



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

1. Ajustes de *background*, margem, *padding* e bordas
2. Exibição e posicionamento de elementos
3. Centralização de Elementos
4. Pseudo-Classes
5. Pseudo-Elementos
6. Cascade, Especificidade e Herança
7. Introdução ao Design Responsivo
8. Introdução à Media Queries
9. 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

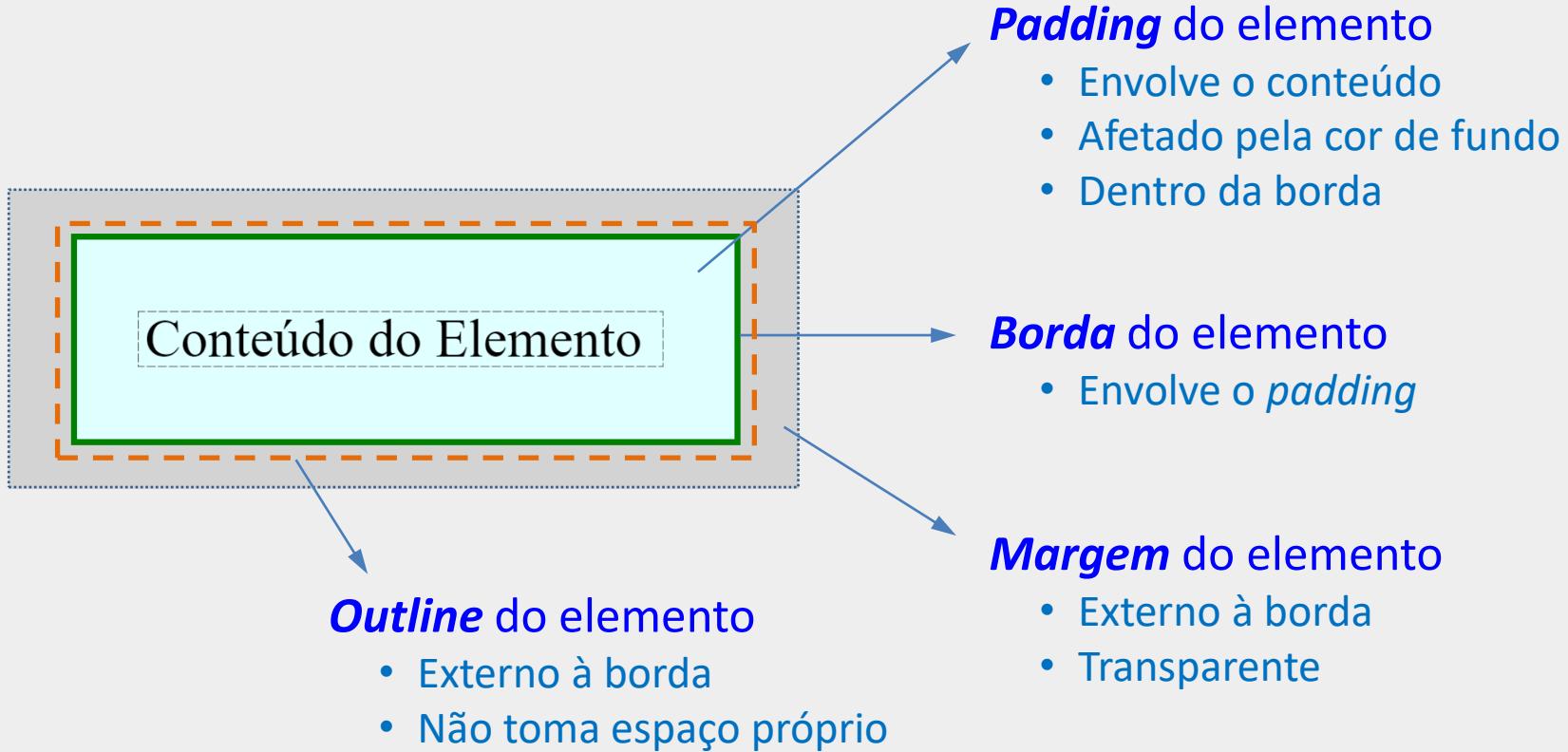
```
<style>
  body {
    background-image: url("images/bg2.jpg");
    background-repeat: no-repeat;
    background-size: cover;
  }
</style>
```

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`

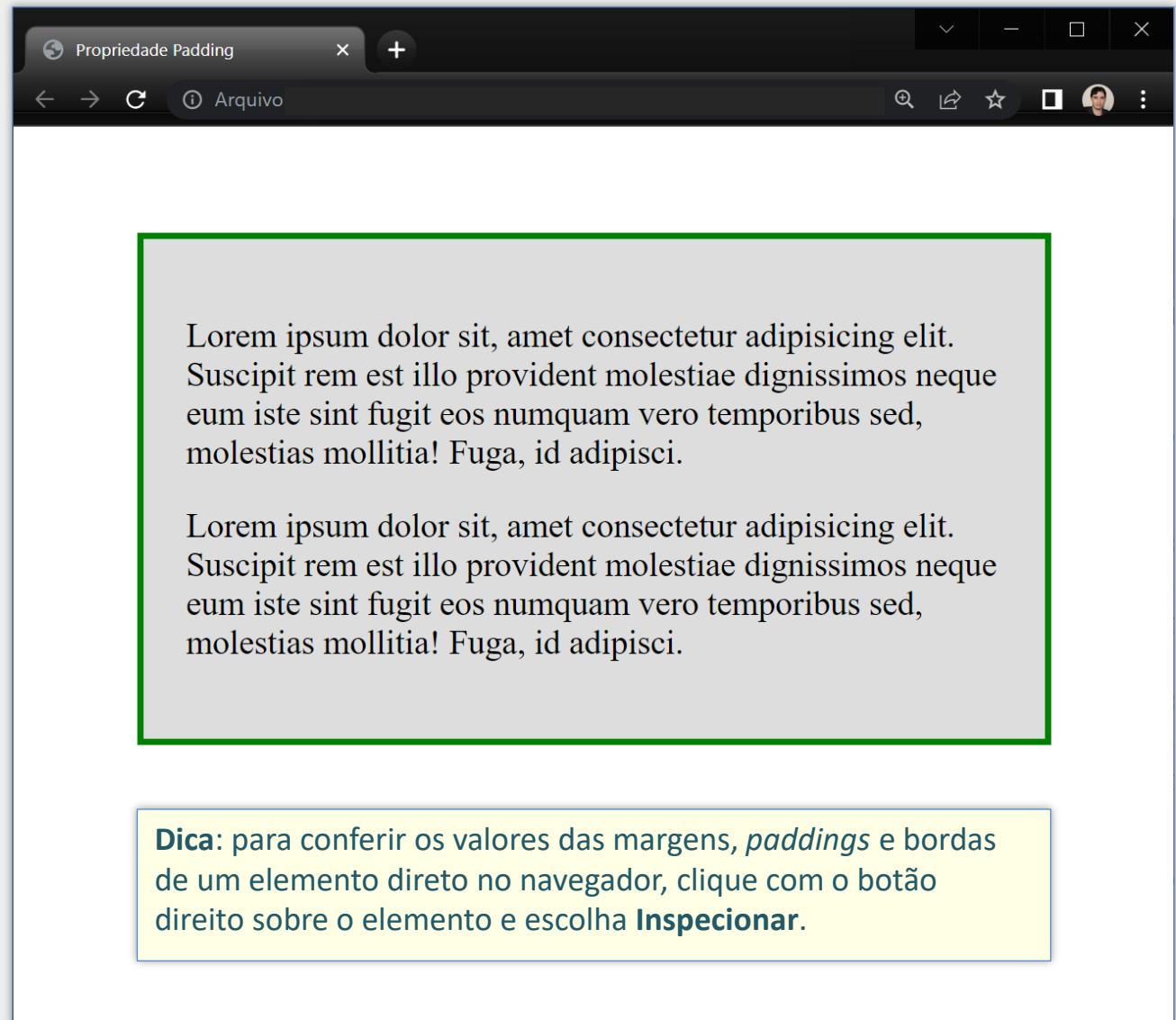
`margin-left`

`margin-right`

Ajuste de Margem e Padding - Exemplo

```
<style>
  body {
    margin: 0;
  }

  main {
    background-color: #ddd;
    margin: 50px;
    padding: 20px;
    border: 3px solid green;
  }
</style>
</head>
<body>
  <main>
    <p>Lorem ipsum dolor sit, amet conse-
      molestiae dignissimos neque eum is-
      mollitia! Fuga, id adipisci.</p>
    <p>Lorem ipsum dolor sit, amet conse-
      molestiae dignissimos neque eum is-
      mollitia! Fuga, id adipisci.</p>
  </main>
</body>
```



Dica: para conferir os valores das margens, *padding*s e bordas de um elemento direto no navegador, clique com o botão direito sobre o elemento e escolha **Inspecionar**.

The screenshot shows a web development environment with two main panels: the Elementos panel and the Element Inspector panel.

Elementos Panel: This panel displays the DOM structure. The `<main>` element is selected and highlighted with a red box. The tree structure includes the `<head>`, `<body>`, and the selected `<main>` node.

Element Inspector Panel: This panel provides detailed information about the selected element (`<main>`). It shows the following properties:

- margin:** 50
- border:** 3
- padding:** 20
- content size:** 454.104x157.714
- inner padding:** 20
- inner border:** 3
- outer border:** 50

The `display: block;` style rule is also visible in the inspector.

The screenshot shows a web browser window with the title "Propriedade Padding". The address bar says "Arquivo |". The page content is an orange box containing two identical paragraphs of placeholder text. A green border surrounds the entire content area. A yellow callout box with a blue arrow points from the text "Dica:" to the top margin of the main content area. The browser's developer tools are open, specifically the "Elementos" (Elements) panel. The "main" element is selected. The "Estilos" (Styles) panel shows a CSS rule with `display: block;`. The "Computado" (Computed) panel shows the visual representation of the element's styling. A large dashed orange box represents the element's bounding box, with dimensions 454.104x157.714 and a 3px border. The margins are 50px on all sides, and the padding is 20px on all sides.

Dica: Ao passar o ponteiro do mouse sobre a região da margem no quadro ao lado, o navegador mostra na página o respectivo espaço ocupado pelo elemento.

Propriedade Padding

Arquivo |

Elementos

<head>...</head>

<body>

... <main> == \$0

html body main

Estilos Computado Layout

Filtro :hov .cls +

```
display: block;
```

margin 50

border 3

padding 20

50 3 20 20 454.104x157.714 20 3 50

20

3

50

Propriedade *margin* - Observação

- Margens entre dois elementos de bloco vizinhos não se somam
- Prevalecerá a maior delas
- Conhecido como sobreposição de margens ou *margin collapsing*

