



# WACAD002

# Fundamentos de CSS

---

Prof. MSc. Arcanjo Miguel Mota Lopes

[amml@icomp.ufam.edu.br](mailto:amml@icomp.ufam.edu.br)

# 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 (border-\*)

---

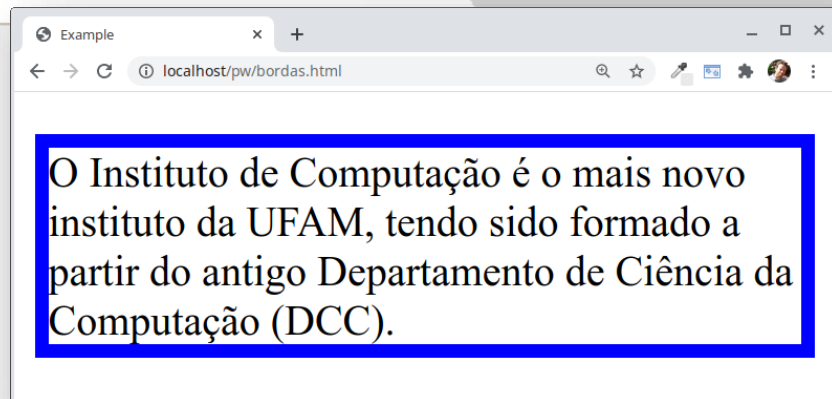


# Bordas (border-\*)

- 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>
```



# border-style

- 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.

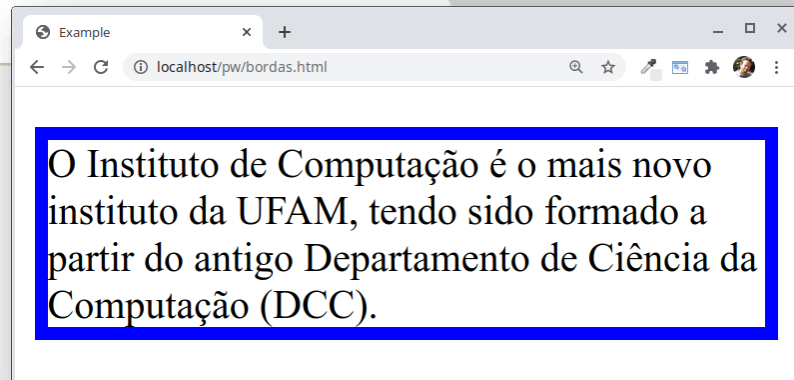
dashed
dotted
double
groove
inset
outset
ridge
solid

# border-style

- 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
  da Computação (DCC).
</p>
```



# border-style

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

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

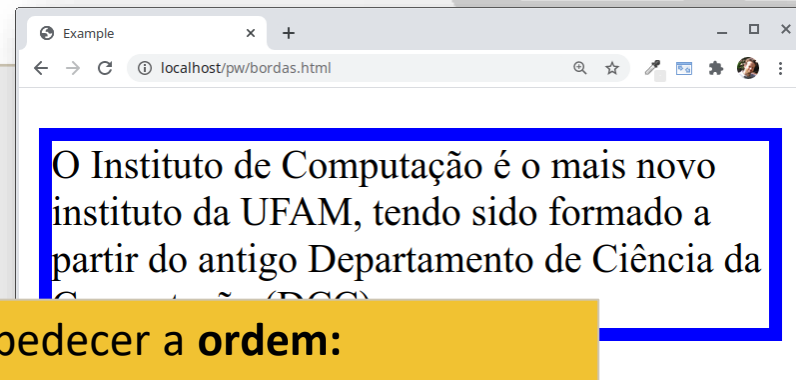
```
<p>
```

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>
```

Para usar o atalho, é preciso obedecer a **ordem**:

**width style color**



# border-style

- 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.



# border-style

```
<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.

# border-radius

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

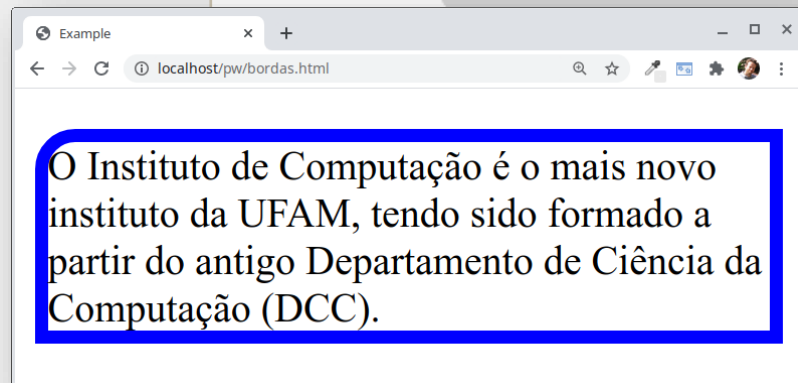
Property	Description	Values
border-top-left-radius border-top-right-radius border-bottom-left-radius border-bottom-right-radius	Sets the radius for a single corner.	A pair of length or percentage values. The percentages relate to the width and height of the border box.
border-radius	This shorthand property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

# border-radius

- 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>
```



# Padding (acolchoamento)

---



# 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>&lt;length&gt;</i> or <i>&lt;%&gt;</i>
padding-right	Sets the padding for the right edge.	<i>&lt;length&gt;</i> or <i>&lt;%&gt;</i>
padding-bottom	Sets the padding for the bottom edge.	<i>&lt;length&gt;</i> or <i>&lt;%&gt;</i>
padding-left	Sets the padding for the left edge.	<i>&lt;length&gt;</i> or <i>&lt;%&gt;</i>
padding	This shorthand property sets the padding for all edges in a single declaration.	1–4 <i>&lt;length&gt;</i> or <i>&lt;%&gt;</i> values

# Padding

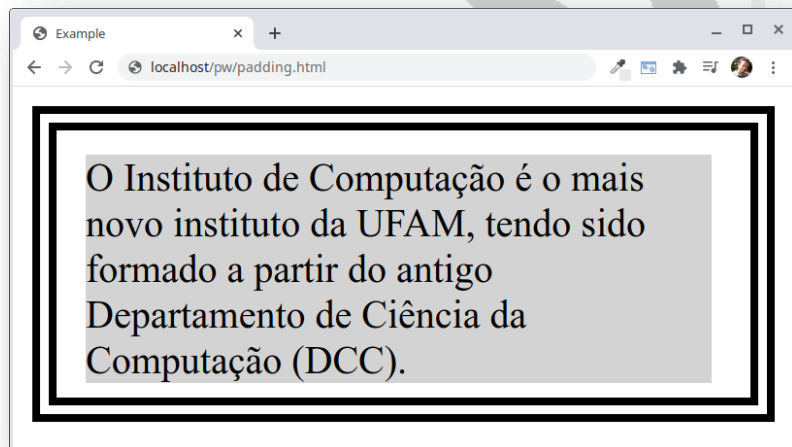
- É possível atribuir um valor de **padding** para cada lado de um elemento caixa, ou atribuir um único valor de **padding** para o elemento inteiro

Property	Default	Values
padding-top	Quando informamos o <b>padding</b> usando porcentagens, a porcentagem é sempre relativa ao comprimento do elemento pai. A altura não é levada em conta.	<length> or <%>
padding-right		<length> or <%>
padding-bottom		<length> or <%>
padding-left		<length> or <%>
padding	This shorthand property sets the padding for all edges in a single declaration.	1–4 <length> or <%> values

# 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>
```





