



Programação para Internet

Módulo 2 – Parte 2

Introdução à Linguagem CSS

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo protegido por direito autoral, nos termos da Lei nº 9 610/98

A cópia, reprodução ou apropriação deste material, total ou parcialmente, é proibida pelo autor

Conteúdo da Aula

1. Propriedades CSS
2. Centralização de Conteúdo
3. Pseudo-Classes
4. Pseudo-Elementos
5. Cascade, Especificidade e Herança
6. Design Responsivo
7. Media Queries
8. Dicas e Validação

Algumas Propriedades de Ajuste de Fundo

Propriedade	Descrição	Exemplos de Valores
background-color	Define a cor de fundo	<i>lightgray</i>
background-image	Define uma imagem de fundo	<i>url("images/bgImage.png")</i>
background-repeat	Modo de repetição da imagem de fundo	<i>no-repeat, repeat-x, repeat-y, repeat</i>
background-size	Tamanho da imagem de fundo	<i>100% 100%, cover</i>
background-position	Posição da imagem de fundo	<i>right top</i>

Há também a propriedade abreviada **background** para definir tudo de uma vez

Propriedades de Ajuste de Fundo - Exemplo

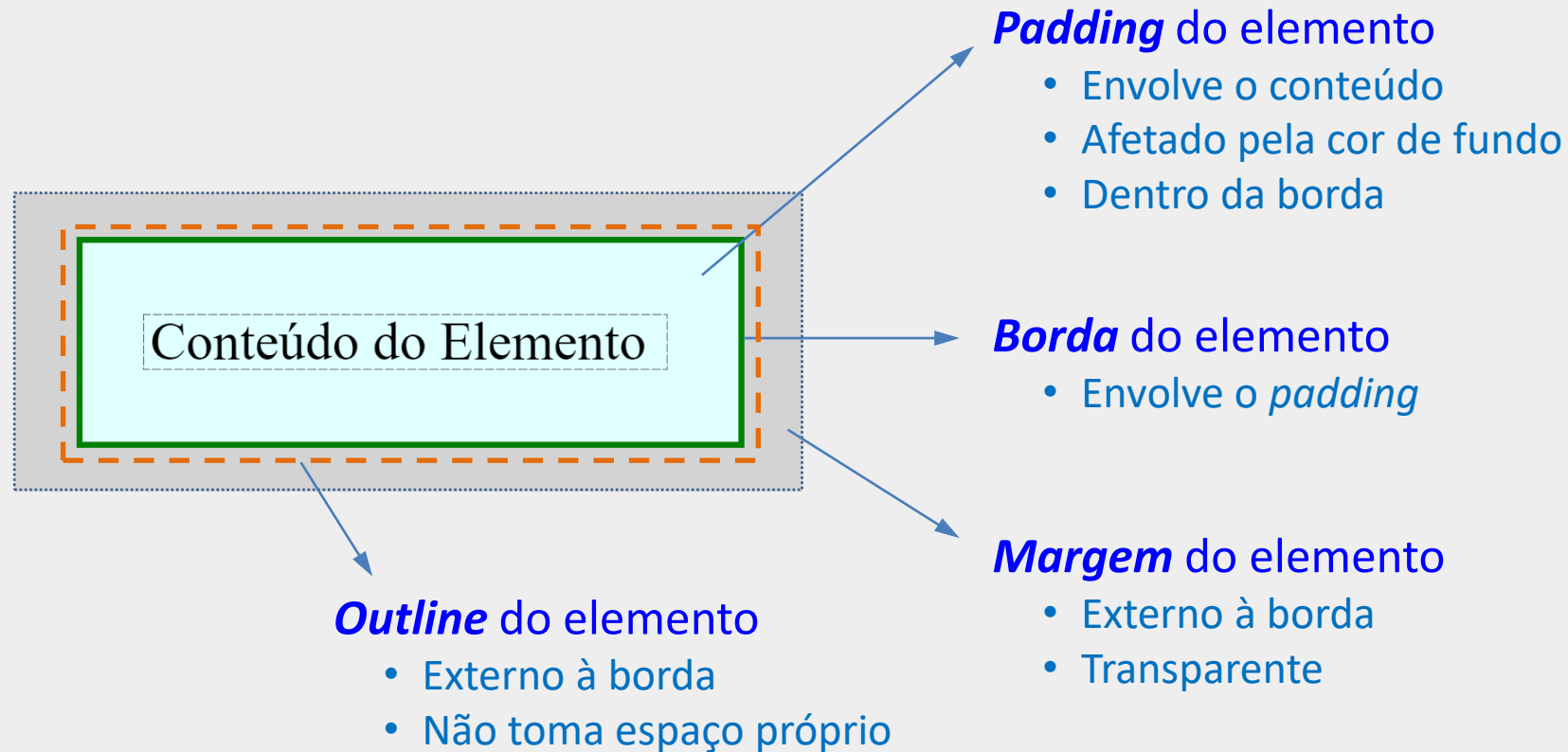
```
<style>
  body {
    background-image: url("images/bg2.jpg");
    background-repeat: no-repeat;
    background-size: cover;
  }
</style>
```

Testando imagem de fundo

Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero numquam at atque ipsam voluptas magni in nihil laborum est, magnam quia repellat quisquam recusandae dolor nisi possimus quas? Exercitationem, assumenda!



Conteúdo, Padding, Borda e Margem



Propriedade *padding*

- Propriedade Abreviada

`padding: 20px 50px 80px 100px;` (superior, direita, inferior e esquerda)

`padding: 20px 40px 20px;` (superior, laterais, inferior)

`padding: 20px 40px;` (superior/inferior, laterais)

`padding: 20px;` (todos os *padding*s)

- Propriedades Individuais

`padding-top`

`padding-bottom`

`padding-left`

`padding-right`

Propriedade *margin*

- Propriedade Abreviada

`margin: 20px 50px 80px 100px;` (superior, direita, inferior e esquerda)

`margin: 20px 40px 20px;` (superior, laterais, inferior)

`margin: 20px 40px;` (superior/inferior, laterais)

`margin: 20px;` (todas as margens)

- Propriedades Individuais

`margin-top`

`margin-bottom`

`margin-left`

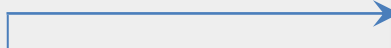
`margin-right`

Propriedade *margin* - Observação

- Margens entre dois elementos de bloco vizinhos não se somam
- Prevalecerá a maior delas

Propriedade *border*

- Propriedade Abreviada
- Permite definir a borda por completo (4 lados)
- Sintaxe mais comum:
`border: espessura estilo cor;`
Ex.: `border: 1px solid blue;`
- A ordem dos valores não importa



solid
dotted
dashed
double
none

Definindo as Bordas Individualmente

- Outras propriedades abreviadas (semelhantes a *border*)
 - `border-top`
 - `border-bottom`
 - `border-left`
 - `border-right`
- Propriedades constituintes
 - `border-top-color`
 - `border-top-style`
 - `etc.`

Definindo Aspectos Específicos das Bordas

- **border-color** define apenas a cor da borda
- **border-style** define apenas o estilo da borda
- **border-width** define apenas a espessura da borda

Exemplos:

- `border-color: blue;`
- `border-color: red green;`
- `border-width: 1px;`
- `border-top-color: gray;`
- `border-top-width: 2px;`

Ajustes de Borda - Exemplo

```
#p1 {  
  border: 1px solid blue;  
}  
  
#p2 {  
  border-color: red green blue black;  
  border-style: dashed dotted double solid;  
  border-width: 4px 6px 8px 10px;  
}  
  
#p3 {  
  border-color: gray;  
  border-style: solid none;  
  border-width: 2px;  
}  
  
#p4 {  
  border-top: 1px solid blue;  
}
```

Programação para Internet

Programação para Internet

Programação para Internet

Programação para Internet

Elements Console

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>...</head>
  <body>
    ... <p id="p1"> Programação para Internet </p> == $0
    <p id="p2"> Programação para Internet </p>
    <p id="p3"> Programação para Internet </p>
  </body>
</html>
```

html body p#p1

Styles Computed Layout Event Listeners

Filter :hov .cls +

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

margin 40

border 0.992

padding 10

18.046 x 54.404

10 10 10 10

0.992 0.992 0.992 0.992

40 40 40 40

Conferindo os valores de margens, paddings e bordas no Google Chrome. Clique com o botão direito sobre o elemento que desejar verificar e escolha **Inspecionar**.

Propriedade ***border-radius***

- Define a **borda** e o ***background*** com cantos arredondados
- Algumas formas de uso:

`border-radius: 3px;`

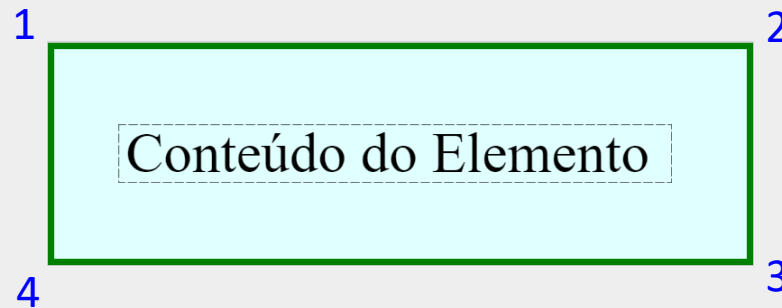
4 cantos com mesma curvatura, em pixels

`border-radius: 10%;`

4 cantos com mesma curvatura, em porc.

`border-radius: 10px 20px 30px 40px;`

cada canto com curvatura própria,
na ordem dos números a seguir:



Propriedade *border-radius*

