



# Programação em Microinformática

- Div
  - Float
  - Display
  - Overflow
-

# HTML <DIV>



A <div>tag define uma divisão ou seção em um documento HTML.



A <div>tag é usada como um contêiner para elementos HTML - que é estilizado



com CSS ou manipulado com JavaScript.



A <div>tag é facilmente estilizada usando o atributo class ou id.



Qualquer tipo de conteúdo pode ser colocado dentro da <div> tag!



**Nota:** Por padrão, os navegadores sempre colocam uma quebra de linha antes e depois do <div>elemento.



Tag <DIV>

**<div>**

Todo, qualquer e quantos  
elementos do HTML você queira  
colocar aqui.

**</div>**

- O nome **div** vem de divisão, e esta tag vai lhe dar o simples poder de dividir qualquer trecho de seu código.
- a tag <div> é um container para armazenar diversos elementos
- Sintaxe da tag <div>

## <div> e classes (Exemplo 1)

```
<!DOCTYPE HTML>
<html>
<head>
<style>

.nomeDaClasse {
  background: gray;
  margin: 10px;
  font-size: 150%;
}

.nomeDaSegundaClasse {
  background: green;
  margin: 10px;
}

</style>

<head>

<body>

<div class="nomeDaClasse"> Olá mundo </div>

<div class="nomeDaSegundaClasse"> Bom fim de Semana!!! </div>

</body>
</html>
```

Testar!!!

## Exemplo 2

<div>

```
<!DOCTYPE html>
<html>

<head>

<style>
div.cities {
    background-color:black;
    color:white;
    margin:20px;
    padding:20px;
}
</style>

</head>

<body>

<div class="cities">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e do Reino Unido.
<p> O centro de Londres, a antiga City of London, també

</div>

<div class="cities">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e do Reino Unido.
<p> O centro de Londres, a antiga City of London, també
</div>

<div class="cities">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e do Reino Unido.
<p> O centro de Londres, a antiga City of London, també
</div>
```

Testar!!!

```
<html>

<head>

<style>

div.cities {
  background-color:black;
  color:white;
  margin:20px;
  padding:20px;
}

div.cities2 {
  background-color:red;
  color:white;
  margin:20px;
  padding:20px;
}

</style>

</head>

<body>

<div class="cities">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e ...
<p> O centro de Londres, a antiga City ...
</div>

<div class="cities2">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e ...
<p> O centro de Londres, a antiga City ...
</div>

<div class="cities">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra e ...
<p> O centro de Londres, a antiga City ...
</div>

</body>
</html>
```

Exemplo 3  
<div>

# Controle do fluxo do documento no HTML

## Parte I

---



# CSS: Propriedade float

- *Float* é uma propriedade CSS que usamos para posicionar algo na tela, seja um elemento em bloco ou em linha.
  - Com a propriedade *float* definida, um elemento pode ser empurrado para esquerda ou para a direita, permitindo que os outros elementos possam organizar em torno dele.
-



# CSS Float

A **float** propriedade CSS especifica como um elemento deve flutuar.

```
<div>
  <figure>
    
    <figcaption>Legenda da imagem</figcaption>
  </figure>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing
</div>
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est. Lorem ipsum dolor sit amet, consectetur

Testar!!!

# CSS Float

```
figure {  
  float: left;  
  margin: 0 10px 10px 0;  
}
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est. Lorem

ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.



# CSS Float

- Utilizado para posicionar elementos na esquerda ou direita
    - **left** - O elemento flutua à esquerda do contêiner
    - **right**- O elemento flutua à direita do seu contêiner
    - **none** - O elemento não flutua (será exibido exatamente onde ocorre no texto). Este é o padrão
-

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    float: right;
    margin: 0 0 10px 10px;
    width:170px;
    height:170px;
}

p {
text-align:justify;
}

</style>
</head>
```

Testar!!!

```
<body>

<p>a imagem flutuará para a direita no parágrafo e o texto no parágrafo irá
envolver a imagem.</p>



<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
imperdiet,
nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim
ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor
vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula,
facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus
interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed
ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed
ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer
fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta.
Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```

# Praticando....

## Inserindo Imagens lado a lado



O Parque Nacional da Serra da Canastra é um dos mais importantes parques nacionais brasileiros, criado em 1972 através do decreto 70.355 de 1972. Dentro do parque está localizada a nascente histórica do rio São Francisco, no município de São Roque de Minas.