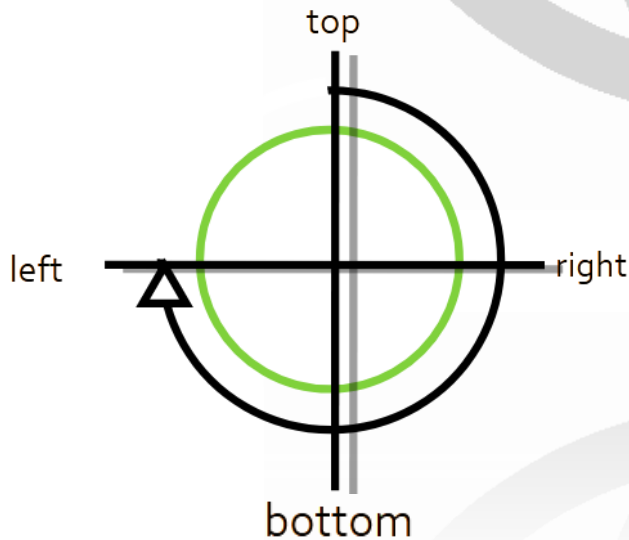
# Padding

- 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;
```

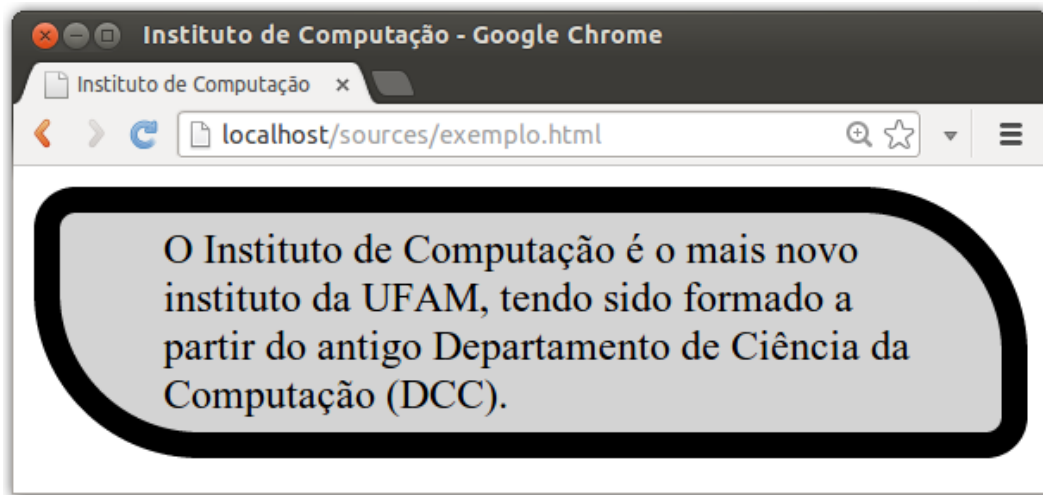
```
padding: 10px 16px 6px 12px;
```

Os estilos acima  
produzem o  
mesmo efeito



# Atividade 01

- Codifique um documento **html/css** cujo resultado seja similar ao da figura abaixo



# Margem (margin-\*)

---



# Margem (margin-\*)

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



# Margem (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

# Margem (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	<p>Quando informamos a <b>margem</b> usando Porcentagens (%), a porcentagem é sempre relativa ao comprimento do elemento pai. A altura não é levada em conta.</p>	<length> or <%>
margin-right		<length> or <%>
margin-bottom		<length> or <%>
margin-left		<length> or <%>
margin	This shorthand property sets the margin for all edges in a single declaration.	1–4 <length> or <%> values

# Margem (margin-\*)

- `margin: 5%; /* todos os lados*/`
- `margin: 10px; /* todos os lados*/`
- `margin: 1.6em 20px; /* topo e inferior | esquerda e direita */`
- `margin: 10px 3% 1em; /* topo | esquerda e direita | inferior */`
- `margin: 10px 3px 30px 5px; /* topo | direita | inferior | esquerda */`

# Margem (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>
```





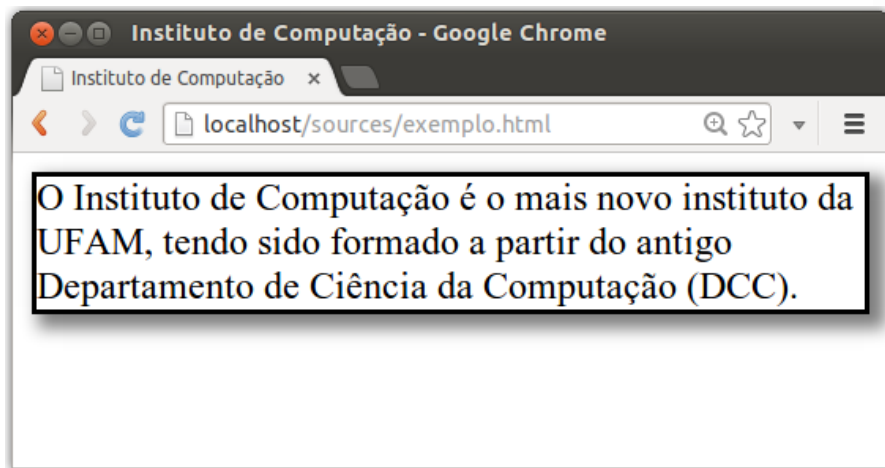
# Sombreamento

---



# Box-shadow

- 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



# Box-shadow

- 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

# Box-shadow

- 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.

