



CSS

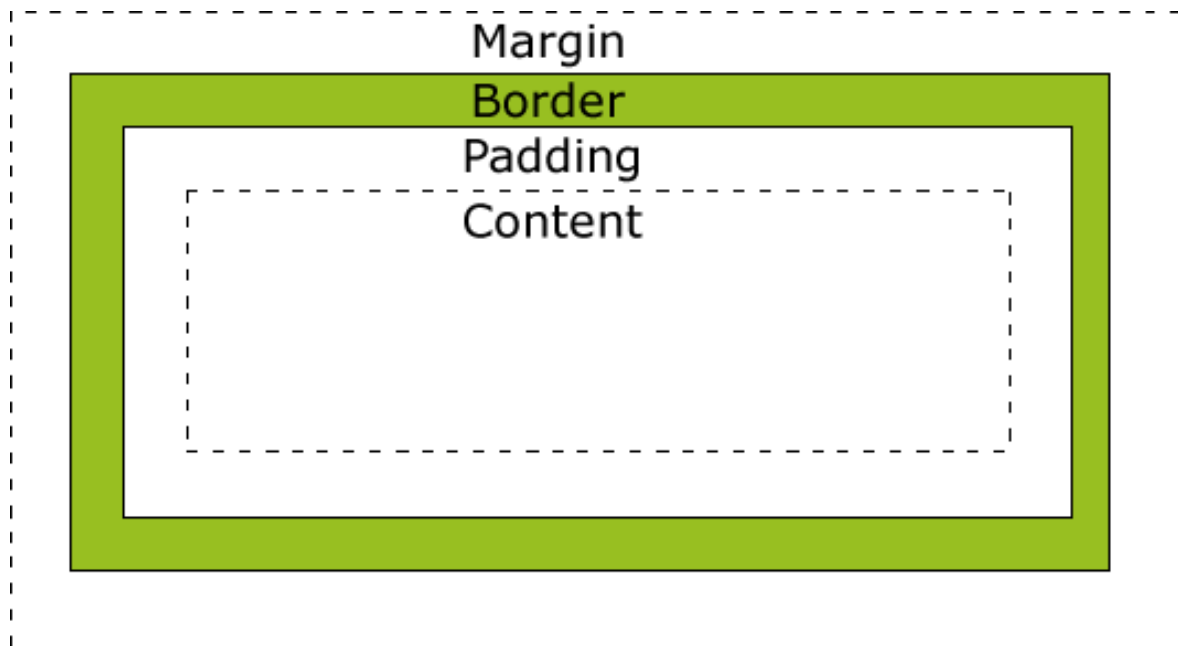


Fundamentos de CSS

Msc. Arcanjo Miguel Mota Lopes

Box Model

- O box model do CSS é essencialmente uma caixa que envolve todos os elementos do tipo block
- Esta caixa possui quatro elementos, conforme mostrado na figura abaixo:



Bordas

- Existem três propriedades CSS básicas para bordas: **border-width**, **border-style**, e **border-color**

```
<style type="text/css">
  p {
    border-width: 5px;
    border-style: solid;
    border-color: blue;
  }
</style>
```

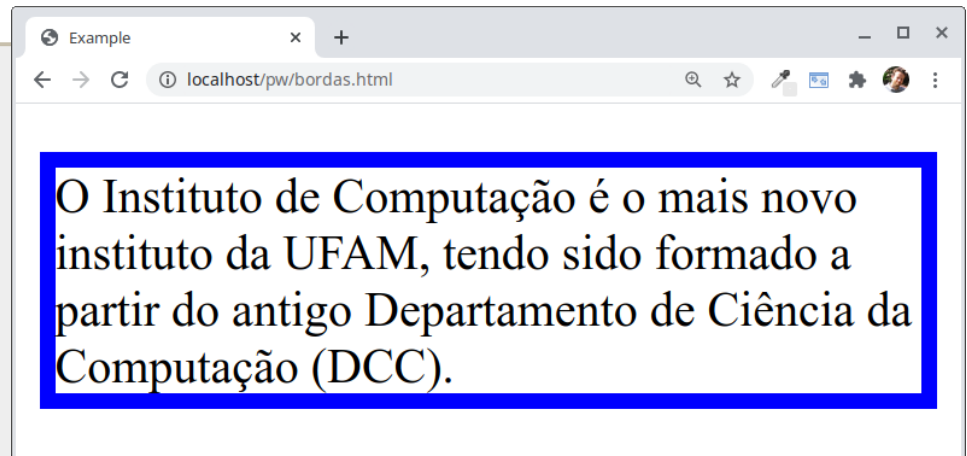
```
<p>
  O Instituto de Computação é o mais novo
  instituto da UFAM, tendo sido formado a
  partir do antigo Departamento de Ciência
  da Computação (DCC).
</p>
```

Bordas

- Existem três propriedades CSS básicas para bordas: **border-width**, **border-style**, e **border-color**

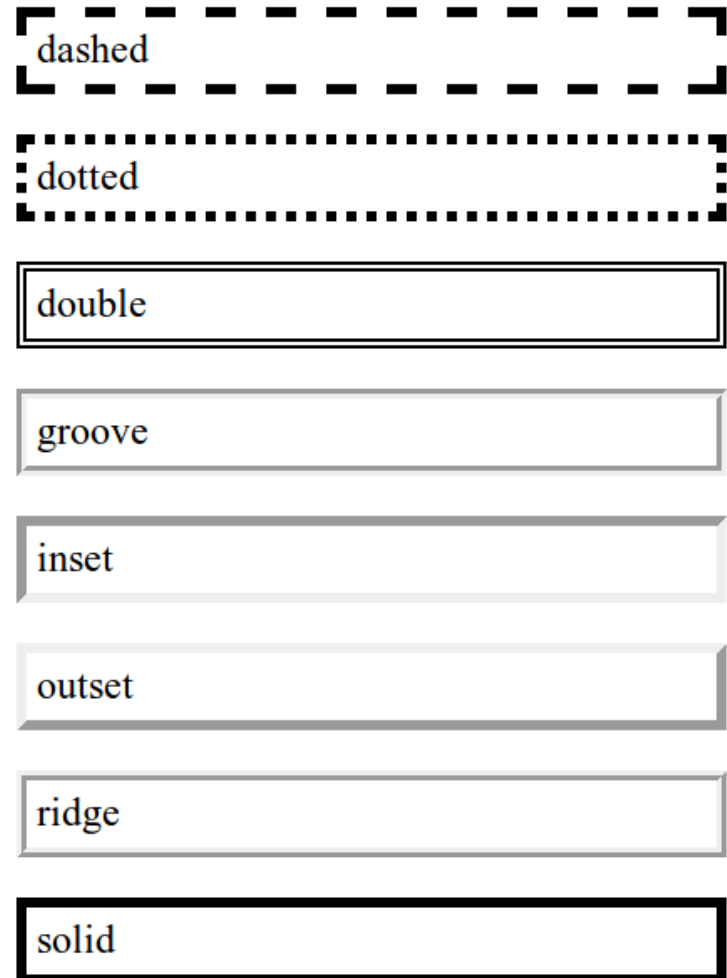
```
<style type="text/css">
  p {
    border-width: 5px;
    border-style: solid;
    border-color: blue;
  }
</style>
```

```
<p>
  O Instituto de Computação é o mais novo
  instituto da UFAM, tendo sido formado a
  partir do antigo Departamento de Ciência
  da Computação (DCC).
</p>
```



Bordas

- A tabela ao lado mostra as opções disponíveis para a propriedade **border-style**
- Além destas, existe a opção none, que força a não apresentação da borda.

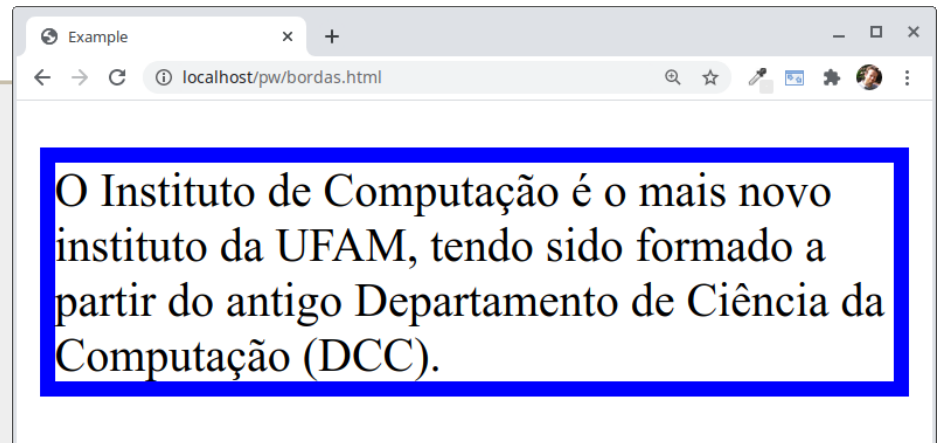


Bordas

- Ao invés de usar cada propriedade individualmente, podemos adotar um atalho:

```
<style type="text/css">
  p {
    border: 5px solid blue;
  }
</style>
```

```
<p>
  O Instituto de Computação é o mais novo
  instituto da UFAM, tendo sido formado a
  partir do antigo Departamento de Ciência
  da Computação (DCC).
</p>
```

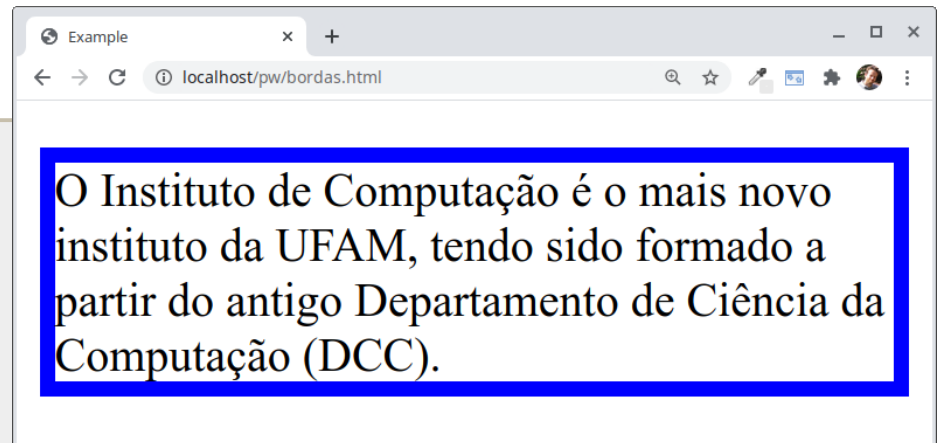


Bordas

- Ao invés de usar cada propriedade individualmente, podemos adotar um atalho:

```
<style type="text/css">  
  p {  
    border: 5px solid blue;  
  }  
</style>
```

```
<p>  
  Para usar o atalho, é preciso  
  obedecer a ordem: width style  
  color  
</p>
```



Borda para cada um dos lados

- CSS fornece propriedades que especificam cada borda (direita, esquerda, inferior e superior).

```
<style type="text/css">
  p {
    border-top-style: double;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: groove;
  }
</style>
```

```
<p>
  Example with border individual sides.
</p>
```

Example with border individual sides.



Borda para cada um dos lados

```
<style type="text/css">
  p.border-all {
    border: 3px solid red;
  }
  p.border-left{
    border-left: 4px solid blue;
    background-color: #dcdcdc;
  }
  p.border-top{
    border-top: 6px solid green;
    background-color: #dcdcdc;
  }
</style>
```

```
<p class="border-all">
  Example with a shorthand property for border-width, border-style,
  and border-color..
</p>
<p class="border-left">
  Example with a shorthand property for border-left-width, border-left-style,
  and border-left-color.
</p>
<p class="border-top">
  Example with a shorthand property for border-top-width, border-top-style,
  and border-top-color.
</p>
```

The border Shorthand Property

Example with a shorthand property for border-width, border-style, and border-color.

Example with a shorthand property for border-left-width, border-left-style, and border-left-color.

Example with a shorthand property for border-top-width, border-top-style, and border-top-color.

Bordas Arredondadas

- Podemos criar bordas arredondadas usando as propriedades border radius
- Existem 5 propriedades associadas com essa capacidade

Property	Description	Values
border-top-left-radius border-top-right-radius border-bottom-left-radius border-bottom-right-radius	Sets the radius for a single corner.	A pair of length or percentage values. The percentages relate to the width and height of the border box.
border-radius	This shorthand property sets all corners at once.	One or four pairs of length or percentage values, separated by a / character.

