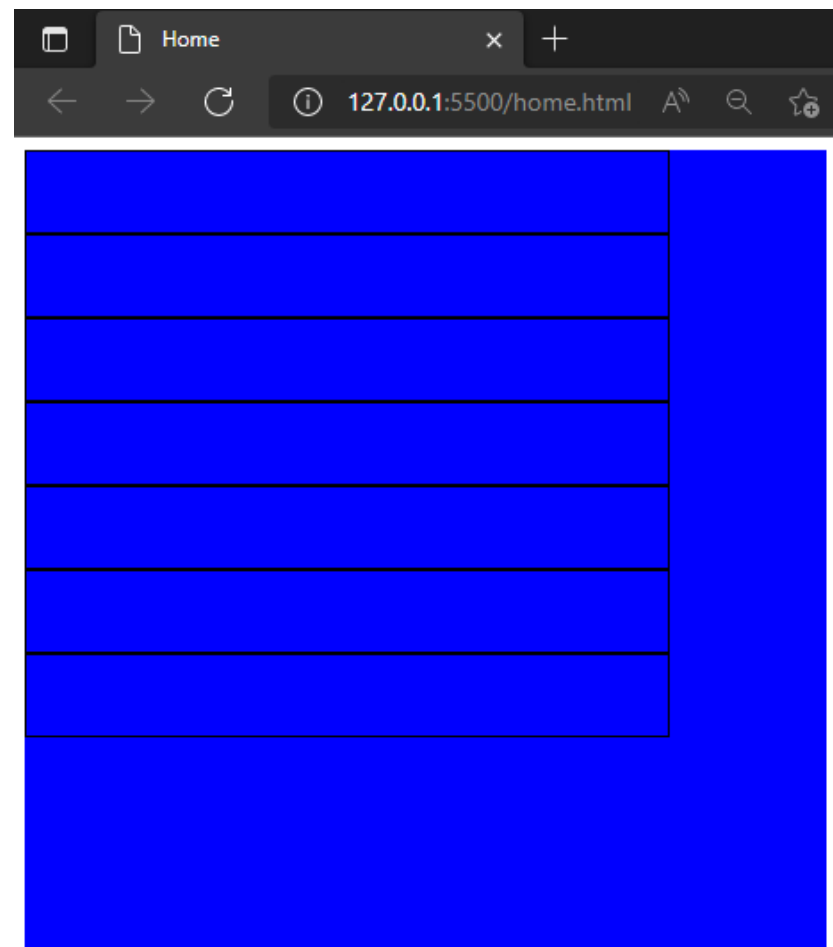
CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.irmao {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.filho ~ div {
  background-color: yellow;
}
```



CSS: seletor geral de irmãos

Considerando, agora, esta outra implementação em CSS, tem-se a seguinte alteração visual:

HTML:

```
<body>
  <div class="pai">
    <div class="filho"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
    <div class="irmao"></div>
  </div>
</body>
```

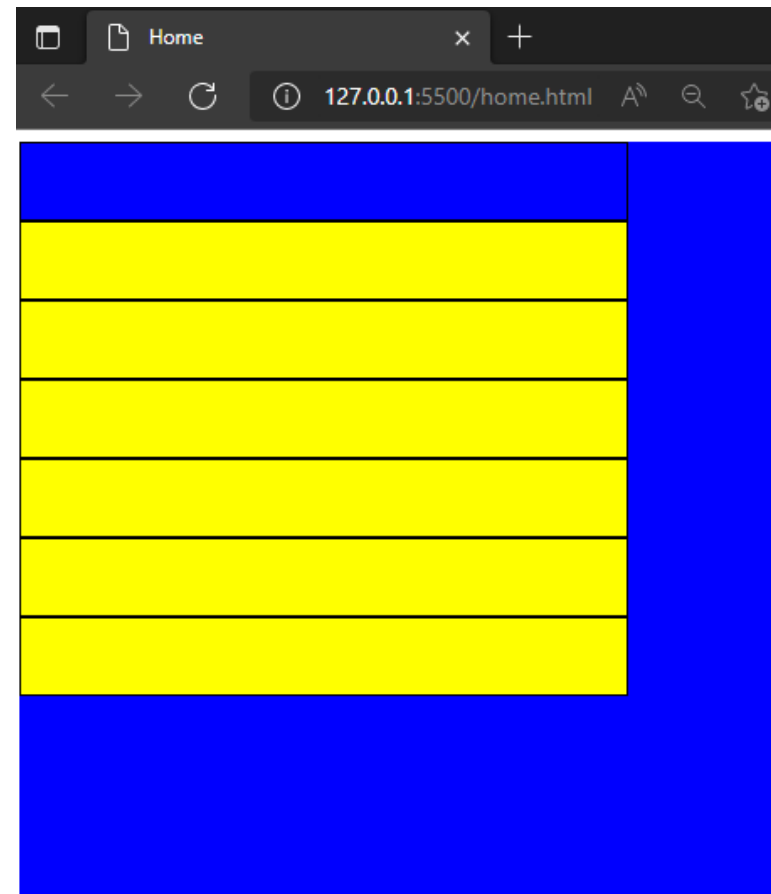
CSS:

```
.pai {
  width: 500px;
  height: 500px;
  background-color: blue;
}

.filho {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

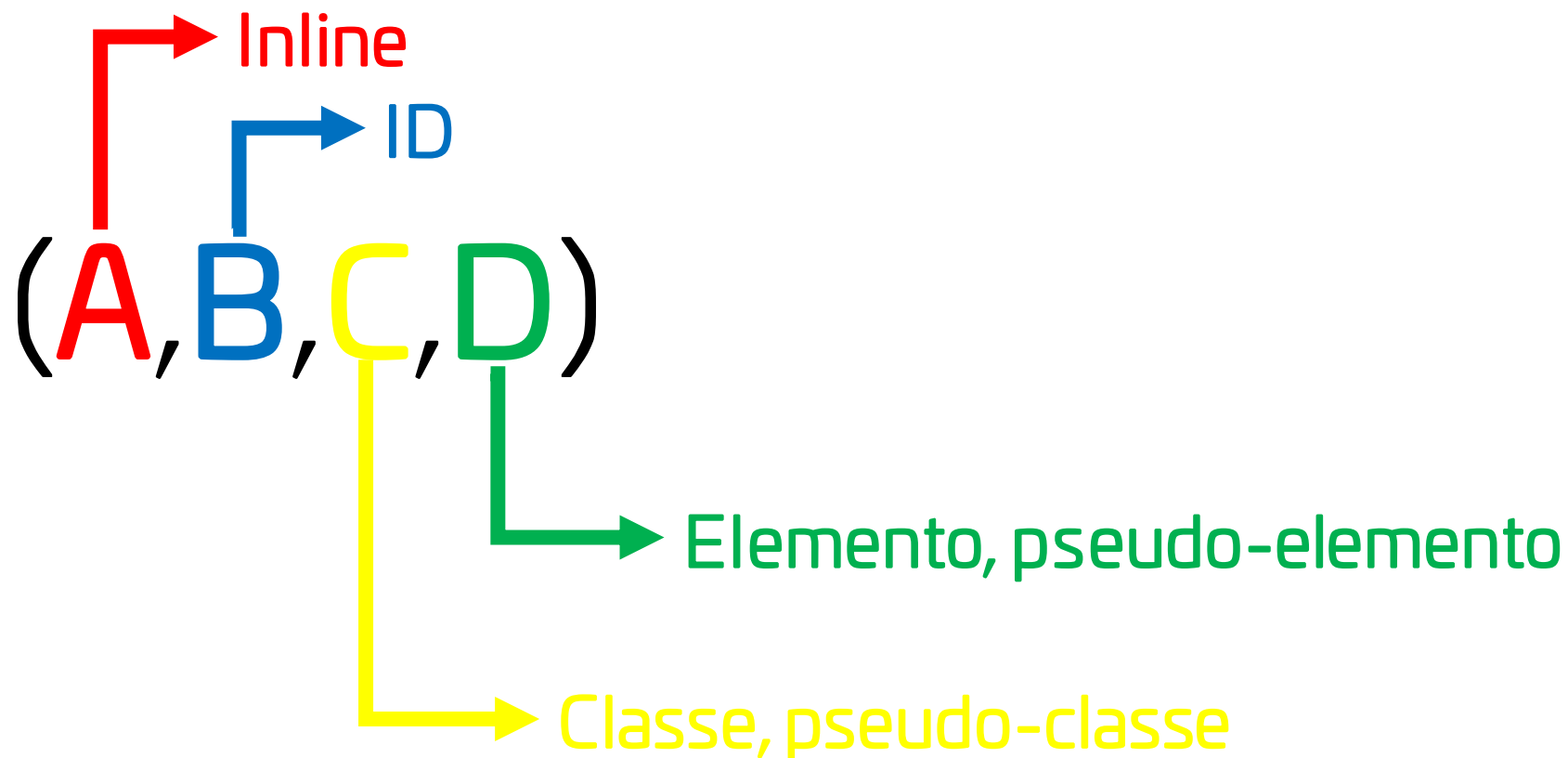
.irmao {
  width: 400px;
  height: 50px;
  border: 1px solid black;
}