# Box-shadow

- 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**

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

sombreamento

# Box-shadow

- Exemplo de sombra externa 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:  
**offset 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).

Sombreamento  
para dentro (inset)

# Box-shadow

- Exemplo de sombra externa 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:  
**offset voffset blur  
spread color inset**

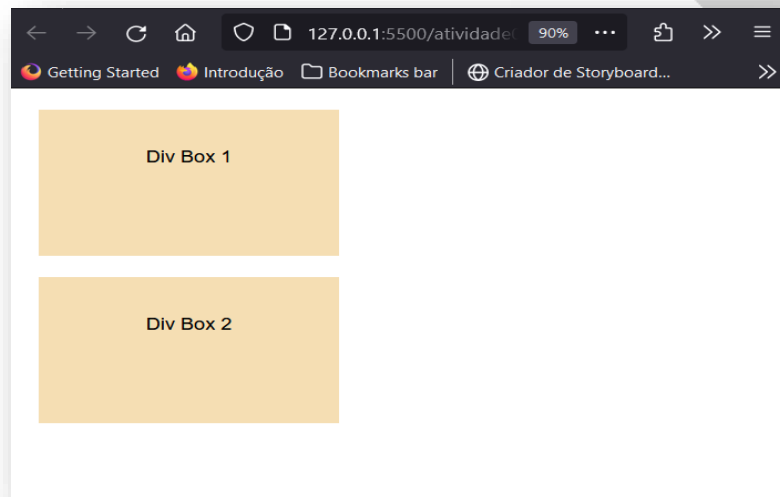
O Instituto de Computação é o mais novo instituto da UFAM, tendo sido formado a partir do antigo Departamento de Ciência da Computação (DCC).

Sombreamento  
para dentro (inset)

sombreamento

# Atividade 02

- Crie um documento HTML chamado **box-shadow.html** e um arquivo CSS chamado **styles.css**.
- No arquivo HTML, crie 2 (dois) elementos HTML `<div>` com um `<p>` cada, com um texto.
- No arquivo CSS (`styles.css`), aplique o seguinte estilos para os elementos criado. **DICA: use class**
  - Largura de 200px;
  - Altura de 100px
  - Padding: 20px
  - Margem de 20px
  - Text-align: center
  - Background na cor wheat

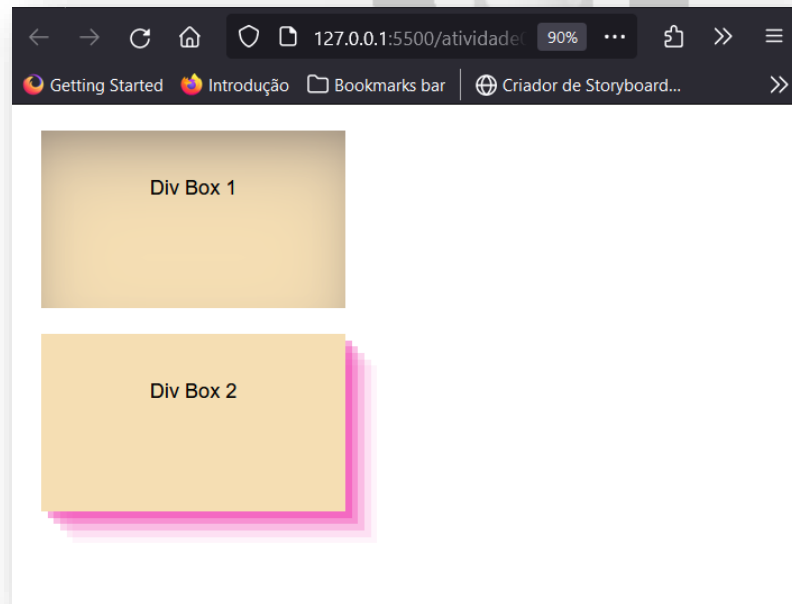


**Resultado esperado**



## Atividade 02 (parte 2)

- Acesse o link:  
<https://getcssscan.com/css-box-shadow-examples>
- No primeiro elemento <div>, aplique o box-shadow valor #25
- No segundo elemento <div>, aplique o box-shadow valor #48



**Resultado esperado**

# Transbordamento (overflow)

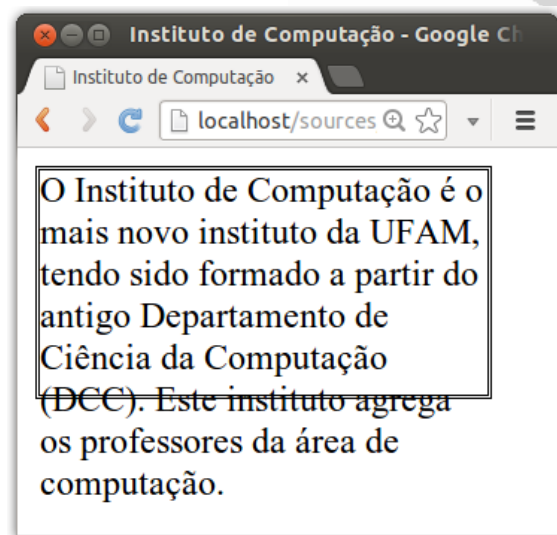
---



# Transbordamento (overflow)

- Quando diminuimos 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>
```

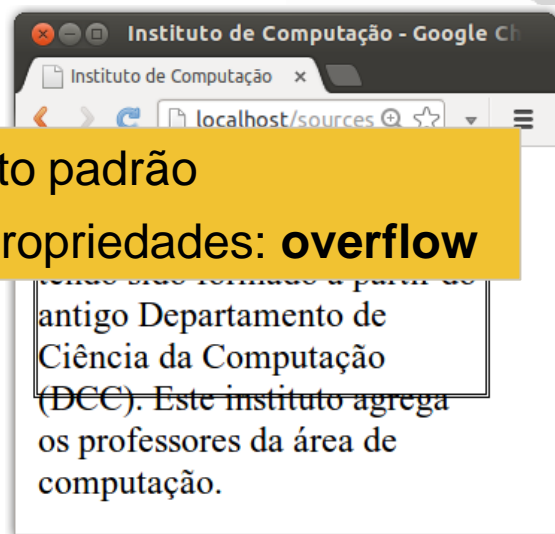


# Transbordamento (overflow)

- Quando diminuimos 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:
  height:
  border: medium double;
}
</style>
```