```
#p1 {  
  border-radius: 10px;  
}  
  
#p2 {  
  border-radius: 10px 20px 40px 60px  
}  
  
#p3 {  
  border-radius: 10px 40px;  
  border: none;  
}  
  
#p4 {  
  border-radius: 20% 50% 50% 20%;  
}
```

Programação para Internet (P1)

Programação para Internet (P2)


Programação para Internet (P3)

Programação para Internet (P4)

Propriedade *box-shadow*


`box-shadow: 20px 10px;`

Programação para Internet




`box-shadow: 5px 5px gray;`

Programação para Internet



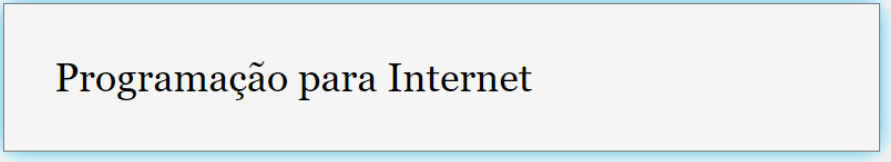
`box-shadow: 5px 5px 5px gray;`

Programação para Internet



`box-shadow: 0 0 10px cyan;`

Programação para Internet



Ajustes de Largura e Altura dos Elementos

- As propriedades a seguir permitem definir, respectivamente, a largura, altura, mínima largura, mínima altura, máxima largura e máxima altura:
 - `width`, `height`,
 - `min-width`, `min-height`,
 - `max-width`, `max-height`
- Observações
 - Não contabilizam margens, bordas e *padding*s
 - Valores em pixels, porcentagem, em, rem, etc.
 - Outros valores: *auto*, *max-content*, *min-content*, *fit-content*

Propriedade *display*

- Altera o modo de **apresentação** do elemento
- Um elemento de linha pode ser exibido como bloco e vice-versa
- Permite ocultar um elemento, removendo do layout
- Alguns valores possíveis:
 - `none`
 - `block`
 - `inline`
 - `inline-block`
 - `flex`
 - `grid`
 - `inline-flex`
 - `inline-grid`

Propriedade *display*

`display: none`

- oculta completamente o elemento, removendo do layout
- libera o espaço para outros

`display: block`

- exibição em nível de bloco (como o <div>)
- começará em nova linha e ocupará, por padrão, toda a largura

Propriedade *display*

`display: inline`

- exibição em nível de linha (como o ``)
- *width* e *height* não terão efeito em alguns elementos
- margens e *padding*s **superior** e **inferior** de alguns elementos **não são** respeitados

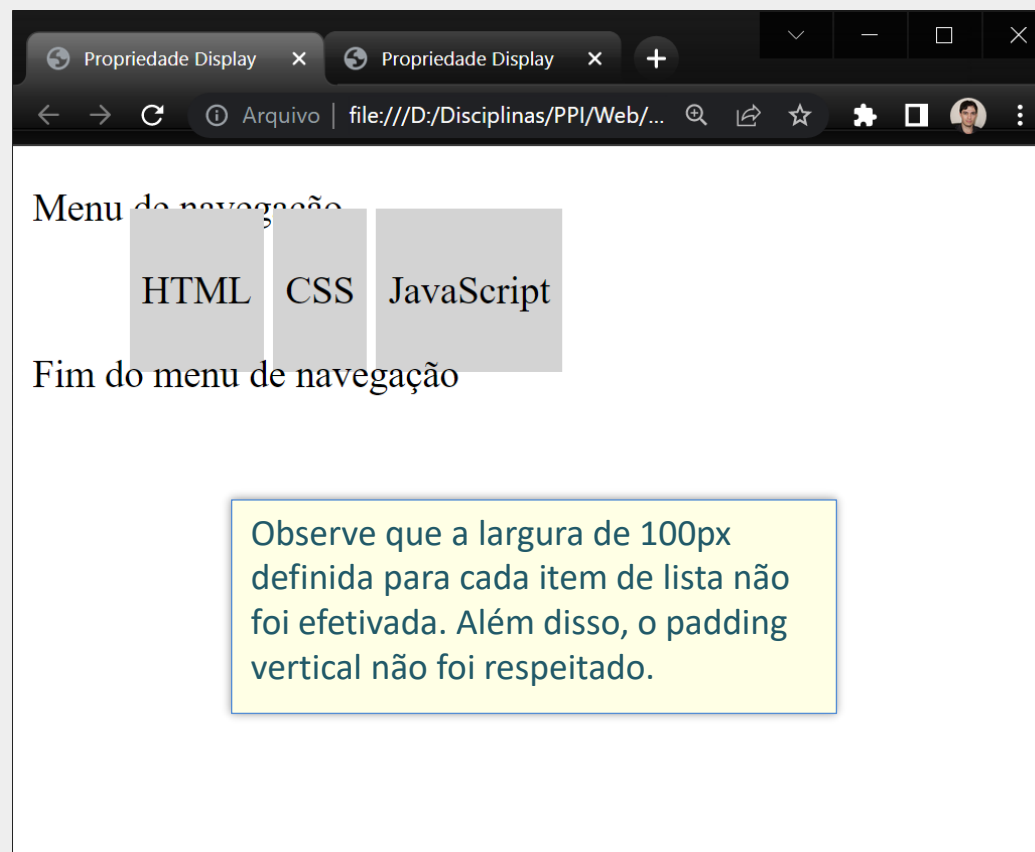
`display: inline-block`

- exibição em nível de linha
- possibilidade de usar *width* e *height*
- margens e *padding*s **superiores** e **inferiores** **respeitados**

Item de Lista com **display: inline**

```
<style>
  nav li {
    background-color: lightgray;
    padding: 25px 5px;
    width: 100px;
    display: inline;
  }
</style>
</head>

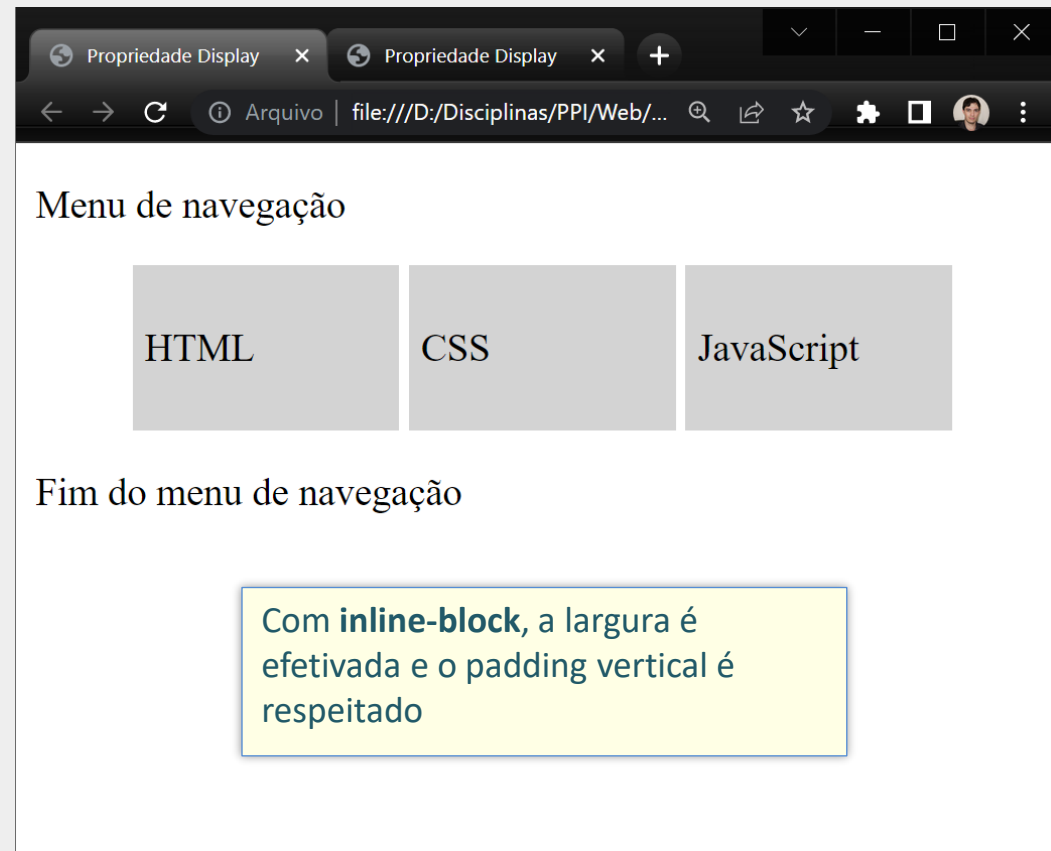
<body>
  <nav>
    <p>Menu de navegação</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
    </ul>
    <p>Fim do menu de navegação</p>
  </nav>
</body>
```



Item de Lista com **display: inline-block**

```
<style>
  nav li {
    background-color: lightgray;
    padding: 25px 5px;
    width: 100px;
    display: inline-block;
  }
</style>
</head>

<body>
  <nav>
    <p>Menu de navegação</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
    </ul>
    <p>Fim do menu de navegação</p>
  </nav>
</body>
```



Propriedade *visibility*

- Mostra ou oculta um elemento **sem alterar o layout**
- Valores possíveis
 - **visible** elemento visível
 - **hidden** elemento oculto, mas espaço ocupado permanece
 - **collapse** comumente usado para ocultar linhas e/ou colunas de tabelas. O espaço é removido sem alterar os tamanhos das linhas e colunas.

Propriedade *float*

- Posiciona o elemento no lado esquerdo ou direito de seu container
- Permite que texto e elementos *inline* se posicionem à sua volta
- Alguns valores
 - **left** elemento "flutua" no **lado esquerdo** do *container*
 - **right** elemento "flutua" no **lado direito** do *container*
 - **none** elemento não "flutua"

Propriedade *float* - Exemplo