Exemplo de Sobreposição de Margens

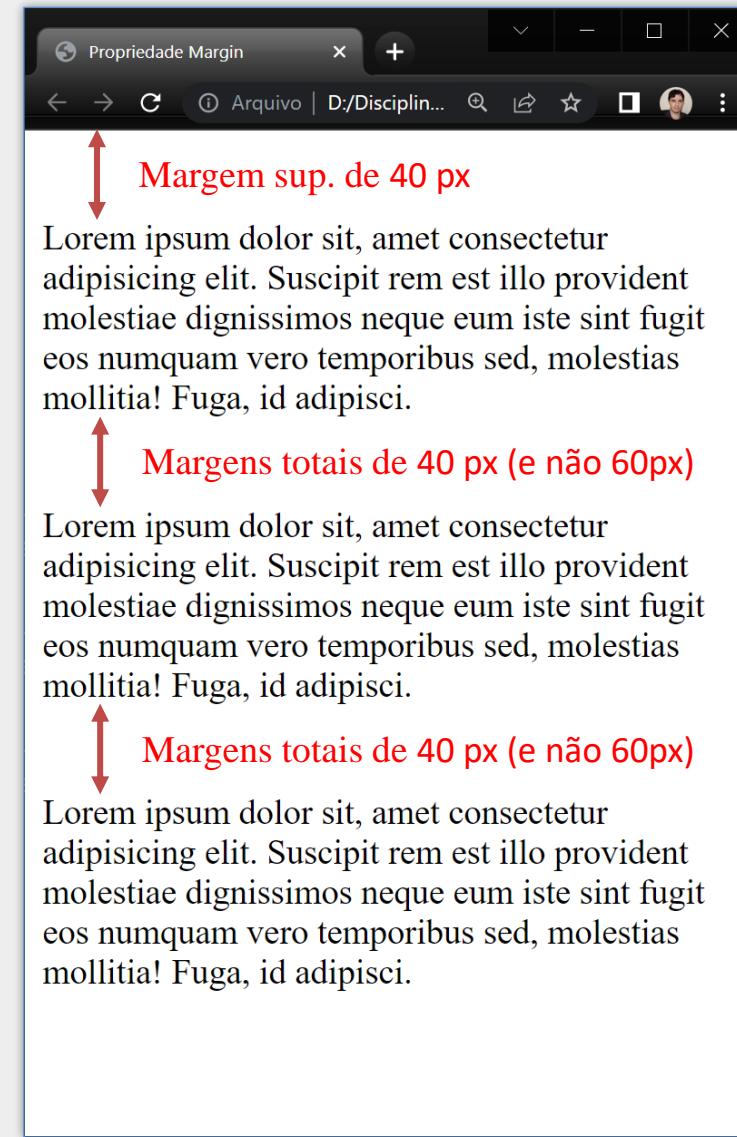
```
<style>
  p {
    margin-top: 40px;
    margin-bottom: 20px;
  }
</style>
</head>

<body>

  <p>Lorem ipsum dolor sit, amet cons
      molestiae dignissimos neque eum i
      mollitia! Fuga, id adipisci.</p>
  <p>Lorem ipsum dolor sit, amet cons
      molestiae dignissimos neque eum i
      mollitia! Fuga, id adipisci.</p>
  <p>Lorem ipsum dolor sit, amet cons
      molestiae dignissimos neque eum i
      mollitia! Fuga, id adipisci.</p>

  <!-- vertical margin collapsing (bl

</body>
```



Definindo Todas as Bordas

- Propriedade Abreviada `border`
- Permite definir a borda por completo (4 lados)
- Sintaxe comum: `border: espessura estilo cor;`
- A ordem dos valores não importa
- Exemplos:
 - `border: 1px solid blue;`
 - `border: 2px double red;`
 - `border: none;`

Estilos

`solid`
`dotted`
`dashed`
`double`
`none`

Definindo as Bordas Individualmente

- Propriedades

- border-top
- border-bottom
- border-left
- border-right

- Exemplos

```
border-top: 1px solid red;
```

```
border-left: 2px dotted gray;
```

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

Propriedade *border-radius*

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

`border-radius: 3px;`

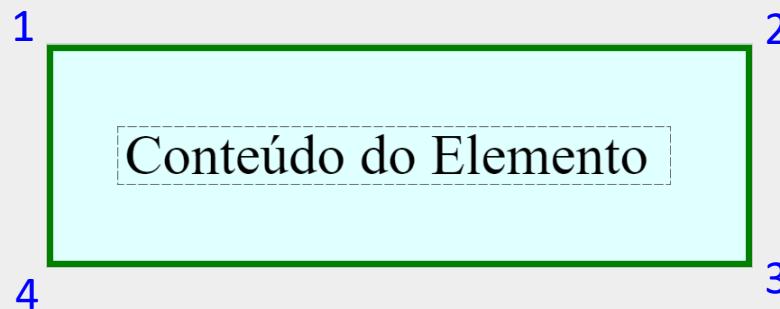
4 cantos com mesma curvatura, em pixels

`border-radius: 10%;`

4 cantos com mesma curvatura, em porcentagem

`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



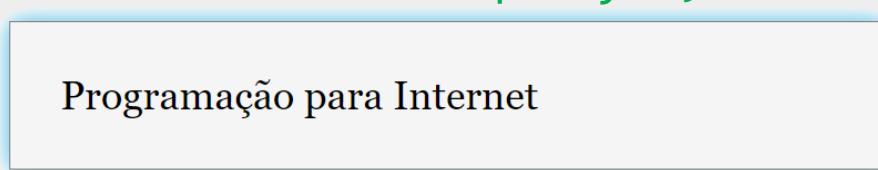
`box-shadow: 5px 5px 5px gray;`

Programação para Internet



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

Programação para Internet

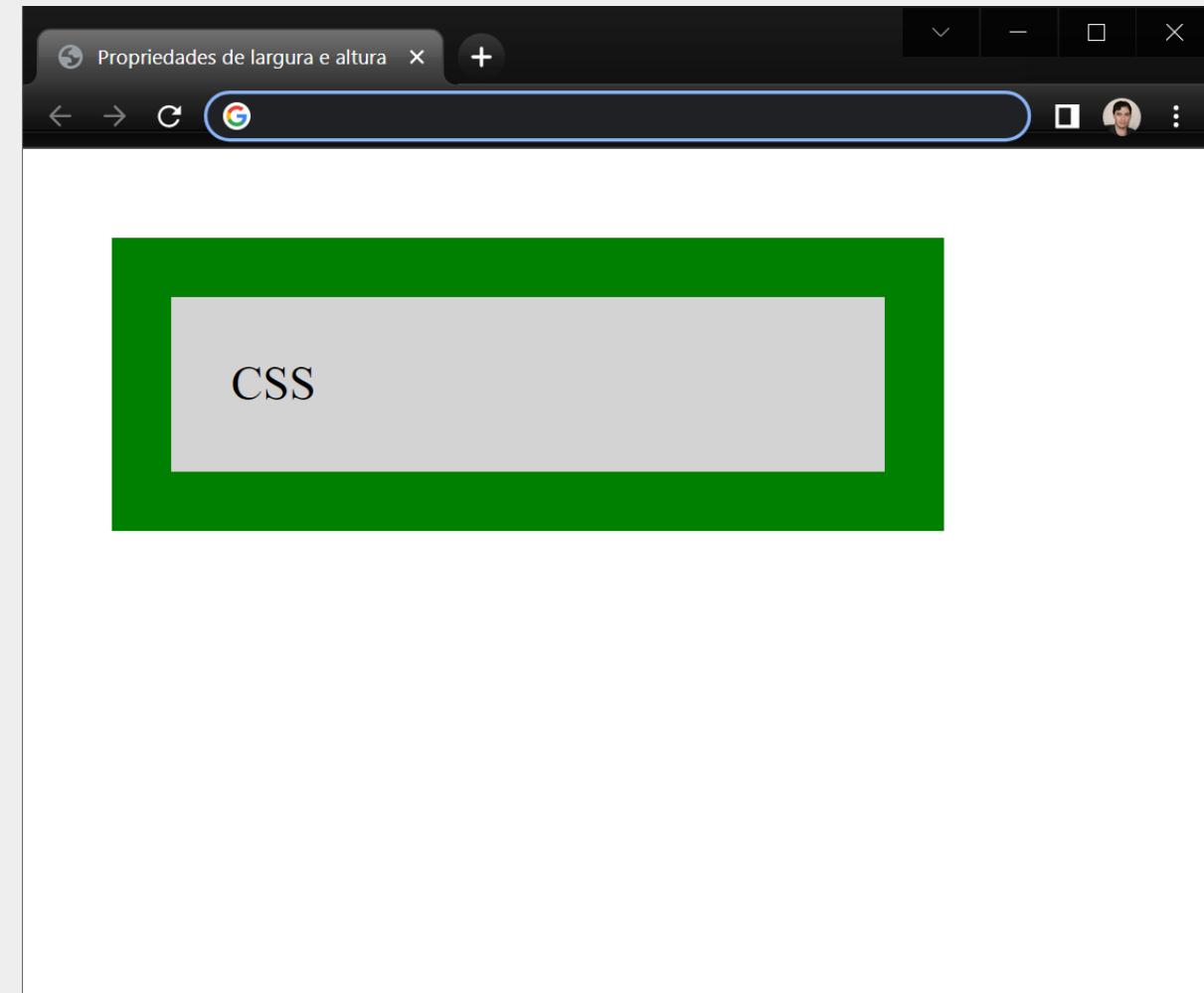


Ajustes de Largura e Altura dos Elementos

- 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
 - Definem tamanhos para a região de **conteúdo** do elemento, sem contabilizar margens, bordas e *paddings*
 - Valores em pixels, porcentagem, em, rem, etc.
 - Outros valores: `auto`, `max-content`, `min-content`, `fit-content`

Ajuste de largura - Exemplo

```
8  <style>
9      body { margin: 0 }
10     div {
11         background-color: lightgray;
12         width: 200px;
13         margin: 30px;
14         padding: 20px;
15         border: 20px solid green;
16     }
17 </style>
18 </head>
19
20 <body>
21     <div>
22         CSS
23     </div>
24 </body>
```



Largura Total do Elemento

The screenshot illustrates the concept of total element width in a browser's developer tools. On the left, a green box contains the text "CSS". A light gray border surrounds the box. Red dashed lines indicate the element's bounding box with dimensions: top and bottom padding of 30px, left and right margins of 20px each, and a central content area of 200px width. A red dashed arrow points from the text "Largura def. com width: 200px" to the central content area. Another red dashed arrow points from the text "Largura total do elemento: 340px" to the entire bounding box. On the right, the browser's developer tools interface shows the "Elementos" tab selected. The "Computado" tab displays the element's total dimensions: margin (30px), border (19.998px), padding (20px), and content (199.998x18.523px). The total width is shown as 340px.

Propriedades de largura e altura

Arquivo |

CSS

Largura def. com width: 200px

Largura total do elemento: 340px

margin 30
border 19.998
padding 20
199.998x18.523
20 20 19.998
20
19.998
30

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 *paddings superiores e inferiores* de alguns elementos **não são** respeitados

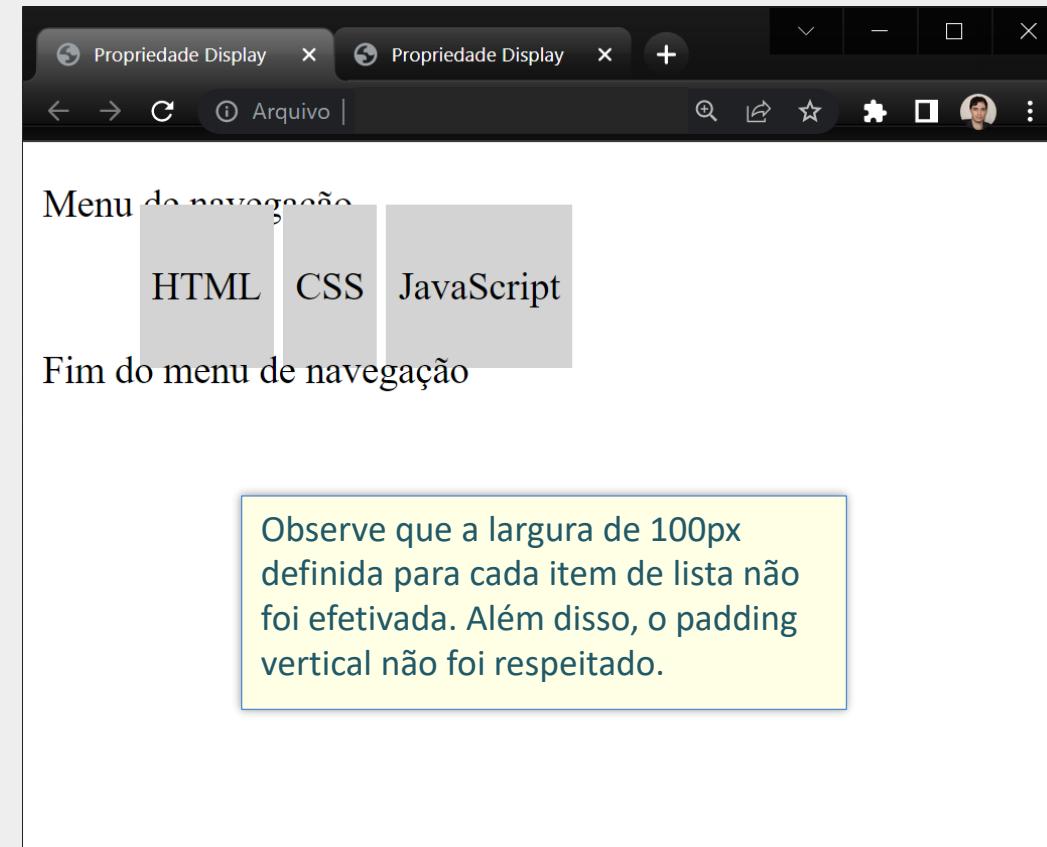
display: inline-block

- exibição em nível de linha
- possibilidade de usar *width* e *height*
- margens e *paddings 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>