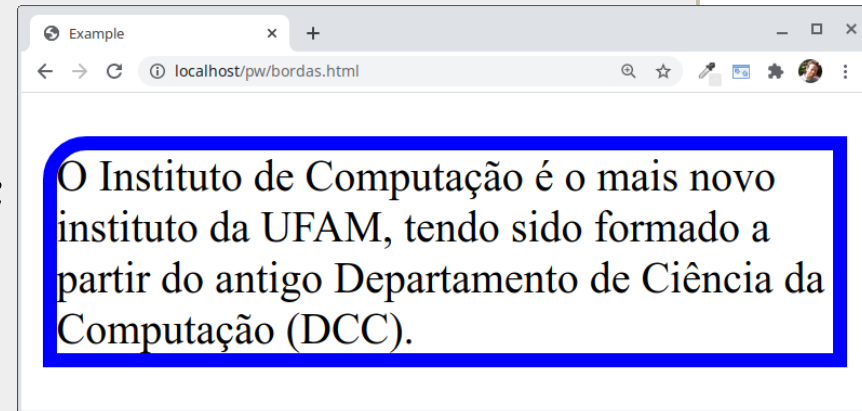


Bordas Arredondadas

- No exemplo abaixo, apenas a borda superior à esquerda será arredondada

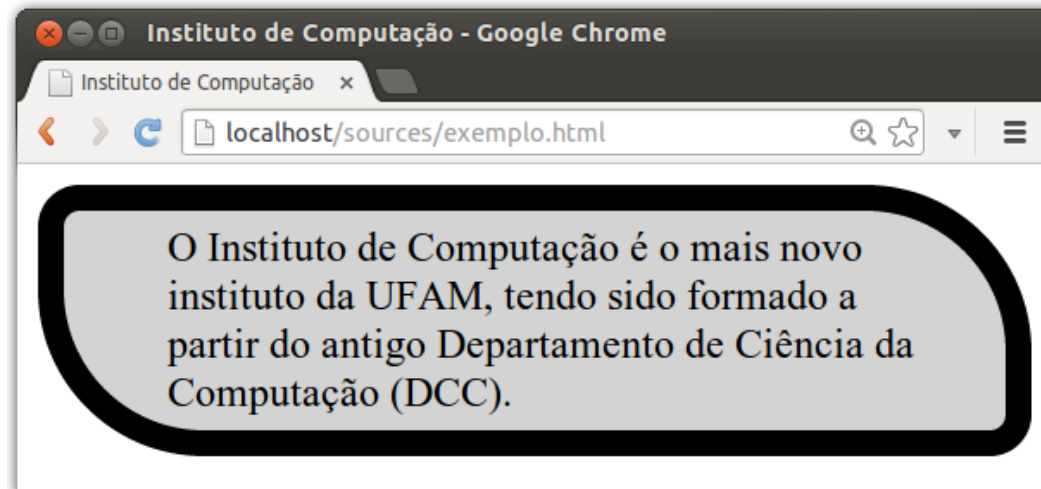
```
<style type="text/css">
  p {
    border: 5px solid blue;
    border-top-left-radius: 16px;
  }
</style>

<p>
  O Instituto de Computação é o mais novo
  instituto da UFAM, tendo sido formado a
  partir do antigo Departamento de Ciência
  da Computação (DCC).
</p>
```



Exercício

- Codifique uma página **html/css** cujo resultado seja similar ao da figura abaixo



Padding

- Padding (acolchoamento) é o espaço entre o conteúdo do elemento e sua borda



Padding

- É possível atribuir um valor de padding para cada lado de um elemento caixa, ou atribuir um único valor de padding para o elemento inteiro

Property	Description	Values
padding-top	Sets the padding for the top edge.	<i><length></i> or <i><%></i>
padding-right	Sets the padding for the right edge.	<i><length></i> or <i><%></i>
padding-bottom	Sets the padding for the bottom edge.	<i><length></i> or <i><%></i>
padding-left	Sets the padding for the left edge.	<i><length></i> or <i><%></i>
padding	This shorthand property sets the padding for all edges in a single declaration.	1–4 <i><length></i> or <i><%></i> values



Padding

- É possível atribuir um valor de padding para cada lado de um elemento caixa, ou atribuir um único valor de padding para o elemento inteiro

Property	Description	Values
padding-top	Sets the padding for the top edge	<length> or <%>
padding-right	Sets the padding for the right edge	<length> or <%>
padding-bottom	Sets the padding for the bottom edge	<length> or <%>
padding-left	Sets the padding for the left edge	<length> or <%>
padding	This shorthand property sets the padding for all four sides of the element. This shorthand property can be used with 1 to 4 values. If 1 value is used, it applies to all four sides. If 2 values are used, the first value applies to the top and bottom sides, and the second value applies to the left and right sides. If 3 values are used, the first value applies to the top and bottom sides, the second value applies to the left side, and the third value applies to the right side. If 4 values are used, each value applies to a specific side: top, right, bottom, and left, in that order.	1–4 <length> or <%> values

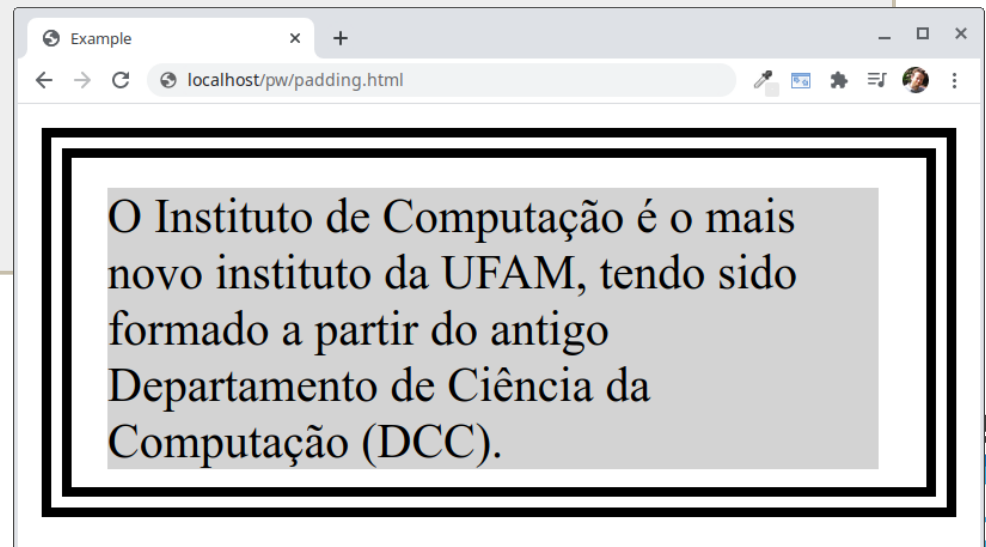
Quando informamos o **padding** usando porcentagens, a porcentagem é sempre relativa ao comprimento do elemento pai. A altura não é levada em conta.



Padding

- Declarando os valores de padding para cada lado

```
<style type="text/css">
div {
  border: 10px double black;
  background-color: lightgray;
  background-clip: content-box;
  padding-top: 10px;
  padding-right: 16px;
  padding-bottom: 6px;
  padding-left: 12px;
}
</style>
```

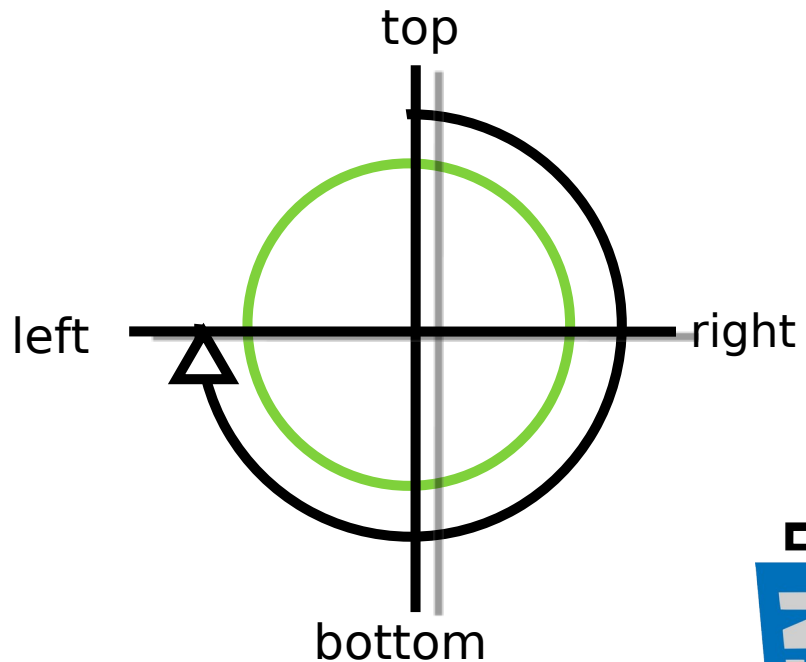


Padding

- O **padding** também pode ser feito em uma única declaração
- Para isso, basta usar o comando padding e definir os valores na seguinte ordem: topo, direita, fundo e esquerda

```
padding-top: 10px;  
padding-right: 16px;  
padding-bottom: 6px;  
padding-left: 12px;
```

```
padding: 10px 16px 6px 12px;
```



Os estilos acima
produzem o
mesmo efeito

Margin

- Margin é o espaço entre a borda do elemento e aquilo que o rodeia na página Web



Margin

- É possível atribuir um valor de margin para cada lado de um elemento caixa, ou atribuir um único valor de margin para o elemento inteiro

Property	Description	Values
margin-top	Sets the margin for the top edge.	<length> or <%>
margin-right	Sets the margin for the right edge.	<length> or <%>
margin-bottom	Sets the margin for the bottom edge.	<length> or <%>
margin-left	Sets the margin for the left edge.	<length> or <%>
margin	This shorthand property sets the margin for all edges in a single declaration.	1–4 <length> or <%> values



Margin

- É possível atribuir um valor de margin para cada lado de um elemento caixa, ou atribuir um único valor de margin para o elemento inteiro

Property	Description	Values
margin-top	Sets the margin for the top edge.	<length> or <%>
margin-right	Sets the margin for the right edge.	<length> or <%>
margin-bottom	Sets the margin for the bottom edge.	<length> or <%>
margin-left	Sets the margin for the left edge.	<length> or <%>
margin	This property sets the margin for all four sides of the element. This is a shorthand declaration.	1–4 <length> or <%> values

Quando informamos a **margem** usando porcentagens, a porcentagem é sempre relativa ao comprimento do elemento pai. A altura não é levada em conta.



Margin

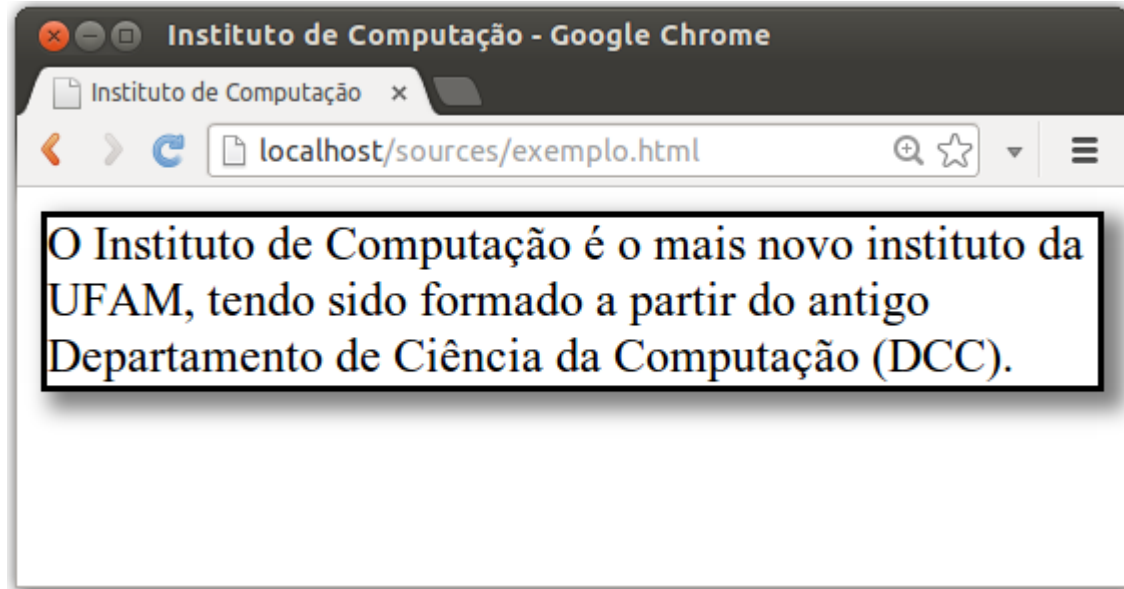
- Declarando os valores de margin vertical e horizontal

```
<style type="text/css">
img {
  border: 4px solid black;
  background: lightgray;
  margin: 4px 6px;
}
</style>
...
<body>
  
  
</body>
```



Sombra de Caixas

- Uma das ferramentas mais aguardadas do CSS3 foi a habilidade de adicionar sombras aos elementos
- A sombra é feita através da propriedade **box-shadow**, que veremos a seguir



Sombra de Caixas

- Os valores da propriedade box-shadow seguem o seguinte formato:

```
box-shadow: hoffset voffset blur spread color inset
```

- **hoffset**, a offset horizontal, que é um valor de comprimento
 - Valores positivos criam uma sombra pra direita, e negativos para a esquerda
- **voffset**, a offset vertical, que é um valor de comprimento
 - Valores positivos criam uma sombra pra baixo, e negativos para cima



Sombra de Caixas

- Os valores da propriedade box-shadow seguem o seguinte formato:

```
box-shadow: hoffset voffset blur spread color inset
```

- **blur**, especifica o tamanho do borrão. Quanto maior o valor, mais borrada fica a sombra. Opcional.
- **spread**, especifica o grau de espalhamento. Opcional.
- **color**, especifica a cor da sombra. Se omitido, o browser irá escolher a cor. Opcional.
- **inset**, causa uma sombra dentro da caixa, e não fora. Opcional.



Sombra de Caixas

- Exemplo de sombra externa à caixa:

```
<style type="text/css">
div {
  margin: 100px;
  border: medium solid black;
  box-shadow: 4px 4px 10px 2px gray;
}
</style>
```

Ordem dos valores:
**hoffset voffset blur
spread color inset**

O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação (DCC).