```
<style>
  .esquerdo {
    background-color: #ddd;
    width: 15%;
    height: 300px;
    margin: 5px;
    float: left;
  }
  .direito {
    background-color: #ddd;
    width: 25%;
    height: 300px;
    float: right;
  }
</style>
</head>
<body>
  <aside class="esquerdo">Esquerdo</aside>
  <aside class="esquerdo">Esquerdo</aside>
  <aside class="direito">Direito</aside>
  <main><p>Lorem ipsum dolor...</p></main>
</body>
```

Esquerdo

Esquerdo

Lorem ipsum dolor
sit amet consectetur,
adipisicing elit. Illum
deserunt natus earum
facilis quas ipsa
numquam id omnis
eius iusto.

Direito

Propriedade *overflow*

- Trata o conteúdo do elemento quando ele extrapola a borda
- Propriedade abreviada de `overflow-x` e `overflow-y`
- Alguns valores
 - `visible` conteúdo sempre visível, ainda que fora dos limites (*default*)
 - `hidden` conteúdo cortado, se necessário, para caber no espaço
 - `scroll` barras de rolagens são sempre apresentadas
 - `auto` barras de rolagens apenas se necessário

Propriedade *overflow* - Exemplo

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

`overflow: visible;`

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido pela Faculdade de Computação

`overflow: hidden;`

Sistemas de Informação

Curso de graduação oferecido pela Faculdade de Computação da Universidade Federal de Uberlândia. Possui duração de 4 anos.

Curso de graduação oferecido

`overflow: auto;`

Propriedade *position*

- Define como o elemento é **posicionado** na página