No entanto, esse comportamento padrão  
pode ser mudado através das propriedades: **overflow**



# 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

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

# Overflow

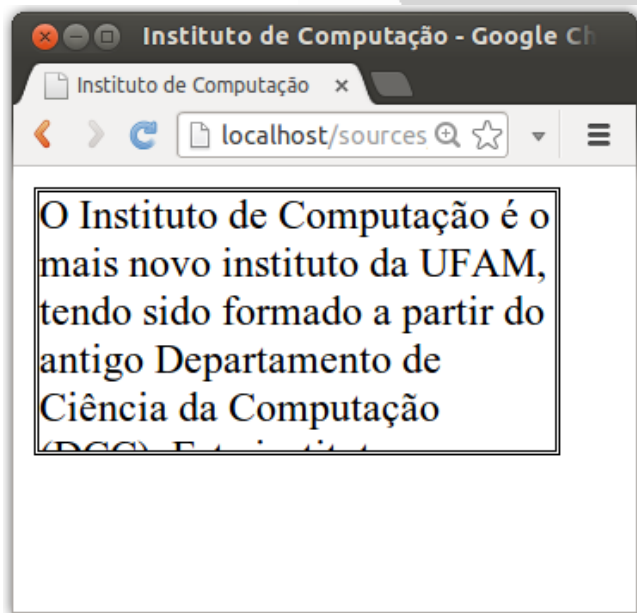
- Valores para a **propriedade overflow**

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

# Overflow: hidden

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

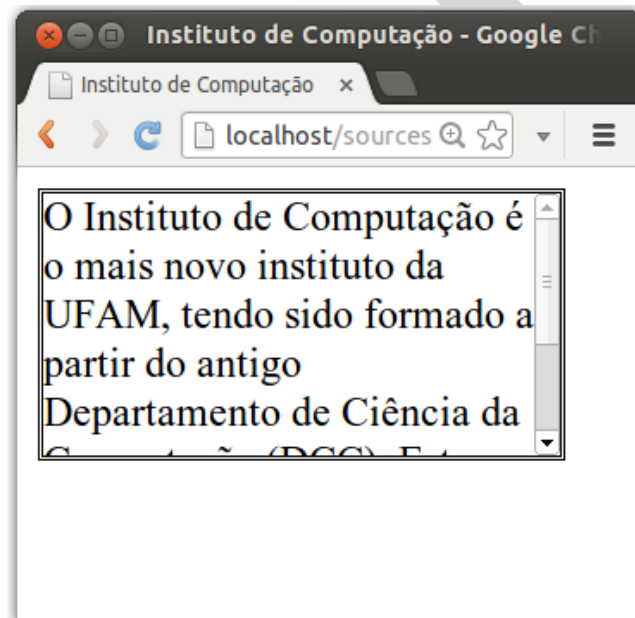
```
<style type="text/css">
div {
  width: 200px;
  height: 100px;
  overflow: hidden;
  border: medium double;
}
</style>
```



# Overflow: auto

- **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>
```

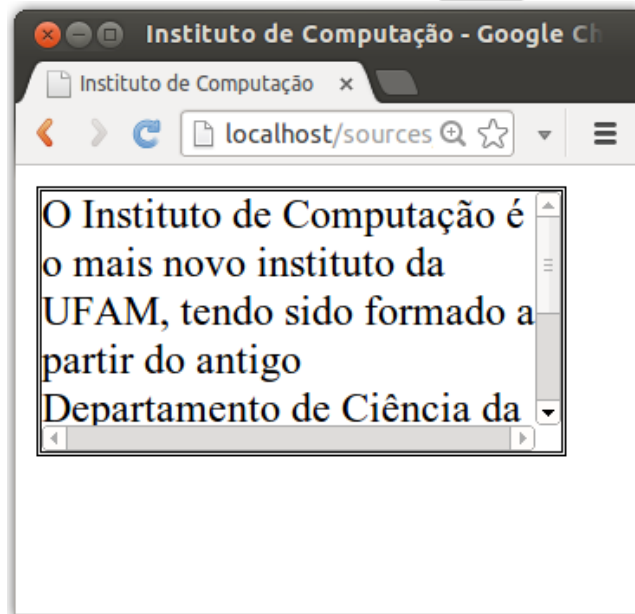




# Overflow: scroll

- **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.

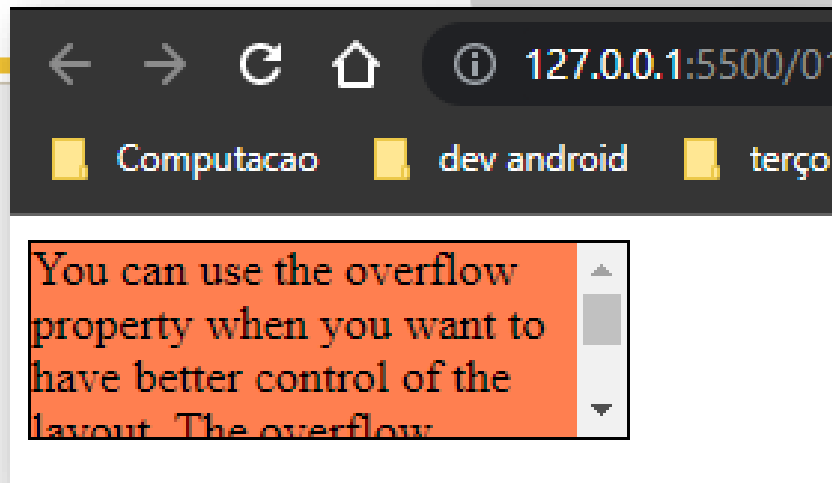
# Overflow-x & Overflow-y