Sombra de Caixas

- Exemplo de sombra interna à caixa:

```
<style type="text/css">
div {
  margin: 100px;
  border: medium solid black;
  box-shadow: 2px 2px 4px 2px gray inset;
}
</style>
```

Ordem dos valores:
**hoffset voffset blur
spread color inset**

O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação (DCC).



Sombra de Caixas

- Exemplo de sombra interna à caixa

```
<style type="text/css">
div {
  margin: 100px;
  border: medium solid black;
  box-shadow: 4px 4px 10px 2px gray,
              2px 2px 4px 2px gray inset;
}
</style>
```

É possível definir múltiplas sombras em uma única declaração da propriedade box-shadow

Ordem dos valores:
hoffset voffset blur spread color inset

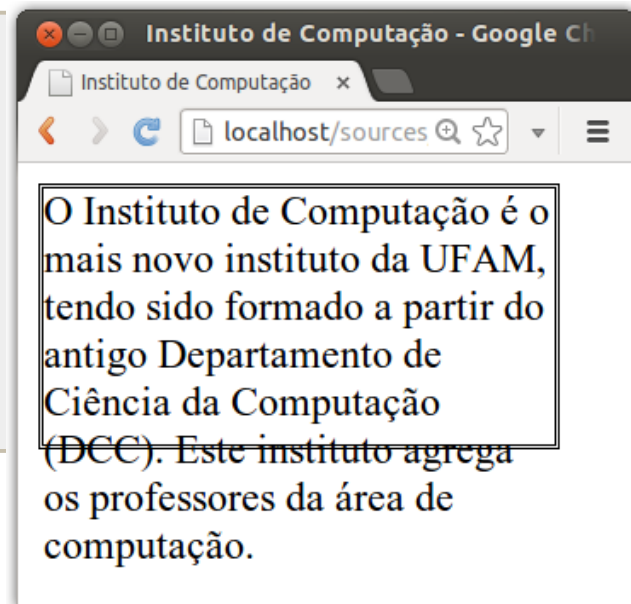
O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação (DCC).



Lidando com Transbordamento

- Quando diminuámos o tamanho de um elemento, seu conteúdo pode não caber mais em seu interior
- Por padrão, quando isso ocorre, o browser deixa o conteúdo do elemento transbordar para fora

```
<style type="text/css">
div {
  width: 200px;
  height: 100px;
  border: medium double;
}
</style>
```

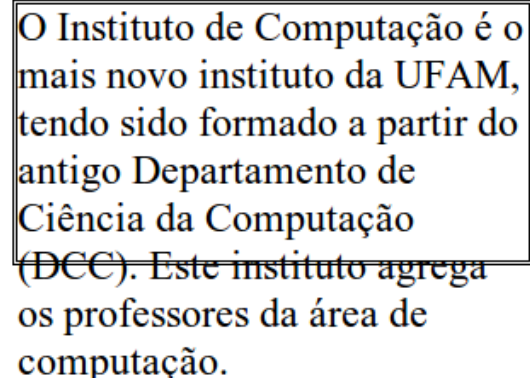


Lidando com Transbordamento

- Quando diminuimos o tamanho de um elemento, seu conteúdo pode não caber mais em seu interior

- Por padrão, esse comportamento pode ser mudado através das propriedades **overflow**

```
<style>  
div {  
  width: 200px;  
  height: 100px;  
  border: medium double;  
}  
</style>
```



O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação (DCC). Este instituto agrega os professores da área de computação.



CSS overflow

- A **propriedade overflow** define o conteúdo do comportamento que transborda a caixa do elemento. Esta propriedade funciona apenas para elementos de bloco que têm uma altura especificada
- Ele especifica se o conteúdo deve ser cortado para caber na caixa ou se as barras de rolagem devem ser adicionadas ao elemento



CSS overflow

- Propriedades **overflow** existentes

Property	Description
overflow-x overflow-y	Set the horizontal or vertical overflow style.
overflow	Shorthand property.



CSS overflow

- Possíveis valores para a propriedade overflow:

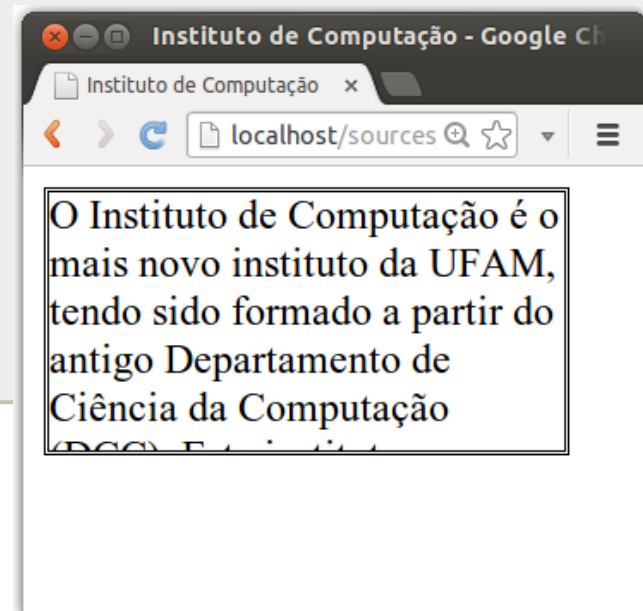
Valor	Descrição
visible	<i>Default.</i> O estouro não é cortado. O conteúdo renderiza fora da caixa do elemento
hidden	O estouro é cortado e o restante do conteúdo ficará invisível
scroll	O estouro é cortado e uma barra de rolagem é adicionada para ver o restante do conteúdo
auto	Semelhante ao scroll, mas adiciona barras de rolagem somente quando necessário



CSS Overflow

- **overflow: hidden** - o conteúdo é truncado, e apenas a porção dentro da caixa é apresentada

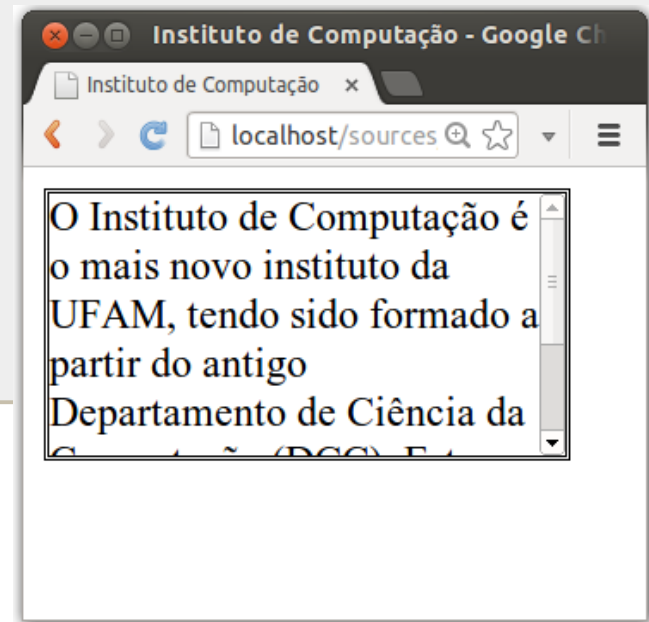
```
<style type="text/css">
div {
  width: 200px;
  height: 100px;
  overflow: hidden;
  border: medium double;
}
</style>
```



CSS Overflow

- **overflow: auto** – a barra de rolagem aparece caso haja transbordamento

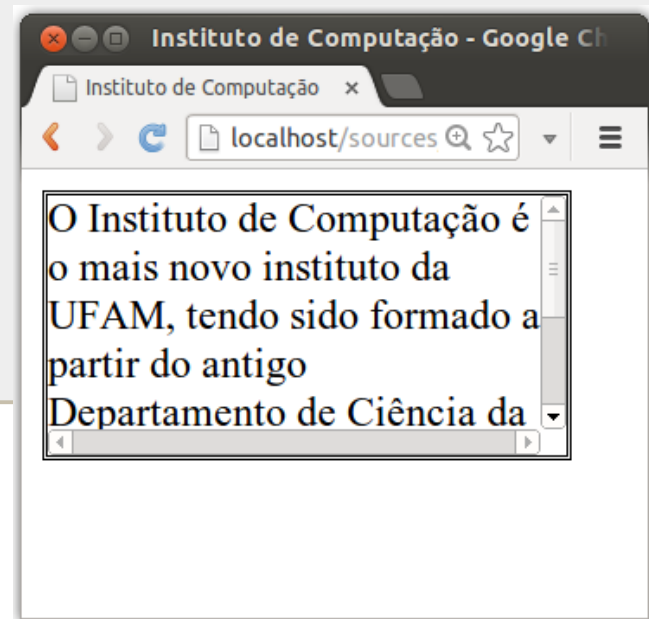
```
<style type="text/css">
div {
  width: 200px;
  height: 100px;
  overflow: auto;
  border: medium double;
}
</style>
```



CSS Overflow

- **overflow: scroll** – a barra de rolagem aparece, havendo transbordamento ou não

```
<style type="text/css">
div {
  width: 200px;
  height: 100px;
  overflow: scroll;
  border: medium double;
}
</style>
```



Overflow-x & overflow-y

- As propriedades **overflow-x** e **overflow-y** especificam se o estouro do conteúdo deve ser alterado apenas horizontalmente ou verticalmente (ou ambos)
- **Overflow-x:** especifica o que fazer com as bordas esquerda/direita do conteúdo.
- **Overflow-y:** especifica o que fazer com as bordas superior/inferior do conteúdo.



Overflow-x & overflow-y

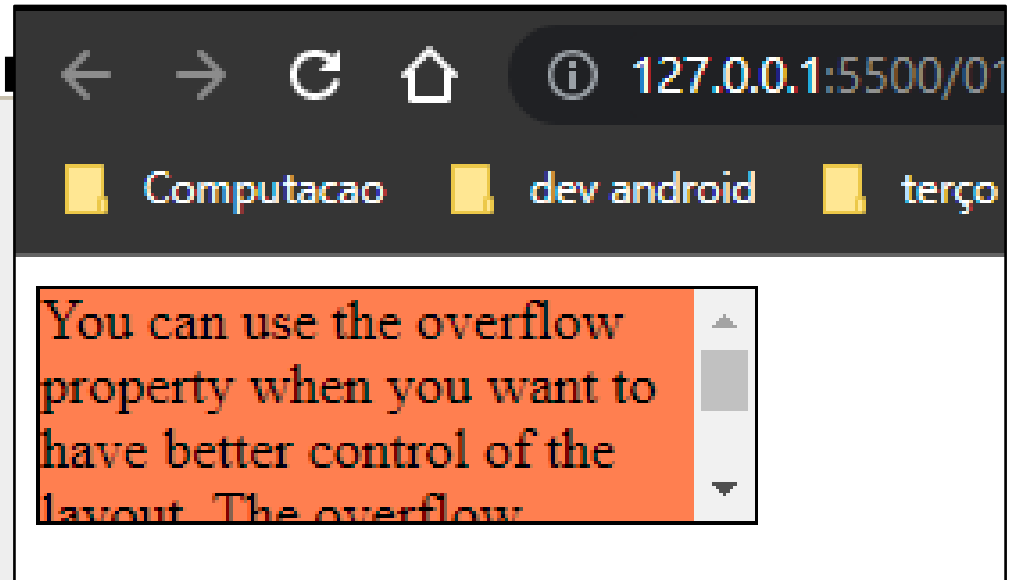
- As propriedades **overflow-x** e **overflow-y**

```
<style type="text/css">
div {
  background-color: coral;
  Width: 200px;
  height: 65px;
  overflow-x: hidden;
  overflow-y: scroll;
  border: 1px solid black;
}
</style>
```

```
<div>
```

You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's Box.

```
</div>
```



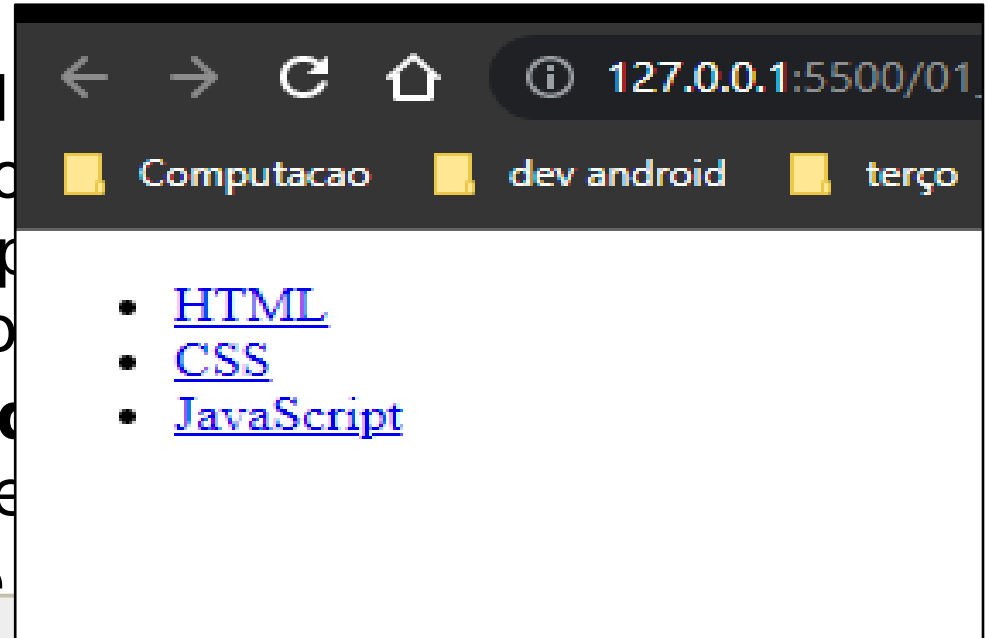
CSS Display

- Cada elemento HTML tem um valor de exibição padrão, dependendo do tipo de elemento. O valor de exibição padrão para a maioria dos elementos é bloco ou embutido.
- Ou seja, **propriedade display** define os tipos de exibição internos e externos de um elemento.
 - **Externo:** define a participação de um elemento no fluxo do layout
 - **Interno:** define o layout dos filhos.
- Ou seja, **propriedade display** define os tipos de exibição internos e externos de um elemento.