- Normalmente utilizada em conjunto com `top`, `left`, `right` e `bottom`
- Valores possíveis
 - `static`
 - `relative`
 - `absolute`
 - `fixed`
 - `sticky`
- O elemento é dito **posicionado** quando *position* tem valor diferente de `static`

Propriedade *position*

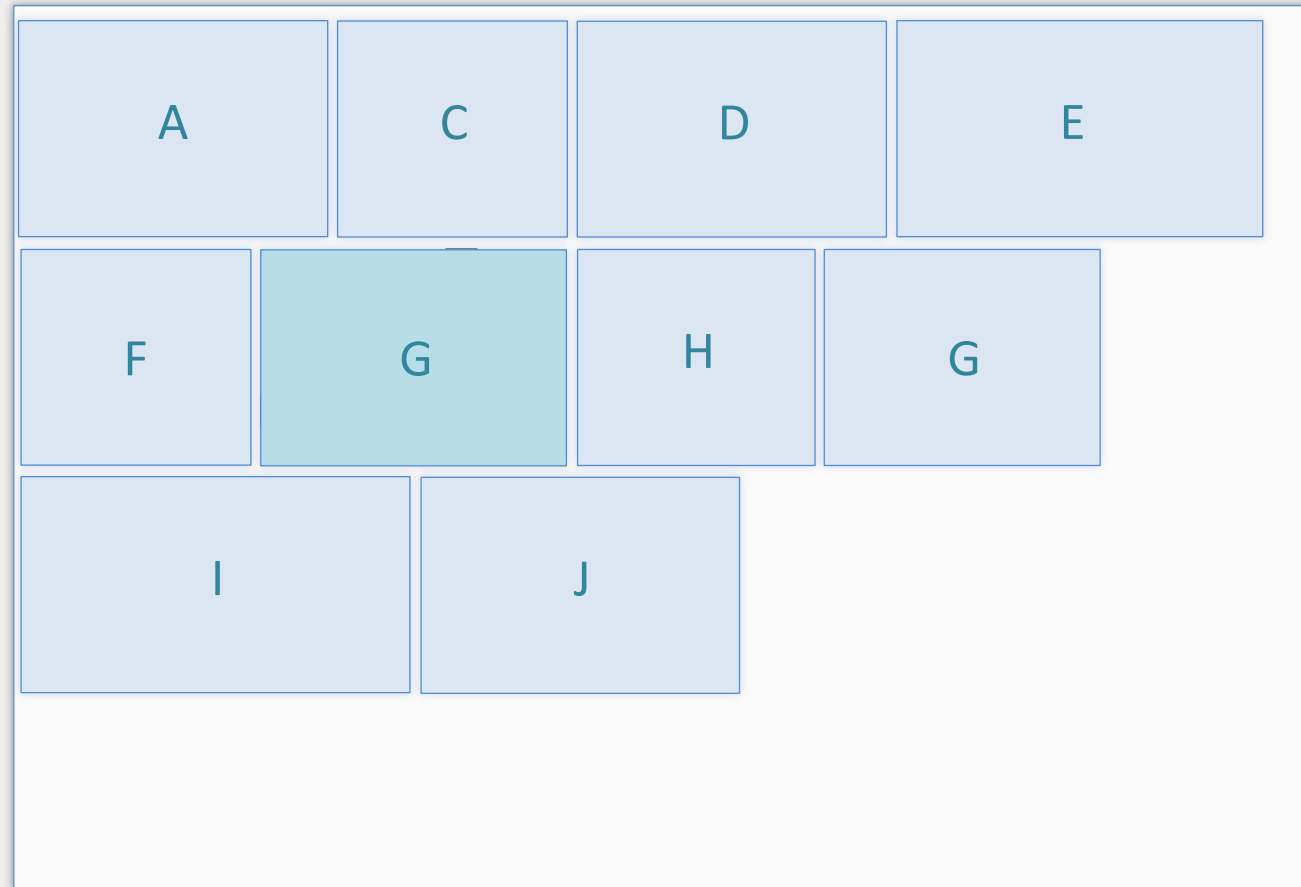
position: *static*;

- Valor padrão
- Elemento posicionado de acordo com fluxo normal do documento

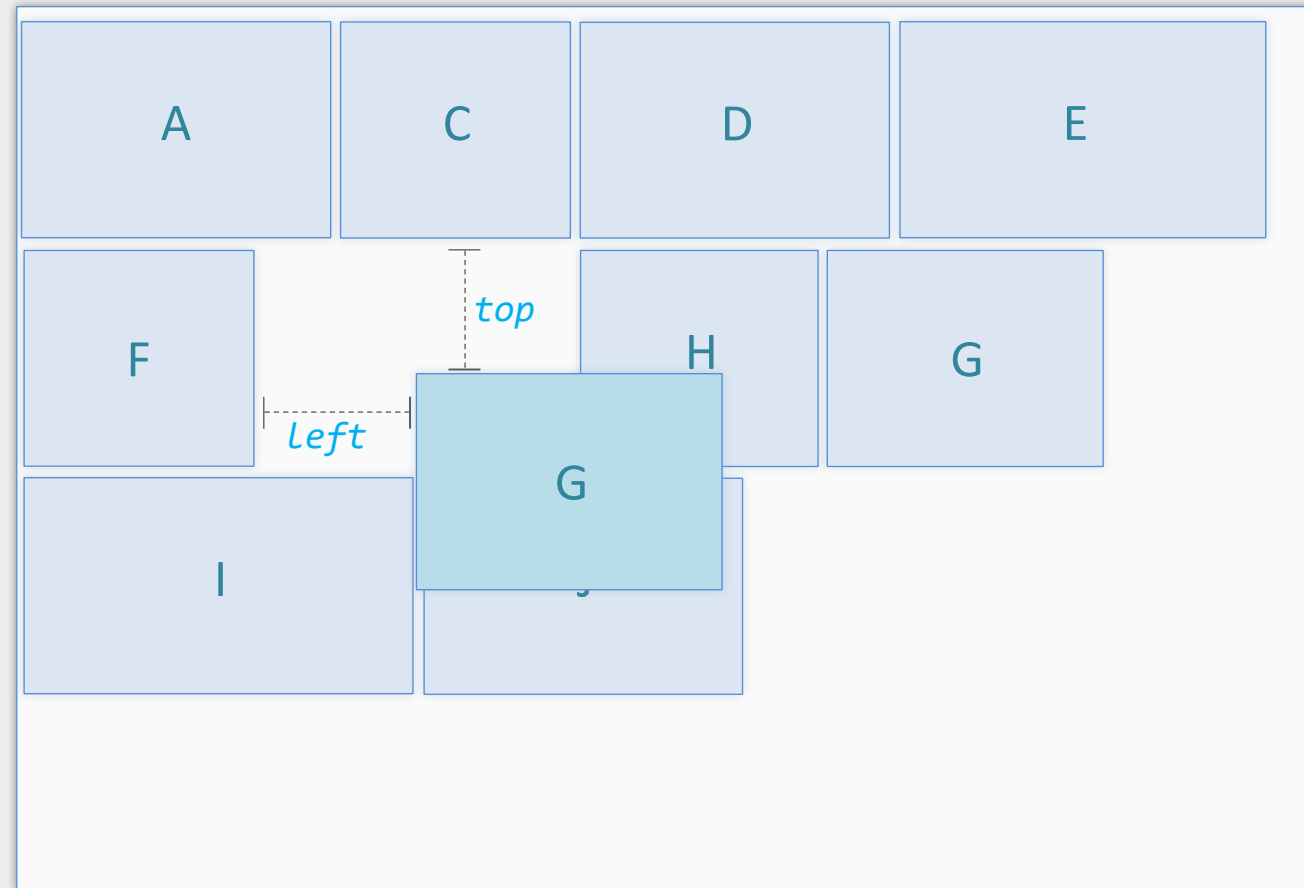
position: *relative*;

- Primeiramente posicionado de acordo com o fluxo normal
- Em seguida, é deslocado da sua posição com *top*, *left*, *right* e *bottom*
- O deslocamento não afeta a posição dos demais elementos
 - Layout geral é o mesmo do posicionamento *static*

Propriedade *position: relative*



Propriedade *position*: *relative*



Propriedade ***position*** - Valores

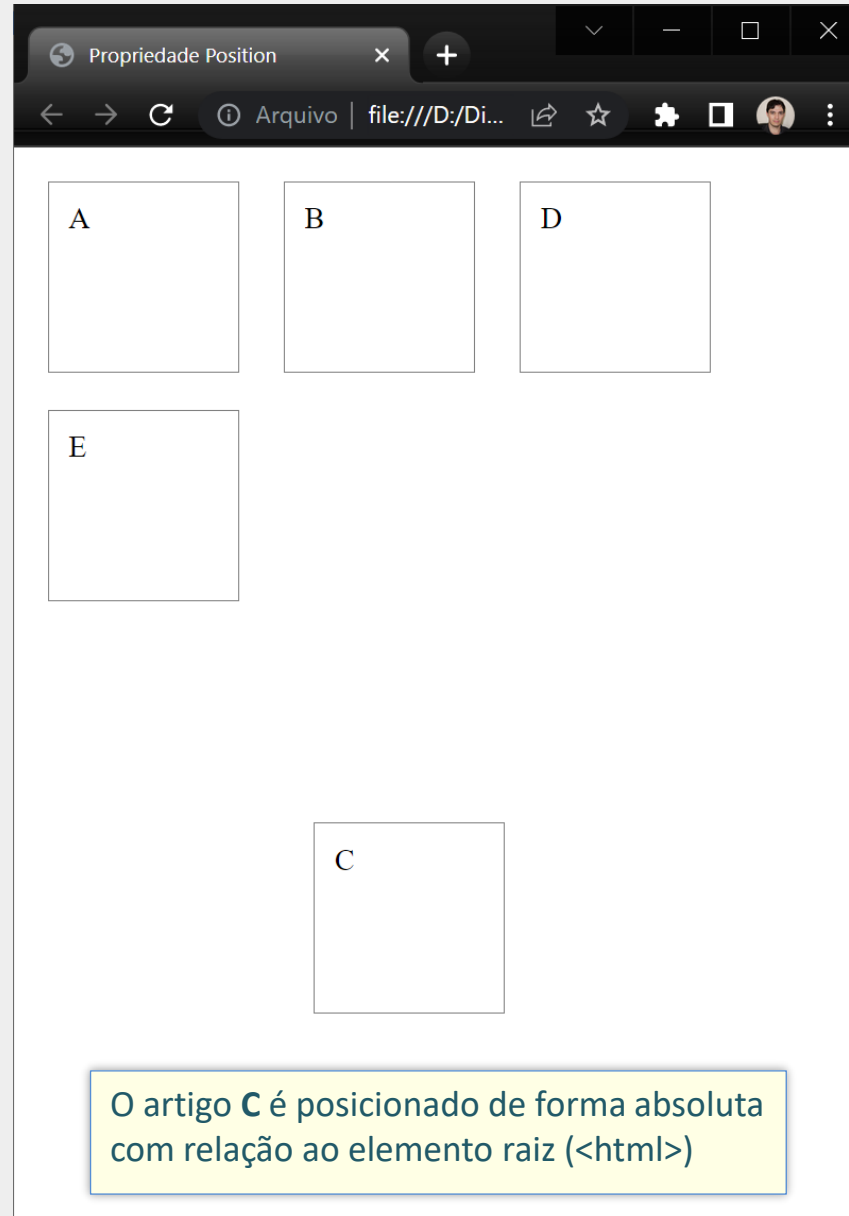
position: **absolute**;

- Elemento removido do fluxo normal
- Nenhum espaço no layout é criado para o elemento
- Pode ser utilizado para centralizar um elemento de bloco
- Posicionamento com *top*, *left*, *right* e *bottom* relativo ao:
 - Ancestral mais próximo **posicionado**, se houver
 - Caso contrário, com relação ao container inicial (elemento <html>)

position: absolute

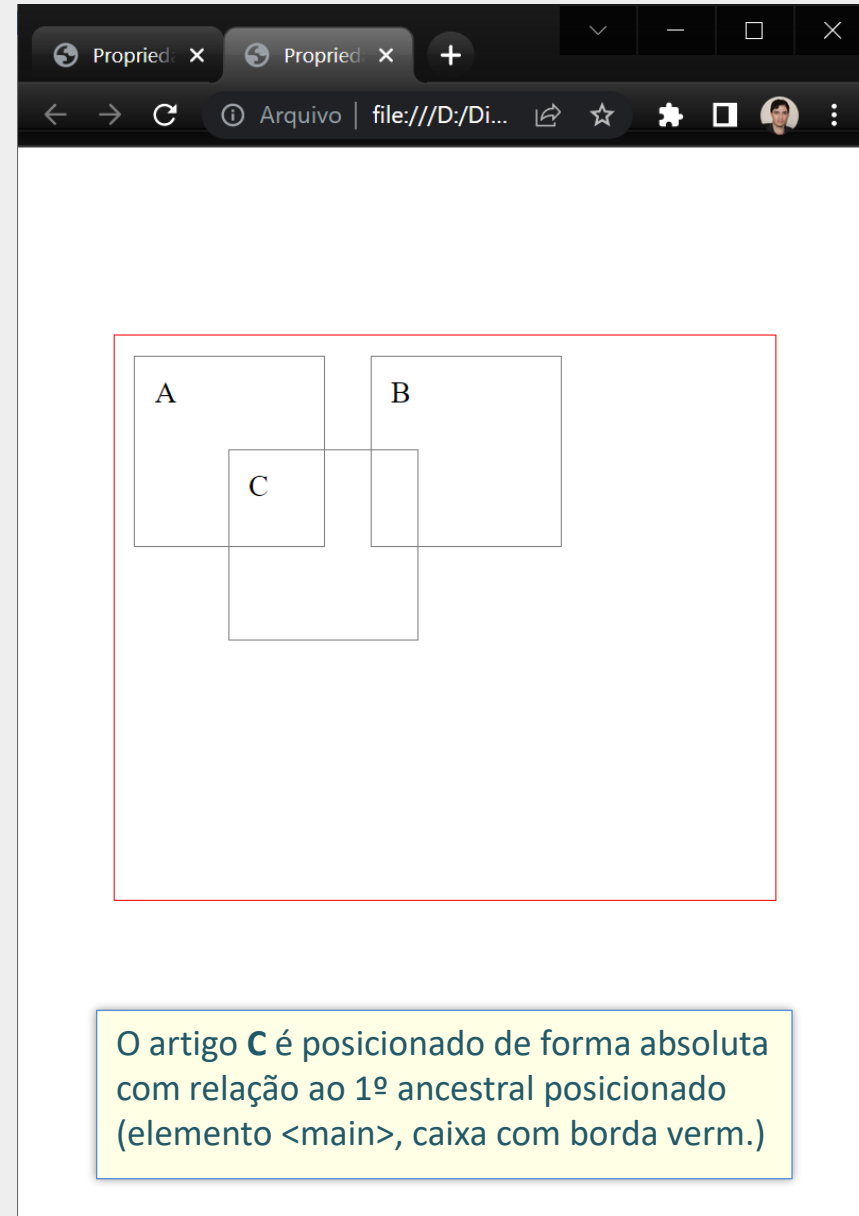
```
<style>
  article {
    margin: 10px; padding: 10px;
    border: 1px solid gray;
    width: 80px; height: 80px;
    display: inline-block;
  }