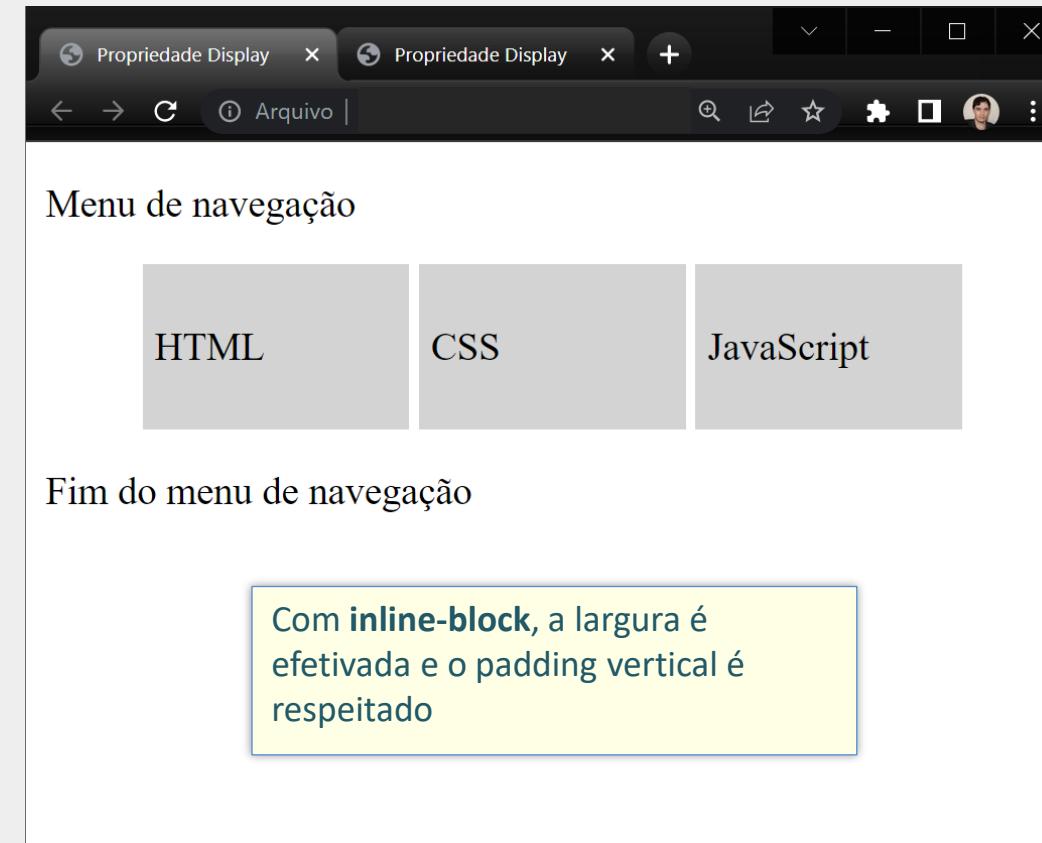
<body>
  <nav>
    <p>Menu de navegação</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
    </ul>
    <p>Fim do menu de navegação</p>
  </nav>
</body>
```



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>
```



Com **inline-block**, a largura é efetivada e o padding vertical é respeitado

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 (veja exemplo no próximo slide)

```


| Aluno          | Prova 1 | Prova 2 |
|----------------|---------|---------|
| Fulano         | 7,0     | 8,0     |
| Ciclano        | 6,0     | 9,0     |
| Media da Turma | 6,5     | 8,5     |


```

| Aluno | Prova 1 | Prova 2 |
|----------------|---------|---------|
| Fulano | 7,0 | 8,0 |
| Ciclano | 6,0 | 9,0 |
| Media da Turma | 6,5 | 8,5 |

Tabela original com todas as linhas e colunas sendo exibidas

```

.collapse { visibility: collapse; }
.displayNone { display: none; }

```

</style>

</head>

<body>

```


| Media da Turma | 6,5 | 8,5 |
|----------------|-----|-----|
|----------------|-----|-----|


```

| Aluno | Prova 1 | Prova 2 |
|---------|---------|---------|
| Fulano | 7,0 | 8,0 |
| Ciclano | 6,0 | 9,0 |

Última linha ocultada com a propriedade **visibility: collapse**. Repare que as larguras das colunas permanecem inalteradas.

```

.collapse { visibility: collapse; }
.displayNone { display: none; }

```

</style>

</head>

<body>

```


| Media da Turma | 6,5 | 8,5 |
|----------------|-----|-----|
|----------------|-----|-----|


```

| Aluno | Prova 1 | Prova 2 |
|---------|---------|---------|
| Fulano | 7,0 | 8,0 |
| Ciclano | 6,0 | 9,0 |

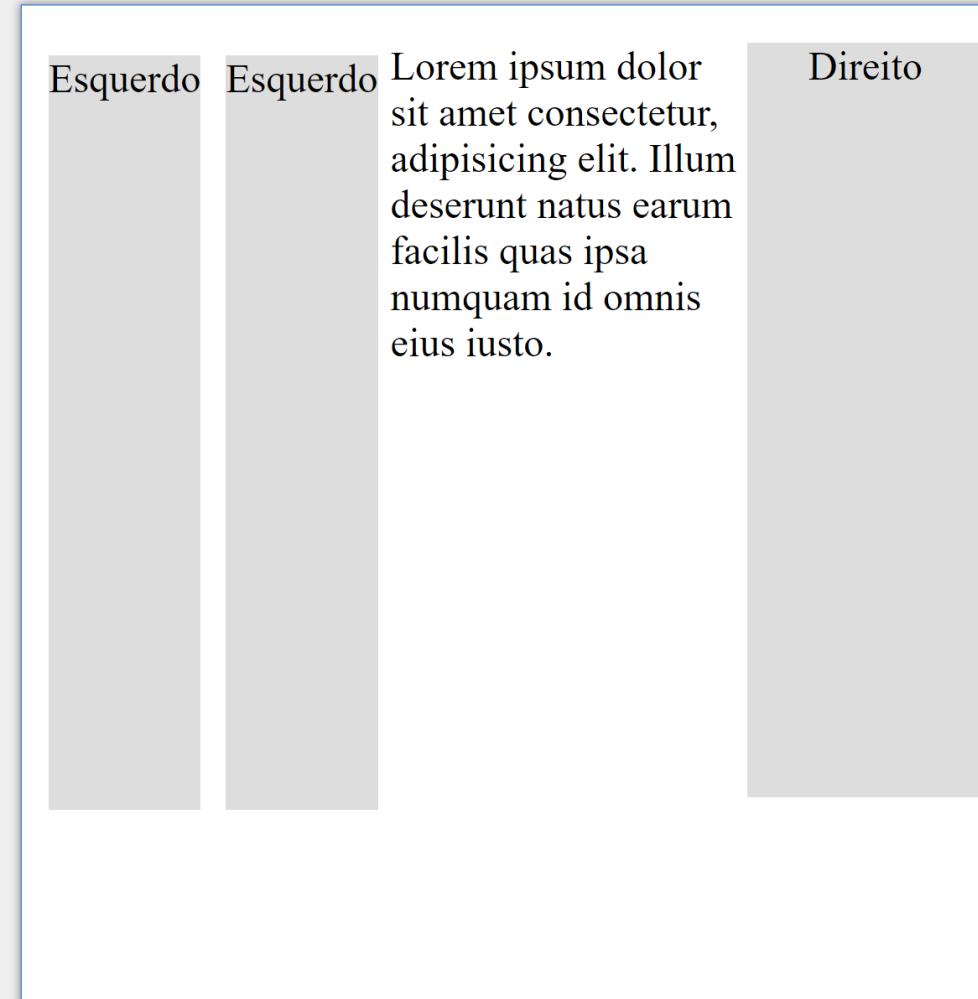
Última linha ocultada com **display: none**. Repare que as larguras das colunas são reajustadas conforme conteúdo (col. **Aluno**)

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>
```



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 é 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***

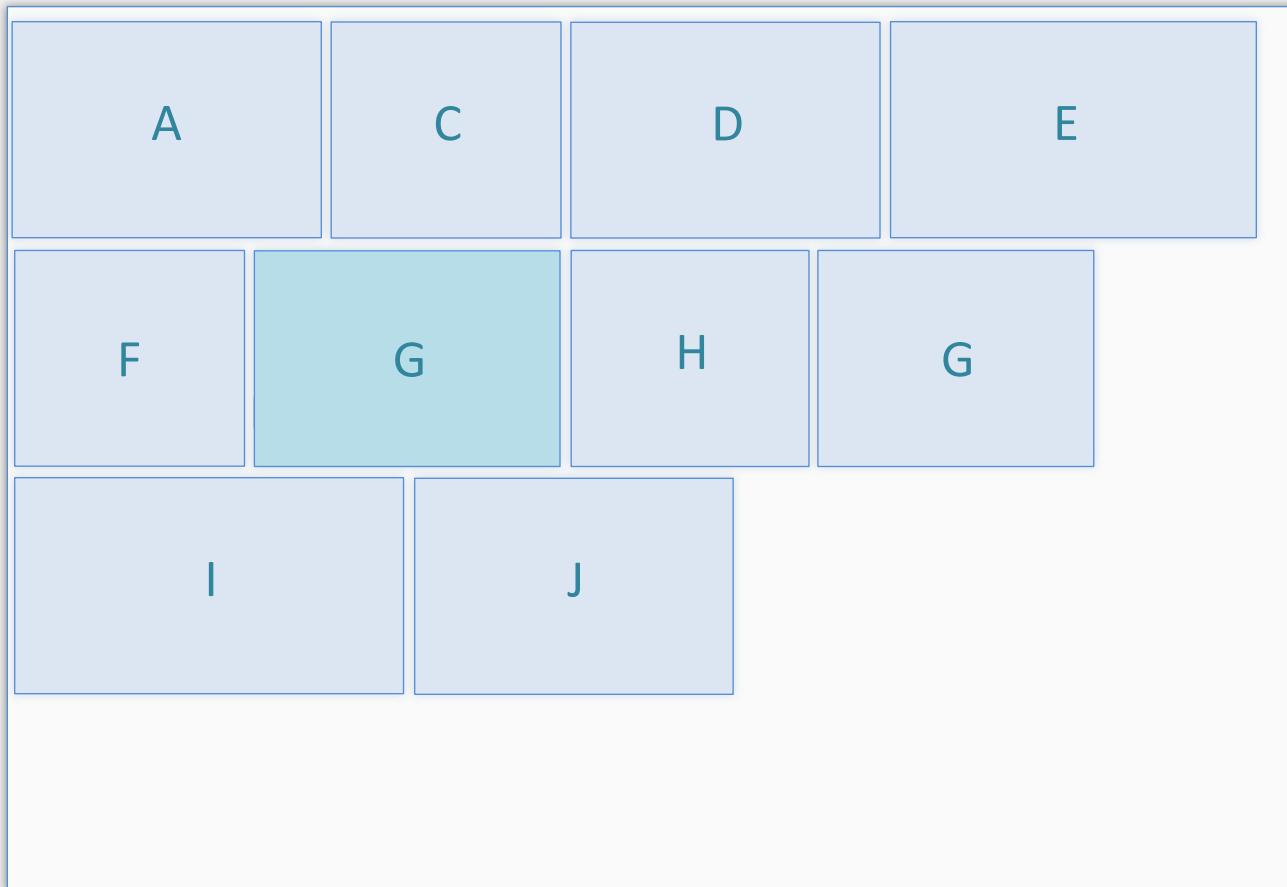
`position: static;`

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

`position: relative;`

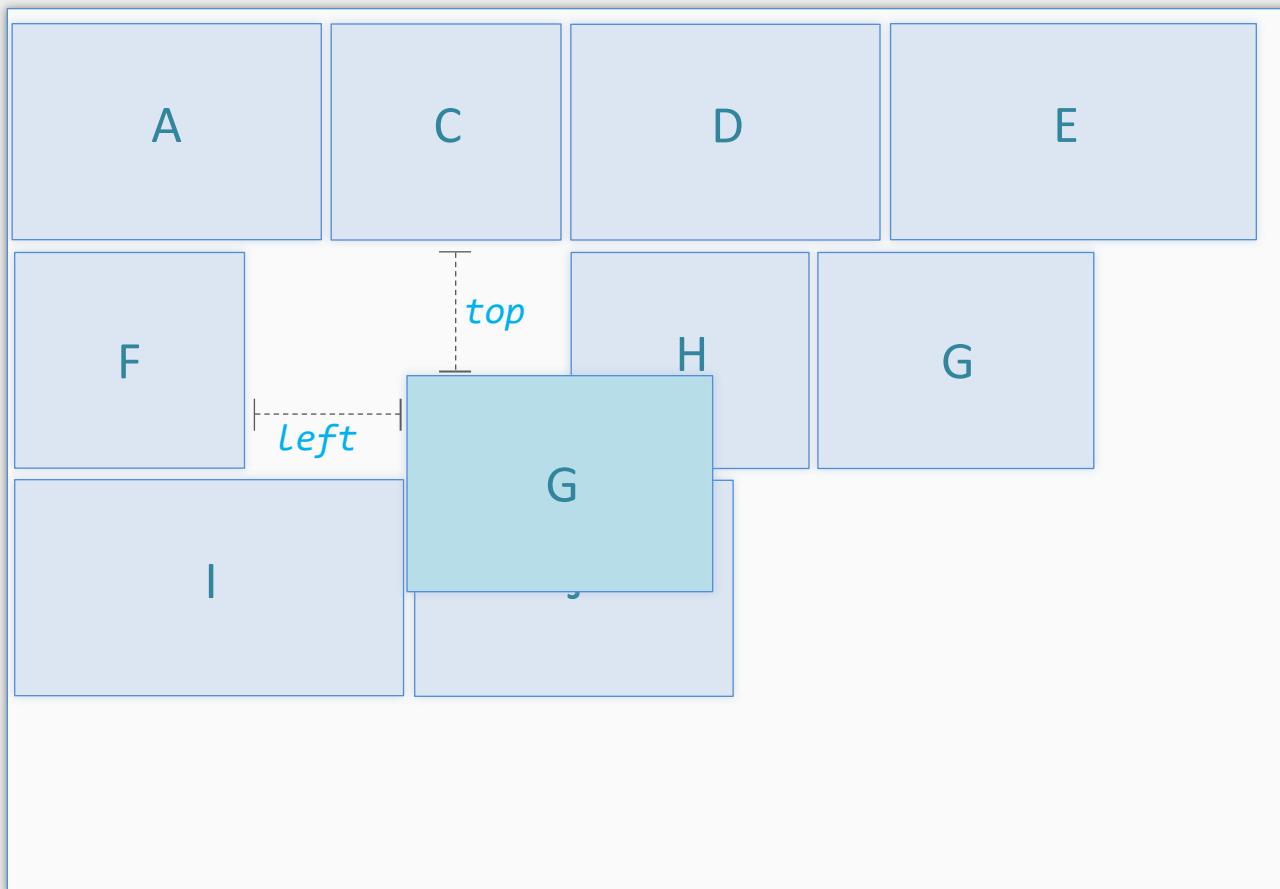
- Primeiramente posicionado de acordo com o fluxo normal
- Em seguida, é deslocado da sua posição com `top`, `left`, `right` e `bottom`
- O deslocamento não afeta a posição dos demais elementos
 - Layout geral é o mesmo do posicionamento `static`

Propriedade ***position: static***