```
<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>
```



# Display

---

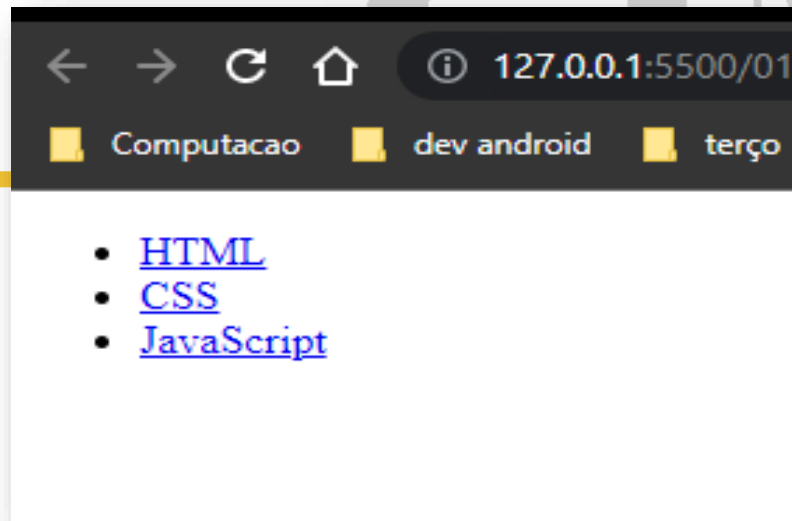


# Display

---

- 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.

# Display



```
<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>
```

# Display

```
<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>
```

← → ↻ 🏠 ⓘ 127.0.0.1:5500/01\_

Computacao dev android terça

[HTML](#) [CSS](#) [JavaScript](#)

# Display

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

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

[HTML](#) [CSS](#) [JavaScript](#)

**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.**



# display: block

---

- 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

# display: block

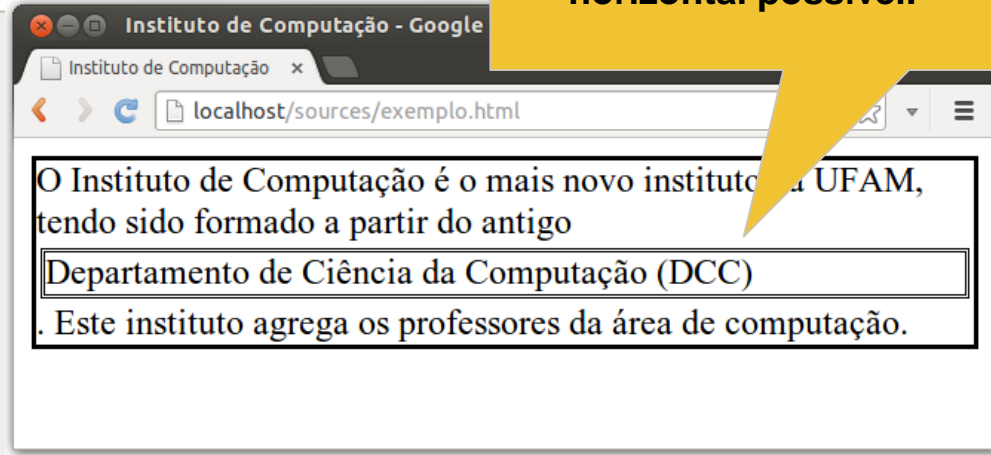
Por padrão, ele ocupam o máximo de espaço horizontal possível.

```
<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.

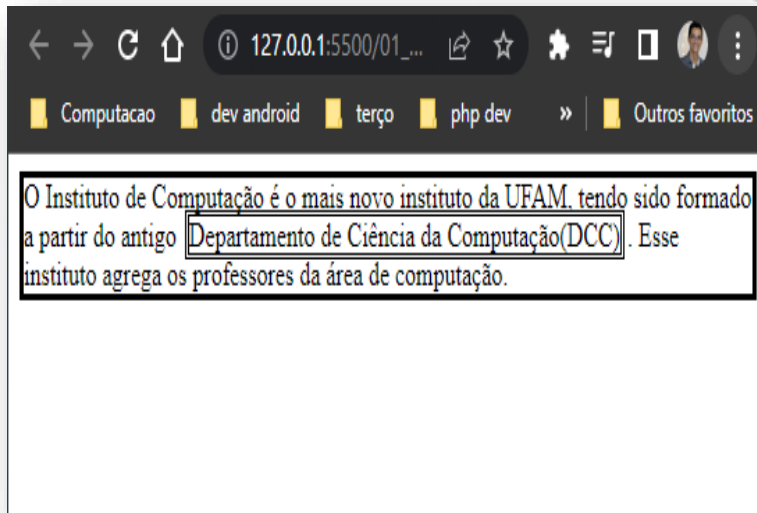
```
</div>
```



# display: inline

- 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

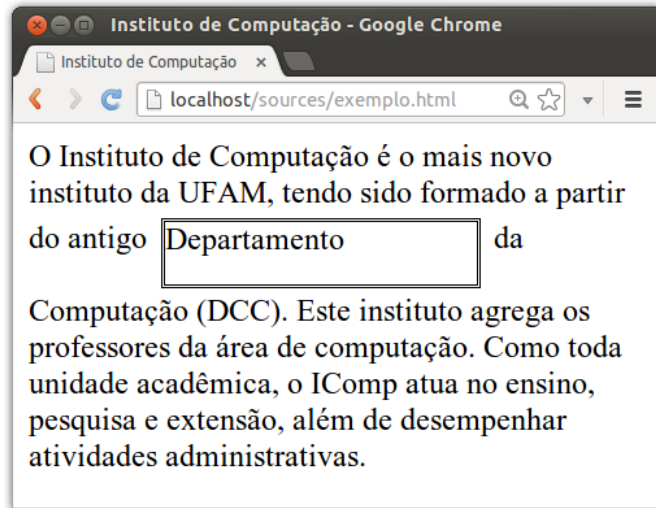
```
<style type="text/css">
div {
  display: inline;
}
span {
  display: inline;
  border: medium double black;
  margin: 20em;
  width: 100em;
  height: 200em;
}
</style>
```



# display: inline-block

- 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>
```



# display: none

- Usando **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>
```

# display:

- Usando

```
<p id="togo"
tendo sido
(DCC).</p>
<p>Este in
Como toda
e extensão
<button>Blo
<script>
var butto
for (var
button:
docu
e.t:
};
}
</script>
```

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. Como toda unidade acadêmica, o IComp atua no ensino, pesquisa e extensão, além de desempenhar atividades administrativas.

Block

None

Example - Google Chrome

Example

localhost/sources/exempl

Este instituto agrega os professores da área de computação. Como toda unidade acadêmica, o IComp atua no ensino, pesquisa e extensão, além de desempenhar atividades administrativas.

Block

None

## Atividade 03

- Crie os seguintes blocos. Cada bloco é uma `<div>`



- **#DICA:** utilize as técnicas de display para encontrar o formato correto.
- **Cores utilizadas:**
  - Quadro1 `<div> rgb(239, 116, 111);`
  - Quadro2 `<div> #ECF081;`
  - Quadro3 `<div> MediumSeaGreen;`
  - Quadro4 `<div> rgb(171, 62, 91);`
- **Medidas:** largura 200px; Altura 100px;

# Posicionamento de Conteúdo

---





# Propriedade **position**

- 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.