CSS Display

- Cada elemento HTML tem um padrão, dependendo de exibição padrão padrão é bloco ou embutido
- Ou seja, **propriedades** de exibição internos e externos
 - **Externo:** define



```
<ul>
  <li><a href="#" target="_blank">HTML</a></li>
  <li><a href="#" target="_blank">CSS</a></li>
  <li><a href="#" target="_blank">Javascript</a></li>
</ul>
```

de

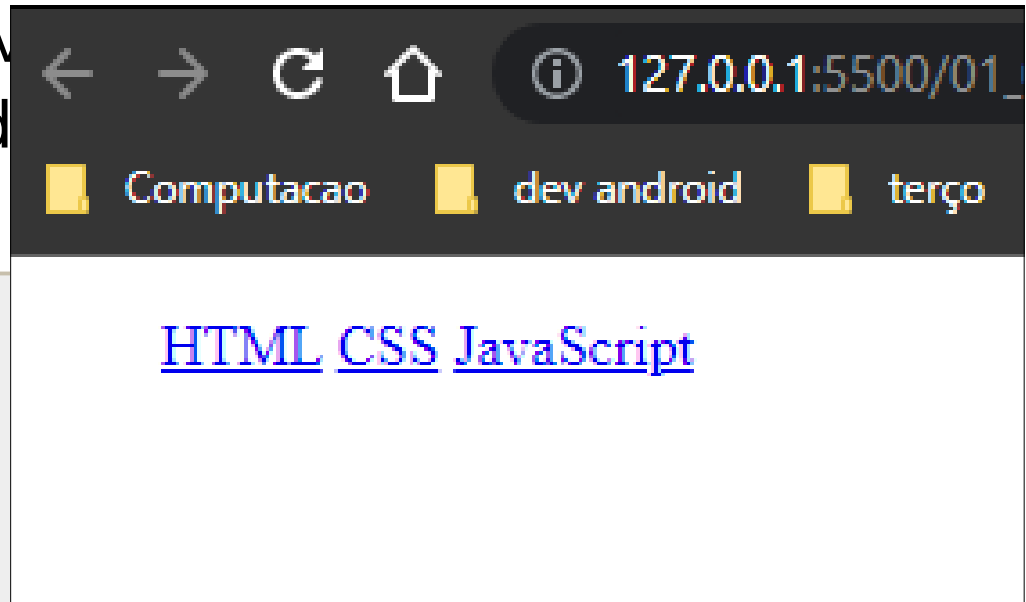


CSS Display

- Cada elemento HTML tem um padrão, dependendo de exibição padrão. É `block` ou `inline`.

```
<style type="text/css">
  li {
    display: inline;
  }
</style>
```

```
<ul>
  <li><a href="#" target="_blank">HTML</a></li>
  <li><a href="#" target="_blank">CSS</a></li>
  <li><a href="#" target="_blank">Javascript</a></li>
</ul>
```



s de

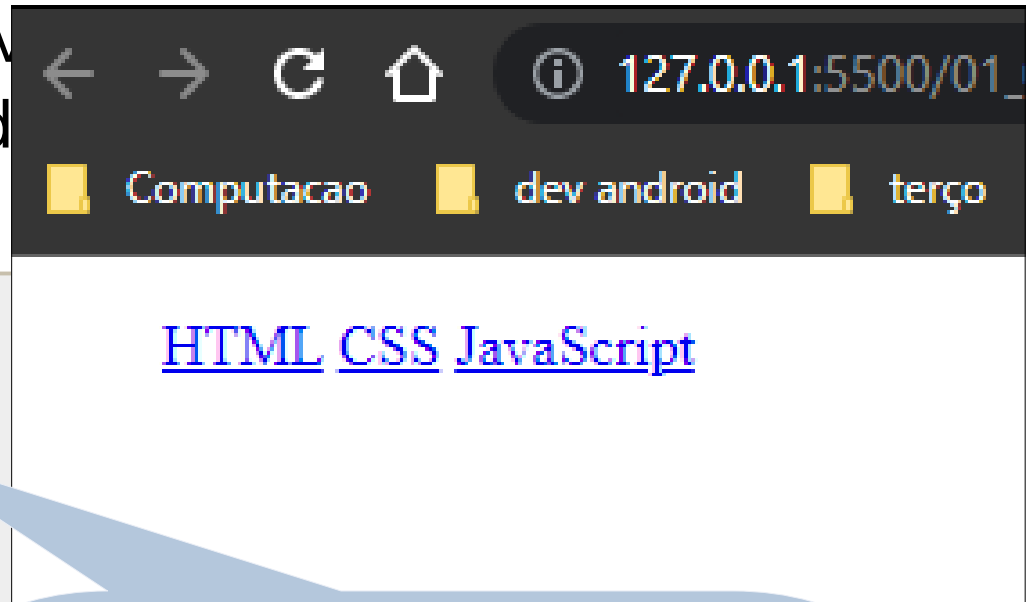


CSS Display

- Cada elemento HTML tem um padrão, dependendo de exibição padrão. É bloco ou embutido.

```
<style type="text/css">
  li {
    display: inline;
  }
</style>
```

```
<ul>
  <li><a href="#" target="#">
  <li><a href="#" target="#">
  <li><a href="#" target="#">
</ul>
```



Definir a propriedade de exibição de um elemento altera apenas como o elemento é exibido, NÃO o tipo de elemento. Então, um elemento inline com display: block; não é permitido ter outros elementos de bloco dentro dele.

de



CSS Display

- Quando usamos **display: block**, é criado um elemento do tipo bloco
 - Este elemento é horizontalmente separado dos demais elementos que estão à sua volta
 - O elemento faz uma quebra de linha antes e depois de si mesmo
- O elemento p, usado para criar parágrafos, inclui essa propriedade/valor em seu estilo padrão



CSS Display

- Usando o valor **block** para a propriedade display

```
<style type="text/css">
div {
  border: medium solid black
}
span {
  display: block;
  border: medium double black;
  margin: 2px;
}
</style>
```

...

```
<div>
```

O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo ``Departamento de Ciência da Computação(DCC)``. Esse instituto agrega os professores da área de computação.

```
</div>
```

CSS Display

- Usando o valor **block** para a propriedade display

```
<style type="text/css">
```

```
div {  
  border: medi
```

```
span {  
  display: blo  
  border: medi  
  margin: 2px;  
}
```

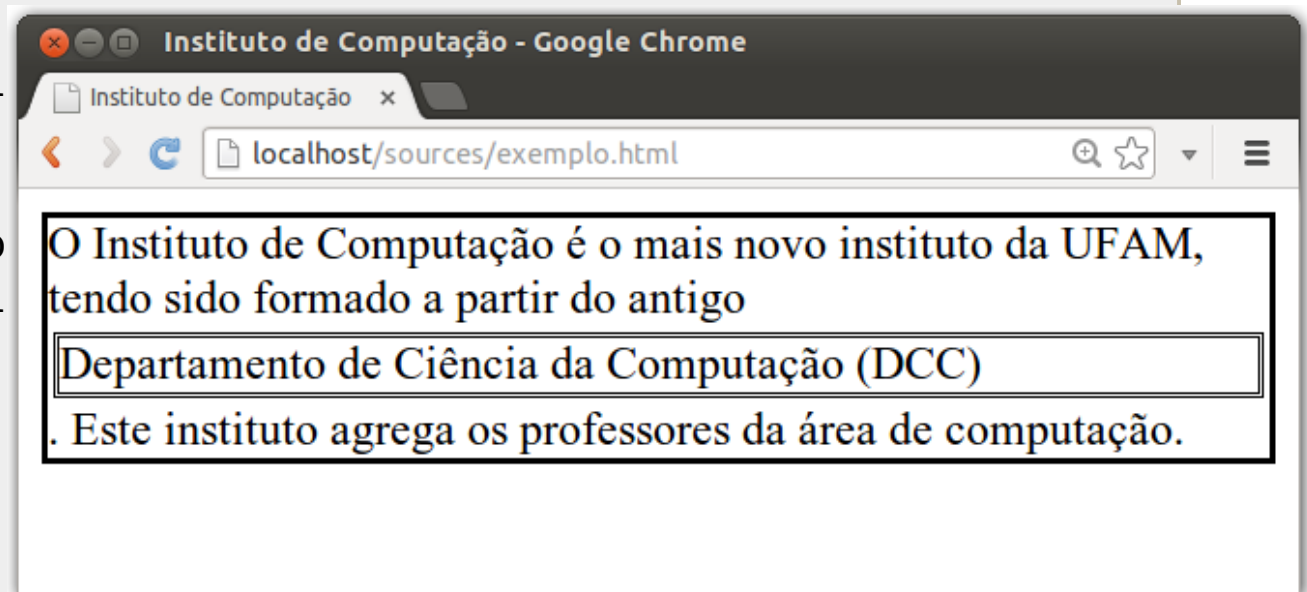
```
</style>
```

```
...
```

```
<div>
```

O Instituto de Computação e o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação(DCC). Esse instituto agrega os professores da área de computação.

```
</div>
```



CSS Display

- Usando o valor **block** para a propriedade display

```
<style type="text/css">
```

```
div {  
  border: medi
```

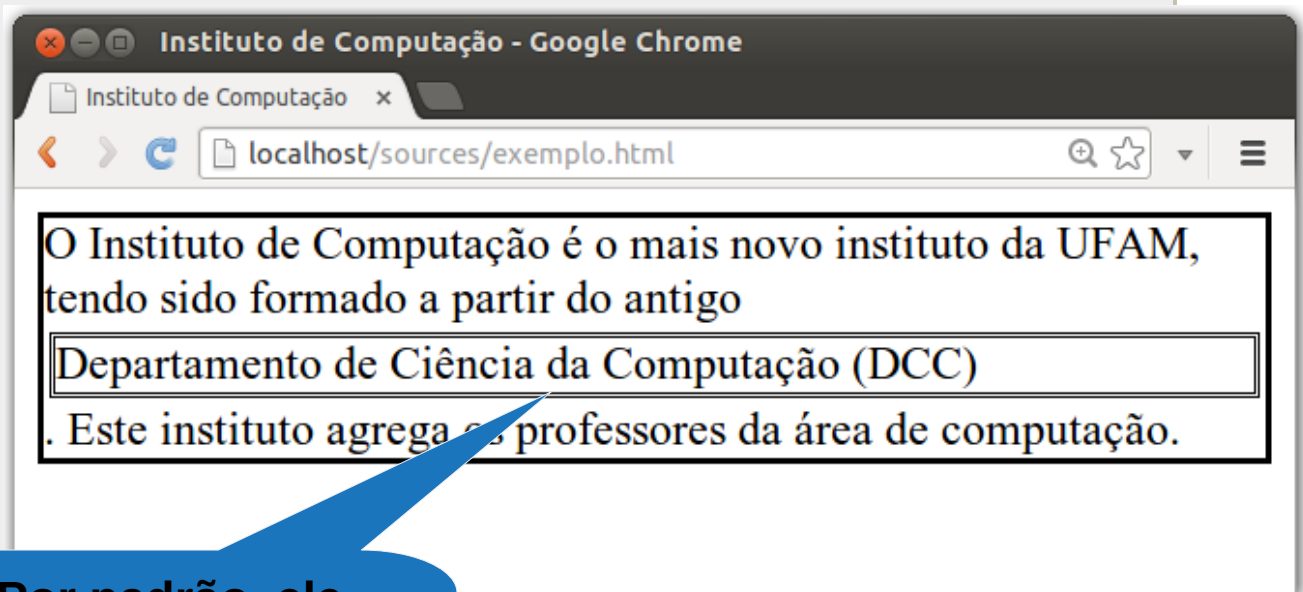
```
span {  
  display: blo  
  border: medi  
  margin: 2px;  
}  
</style>
```

```
...
```

```
<div>
```

```
O Institut  
tendo sid  
de Ciênci  
agrega os  
</div>
```

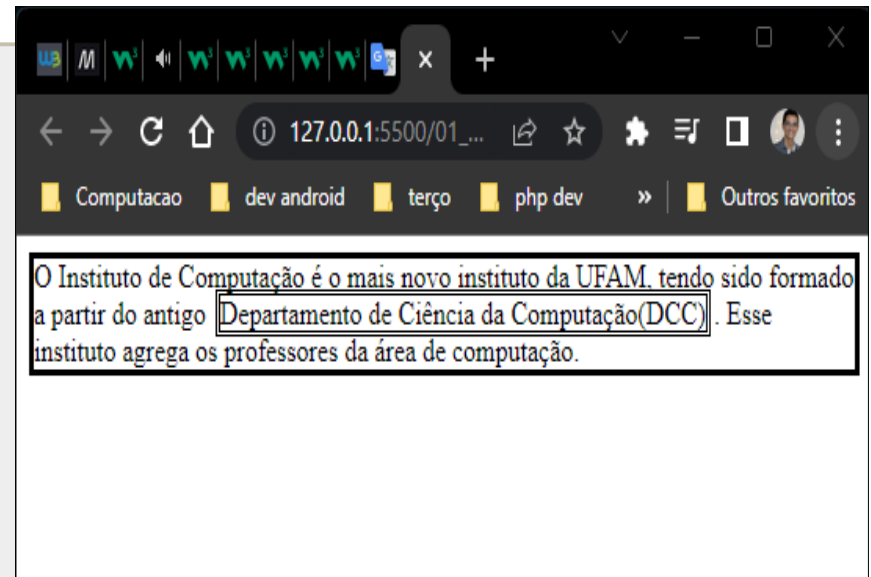
Por padrão, ele
ocupam o máximo de
espaço horizontal
possível.