```
#Painel_G {  
    display: static;  
    background-color: aqua;  
}
```

Propriedade ***position: relative***



```
#Painel_G {  
    display: relative;  
    left: 50px;  
    top: 50px;  
    background-color: aqua;  
}
```

Propriedade ***position*** - Valores

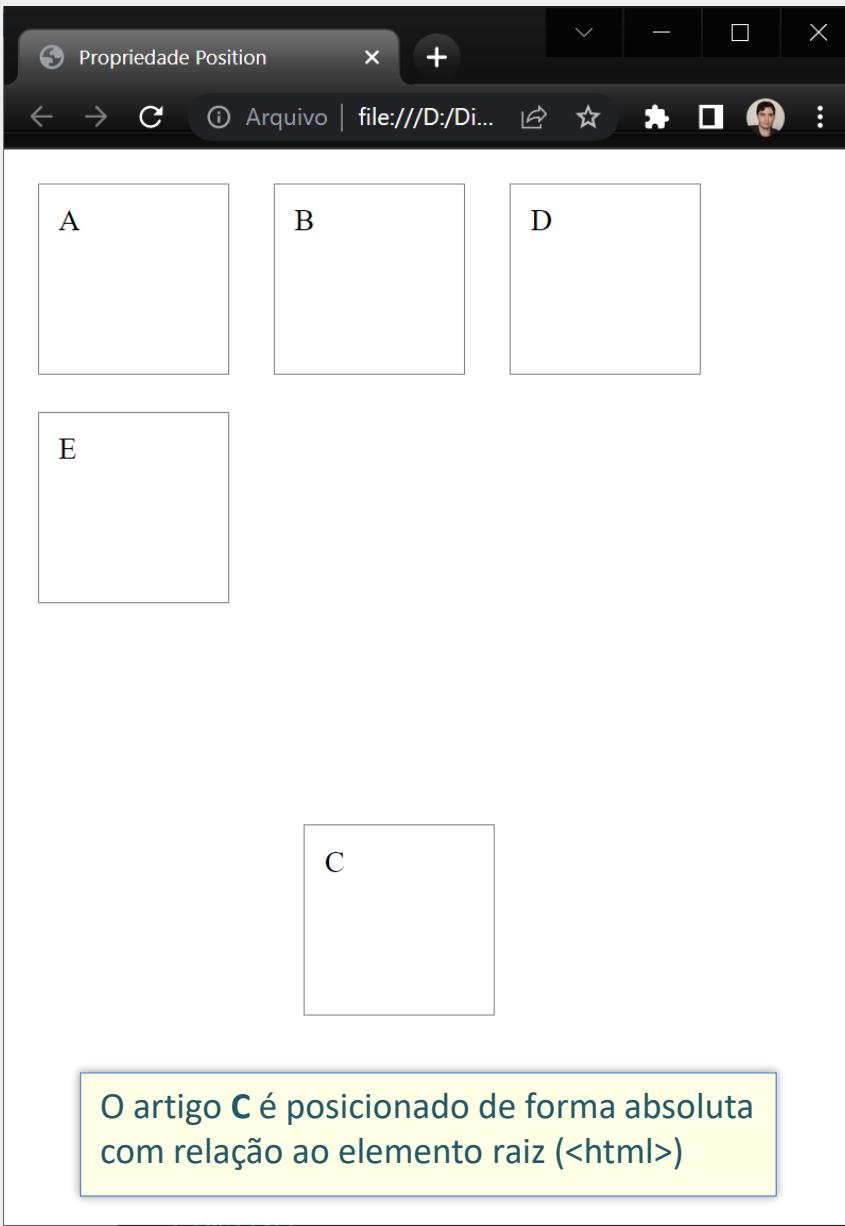
position: absolute;

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

position: absolute

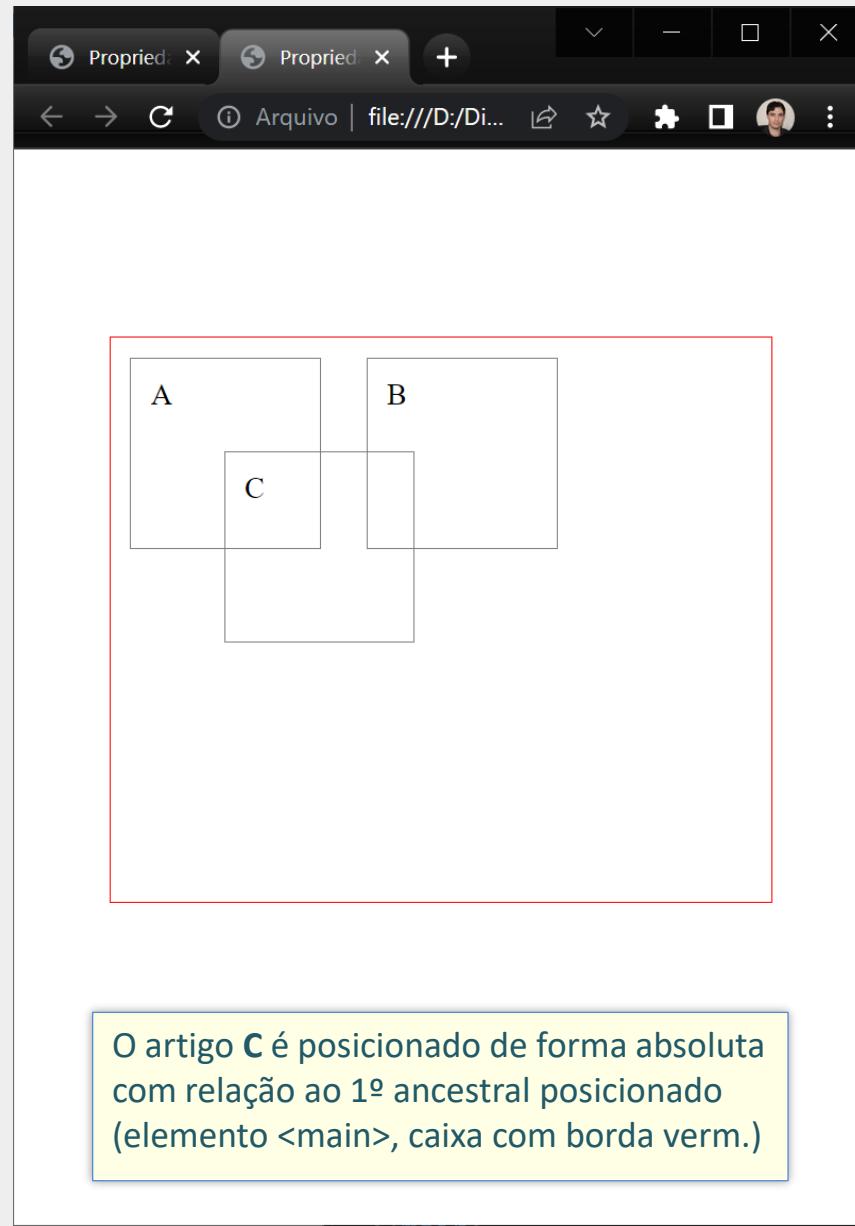
```
<style>
  article {
    margin: 10px; padding: 10px;
    border: 1px solid gray;
    width: 80px; height: 80px;
    display: inline-block;
  }