# Propriedade **position**

- 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;  
  };  
}
```

# 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

```
g" alt="css"/>
```

```
/TagName("BUTTON");  
Id("css-logo");  
++) {  
    .innerHTML;
```

# Instituto de C Instituto de Computação

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

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



No ensino de graduação, o D cursos de Ciência da Comput Engenharia de Software e En Também oferece disciplinas b cursos de graduação da UFAM

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

Static

Relative

Absolute

Fixed

# Instituto de C

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

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



No ensino de graduação, o D  
cursos de Ciência da Comput  
Engenharia de Software e En  
Também oferece disciplinas b  
cursos de graduação da UFAM

**Static** Relative Absolute

# Instituto de

O Instituto de Computaç  
de Ciência da Computaç  
que agrega os professore

Como todo instituto acad  
pesquisa e extensão, alér  
administrativas.

No ensino de graduação, o  
cursos de Ciência da Co  
Engenharia de Software  
Também oferece discipli  
cursos de graduação da U

Static **Relative** Absolute Fixed

# Instituto de Computação



O Instituto de Computação (ICon  
de Ciência da Computação (DCC),  
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

# Instituto de C

O Instituto de Computação (ICom) é um instituto acadêmico que agrega os professores da

Como todo instituto acadêmico, o ICom 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, Engenharia de Software e Engenharia de Sistemas. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

**Static** Relative Absolute

# Instituto d

O Instituto de Computação (ICom) é um instituto acadêmico que agrega os professores da

Como todo instituto acadêmico, o ICom 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, Engenharia de Software e Engenharia de Sistemas. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

Static **Relative** Absolute Fixed

# Instituto de Com

O Instituto de Computação (ICom) é um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico, o ICom 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, Engenharia de Software e Engenharia de Sistemas. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

Static Relative **Absolute** Fixed

O Instituto de Computação (ICom) é um instituto acadêmico que agrega os professores da área de computação.

Como todo instituto acadêmico, o ICom 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, Engenharia de Software e Engenharia de Sistemas. Também oferece disciplinas básicas de computação para 30 cursos de graduação da UFAM em Manaus.

Static Relative Absolute **Fixed**



# position:absolute

---

- 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

# position:absolute

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



bg.jpg



css.png



# position:absolute

- 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>
```



# position:absolute (z-index)

- 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>
```

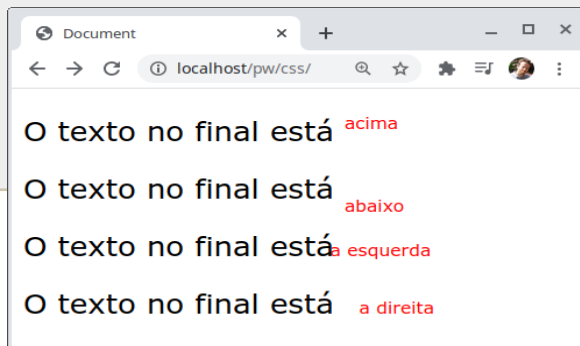


# position:relative

- 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>
```

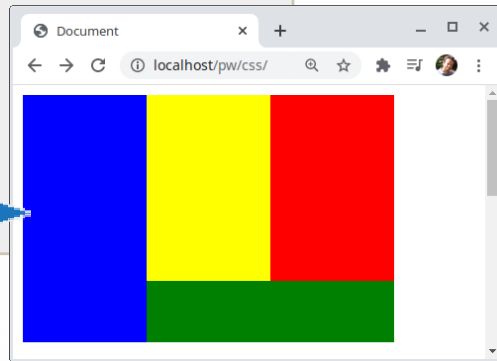
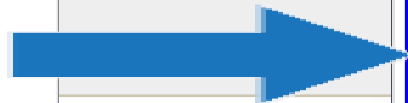
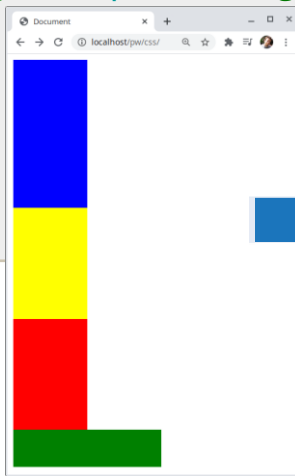


# Atividade 04

- O código abaixo gera a página da esquerda. Use **posicionamento relativo** para transformá-la na página da direita.

```
<style>
  div.blue { width:100px; height:200px; background-color:blue; }
  div.yellow { width:100px; height:150px; background-color:yellow; }
  div.red { width:100px; height:150px; background-color:red; }
  div.green { width:200px; height:50px; background-color:green; }
</style>
```

```
<div class="blue"></div>
<div class="yellow"></div>
<div class="red"></div>
<div class="green"></div>
```



# Elementos Flutuantes

---



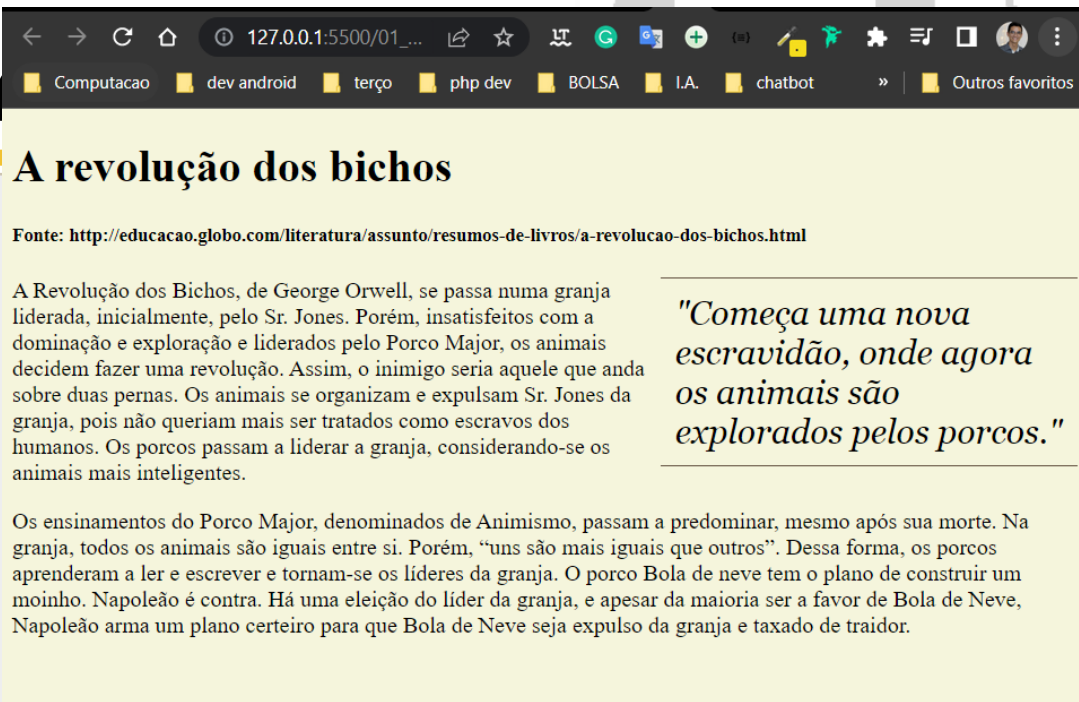
# Elementos flutuantes (float)