# Praticando ...

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.img-container {
  float: left;
  width: 33.33%;
  padding: 5px;
}

</style>
</head>
```

Testar!!!

```
<body>

<h2>Inserindo Imagens lado a lado</h2>

<div class="clearfix">
  <div class="img-container">
    
  </div>
  <div class="img-container">
    
  </div>
  <div class="img-container">
    
  </div>
</div>
<p> O Parque Nacional da Serra da Canastra é
</body>
</html>
```

# CSS Clear

- A propriedade `clear` é usada para controlar o comportamento de elementos flutuantes.
- Os elementos após um elemento flutuante fluirão ao redor. Para evitar isso, use a propriedade `clear`.
- **left:** Elemento é empurrado para baixo de elementos com *float left*;
- **right:** Elemento é empurrado para baixo de elementos com *float right*;
- **both:** Elemento é empurrado para baixo de elementos com *float left ou right*;
- **none:** Elemento não é empurrado para baixo de elementos com *float*.



### Sem clear

Texto 1 - Aprendendo a utilizar o clear.

Texto 1

### Com clear

Texto 2

Texto 2- Aprendendo a utilizar o clear.



```
<!DOCTYPE html>
<html>
<head>
<style>
.div1 {
    float: right;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}

.div2 {
    border: 1px solid red;
}

.div3 {
    float: right;
    width: 100px;
    height: 50px;
    margin: 10px;
    border: 3px solid #73AD21;
}

.div4 {
    border: 1px solid red;
    clear: both;
}
</style>
</head>
```

```
<body>

<h2>Sem clear</h2>
<div class="div1">Texto 1</div>
<div class="div2">Texto 1 - Aprendendo a utilizar o clear.</div>

<br>
<br>
<br>
<br>

<h2>Com clear</h2>
<div class="div3">Texto 2</div>
<div class="div4">Texto 2- Aprendendo a utilizar o clear.</div>

</body>
</html>
```

Testar!!!

# CSS Overflow

- A propriedade overflow especifica quando o conteúdo de um elemento de nível de bloco deve ser cortado, exibido barra de rolagem ou se transborda do elemento
- Utiliza-se a propriedade overflow quando um elemento for maior que o elemento que o contém

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



# CSS Overflow

- **Visible:** É o valor padrão e define que o conteúdo não será cortado se exceder o tamanho do elemento.

```
1 | p {  
2 |     width: 12em;  
3 |     border: dotted;  
4 |     overflow: visible;  
5 | }
```

`visible` (default)

Sed ut perspiciatis unde  
omnis iste natus error sit  
voluptatem accusantium...  
doloremque laudantium.

**Hidden:** O conteúdo excedido não ficará visível

```
1 | p { overflow: hidden;
```

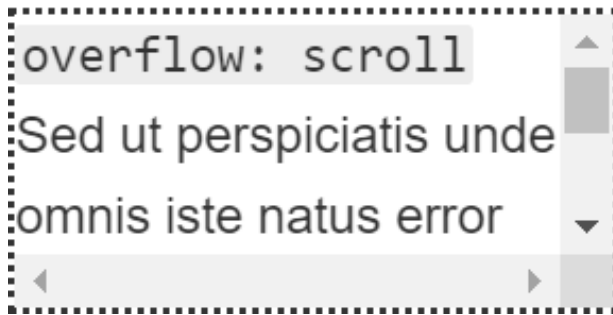
`overflow: hidden`

Sed ut perspiciatis unde  
omnis iste natus error sit  
voluptatem accusantium...

# CSS Overflow

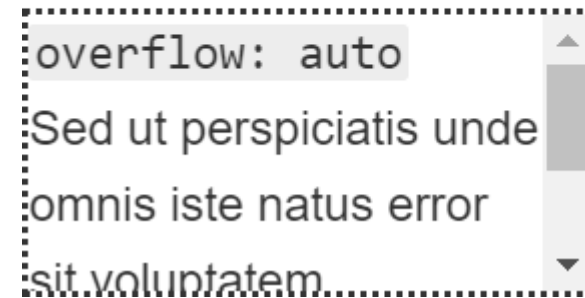
- **Auto:** A barra de rolagem será adicionada automaticamente se o conteúdo exceder o tamanho do elemento.

```
1 | p { overflow: scroll;
```



- **Scroll:** O conteúdo excedido será cortado mas haverá uma barra de rolagem para permitir a visualização completa do mesmo.