  .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>
```



O artigo **C** é posicionado de forma absoluta com relação ao 1º ancestral posicionado (elemento `<main>`, caixa com borda vermelha.)

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

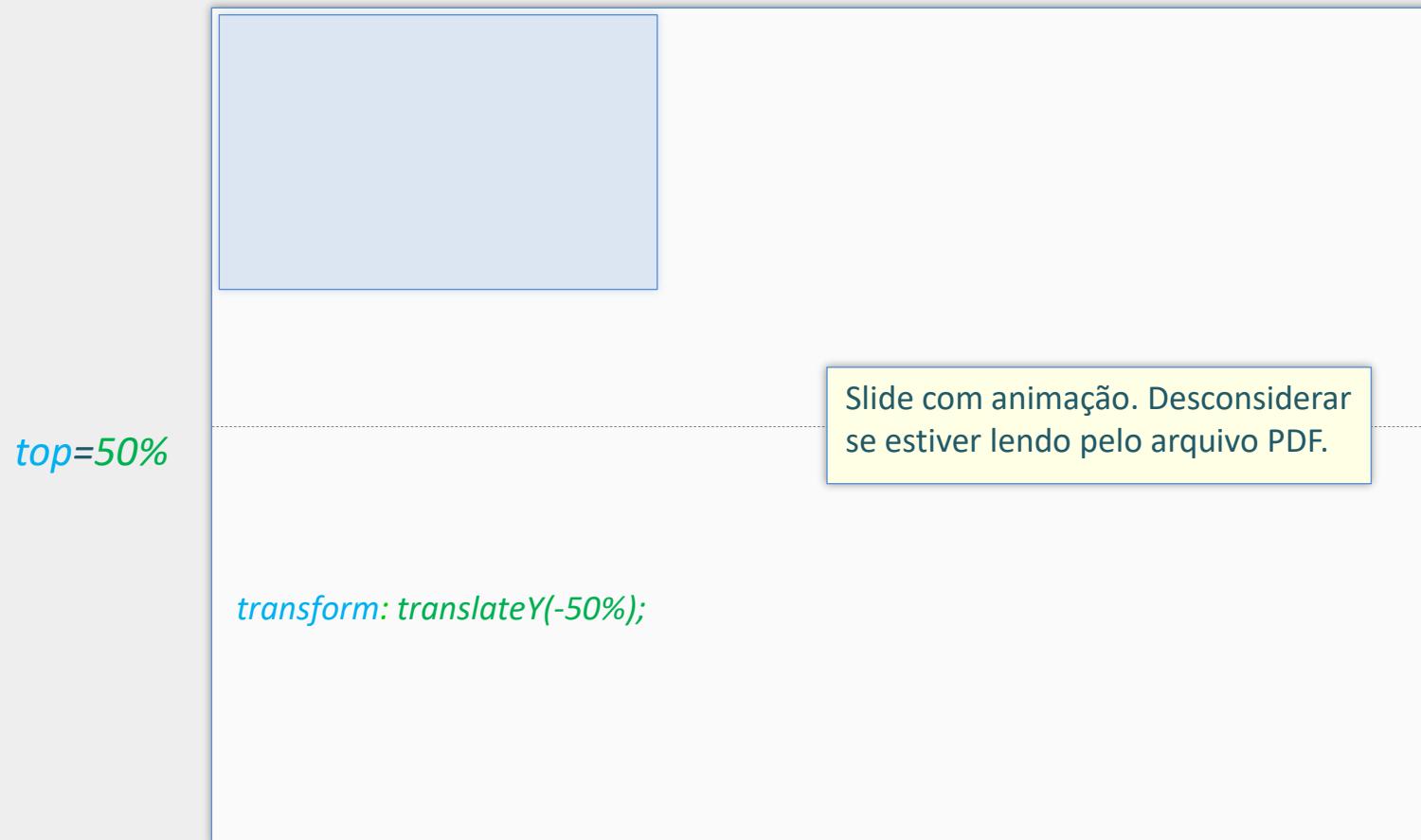
Propriedade *transform*

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

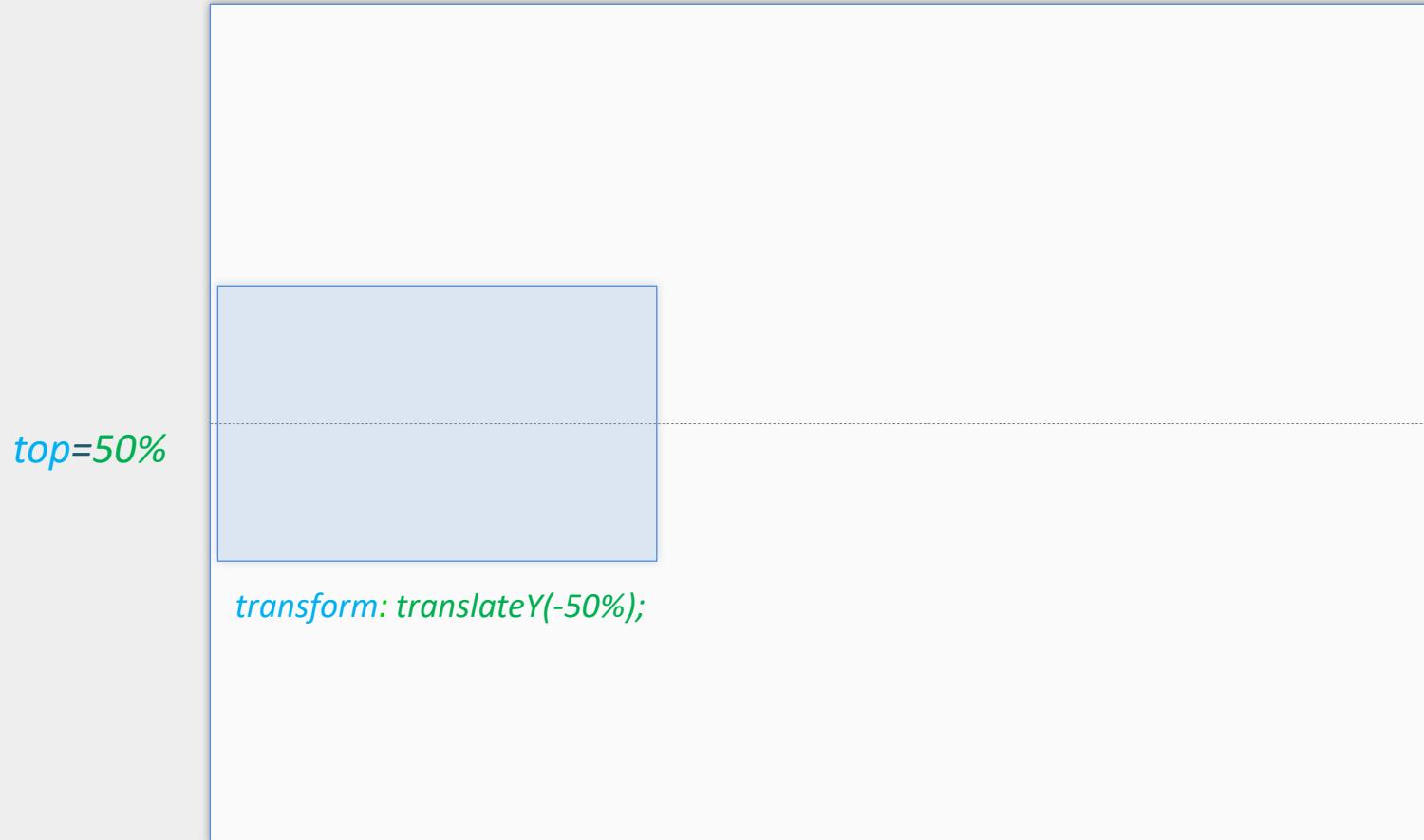
| | |
|---|--|
| <code>transform: translateX(50px);</code> | move o elemento 50 pixels em x (hor.) |
| <code>transform: translateY(50px);</code> | move o elemento 50 pixels em y (vert.) |
| <code>transform: translate(30px,10px);</code> | move o elemento 30 pixels em x e 10 em y |
| <code>transform: rotate(45deg);</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



Centralizando Bloco na Vertical com Pos. Absoluto



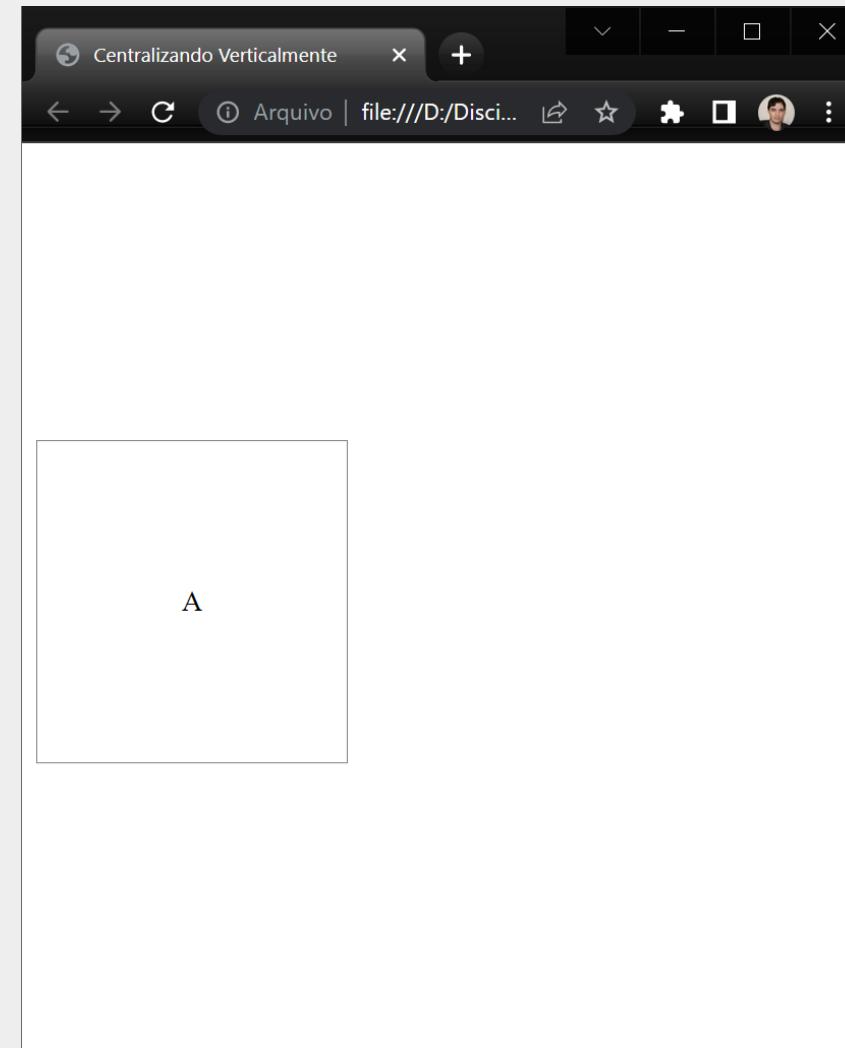
Centralizando Bloco na Vertical com Pos. Absoluto

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

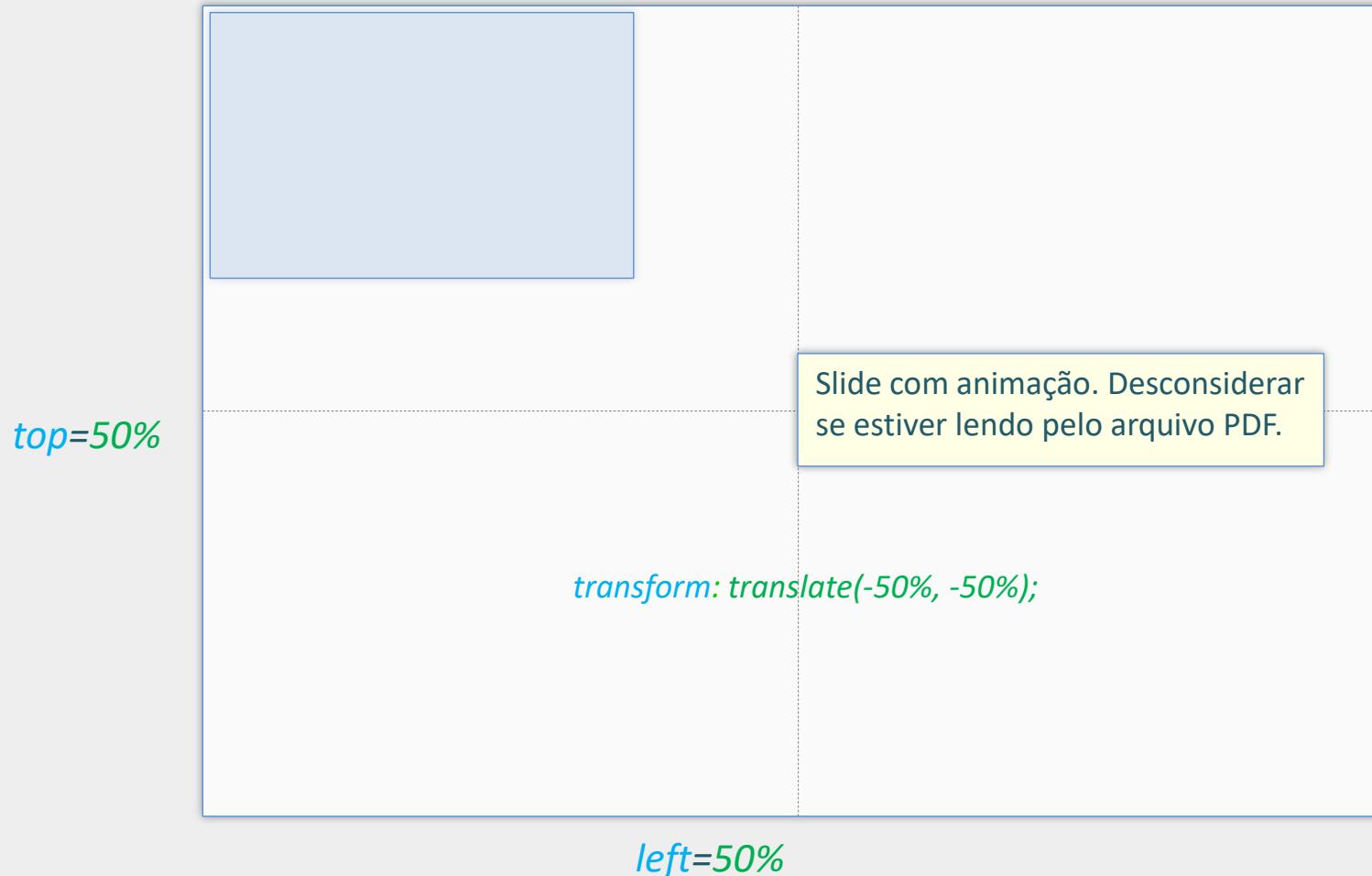
    position: absolute;
    top: 50%;
    transform: translateY(-50%);

  }
</style>
</head>

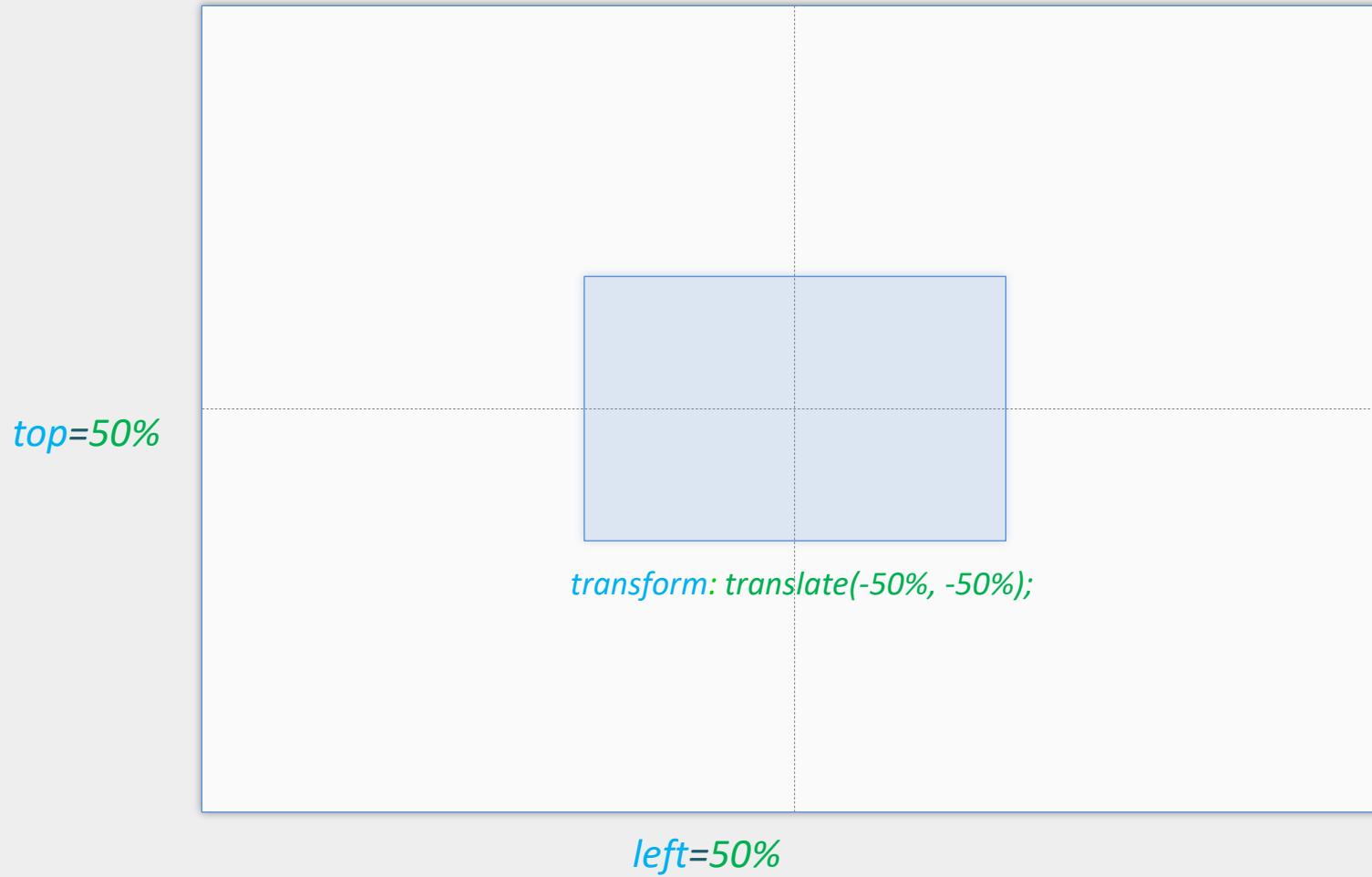
<body>
  <main>A</main>
</body>
```