CSS Display

- Quando usamos **display: inline**, o elemento não se separa do fluxo do conteúdo
- Certas propriedades típicas de blocos, tais como **width** e **height**, são ignoradas

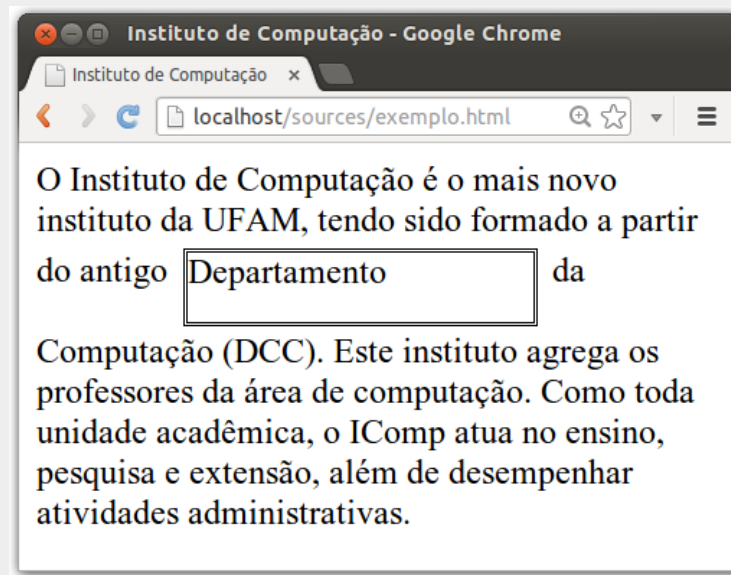
```
<style type="text/css">
div {
  display: inline;
}
span {
  display: inline;
  border: medium double black;
  margin: 20em;
  width: 100em;
  height: 200em;
}
</style>
```



CSS Display

- Usando **display: inline-block**, criamos um elemento com características bloco e inline
 - Não existe quebras de linha antes e depois do elemento
 - Entretanto, dentro do elemento, propriedades tais como **width** e **height** podem ser aplicadas

```
<style type="text/css">
div {
  display: inline;
}
span {
  display: inline-block;
  border: medium double;
  width: 10em;
  height: 2em;
  margin: 3px;
}
</style>
```



CSS Display

- Quando **display: none**, o elemento não aparece na página

```
<p id="toggle">O Instituto de Computação é o mais  
novo instituto a UFAM, tendo sido formado a partir  
do antigo Departamento de Ciência da Computação (DCC).</p>  
  
<p>Este instituto agrega os professores da área de computação.  
Como toda unidade acadêmica, o IComp atua no ensino, pesquisa  
e extensão, e desempenha atividades administrativas.</p>  
  
<button>Block</button><button>None</button>  
  
<script>  
  var buttons = document.getElementsByTagName("BUTTON");  
  for (var i = 0; i < buttons.length; i++) {  
    buttons[i].onclick = function(e) {  
      document.getElementById("toggle").style.display=  
        e.target.innerHTML;  
    };  
  }  
</script>
```



CSS Display

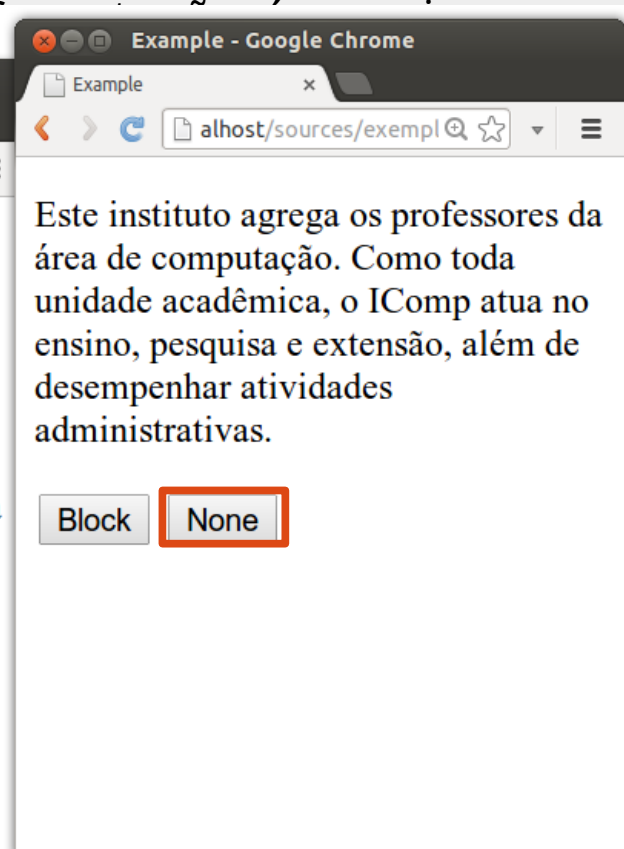
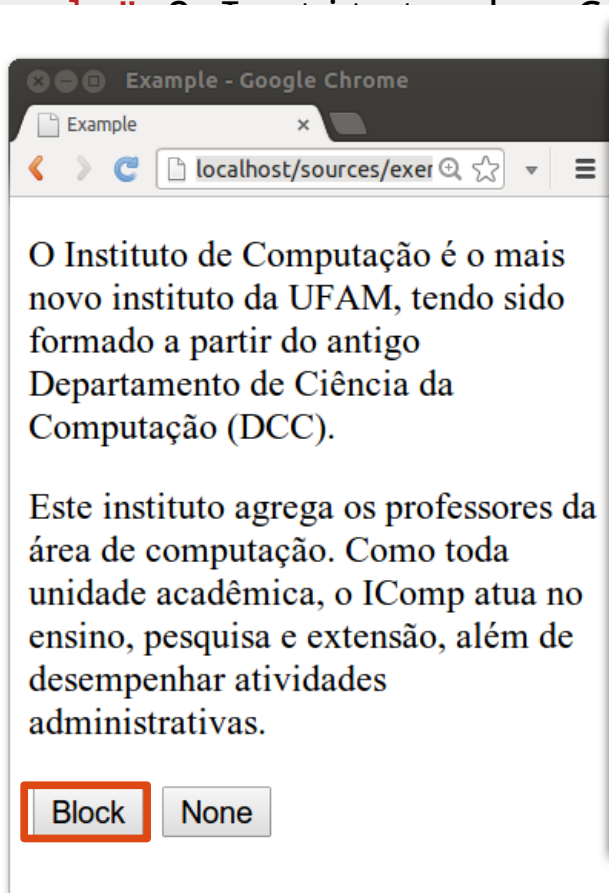
- Quando **display: none**, o elemento não

```
<p id="t  
novo ins  
do antig
```

```
<p>Este  
Como tod  
e extens
```

```
<button>
```

```
<script>  
var bu  
for (v  
butt  
d  
e  
};  
}  
</script>
```



>
ação.
uiza



CSS Display

- Crie os seguintes blocos.
- **#DICA:** utilize as técnicas de **display** para encontrar o formato correto.
- **Cores utilizadas:** `rgb(239, 116, 111); #ECF081; MediumSeaGreen; rgb(171, 62, 91);`
- **Medidas:** largura 200px; Altura 100px;



Posicionamento de conteúdo

- A propriedade **position** define o método com que um elemento é posicionado na página
- Possíveis valores da propriedade **position**:

Valor	Descrição
<code>static</code>	Os elementos são colocados de acordo com o fluxo normal do documento. Este é o valor padrão desta propriedade
<code>absolute</code>	Os elementos são removidos do fluxo do documento e posicionados em relação ao seu elemento ancestral posicionado
<code>fixed</code>	Os elementos são removidos do fluxo do documento e posicionados em relação à janela do navegador.
<code>relative</code>	Elemento são colocados em relação à sua posição normal.



Posicionamento de conteúdo

- Exemplo de uso do **position**

```
img {  
  top: 20px;  
  left: 200px;  
}
```

Podemos usar as propriedades **top**, **bottom**, **left**, e **right** para deslocar o elemento especificado pela propriedade **position**

```
  
<button>Static</button>  
<button>Relative</button>  
<button>Absolute</button>  
<button>Fixed</button>
```

```
const buttons = document.getElementsByTagName("BUTTON");  
const cssLogo = document.getElementById("css-logo");  
for (var i = 0; i < buttons.length; i++) {  
  buttons[i].onclick = function(e) {  
    cssLogo.style.position = e.target.innerHTML;  
  };  
}
```



Posicionamento de conteúdo

Ex

tion


Document

localhost/pw/css/

Instituto de Computação

O Instituto de Computação (IComp), antigo Departamento de Ciência da Computação (DCC), é um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico o IComp atua no ensino, pesquisa e extensão, além de desempenhar atividades administrativas.



No ensino de graduação, o DCC atua principalmente nos cursos de Ciência da Computação, Sistemas de Informação, Engenharia de Software e Engenharia da Computação. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

Static

Relative

Absolute

Fixed

pq.png" alt="css"/>

```
ntsByTagName("BUTTON");  
ntById("css-logo");  
h; i++) {  
) {  
rget.innerHTML;
```



Posicionamento de conteúdo

Ex


```
im
}
<i
<b
<b
<b
<b
CO
CO
fo
}

```

Instituto de C

O Instituto de Computação (de Ciência da Computação (que agrega os professores da

Como todo instituto acadêm pesquisa e extensão, além de administrativas.




No ensino de graduação, o I cursos de Ciência da Compu Engenharia de Software e E Também oferece disciplinas cursos de graduação da UFA

☒ Static ☐ Relative ☐ Absolute

Instituto de Computação

O Instituto de Computação (IComp), antigo Departamento de Ciência da Computação (DCC), é um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico o IComp atua no ensino, pesquisa e extensão, além de desempenhar atividades administrativas.




No ensino de graduação, o DCC atua principalmente nos cursos de Ciência da Computação, Sistemas de Informação, Engenharia de Software e Engenharia da Computação. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

☐ Static ☒ Relative ☐ Absolute ☐ Fixed

```
SS"/>
BUTTON");
go");

```



Posicionamento de conteúdo

Ex

```
im
}
<i
<b
<b
<b
<b
CO
CO
fo
}

```

Instituto de Computação

O Instituto de Computação (IComp) é o Departamento de Ciência da Computação (DCC), um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico, além de desempenhar atividades administrativas.

No ensino de graduação, o DCC atua principalmente nos cursos de Ciência da Computação, Sistemas de Informação, Engenharia de Software e Engenharia da Computação. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

☐ Static ☐ Relative ☒ Absolute ☐ Fixed

Instituto de Computação

O Instituto de Computação (IComp) é o Departamento de Ciência da Computação (DCC), um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico, além de desempenhar atividades administrativas.

No ensino de graduação, o DCC atua principalmente nos cursos de Ciência da Computação, Sistemas de Informação, Engenharia de Software e Engenharia da Computação. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

☐ Static ☒ Relative ☐ Absolute ☐ Fixed

Instituto de Computação

O Instituto de Computação (IComp) é o Departamento de Ciência da Computação (DCC), um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico, além de desempenhar atividades administrativas.

No ensino de graduação, o DCC atua principalmente nos cursos de Ciência da Computação, Sistemas de Informação, Engenharia de Software e Engenharia da Computação. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

☐ Static ☐ Relative ☒ Absolute ☐ Fixed



Posicionamento de conteúdo

Ex

The image displays three overlapping browser windows, each showing a web page titled "Instituto de Computação" (Institute of Computing). The page content includes a CSS logo, a paragraph about the institute, and a section about undergraduate teaching. The windows illustrate different CSS positioning settings for the content elements:

- Left Window (Static):** The "Static" button is highlighted in red. The content is positioned normally, overlapping the other windows.
- Middle Window (Relative):** The "Relative" button is highlighted in red. The content is slightly offset from its normal position.
- Right Window (Fixed):** The "Fixed" button is highlighted in red. The content is locked in place, appearing on top of the other windows.

Each window also shows a CSS logo and a paragraph of text: "O Instituto de Computação (IComp) é um instituto acadêmico que agrega os professores da área de computação." and "Como todo instituto acadêmico o IComp atua no ensino, pesquisa e extensão, além de desempenhar atividades administrativas."