---

- 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 (float)

```
<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 (float)

```
<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
  </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.

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 porquinho. 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.

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

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



# Elementos flutuantes (float)

---

- A **propriedade float** pode ter um dos seguintes valores:
  - **left**: O elemento flutua à esquerda de seu contêiner (elemento pai)
  - **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

# Contêineres Flutuantes

---



# Contêineres flutuantes (float)

- 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

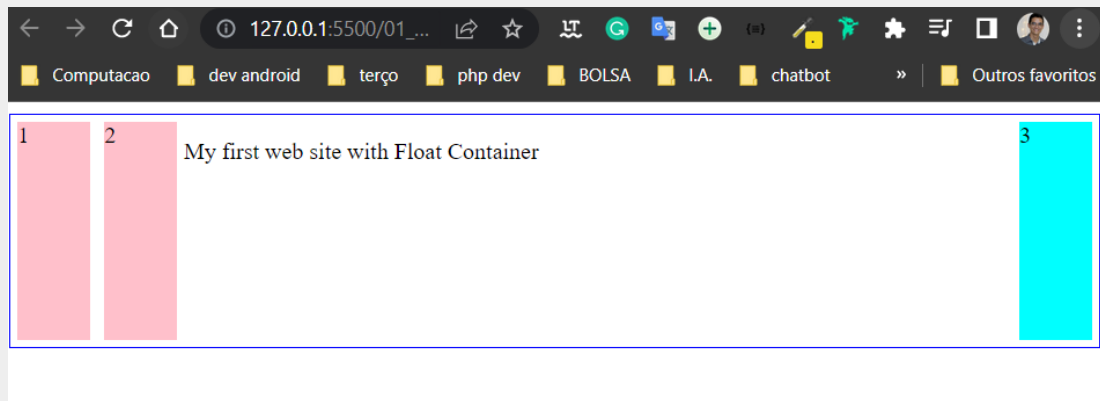
- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycle
- 1876: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

```

<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>

```



# Position ou float?

---

- 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.

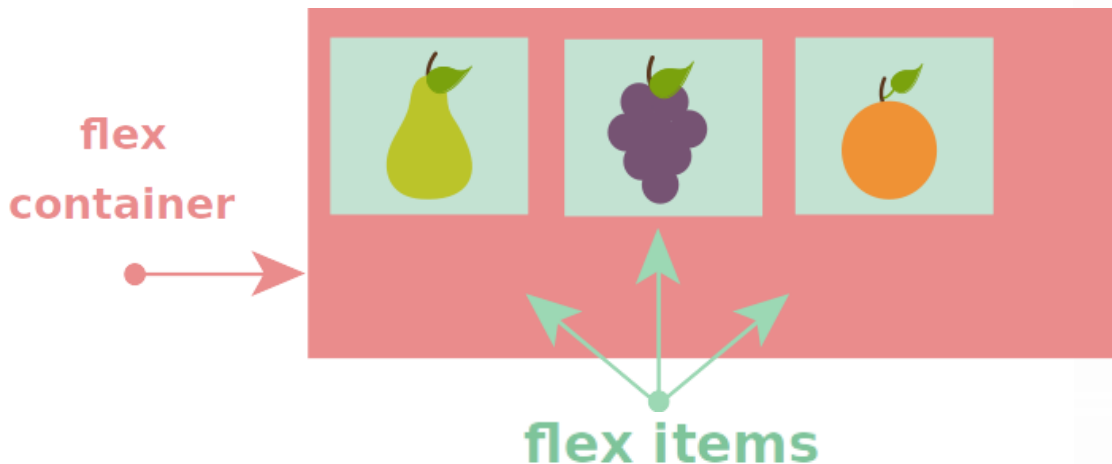
# Flexbox Layout

---



# Flexbox Layout

- Os métodos de display vistos até então, **block**, **inline** e **inline-block**, 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



# Flexbox Layout

- Os métodos de display vistos até então, **block**, **inline** e **inline-block**, 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 contêineres

Algumas propriedades CSS do flexbox estão relacionadas com o **contêiner**, enquanto outras com os **itens**