Centralizando na Vertical e Horizontal com Pos. Absoluto



Centralizando na Vertical e Horizontal com Pos. Absoluto

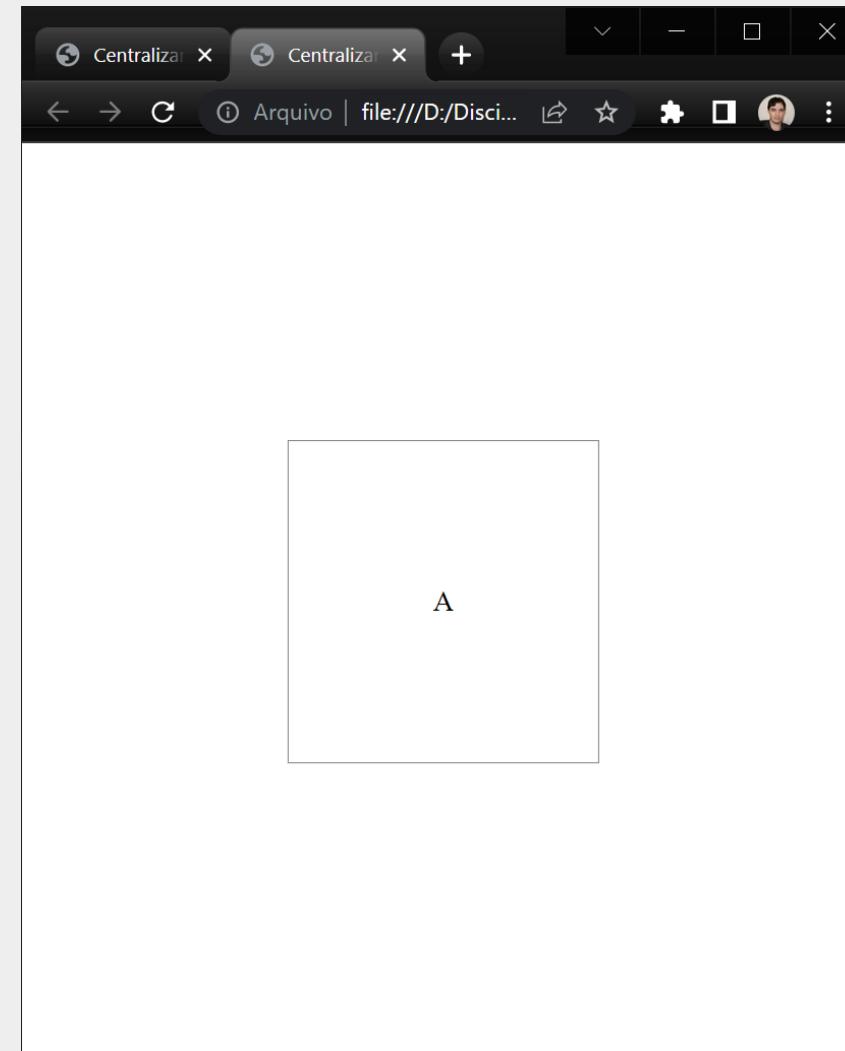


Centralizando na Vertical e Horizontal com Pos. Absoluto

```
<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%);  
}
```



Outra Forma de Centralizar na Horizontal Utilizando **width** e **margin**

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

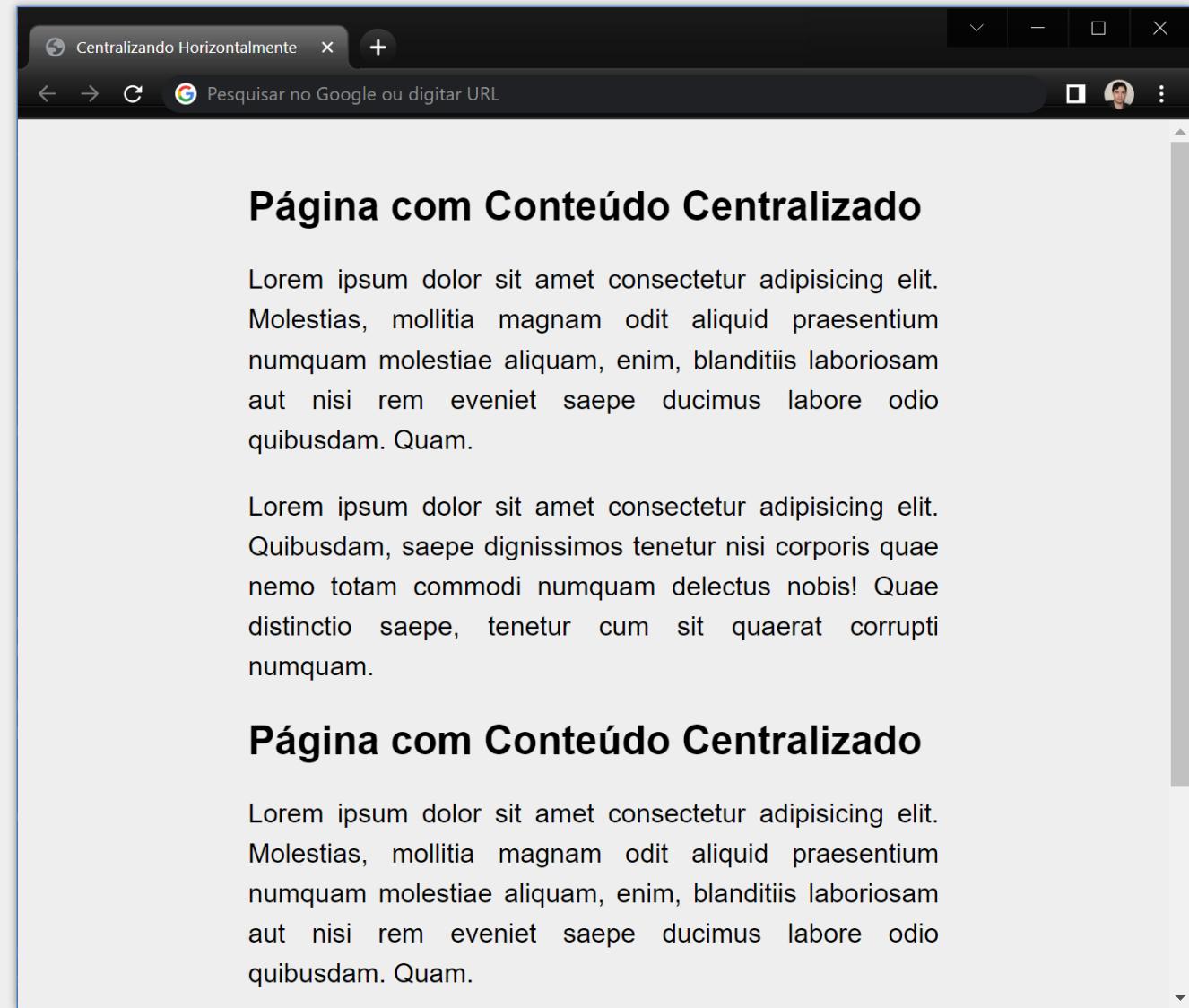
```
body {  
    width: 60%;  
    margin: 0 auto;  
}
```

OBS: para centralizar apenas o texto dentro de um elemento, utilize:
text-align: center;