Posicionamento absoluto

- A propriedade **position:absolute** provê grande controle sobre o local de um elemento na página
- Normalmente, os elementos são posicionados na página na ordem em que aparecem no HTML
- Especificar a posição de um elemento como absoluta ignora o fluxo normal dos elementos
- Os elementos são dispostos de acordo com a distância do topo, da esquerda, da direita ou do fundo das páginas



Posicionamento absoluto

- Para compreender o posicionamento absoluto, considere as duas imagens abaixo



Posicionamento absoluto

- Podemos usar a propriedade **position: absolute** para colocar uma imagem em cima da outra

```
<style type="text/css">
  .bg_image {
    position: absolute;
    top: 10px;
    left: 10px;
    z-index: 1;
  }
  .fg_image {
    position: absolute;
    top: 35px;
    left: 110px;
    z-index: 2;
  }
</style>
```

```
<body>
  
  
</body>
```



Posicionamento absoluto

- A propriedade **z-index** permite criar elementos sobrepostos
 - Elementos com z-index maiores são mostrados na frente dos elementos com z-index menores

```
<style type="text/css">
.bg_image {
  position: absolute;
  top: 10px;
  left: 10px;
  z-index: 1;
}
.fg_image {
  position: absolute;
  top: 35px;
  left: 110px;
  z-index: 2;
}
</style>
```

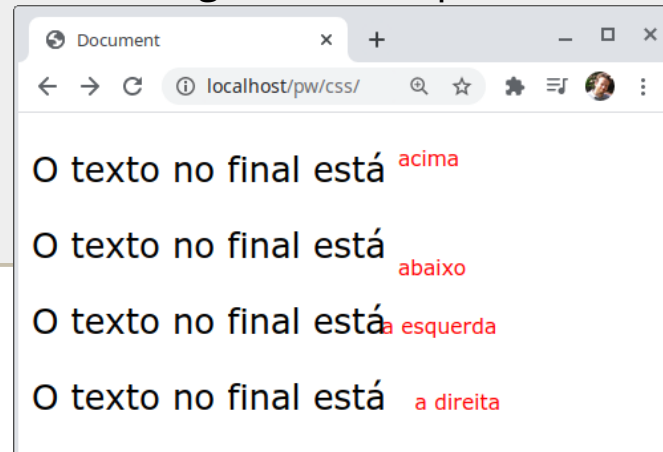


Posicionamento relativo

- A propriedade **position: relative** posiciona um elemento em relação a sua posição original

```
<p>O texto no final está <span class="super">acima</span></p>
<p>O texto no final está <span class="sub">abaixo</span></p>
<p>O texto no final está <span class="shiftleft">a esquerda</span></p>
<p>O texto no final está <span class="shiftright">a direita</span></p>

<style type="text/css">
  p { font-size: 1.3em; font-family: verdana, arial, sans-serif; }
  span { color: red; font-size: 0.6em; height: 1em; position: relative; }
  .super { top: -10px; }
  .sub { bottom: -10px; }
  .shiftleft { left: -10px; }
  .shiftright { right: -10px; }
</style>
```



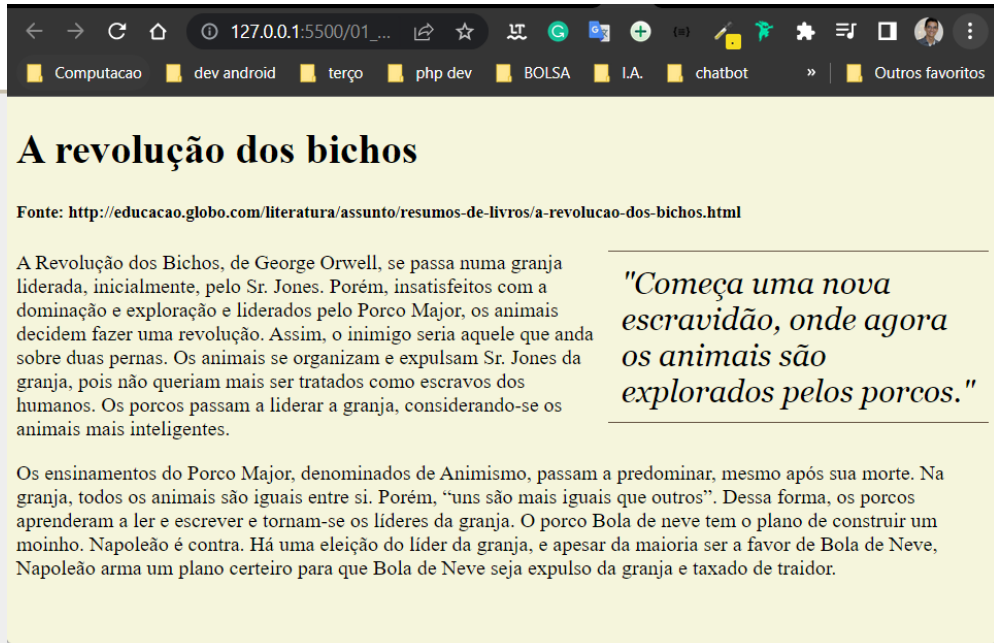
Elementos Flutuantes

- A propriedade float permite pegar um elemento no fluxo normal e colocá-lo o mais à esquerda ou à direita possível do elemento que o contém.
- Qualquer outra coisa que esteja dentro do elemento recipiente irá fluir ao redor do elemento que está flutuando.



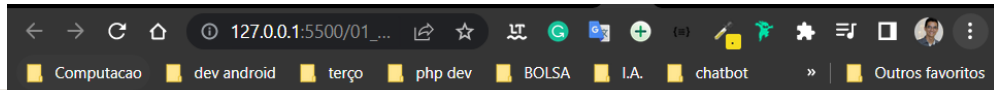
Elementos Flutuantes

```
<style type="text/css">
  body {
    background-color: beige;
  }
  blockquote{
    float: right;
    width: 275px;
    font-size: 150%;
    font-style: italic;
    font-family: Georgia, Times, serif;
    margin: 0px 0px 10px 10px;
    padding: 10px;
    border-top: 1px solid #665544;
    border-bottom: 1px solid #665544;
  }
</style>
<body>
  <h1> A Revolução dos Bichos </h1>
  ...
  <blockquote>
    "Começa uma nova escravidão, onde agora os animais
    são explorados pelos porcos."
  </blockquote>
  ...
  <p> ... </p>
</body>
```



Elementos Flutuantes

```
<style type="text/css">
  body {
    background-color: beige;
  }
  blockquote{
    float: right;
    width: 275px;
    font-size: 150%;
    font-style: italic;
    font-family: Georgia, Times, serif;
    margin: 0px 0px 10px 10px;
    padding: 10px;
    border-top: 1px solid #665544,
    border-bottom: 1px solid #665544,
  }
</style>
<body>
  <h1> A Revolução dos Bichos </h1>
  ...
  <blockquote>
    "Começa uma nova escravidão, onde
    são explorados pelos porcos."
  </blockquote>
  ...
  <p> ... </p>
</body>
```



A revolução dos bichos

Fonte: <http://educacao.globo.com/literatura/assunto/resumos-de-livros/a-revolucao-dos-bichos.html>

A Revolução dos Bichos, de George Orwell, se passa numa granja liderada, inicialmente, pelo Sr. Jones. Porém, insatisfeitos com a dominação e exploração e liderados pelo Porco Major, os animais decidem fazer uma revolução. Assim, o inimigo seria aquele que anda sobre duas pernas. Os animais se organizam e expulsam Sr. Jones da granja, pois não queriam mais ser tratados como escravos dos humanos. Os porcos passam a liderar a granja, considerando-se os animais mais inteligentes.

"Começa uma nova escravidão, onde agora os animais são explorados pelos porcos."

Os ensinamentos do Porco Major, denominados de Animismo, passam a predominar, mesmo após sua morte. Na granja, todos os animais são iguais entre si. Porém, "uns são mais iguais que outros". Dessa forma, os porcos aprenderam a ler e escrever e tornam-se os líderes da granja. O porco Bola de neve tem o plano de construir um moinho. Napoleão é contra. Há uma eleição do líder da granja, e apesar da maioria ser a favor de Bola de Neve, Napoleão arma um plano certo para que Bola de Neve seja expulso da granja e taxado de traidor.

Ao usar a propriedade float, você também deve usar a propriedade width para indicar a largura do elemento flutuante.



Elementos Flutuantes

- A propriedade float pode ter um dos seguintes valores:
- **Left:** O elemento flutua à esquerda de seu contêiner
- **Right:** O elemento flutua à direita de seu contêiner
- **None:** O elemento não flutua (será exibido apenas onde ocorre no texto). Isso é padrão
- **Inherit:** O elemento herda o valor flutuante de seu pai



Containers Flutuantes

- Muitos Layout de páginas web, possuem seções uma do lado das outras.
- Isso é obtido usando um elemento `<div>` para representar cada coluna.

The Evolution of the Bicycle

The First Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

Further Innovations

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

Bicycle Timeline

- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycle
- 1876: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

Containers Flutuantes

```
<style type="text/css">
```

```
section{  
  box-sizing: border-box;  
  border: 1px solid blue;  
  width: 100%;  
  float:left;}
```

```
div{  
  margin: 5px;  
  width: 50px;  
  height: 150px;}
```

```
.left{  
  float: left;  
  background: pink;}
```

```
.right{  
  float: right;  
  background-color: cyan;  
}
```

```
</style>
```

```
<body>
```

```
<section>
```

```
<div class="left">1</div>
```

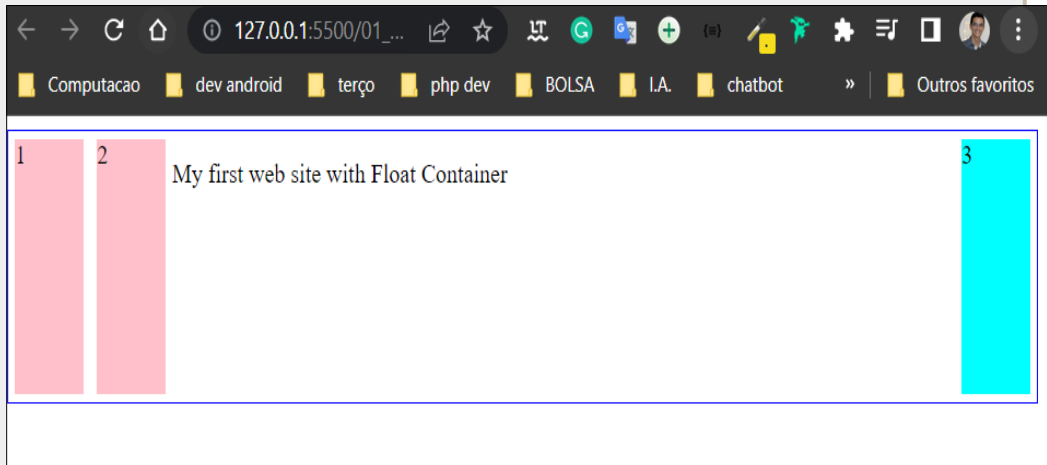
```
<div class="left">2</div>
```

```
<div class="right">3</div>
```

```
<p> My first Web Site with Float Container <p>
```

```
</section>
```

```
</body>
```



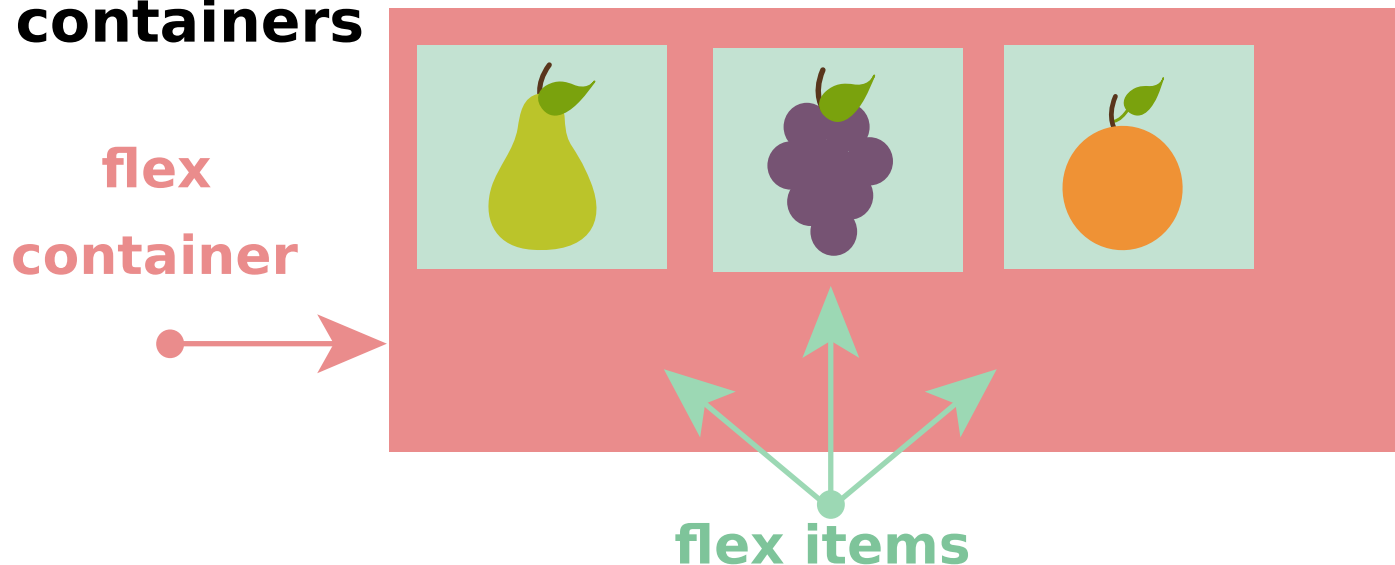
Containers Flutuantes

- Como escolher entre POSITION e FLOAT?
- **float** não interromperá o fluxo do documento
- **position** precisa calcular o fluxo
- **position:relative** move o elemento um pouco para longe da sua localização natural
- **float** faz o elemento aparecer mais à esquerda ou à direita no elemento pai.