  .posAbs {
    position: absolute;
    left: 150px;
    top: 350px;
  }
</style>
</head>
<body>
  <main>
    <article>A</article>
    <article>B</article>
    <article class="posAbs">C</article>
    <article>D</article>
    <article>E</article>
  </main>
</body>
```



position: absolute

```
<style>
  main {
    width: 80%; height: 300px;
    margin: 100px auto;
    border: 1px solid red;
    position: relative;
  }
  article {
    margin: 10px; padding: 10px;
    border: 1px solid gray;
    width: 80px; height: 80px;
    display: inline-block;
  }
  .posAbs {
    position: absolute;
    left: 50px;
    top: 50px;
  }
</style>
</head>
<body>
  <main>
    <article>A</article>
    <article>B</article>
    <article class="posAbs">C</article>
  </main>
</body>
```



Propriedade ***position*** - Valores

position: **fixed**;

- Elemento posicionado com relação à *viewport* (janela do navegador)
- Posição **não se altera com a rolagem** da página
- Posicionamento com as propriedades *top*, *left*, *right* e *bottom*
- Elemento **removido do fluxo normal** - nenhum espaço é criado no layout
- Quando impresso, aparecerá na mesma posição em todas as páginas

position: **sticky**;

- Elemento posicionado de acordo com fluxo normal do documento
- **Ocupa espaço no layout**
- Elemento "gruda" no primeiro ancestral com mecanismo de rolagem

Propriedade *transform*

- Permite mover, rotacionar, torcer e dimensionar um elemento
- Exemplos:

<code>transform: translateX(50px);</code>	move o elemento 50 pixels em x (hor.)
<code>transform: translateY(50px);</code>	move o elemento 50 pixels em y (vert.)
<code>transform: translate(30px,10px);</code>	move o elemento 30 pixels em x e 10 em y
<code>transform: rotate(45deg);</code>	rotaciona o elemento em 45 graus
<code>transform: scale(2);</code>	dobra o tamanho nas duas direções

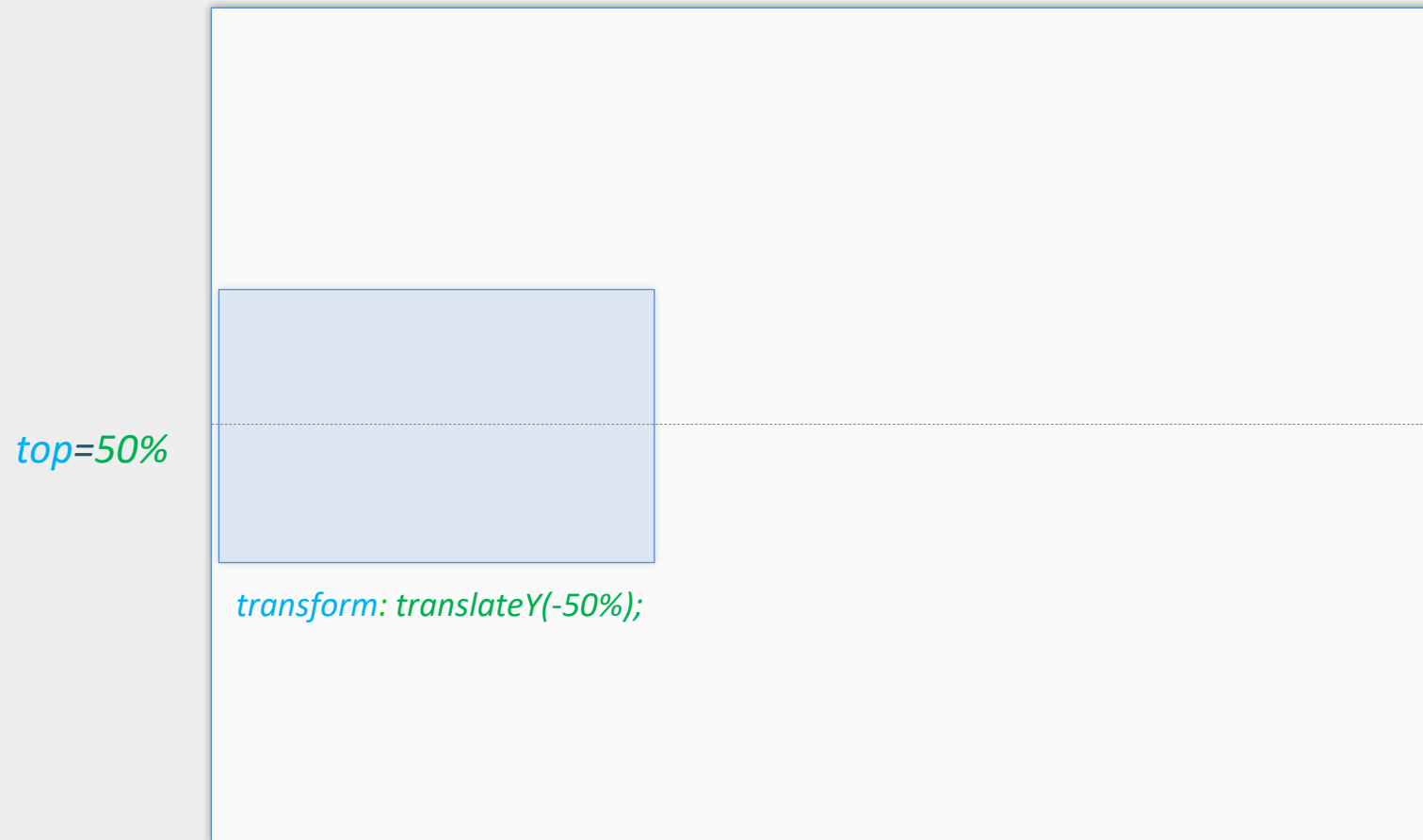
Há muitas outras formas de uso

Centralizando Bloco na Vertical com Pos. Absoluto

top=50%

transform: translateY(-50%);

Centralizando Bloco na Vertical com Pos. Absoluto

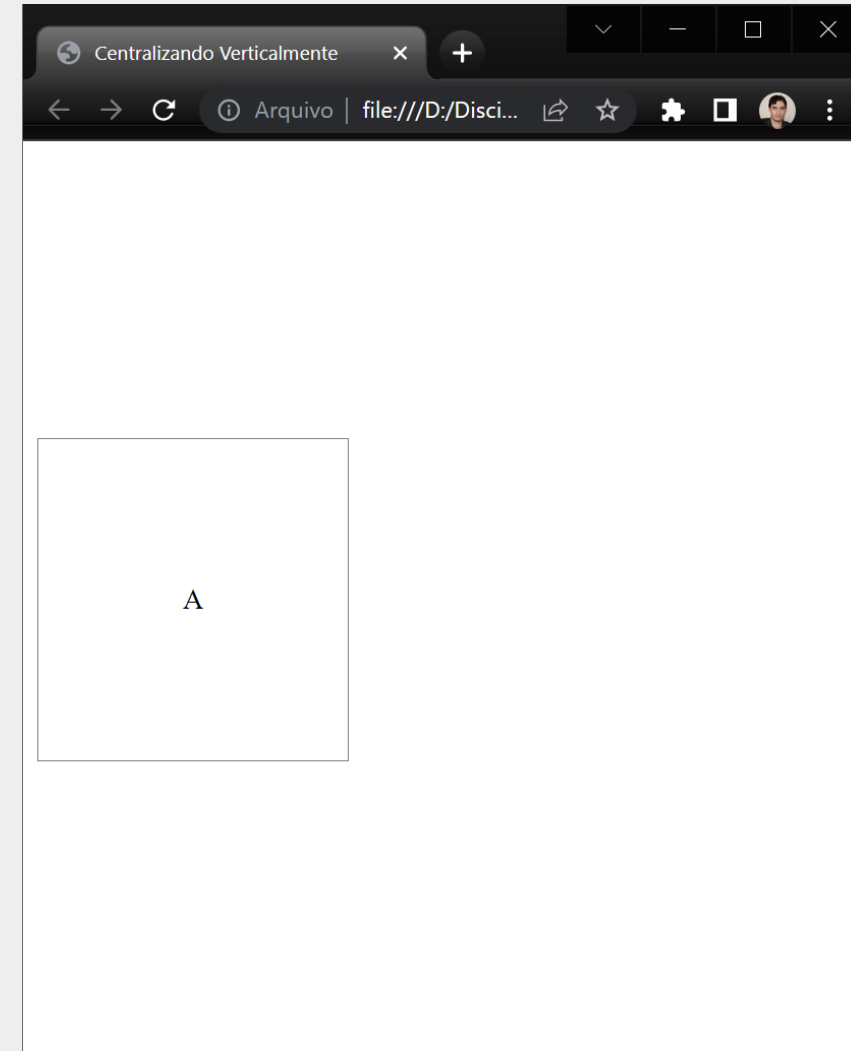


Centralizando Bloco na Vertical com Pos. Absoluto

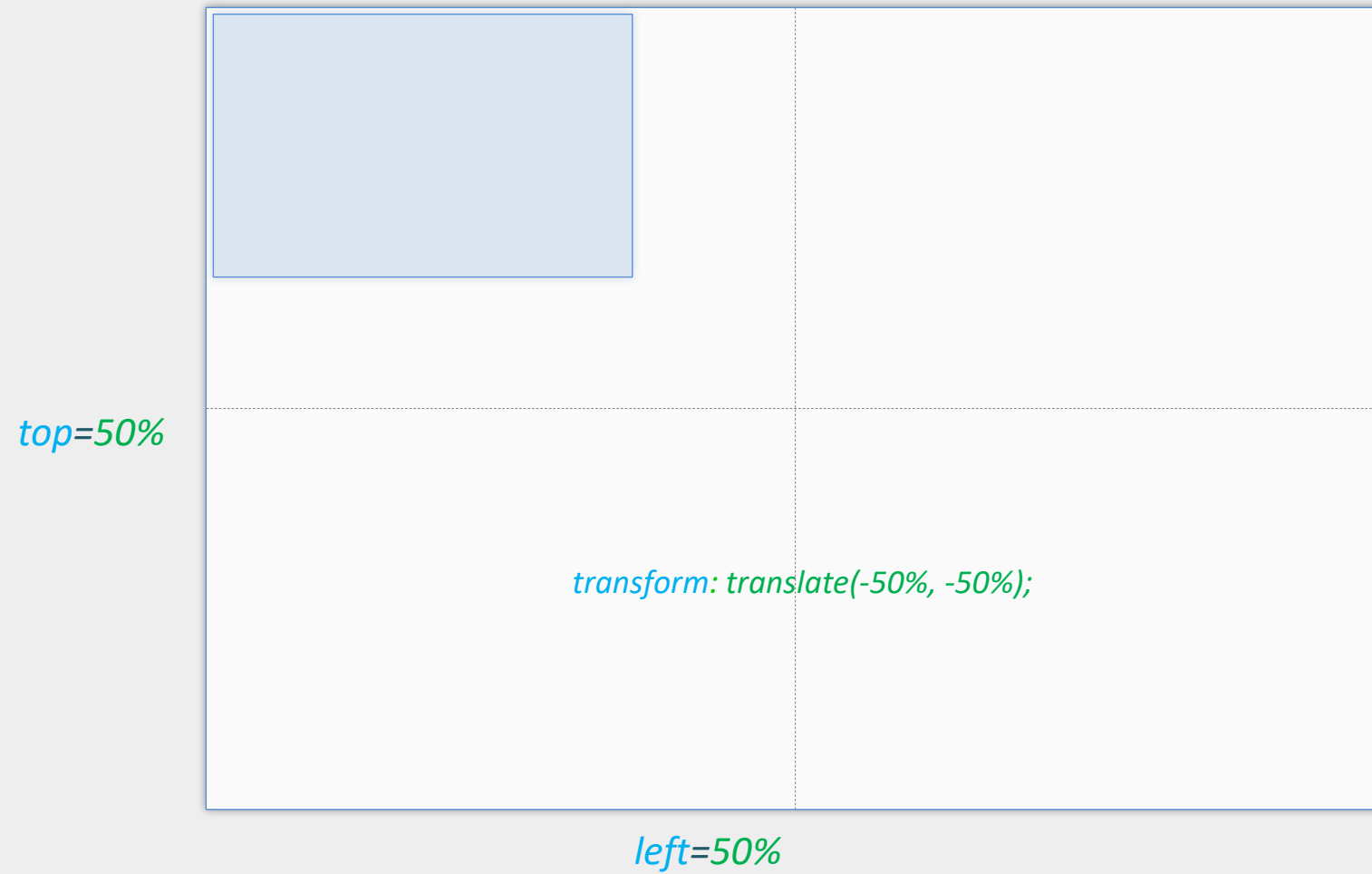
```
<style>
  main {
    padding: 80px;
    border: 1px solid gray;

    position: absolute;
    top: 50%;
    transform: translateY(-50%);
  }
</style>
</head>