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

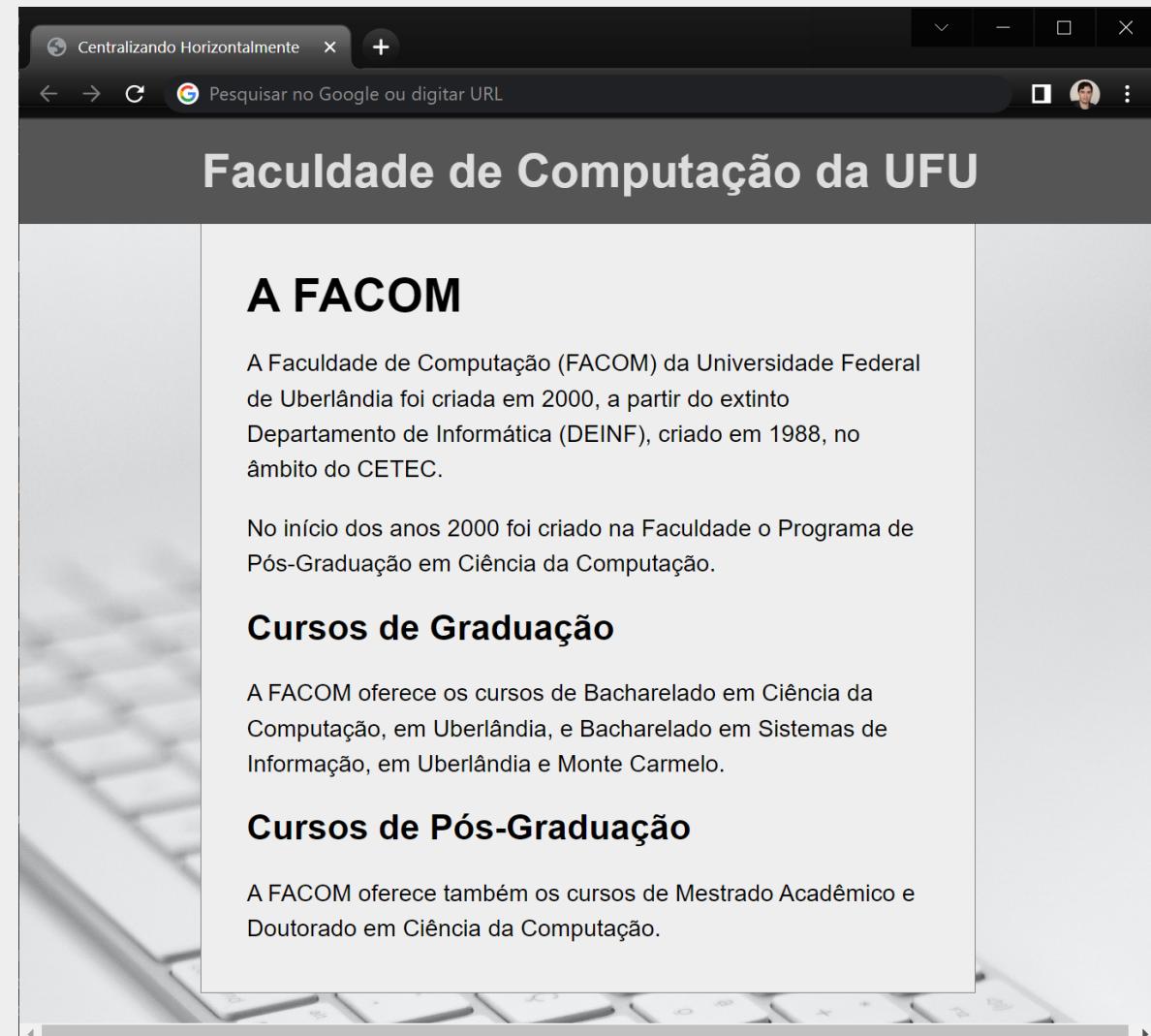
```
<style>
  body {
    text-align: justify;
    width: 60%;
    margin: 40px auto;
  }
</style>
</head>

<body>
  <h2>Página com Conteúdo Centralizado</h2>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisci...
    molestiae aliquam, enim, blanditiis laborios...
  <p>Lorem ipsum dolor sit amet consectetur adipisci...
    nemo totam commodi numquam delectus nobis! ...
  <h2>Página com Conteúdo Centralizado</h2>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisci...
    molestiae aliquam, enim, blanditiis laborios...
  <p>Lorem ipsum dolor sit amet consectetur adipisci...
    nemo totam commodi numquam delectus nobis! ...
</body>
```



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

```
header {  
    background-color: #555;  
    width: 100%;  
}  
  
main {  
    background-color: #eee;  
    width: 60%;  
    margin: 0 auto;  
}  
</style>  
</head>  
<body>  
    <header><h1>Faculdade de Computação da UFU</h1>  
    </header>  
    <main>  
        <h1>A FACOM</h1>  
        <p>A Faculdade de Computação (FACOM) da  
            Universidade Federal de Uberlândia foi criad  
            a partir do extinto Departamento de Informát
```



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 {color: blue;}</i> |
| :hover | Usuário interage com elem. sem ativar Ex.: ponteiro do mouse sobre o elem. | <i>a:hover, button:hover {...}</i> |
| :active | Estado ativo do elemento Ex.: botão do mouse pres. sobre o elem. | <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, radio</i> ou <i>option</i> selecionado | <i>radio:checked {...}</i> |
| :focus | Elemento recebendo foco | <i>input:focus {...}</i> |

Exemplos de Pseudo-Classes

| Pseudo-Classe | Descrição | Exemplos |
|---------------------|--------------------------------|---|
| :first-child | Primeiro filho do elemento pai | <i>li:first-child {color: blue;}</i> |
| :last-child | Último filho do elemento pai | <i>li:last-child {color: blue;}</i> |
| :nth-child | N-ésimo filho do elemento pai | <i>li:nth-child(2) {color: blue;}</i> <i>Altera a cor dos segundos 's</i> |

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

Outros Pseudo-Elementos

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

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

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

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

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

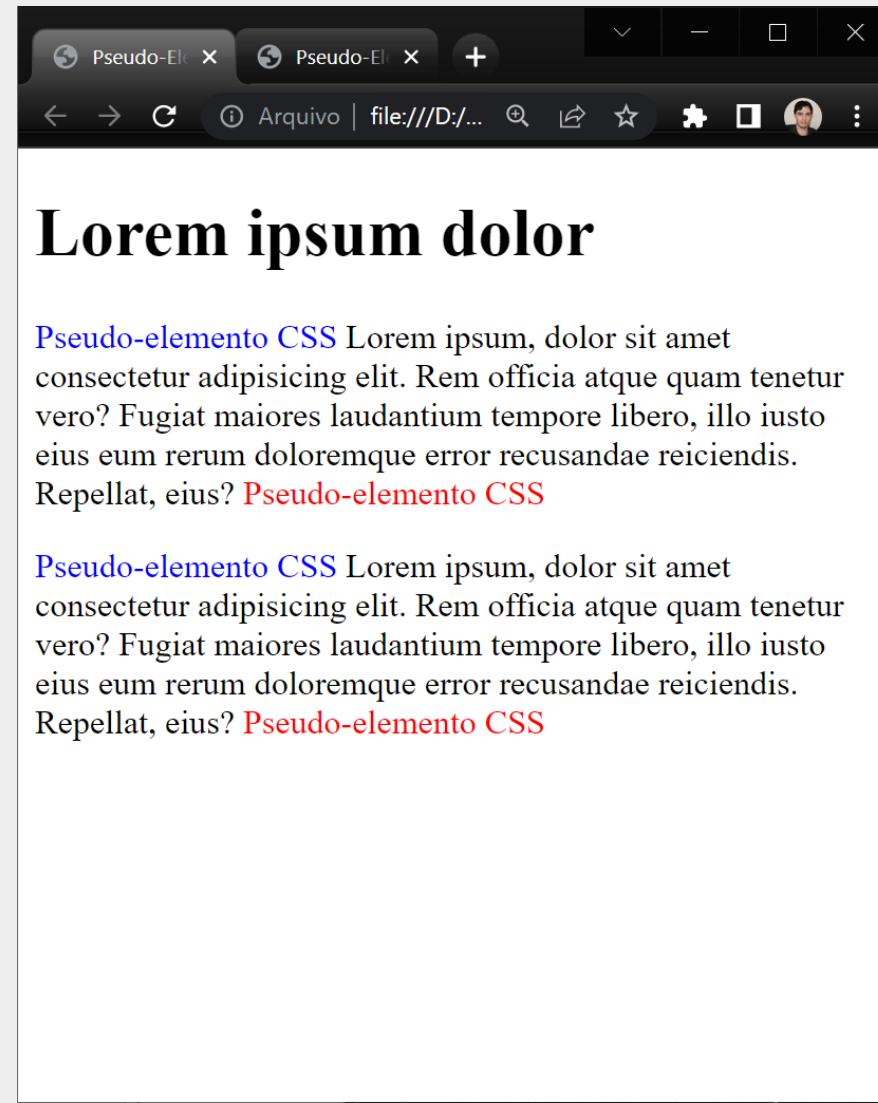
Altera a cor dos textos de **placeholder** dos campos *input*

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

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 iusto.
  <p>Lorem ipsum, dolor sit amet consectetur
    laudantium tempore libero, illo iusto.
```

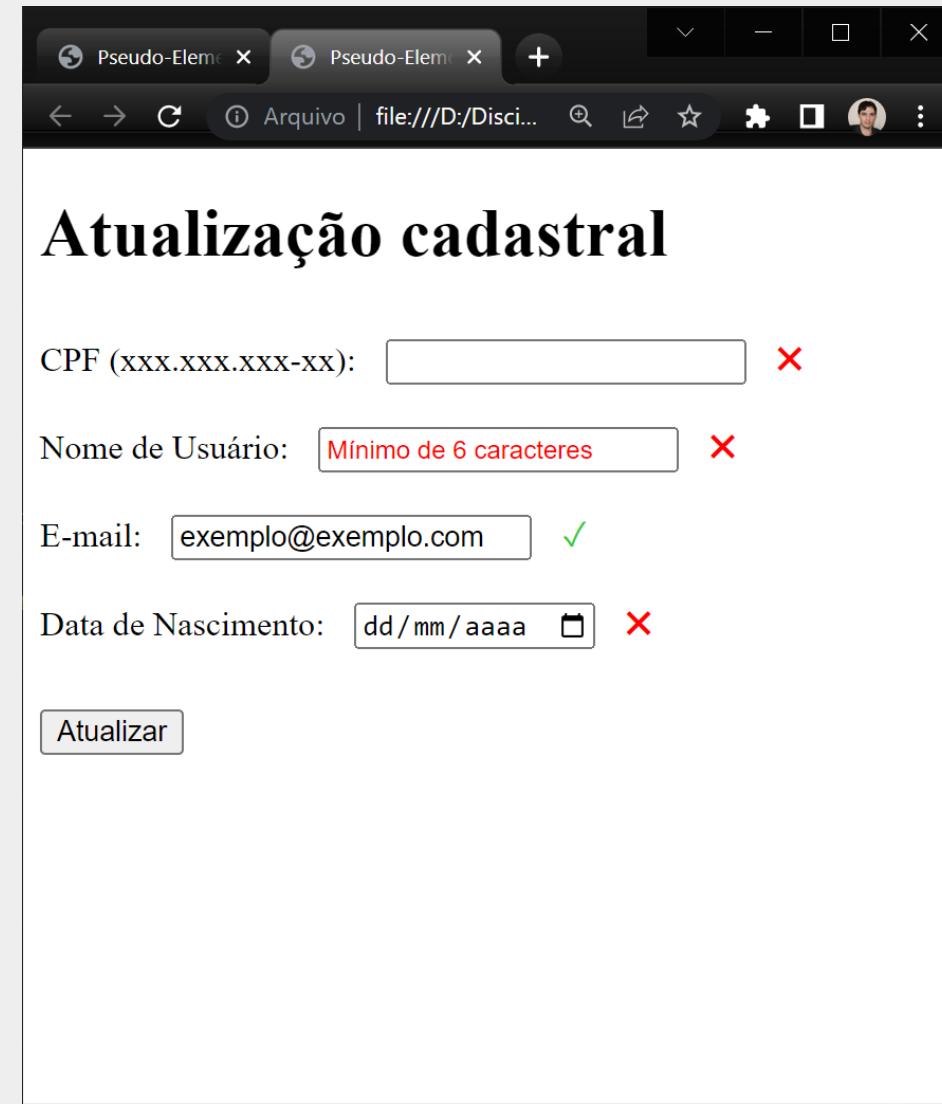


Pseudo-Elementos - Exemplo