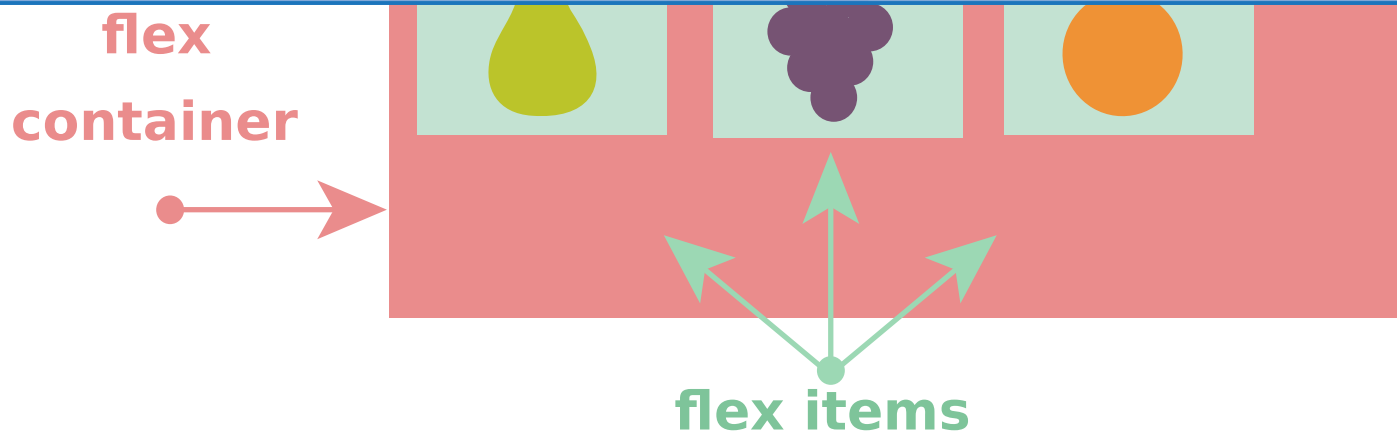
Layout Flexbox

- Os métodos de display vistos até então, block e inline, são pouco flexíveis e não atendem a requisitos muito comuns
- O **Flexbox** é um método de display do CSS3 que permite organizar os itens de uma página dentro de **containers**



Layout Flexbox

- Os métodos de display vistos até então, block e inline, são pouco flexíveis e não atendem a requisitos muito comuns
 - O **Flexbox** é um método de display do CSS3 que
- Algumas propriedades CSS do flexbox estão relacionadas com o **container**, enquanto outras com os **itens**



Layout Flexbox

- A ideia principal por trás do layout flexível é dar ao contêiner a capacidade de alterar a largura/altura (e a ordem) de seus itens para melhor preencher o espaço disponível (principalmente para acomodar todos os tipos de dispositivos de exibição e tamanhos de tela).
- Um contêiner flexível expande os itens para preencher o espaço livre disponível ou os reduz para evitar o estouro.
- É independente de direção, em oposição aos layouts regulares. Embora funcionem bem para páginas, eles carecem de flexibilidade para oferecer suporte a aplicativos grandes ou complexos



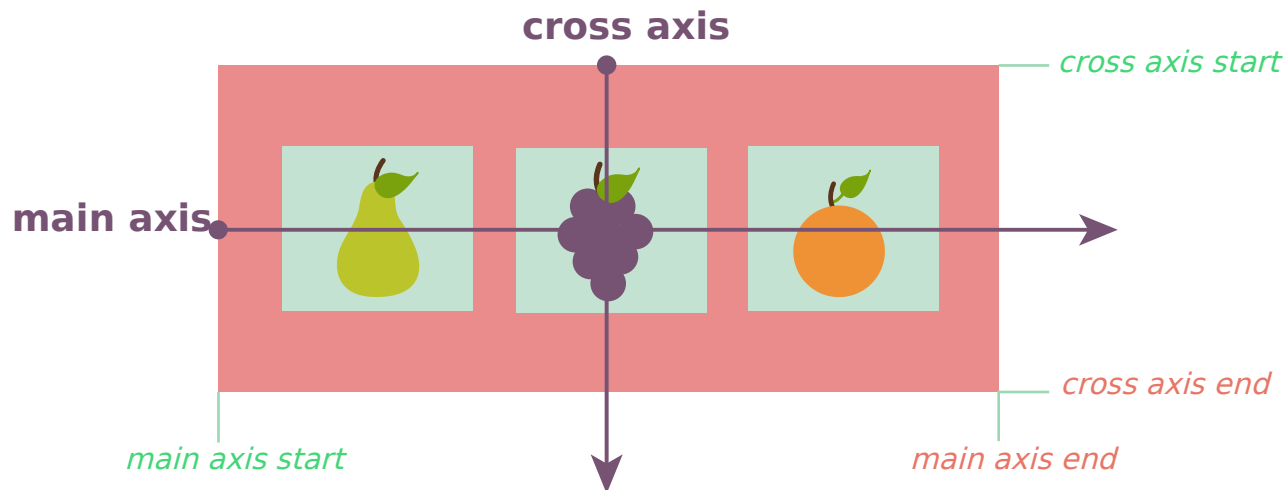
Propriedades do Flex Container

- **display:** define um flex container, habilitando o contexto flex para todos os seus filhos (flex itens)

```
.container { display: flex | inline-flex; }
```

- **flex-direction:** estabelece o main-axis, definindo a direção dos itens dentro de um flex container

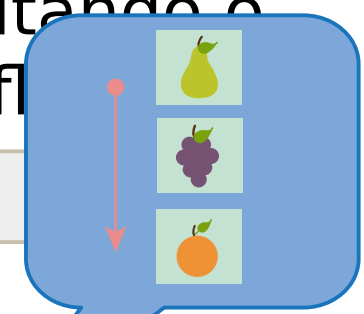
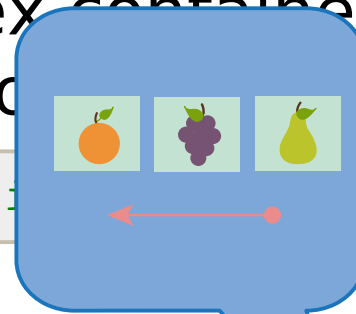
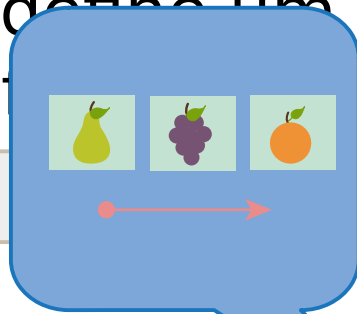
```
.container { flex-direction: row | row-reverse | column | column-reverse; }
```



Propriedades do Flex Container

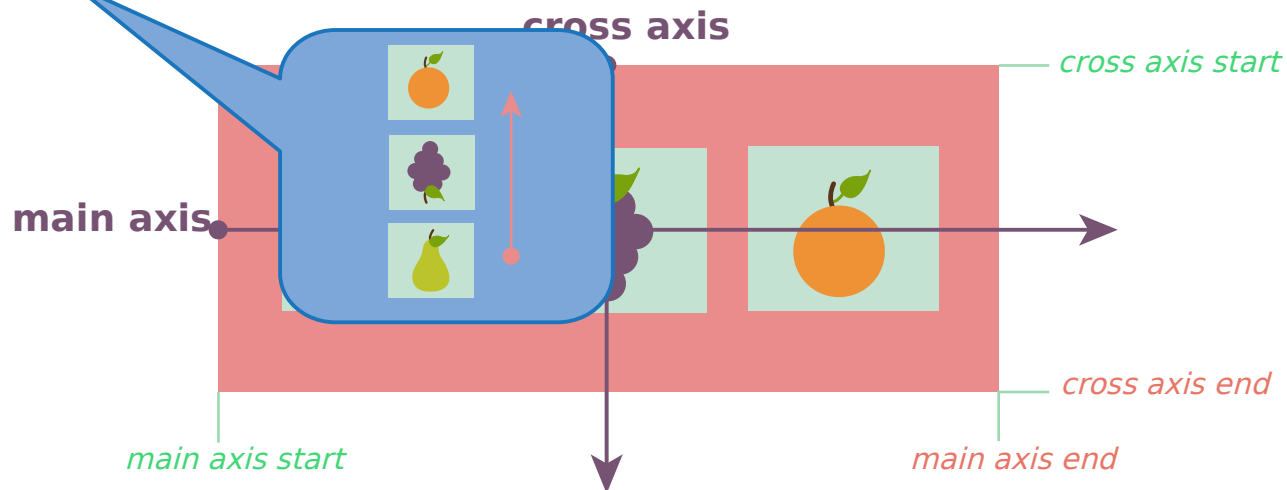
- **display:** define um flex container, habilitando o contexto flexbox e permitindo que os elementos filhos (flex items) sejam distribuídos.

```
.container
```



- **flex-direction:** estabelece o main-axis, definindo a direção da distribuição dos itens flexíveis.

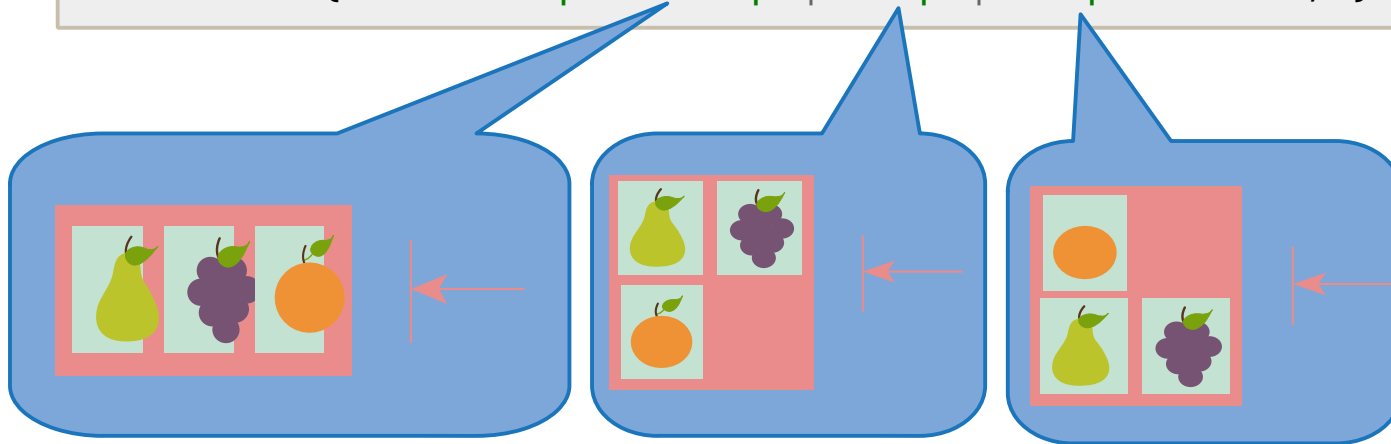
```
.container { flex-direction: row | row-reverse | column | column-reverse; }
```



Propriedades do Flex Container

- **flex-wrap:** define se os itens de um container podem ou não mover para uma nova linha caso não caibam na primeira

```
.container{ flex-wrap: nowrap | wrap | wrap-reverse; }
```



Propriedades do Flex Container

- **justify-content:** define o alinhamento horizontal dos itens quando esses não usam todo o espaço do container

```
.container{ justify-content: flex-start; }
```



```
.container{ justify-content: flex-end; }
```



```
.container{ justify-content: center; }
```



```
.container{ justify-content: space-between; }
```