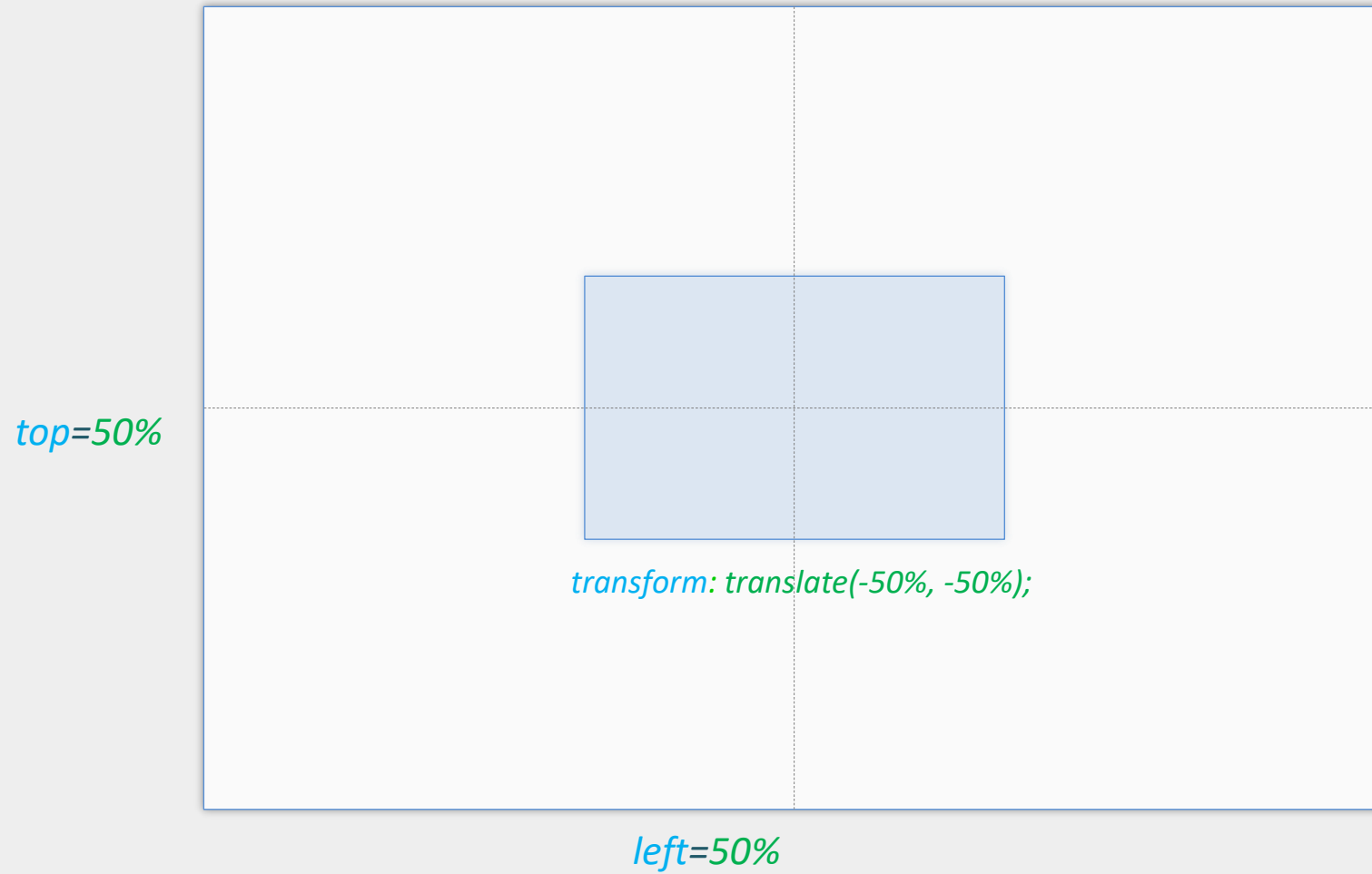
<body>
  <main>A</main>
</body>
```



Centralizando na Vertical e Horizontal com Pos. Absoluto



Centralizando na Vertical e Horizontal com Pos. Absoluto

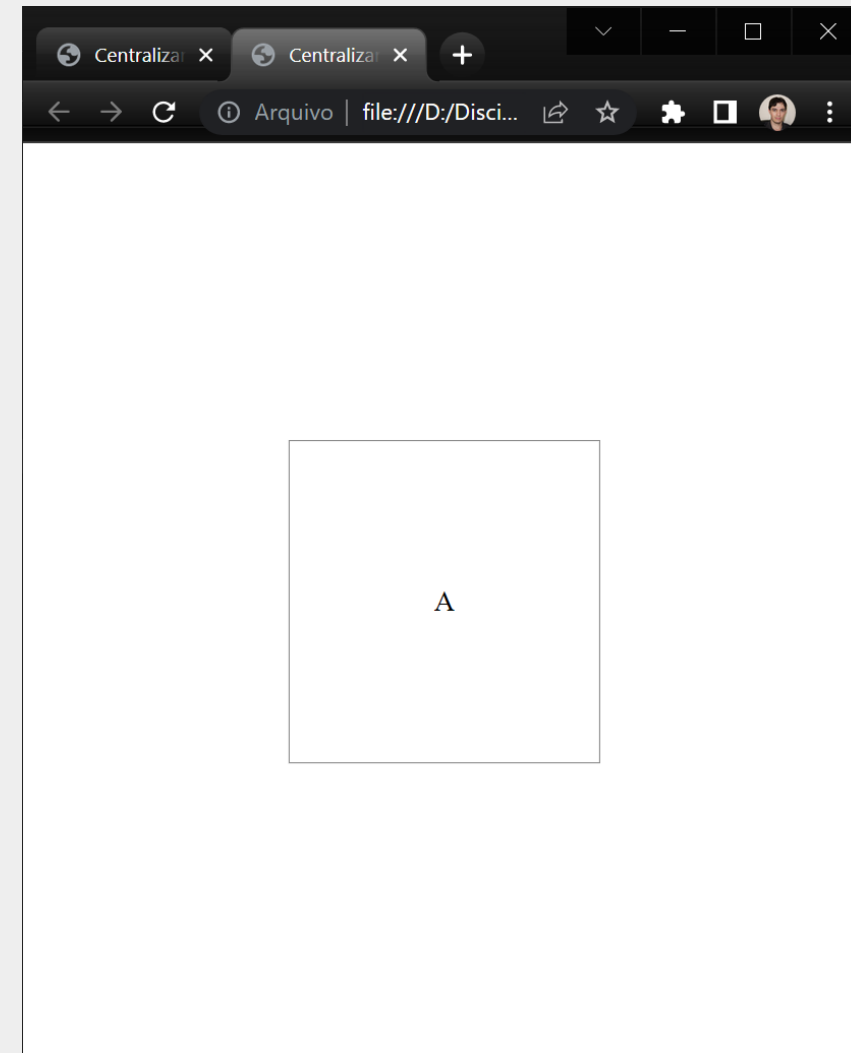


Centralizando na Vertical e Horizontal com Pos. Absoluto

```
<style>
  main {
    padding: 80px;
    border: 1px solid gray;

    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%);
  }
</style>
</head>

<body>
  <main>A</main>
</body>
```



Centralizando na Vertical e Horizontal com Pos. Absoluto

```
main {  
  width: 50%;  
  background-color: #eee;  
  border: 0.5px solid gray;  
  margin: 0;  
  padding: 2% 4%;  
  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```



Centralizando Bloco na Horizontal com **width** e **margin**

- Define-se uma largura com *width*
- Coloca-se as margens laterais no automático (*auto*)

```
body {  
    width: 60%;  
    margin: 0 auto;  
}
```

Para centralizar o texto dentro de um elemento, utilize:
`text-align: center;`

Pseudo-Classes

- Indica um estado em particular do elemento selecionado
- Palavra-chave adicionada a um seletor, precedida por dois-pontos

```
a:visited {  
    color: gray;  
}
```

*Os links **que já foram visitados** aparecerão na cor cinza*

```
input:invalid {  
    border-color: red;  
}
```

*Os campos do tipo input aparecerão com borda vermelha **quando o conteúdo é inválido** (por ex., um campo de e-mail faltando o @)*

Exemplos de Pseudo-Classes

Pseudo-Classe	Descrição	Exemplos
:link	Estilo inicial do link	<i>a:Link</i> {color: blue;}
:hover	Usuário interage com elem. sem ativar Ex.: ponteiro do mouse sobre o elemento	<i>a:hover, button:hover</i> {...}
:active	Estado ativo do elemento Ex.: botão do mouse pres. sobre o elemento	<i>a:active, p:active</i> {...}

Exemplos de Pseudo-Classes

Pseudo-Classe	Descrição	Exemplos
:valid	Campo de form. com conteúdo válido	<i>input:valid</i> {...}
:invalid	Campo com conteúdo inválido	<i>input:invalid</i> {...}
:checked	<i>checkbox</i> , <i>radio</i> ou <i>option</i> selecionado	<i>radio:checked</i> {...}
:focus	Elemento recebendo foco	<i>input:focus</i> {...}
:first-child	Primeiro filho do elemento pai	<i>li:first-child</i> {color: blue;}
:last-child	Último filho do elemento pai	<i>li:last-child</i> {color: blue;}

Pseudo-Elementos

- Permite selecionar **uma parte específica** de um elemento
- Palavra-chave adicionada a um seletor
- Precedida por **::**

```
p::first-line {  
    text-transform: uppercase;  
}
```

A primeira linha de cada parágrafo será apresentada com letras maiúsculas

```
p::selected {  
    color: green;  
    background-color: black;  
}
```

O texto que o usuário selecionar nos parágrafos aparecerá na cor verde com fundo preto

OBS: Para cada pseudo-elemento há uma lista restrita de propriedades que podem ser utilizadas.

Outros Pseudo-Elementos

```
p::after {  
  content: 'exemplo after';  
}
```

Inserir o texto 'exemplo after' como um pseudo-elemento **depois do conteúdo** do parágrafo

```
p::before {  
  content: 'exemplo before';  
}
```

Inserir o texto 'exemplo before' como um pseudo-elemento **antes do conteúdo** do parágrafo

```
input::placeholder {  
  color: red;  
}
```

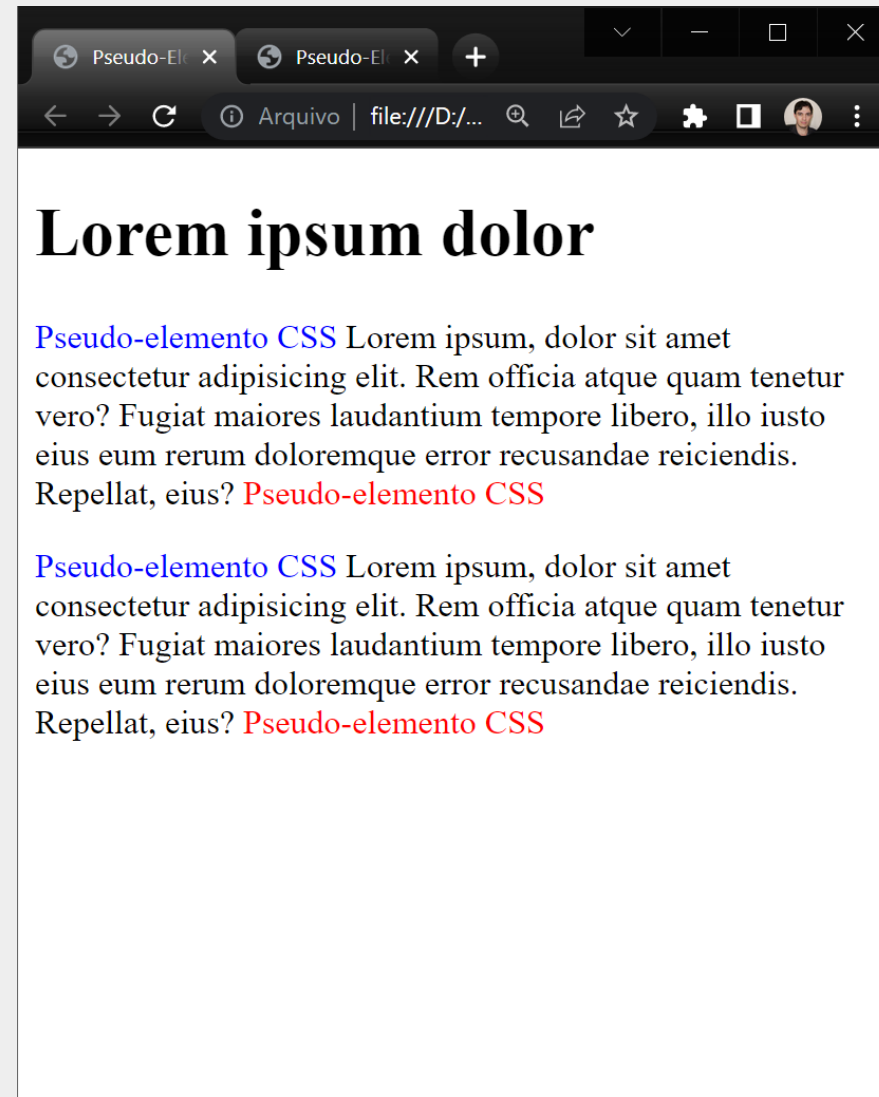
Altera a cor dos textos de **placeholder** dos campos *input*