```
input:invalid + span::after {
    content: 'X';
    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>
```



Observações 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

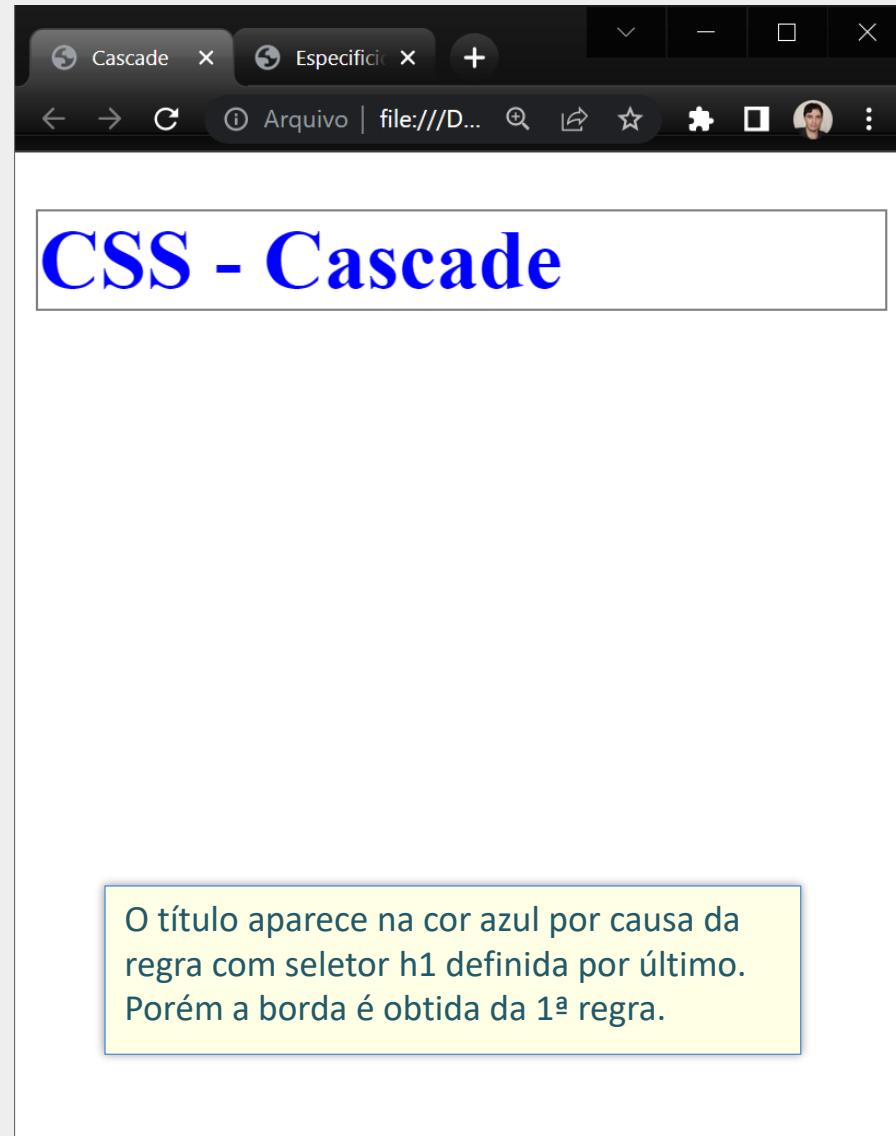
- Caso haja declarações CSS repetidas
- Caso sejam do **mesmo tipo** de seletor
- Prevalecerá a declaração definida por último

Ordem das Regras - Cascade

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

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

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



O título aparece na cor azul por causa da regra com seletor h1 definida por último. Porém a borda é obtida da 1ª regra.

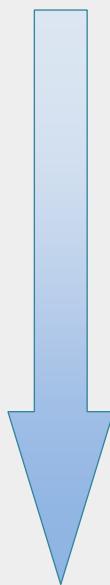
Especificidade

Seletor de Elemento

Seletor de Classe

Seletor de ID

Código Inline



Maior especificidade

Maior peso da declaração CSS

Sobreposição de declarações anteriores

Especificidade das Regras

```
<style>
  h1 {
    color: red;
    text-transform: lowercase;
    margin-top: 200px;
  }

  #tituloVerde {
    color: green;
    border: 1px solid black;
  }

  .tituloAzul {
    color: blue;
    text-transform: uppercase;
  }
</style>
</head>
<body>
  <h1 id="tituloVerde" class="tituloAzul">
    CSS - Especificidade
  </h1>
</body>
```

CSS - ESPECIFICIDADE

Uma cor diferente é definida em cada regra, porém a cor verde, do seletor de ID, é a que prevalece porque tal seletor é mais específico que os outros.

`text-transform` aparece na 1^a e na 3^a regra, porém o valor que prevalece é o da 3^a regra porque ela é mais específica.

Observe que a margem superior vem da regra menos específica pois ela não é redefinida nas demais.

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">
```

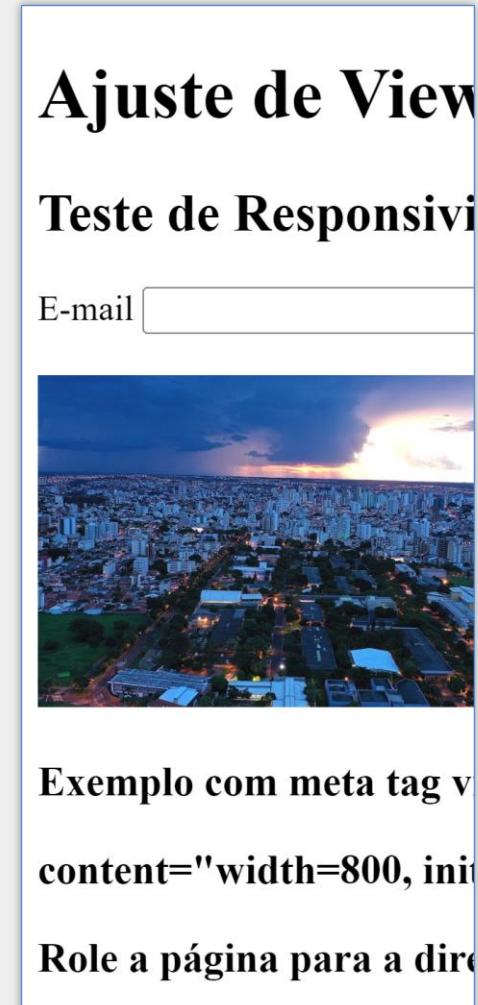
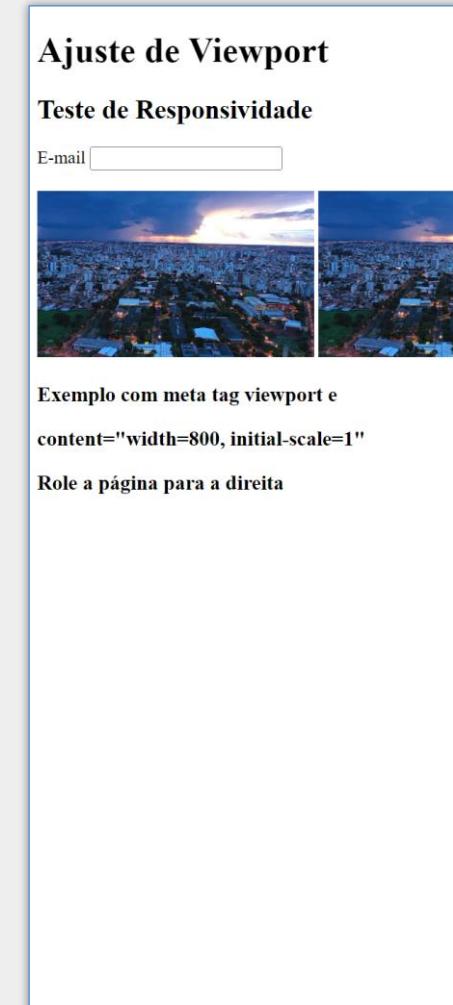
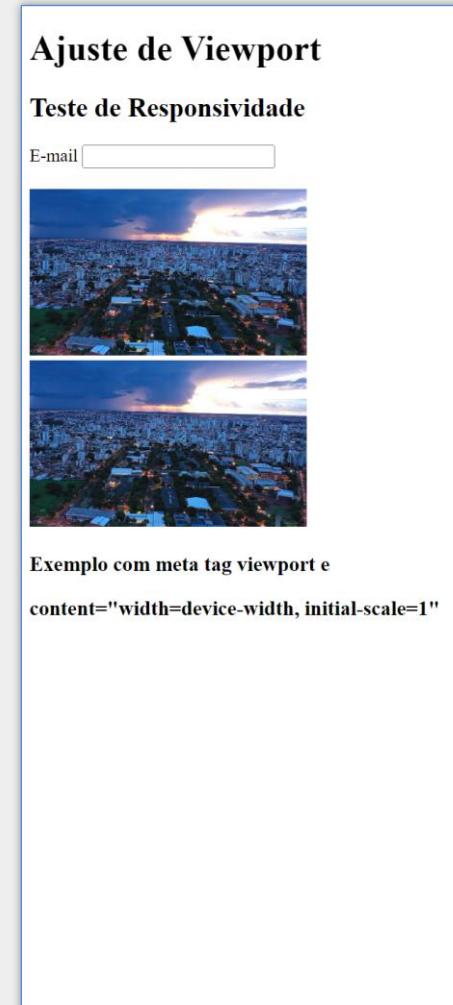
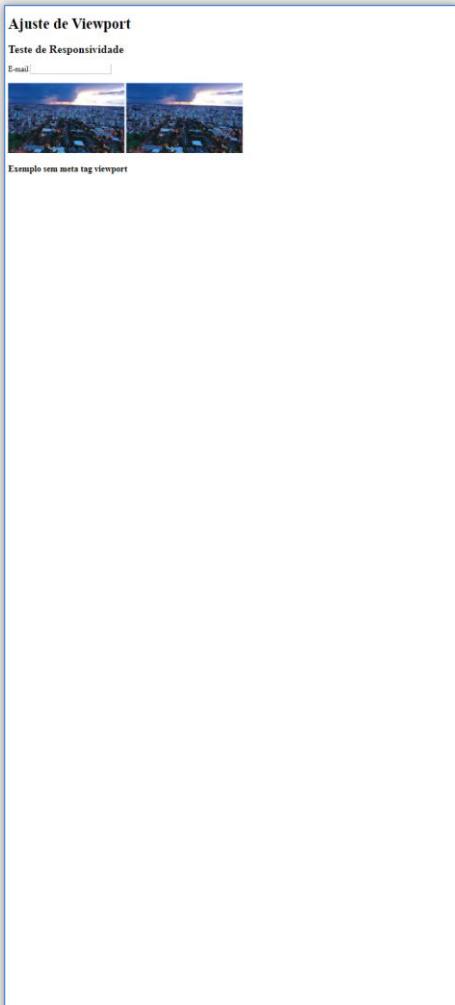
- Viabiliza a responsividade conforme dispositivo/tela
- Página melhor ajustada à tela de smartphones e tablets
- Ativa o fator de escala no dimensionamento em disp. móveis
- *Pixel ratio* do dispositivo é considerado

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)

Página **com** a *meta tag viewport*
e **width** com valor **800**

Página **com** a *meta tag viewport*
e **width=800, initial-scale=2**

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)

MediaType é opcional.

Se omitido será
considerado **all**

```
@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 aplicado quando pelo menos uma das expressões for 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.

OBS: não existe o operador **OR** em media queries, mas a vírgula tem papel similar.

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

- w3.org/Style/CSS/
- developer.mozilla.org/en-US/docs/Web/CSS
- w3.org/Style/CSS/learning
- w3.org/TR/mediaqueries-3/
- **HTML and CSS: Design and Build Websites**, Jon Duckett.