```
1 | p { overflow: auto;
```



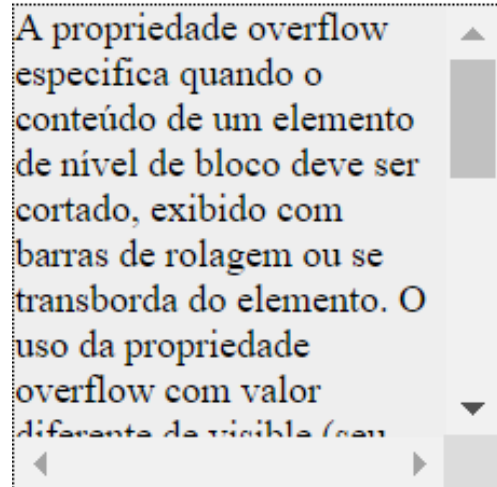
# CSS Overflow-x e Overflow-y

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  background-color: #eee;
  width: 200px;
  height: 200px;
  border: 1px dotted black;
  overflow-x: scroll;
  overflow-y: scroll;
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>A propriedade overflow especifica quando o conteúdo de um elemento de nível de bloco deve ser cortado, exibido com barras de rolagem ou se transborda do elemento. O uso da propriedade overflow com valor diferente de visible (com de rolagem lenta.</p>

</body>
</html>
```

## CSS Overflow



A propriedade overflow especifica quando o conteúdo de um elemento de nível de bloco deve ser cortado, exibido com barras de rolagem ou se transborda do elemento. O uso da propriedade overflow com valor diferente de visible (com

# CSS Overflow

Testar!!!

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 3px solid #4CAF50;
  padding: 5px;
}
```

```
.img1 {
  float: right;
}
```

```
.clearfix {
  overflow: auto;
}
```

```
p{
  clear: both;
}
```

```
</style>
</head>
<body>
```

```
<h1>uso do overflow</h1>
```

```
<div>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum
  interdum...
</div>
```

```
<p> Adicionando a classe clearfix com overflow:auto</p>
```

```
<div class="clearfix">
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum
  interdum...
</div>
```

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

## uso do overflow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



Adicionando a classe clearfix com overflow:auto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



# CSS - Display

---

CSS Layout – Propriedade Display

# Display



A propriedade display é a propriedade CSS mais importante para controlar o layout.



A propriedade display especifica como um elemento é exibido.



Cada elemento HTML tem um valor de exibição padrão, dependendo do tipo de elemento que é.



O valor de exibição padrão da maioria dos elementos é *block* ou *inline*.



# Elemento *block-level*

- 
- Um elemento de nível de bloco sempre inicia em uma nova linha e ocupa toda a largura disponível (estende-se para a esquerda e para a direita o máximo possível)

## Alguns elementos block-level

<div>

<h1> - <h6>

<p>

<form>

<header>

<footer>

<section>

# Elemento In-line

Um elemento “em linha” não inicia em uma nova linha e ocupa apenas a largura necessária.

Exemplos de elementos in-line:

`<span>`

`<a>`

`<img>`

# Substituir o valor de exibição padrão

- Como mencionado, cada elemento possui um valor de exibição padrão. No entanto, você pode substituir isso.
- Alterar um elemento “em linha” para um “elemento de bloco”, ou vice-versa, pode ser útil para fazer com que a página tenha uma aparência específica e ainda siga os padrões da web.
- Um exemplo comum é criar elementos `<li>` embutidos para menus horizontais:

# Display:inline

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
    display: inline;
}
</style>
</head>
<body>
```

```
<p>Mostra os links na posição horizontal</p>
```

```
<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

Mostra os links na posição horizontal

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

Testar!!!

# Display:block

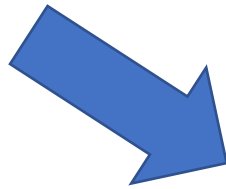
- Uma propriedade de exibição com um valor de "block" resulta em uma quebra de linha entre os dois elementos.