.filho + div {
  background-color: yellow;
}
```

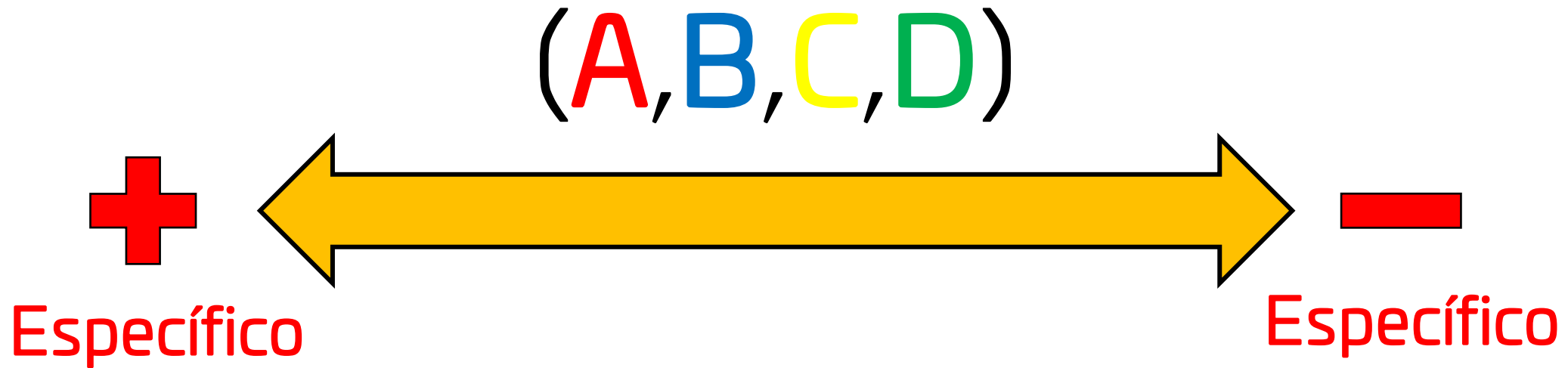


CSS: cálculo da especificidade

O ‘valor’ da especificidade de uma determinada regra pode ser ‘mensurado’ de acordo com a representação abaixo, onde **A**, **B**, **C** e **D** são números inteiros positivos.



CSS: cálculo da especificidade



CSS: cálculo da especificidade – Exemplos

Abaixo alguns exemplos referentes ao ‘cálculo’ de especificidade:

```
* {  
  background-color: blue;  
}
```


(0,0,0,0)

```
p {  
  background-color: blue;  
}
```


(0,0,0,1)

CSS: cálculo da especificidade – Exemplos

Abaixo alguns exemplos referentes ao ‘cálculo’ de especificidade:

```
ul li {  
  background-color:  blue;  
}
```

(0,0,0,2)

```
div .card{  
  background-color:  blue;  
}
```

(0,0,1,1)

CSS: cálculo da especificidade – Exemplos

Abaixo alguns exemplos referentes ao ‘cálculo’ de especificidade:

```
.filho + div {  
  background-color: ☐ blue;  
}
```


(0,0,1,1)

```
header#banner + .info {  
  background-color: ☐ blue;  
}
```


(0,1,1,1)

CSS: cálculo da especificidade – Exemplos

Abaixo alguns exemplos referentes ao ‘cálculo’ de especificidade:

```
<p style="background-color: blue">Lorem ipsum</p>
```

(1,0,0,0)

```
#navegador ul#cl {  
| background-color: blue;  
}
```

(0,2,0,1)

Referências

DUCKETT, Jon. HTML & CSS: design and build websites. Indianapolis, IN: Wiley, 2011.