OBS: Os pseudo-elementos `::after` e `::before` **não podem** ser utilizados em elementos sem conteúdo como `` ou `
`

Pseudo-Elementos - Exemplo

```
<style>
  p::after {
    content: ' Pseudo-elemento CSS ';
    color: ■ red;
  }

  p::before {
    content: ' Pseudo-elemento CSS ';
    color: ■ blue;
  }
</style>
</head>
<body>
  <h1>Lorem ipsum dolor</h1>
  <p>Lorem ipsum, dolor sit amet consectetur
    laudantium tempore libero, illo iust
  <p>Lorem ipsum, dolor sit amet consectetur
    laudantium tempore libero, illo iust
```



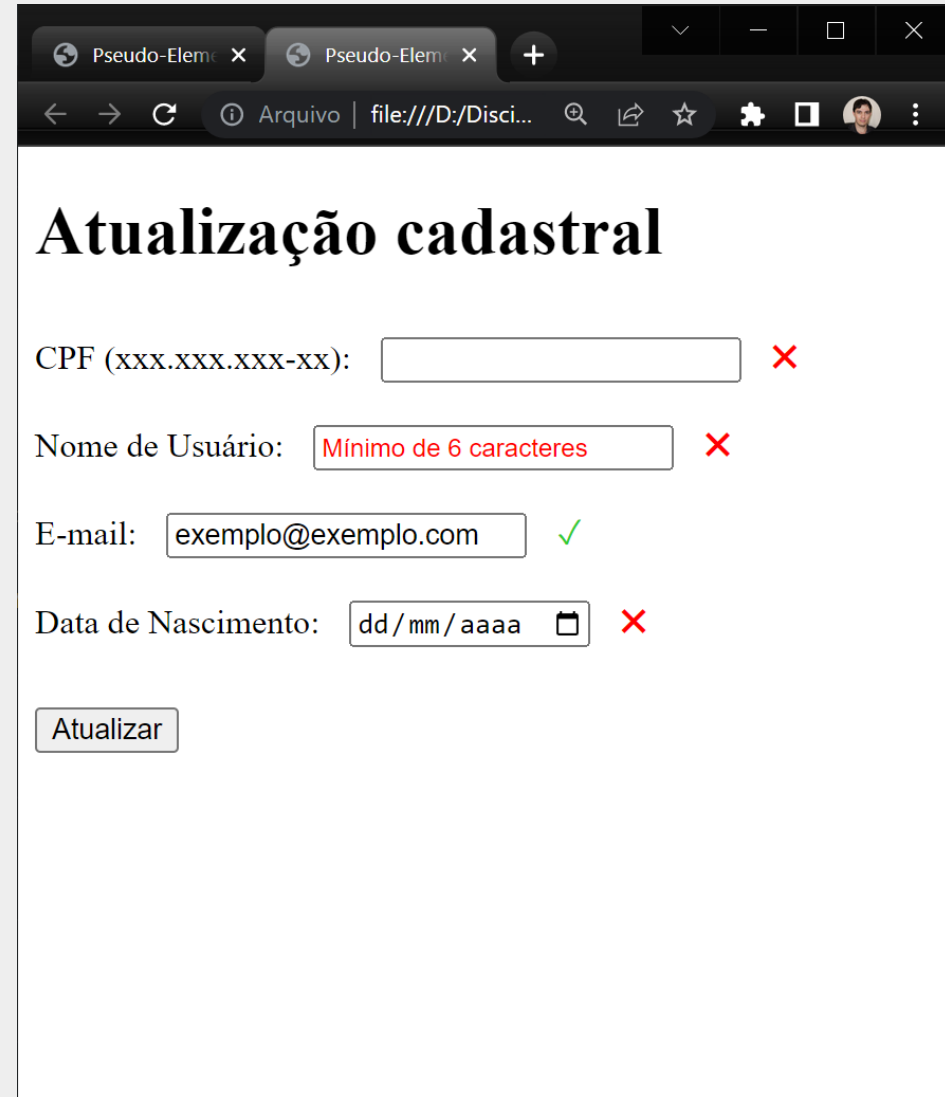
Pseudo-Elementos - Exemplo

```
input:invalid + span::after {
  content: '✗';
  color: red;
}

input:valid + span::after {
  content: '✓';
  color: limegreen;
}

</style>
</head>

<body>
  <form action="action.php" method="post">
    <h1>Atualização cadastral</h1>
    <div>
      <label for="cpf">CPF (xxx.xxx.xxx-xx):</label>
      <input type="text" id="cpf" name="cpf"
        pattern="\d{3}\.\d{3}\.\d{3}-\d{2}">
      <span></span>
    </div>
```



The screenshot shows a web browser window with the title "Atualização cadastral". The form contains the following fields and feedback:

- CPF (xxx.xxx.xxx-xx): An empty text input field with a red "✗" icon to its right, indicating it is invalid.
- Nome de Usuário: A text input field containing "Mínimo de 6 caracteres" with a red "✗" icon to its right, indicating it is invalid.
- E-mail: A text input field containing "exemplo@exemplo.com" with a green "✓" icon to its right, indicating it is valid.
- Data de Nascimento: A date input field with the placeholder "dd/mm/aaaa" and a calendar icon, with a red "✗" icon to its right, indicating it is invalid.
- Atualizar: A button at the bottom of the form.

Observação sobre Pseudo-Elementos

- A notação com 4 pontos (::) foi introduzida na CSS3
- :: é a forma atualmente recomendada para *pseudo-elementos*
- Diferencia **pseudo-elementos** de **pseudo-classes**
- Entretanto, na CSS2 utilizava-se apenas dois pontos (:)
 - Navegadores ainda suportam essa forma

Cascade, Especificidade e Herança

- Em caso de regras conflitantes, qual regra será aplicada?
- Problemas comuns:
 - Regras CSS sem efeito
 - Resultado muito diferente do esperado
- Nessas situações é fundamental conhecer como o navegador decide pelas regras
 - Ordem no código
 - Especificidade
 - Herança

Ordem das Regras

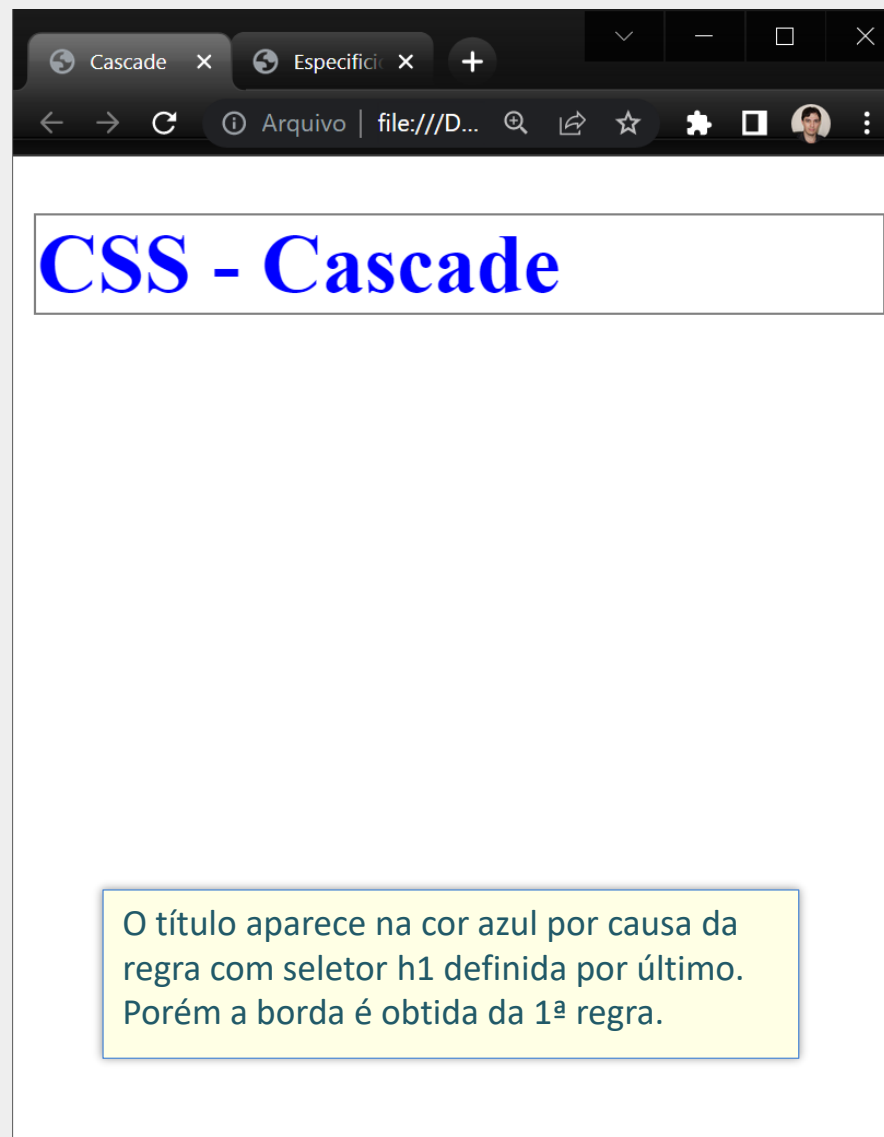
- Caso haja declarações CSS repetidas
- Caso sejam do mesmo tipo de seletor
- Prevalecerá a declaração definida por último