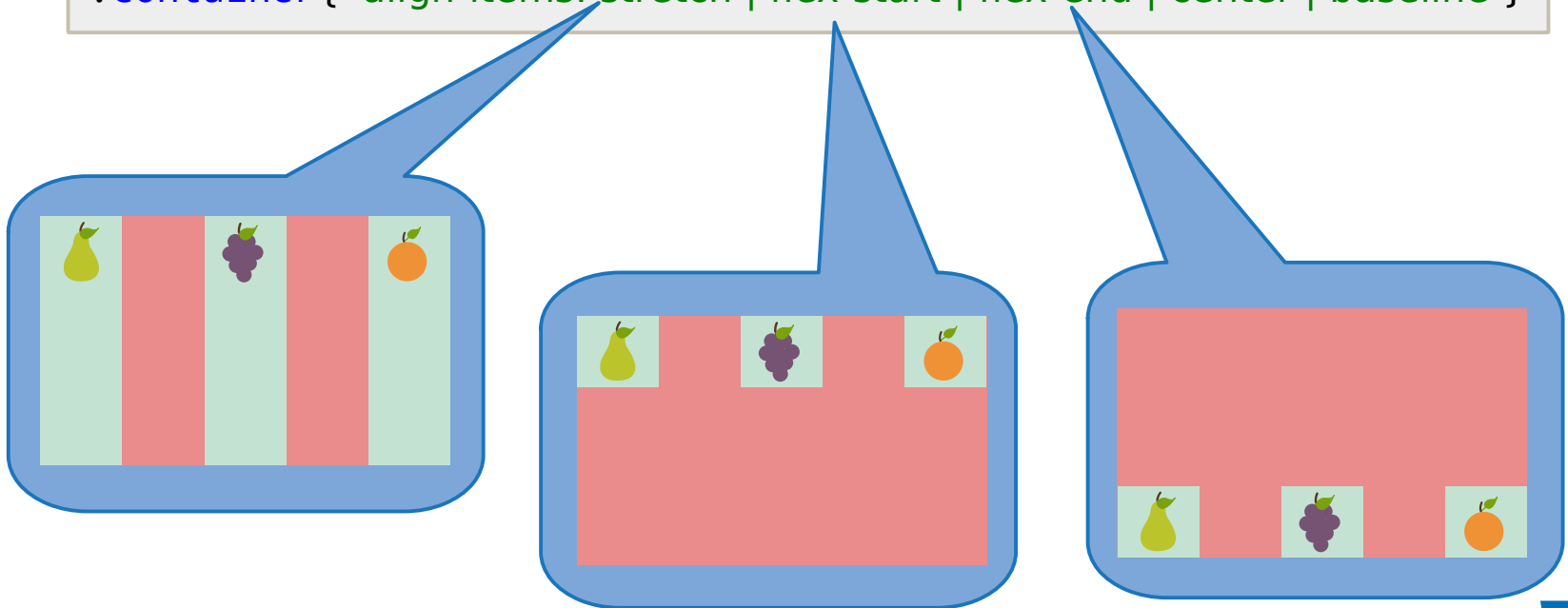
```
.container{ justify-content: space-around; }
```



Propriedades do Flex Container

- **align-items:** Define o alinhamento dos itens no cross-axis. É similar ao justify-content, porém usado no eixo transversal

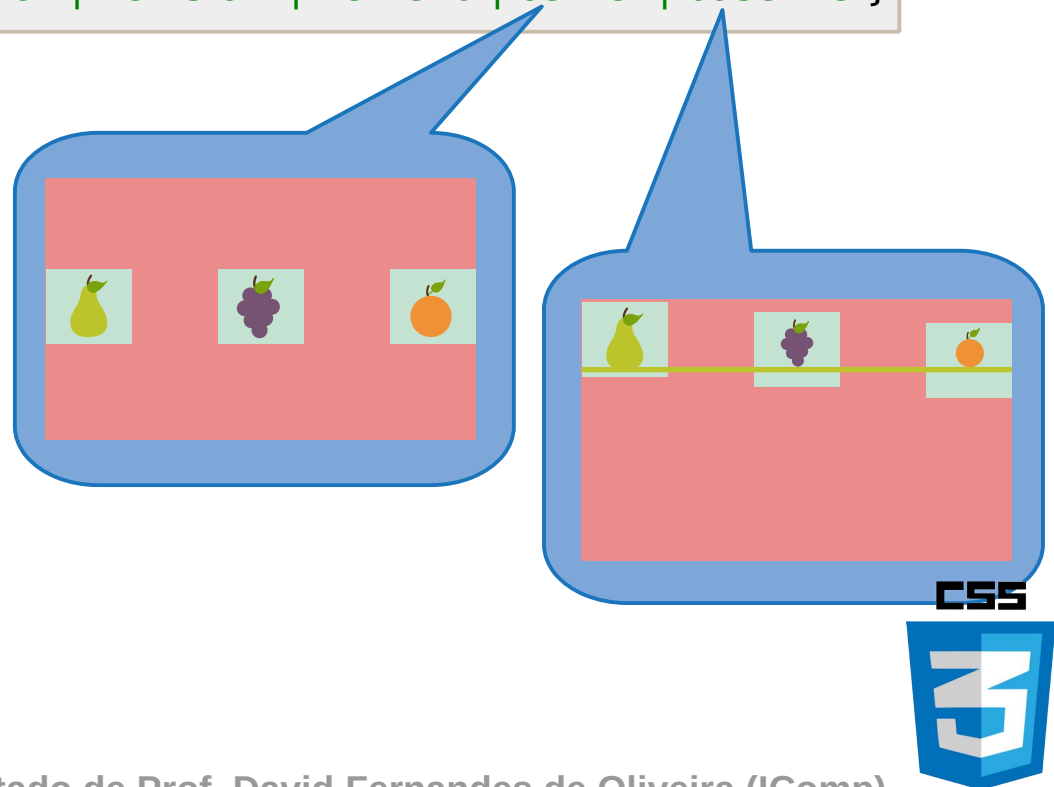
```
.container{ align-items: stretch | flex-start | flex-end | center | baseline }
```



Propriedades do Flex Container

- **align-items:** Define o alinhamento dos itens no cross-axis. É similar ao justify-content, porém usado no eixo transversal

```
.container{ align-items: stretch | flex-start | flex-end | center | baseline }
```



Propriedades do Flex Container

- **align-items**: Define o alinhamento dos itens no cross-axis. É similar ao justify-content, porém usado no eixo transversal

```
.container{ align-items: stretch | flex-start | flex-end | center | baseline }
```

Outras propriedades associadas aos flex containers:

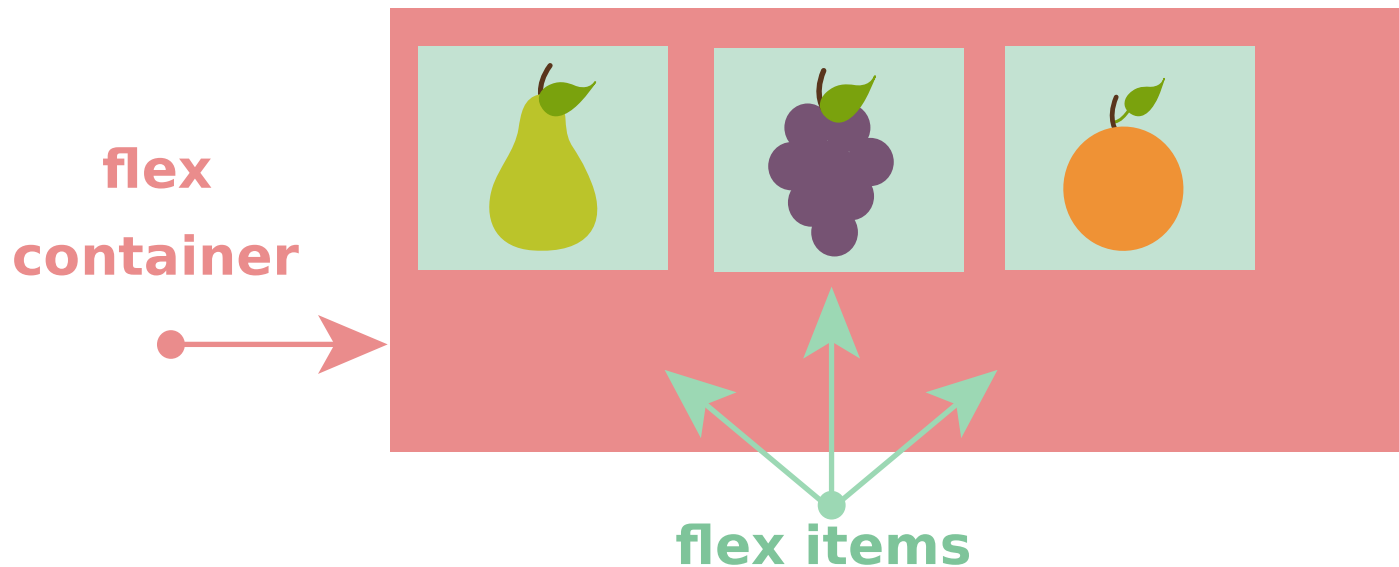
align-content e **flex-flow** (atalho flex-direction e flex-wrap)

CSS



Flex Items

- Os Flex Items são os filhos diretos de um Flex Container
 - Um elemento se torna um flex container com **display: flex**
- Um Flex Item também pode ser um Flex Container, bastando definir **display: flex** nele
 - Assim os filhos desse item também serão flex items.



Propriedades dos Flex Itens

- **flex-grow:** define a proporção com que um item deve crescer, caso seja necessário

```
.item { flex-grow: <número>; /* padrão 0 */ }
```

- **flex-shrink:** define a proporção com que um item deve encolher, caso seja necessário

```
.item { flex-shrink: <número>; /* padrão 1 */ }
```

- **flex-basis:** define o tamanho inicial que um item deve ter antes que o espaço ao seu redor seja distribuído

```
.item { flex-basis: <tamanho> | auto; /* padrão auto */ }
```



Propriedades dos Flex Itens

- **flex:** atalho para as propriedades flex-grow, flex-shrink e flex-basis, nesta ordem

```
.item { flex: <flex-grow> <flex-shrink> <flex-basis> }  
/* valor padrão: 0 1 auto */
```

- **order:** modifica a ordem dos flex itens

```
.item { order: <numero> } /* valor padrão: 0 */
```

- **align-self:** define o alinhamento específico de um único flex item dentro do container

```
.item { order: stretch | flex-start | flex-end | center | baseline }  
/* valor padrão: flex-start */
```



Propriedades dos Flex Itens

```
<style type="text/css">
  html{
    font-family: sans-serif;}

  body{
    width: 70%; max-width: 960px; margin: 20px auto;}

  button{
    font-size: 18px; line-height: 1.5; width: 15%;}

  div {
    height: 100px; border: 1px solid black;}

</style>
<body>
  <div>
    <button> Smile </button>
    <button> Laugh </button>
    <button> Wink </button>
    <button> Shrug </button>
    <button> Blush </button>
  </div>
</body>
```

Propriedades dos Flex Items

```
<style type="text/css">
```

```
html{  
  font-family: sans-serif;
```

```
body{  
  width: 70%;
```

```
button{  
  font-size: 1.2em;
```

```
div {  
  height: 100px;
```

```
</style>
```

```
<body>
```

```
  <div>
```

```
    <button> Smile </button>
```

```
    <button> Laugh </button>
```

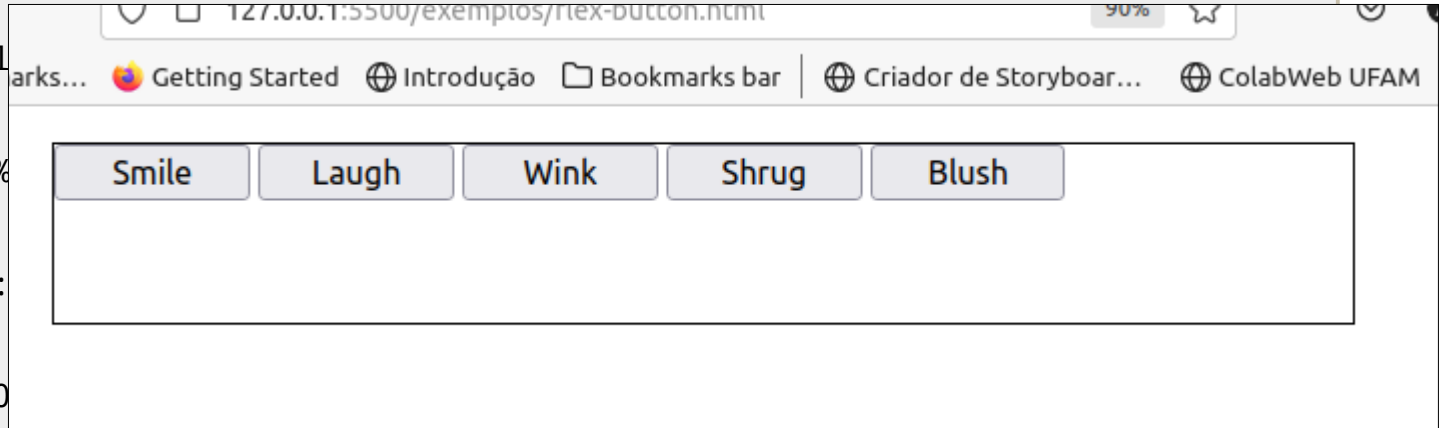
```
    <button> Wink </button>
```

```
    <button> Shrug </button>
```

```
    <button> Blush </button>
```

```
  </div>
```

```
</body>
```



Propriedades dos Flex Itens

```
<style type="text/css">
  html{
    font-family: sans-serif;}

  body{
    width: 70%; max-width: 960px; margin:

  button{
    font-size: 18px; line-height: 1.5; wi

  div {
    height: 100px; border: 1px solid black

</style>
<body>
```

Atividade

- 1) display: flex
- 2) align-items: center
- 3) justify: space-around
- 4) button:first-child self align in flex-end



Exercício

FLEXBOX FROGGY Level 8 of 24

The frogs need to get in the same order as their lilypads using `flex-direction`. This CSS property defines the direction items are placed in the container, and accepts the following values:

- `row`: Items are placed the same as the text direction.
- `row-reverse`: Items are placed opposite to the text direction.
- `column`: Items are placed top to bottom.
- `column-reverse`: Items are placed bottom to top.

```
1 #pond {  
2   display: flex;  
3  
4 }  
5  
6  
7  
8  
9  
10
```

Next

Flexbox Froggy is created by [Codecademy](#) • [GitHub](#) • [Twitter](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).

Acesse o site <https://flexboxfroggy.com/> e ajude os sapos a chegar até as vitórias-régias em um jogo de 24 fases.

CSS 3