# Flexbox Layout

- 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

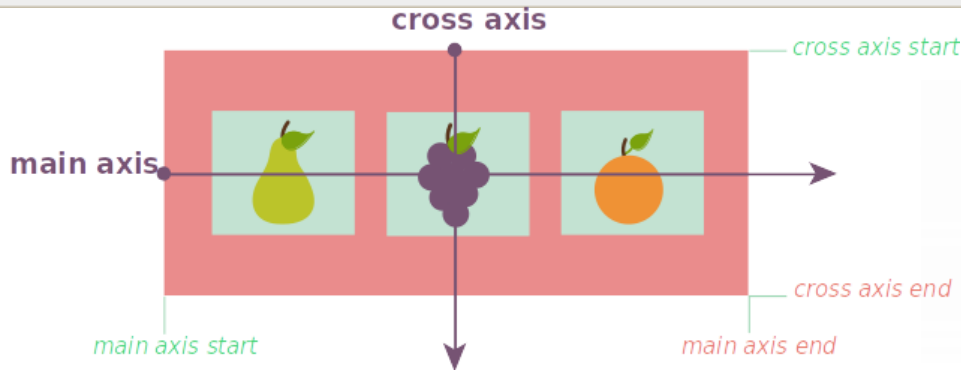
# Propriedades do Flexbox Layout

- **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

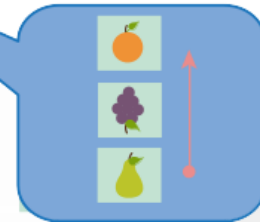
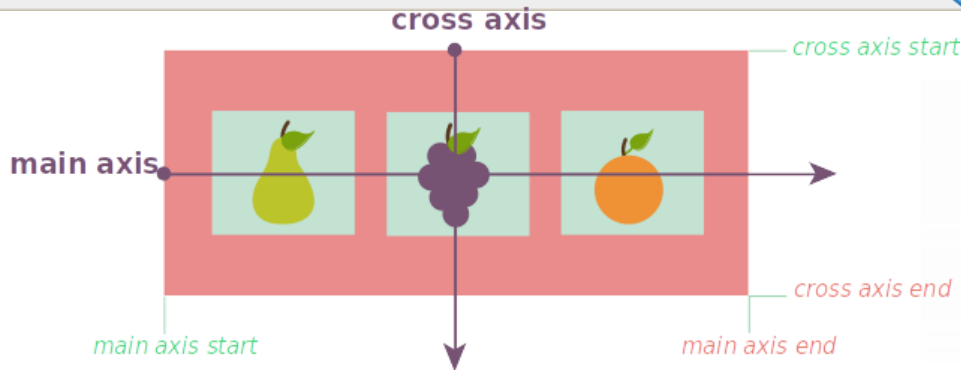
```
.container { flex-direction: row | row-reverse | column | column-reverse; }
```



# Propriedades do Flexbox Layout

- **display** (flex item)   contexto flex p... s filhos
- **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 Flexbox Layout

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

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



# Propriedades do Flexbox Layout

- **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 Flexbox Layout

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

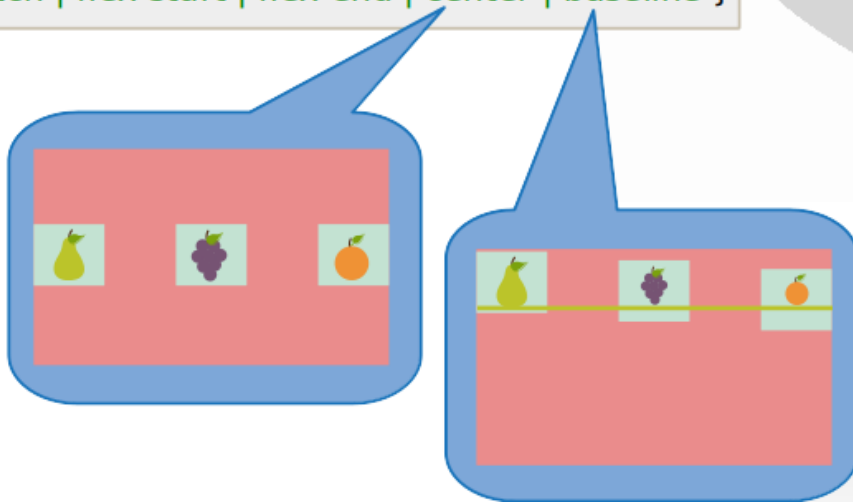
```
.container{ align-items: stretch | flex-start | flex-end | center | baseline }
```



# Propriedades do Flexbox Layout

- **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 }
```



# Flex itens

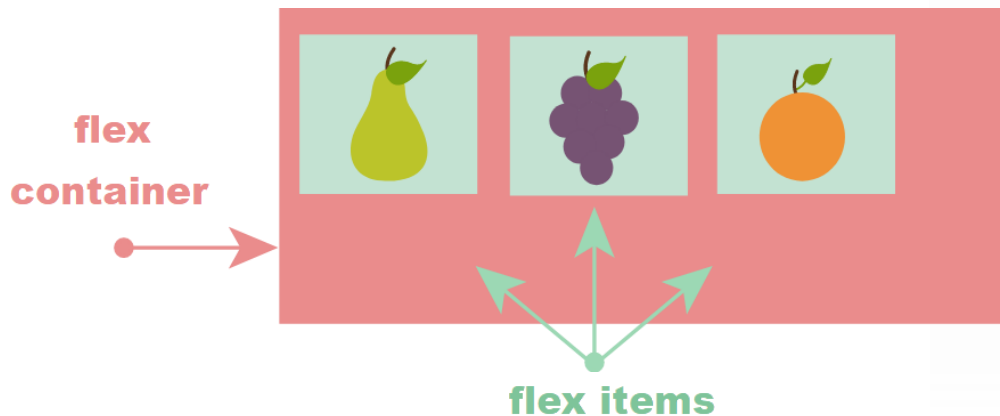
---





# Flex itens

- Os Flex Itens 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**
  - Assim os filhos desse item também serão **flex itens**.



# Propriedades do 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 do 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

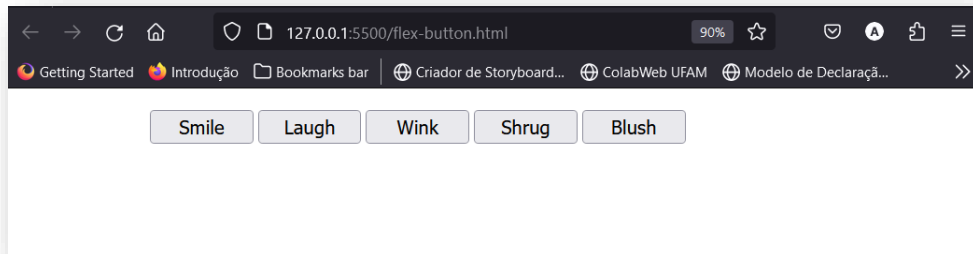
```
.item { order: <numero> } /* valor padrão: 0 */
```

- **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 */
```

# Atividade 05

- Dado o seguinte código, faça o que se pede no próximo slide:



Resultado do código

```
<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;
  }
</style>

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

## Atividade 05 (parte 2)

1. Insira uma borda no container de 1px solid black
2. O container possui layout display flex
3. Os itens precisam estar alinhados no 'center'
4. Os itens do container precisam estar organizados com 'space-around' entre eles
5. E o primeiro <button> 'first-child' precisa estar alinhado na parte inferior do container 'flex-end'

**Resultado esperado!!!**



# Homework