Ordem das Regras - Cascade

```
<style>
  h1 {
    color:  red;
    border: 1px solid  gray;
  }

  h1 {
    color:  blue;
  }
</style>
</head>

<body>
  <h1>CSS - Cascade</h1>
</body>
```



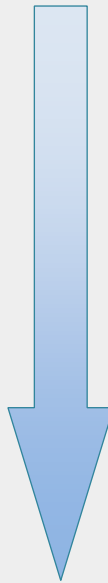
Especificidade

Seletor de Elemento

Seletor de Classe

Seletor de ID

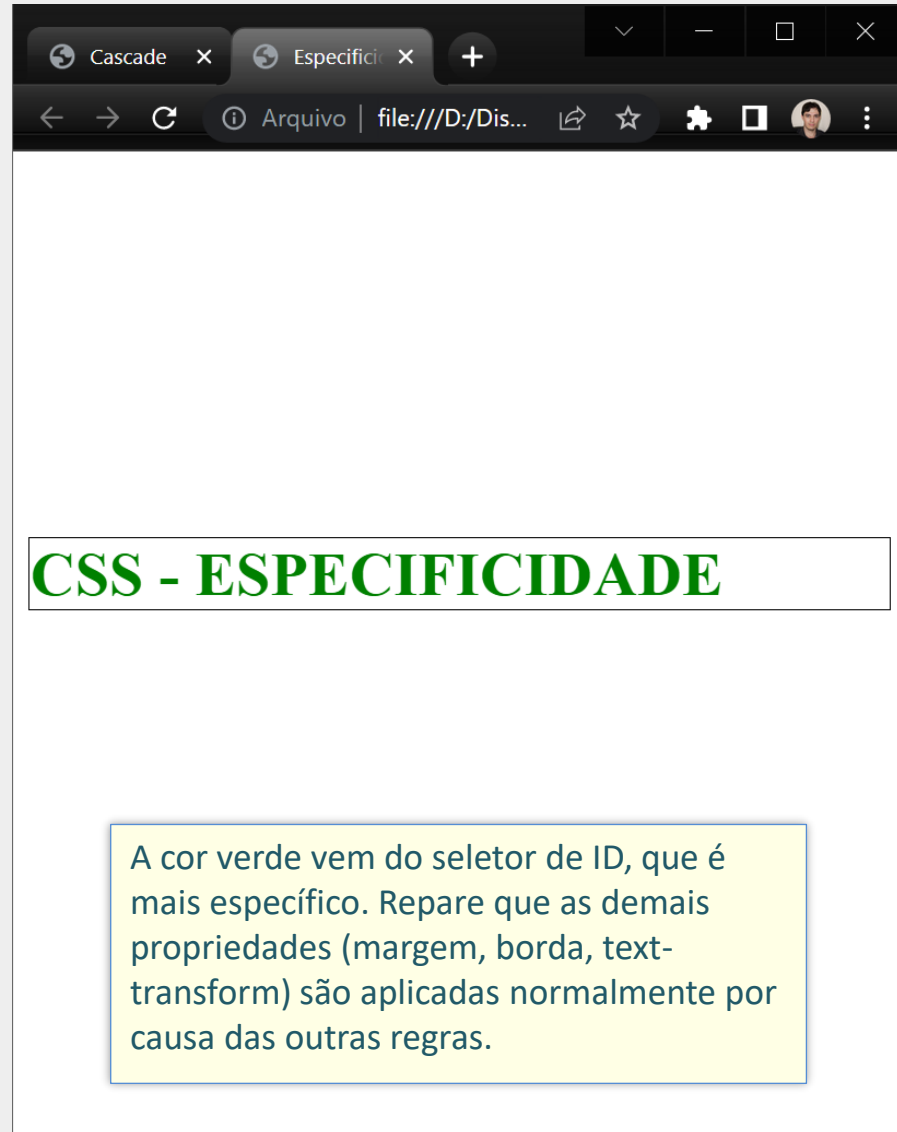
Código Inline



Maior especificidade
Maior peso da declaração CSS
Mais provável que sobrescreva as anteriores

Especificidade das Regras

```
<style>
  h1 {
    color: red;
    text-transform: uppercase;
  }
  #tituloVerde {
    color: green;
    border: 1px solid black;
  }
  .tituloAzul {
    color: blue;
    margin-top: 200px;
  }
</style>
</head>
<body>
  <h1 id="tituloVerde" class="tituloAzul">
    CSS - Especificidade
  </h1>
</body>
```



Herança

- Algumas propriedades herdam os valores do elemento pai
 - Ex.: `font-family`, `color` e `text-align`
 - Ex.: um `<p>` dentro de um `<div>` herda a fonte definida para o `<div>`
- Outras propriedades não herdam
 - Ex.: `width`, `margin` e `padding`
 - Ex.: um `<p>` dentro de um `<div>` não herda as margens definidas para o `<div>`
- Definir uma propriedade para o valor `inherit` ativa a herança

Design Responsivo

- Baseado na ideia de que o *website* deve ser bem exibido em todos os dispositivos (desktop, tablet ou smartphone);
- Evita a necessidade de ter diferentes versões do website para diferentes tamanhos de tela;
- Normalmente envolve a utilização da *meta tag viewport*, *media queries*, unidades relativas, etc.;

Meta Tag Viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

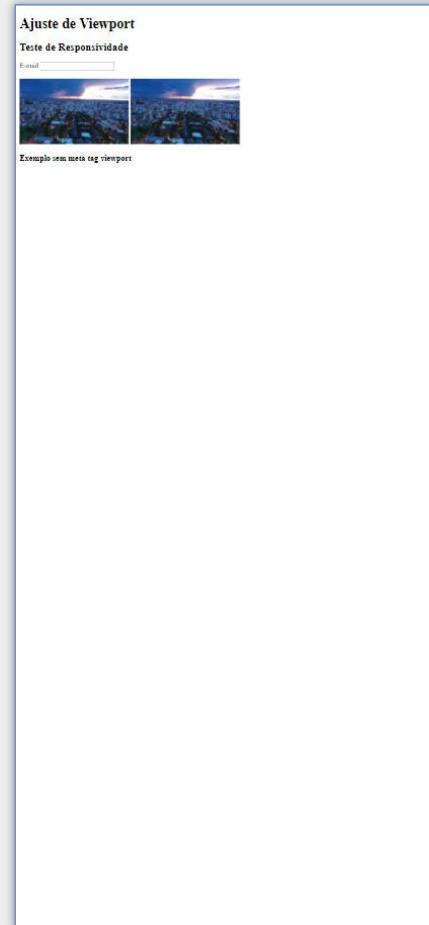
- Viabiliza a responsividade conforme dispositivo/tela
- Página melhor ajustada à tela de smartphones e tablets
- Ativa o fator de escala no dimensionamento em disp. móveis
- *Pixel ratio* do dispositivo é considerado

Utilizando a *Meta Tag Viewport*

```
<head>
  <meta charset="UTF-8">
  <title>Teste de Viewport</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
  <h1>Ajuste de Viewport</h1>
  <h2>Teste de Responsividade</h2>
  <label for="email">E-mail</label>
  <input type="email" id="email">
  <br><br>
  
  
  <h3>Exemplo com meta tag viewport e</h3>
</body>
```

Utilizando a *Meta Tag Viewport*



Página **sem** a *meta tag viewport*
(acessada pelo smartphone)



Página **com** a *meta tag viewport*
(acessada pelo smartphone)

Introdução à **Media Queries**

- Permite ao desenvolvedor testar condições sobre o navegador ou dispositivo do usuário para a aplicação ou não das regras CSS;
- Por exemplo, é possível aplicar estilos CSS apenas quando a tela do dispositivo tenha uma largura mínima ou máxima; ou esteja em determinada posição, como na vertical ou na horizontal;

Introdução à Media Queries

all - para todos os dispositivos
screen - para dispositivos com tela
print - para impressão (ex. modo Ctrl-P)

`@media MediaType AND MediaCondition {`

`/* Código CSS */`

`}`

`(min-width: 400px)`
`(max-width: 900px)`
`(min-width: 400px) and (max-width: 900px)`
`(400px <= width <= 900px)`
`(orientation: portrait)`
`(orientation: landscape)`

Exemplos de expressões (*media condition*)

Introdução à Media Queries

```
body {  
  width: 60%;  
  padding: 2% 0;  
  margin: 0 auto;  
}  
  
@media (max-width: 600px) {  
  body {  
    width: 95%;  
  }  
}
```

O corpo da página ocupará 95% da largura quando a largura da viewport for menor ou igual a 600px.

`max-width` é uma *media feature*. Há outras opções, como `orientation`

Introdução à Media Queries


```
<style>
  body {
    width: 60%;
    padding: 2% 0;
    margin: 0 auto;
    line-height: 2.0;
  }

  @media print and (orientation: portrait) {
    body {
      line-height: 1.1;
      font-size: 10pt;
    }
  }
</style>
```

O espaçamento entre linhas e o tamanho da fonte serão reduzidos quando o documento estiver em modo de impressão na orientação **retrato** (Ctrl-P)

Media Query com Lista de Expressões

- É possível combinar uma lista de media queries separando as expressões com vírgula
- O código CSS será avaliado caso pelo menos uma das expressões seja verdadeira

```
@media screen and (orientation: landscape),  
screen and (min-width: 900px) {  
  body {  
    background-color:  gray;  
  }  
}
```

O cor de fundo será cinza caso a mídia seja do tipo "tela" com orientação "horizontal" ou caso seja do tipo tela com largura mínima de 900px

Validação do Código CSS

- Exibição adequada no navegador não é garantia de código correto
 - O navegador pode ocultar erros e inconsistências
- Código fora da especificação
 - Apresentação inconsistente e imprevisível nos navegadores
- Serviço gratuito oferecido pelo W3C para validação de código CSS
 - jigsaw.w3.org/css-validator/

CSS e Cache do Navegador

- Eventualmente o navegador pode armazenar estilos CSS em memória
- Neste caso, mudanças nas regras CSS podem não ter efeito imediato
- Se preciso, tecle **Ctrl-F5** para forçar o navegador a recarregar os estilos

Referências

- www.w3.org/Style/CSS/
- developer.mozilla.org/en-US/docs/Web/CSS
- www.w3.org/Style/CSS/learning
- **HTML and CSS: Design and Build Websites**, Jon Duckett.