```
<!DOCTYPE html>
<html>
<head>
<style>
span {
    display: block;
}
</style>
</head>
<body>
```

```
<span> Uma propriedade de exibição com um valor de "block" resulta em </ span>
<span> uma quebra de linha entre os dois elementos. </ span>

</body>
</html>
```

Testar!!!



Uma propriedade de exibição com um valor de "block" resulta em uma quebra de linha entre os dois elementos.

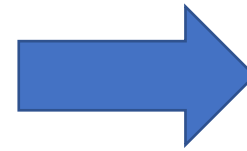
# Display:block

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
    display: block;
}
</style>
</head>
<body>
```

```
<p>Display block</p>
```

```
<a href="/html/default.asp" target="_blank">HTML</a>
<a href="/css/default.asp" target="_blank">CSS</a>
<a href="/js/default.asp" target="_blank">JavaScript</a>
```

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



Display block

[HTML](#)

[CSS](#)

[JavaScript](#)

# Display:inline-block

- Comparado com *display:inline*, a principal diferença é que *display:inline-block* permite definir uma largura e altura no elemento
- Além disso, com *display: inline-block*, as margens superiores e inferiores são respeitadas, mas com *display: inline* elas não são.
- Comparado a *display: block*, a principal diferença é que *display: inline-block* não adiciona uma quebra de linha após o elemento, para que o elemento possa se sentar ao lado de outros elementos.

*inline-block é a junção dos comportamentos inline (ex.: ocupar apenas o espaço do conteúdo, não quebrar linha) e block (ex.: dimensões configuráveis) em um único elemento HTML*

```

<!DOCTYPE html>
<html>
<head>
<style>

.box {
  width: 200px;
  height: 100px;
  margin: 1.5%;
  border: 1px solid green;
}

p{
  text-align: justify;
  margin: 1.5%;
}

p.formata{
  text-align:center;
  line-height:60px;
}

</style>

<body>
<div class="box">
  <p class="formata">Realidade Virtual.</p>
</div>

<div class="box">
  <p class="formata">Realidade Virtual.</p>
</div>

```

Sem uso do float e  
sem uso de display



Testar!!!

Realidade virtual é uma tecnologia de int  
objetivo é criar a sensação de presença en  
técnicas e de equipamentos computaciona  
presença é usualmente referida como imei



```
<!DOCTYPE html>
<html>
<head>
<style>
```

Com uso do float

```
.box {
  float: left;
  width: 200px;
  height: 100px;
  margin: 1.5%;
  border: 1px solid green;
}
```

```
.ajuste{
  clear: both;
}
```

```
p{
  text-align: justify;
  margin: 1.5%;
}
```

```
p.formata{
  text-align:center;
  line-height:60px;
}
```

```
</style>
```

```
<body>
<div class="box">
  <p class="formata">Realidade Virtual.</p>
</div>
```

```
<div class="box">
  <p class="formata">Realidade Virtual.</p>
</div>
```



Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade virtual é uma tecnologia de interface entre um usuário e um sistema operacional através de recursos gráficos 3D ou imagens 360º. O objetivo é criar a sensação de presença em um ambiente virtual diferente do real. Para isso, essa interação é realizada em tempo real, com o uso de técnicas e de equipamentos computacionais que ajudem na ampliação do sentimento de presença do usuário no ambiente virtual. Esta sensação de presença é usualmente referida como imersão.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.box {  
  display: inline-block;  
  width: 200px;  
  height: 100px;  
  margin: 1.5%;  
  border: 1px solid green;  
}
```

```
}
```

```
p{  
  text-align: justify;  
  margin: 1.5%;  
}
```

```
p.formata{  
  text-align: center;  
  line-height: 60px;  
}
```

```
</style>
```

```
<body>
```

```
<div class="box">
```

```
  <p class="formata">Realidade Virtual.</p>
```

```
</div>
```

```
<div class="box">
```

```
  <p class="formata">Realidade Virtual.</p>
```

```
</div>
```

```
<div class="box">
```

```
  <p class="formata">Realidade Virtual.</p>
```

```
</div>
```

```
<div class="box">
```

```
  <p class="formata">Realidade Virtual.</p>
```

```
</div>
```

Com uso do display  
inline-block



Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

Realidade Virtual.

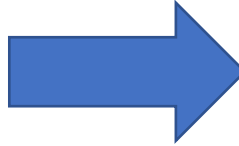
Realidade virtual é uma tecnologia de interface entre um usuário e um sistema operacional através de recursos gráficos 3D ou imagens 360° cujo objetivo é criar a sensação de presença em um ambiente virtual diferente do real. Para isso, essa interação é realizada em tempo real, com o uso de técnicas e de equipamentos computacionais que ajudem na ampliação do sentimento de presença do usuário no ambiente virtual. Esta sensação de presença é usualmente referida como imersão.

# Display: none

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  display: none;
}
</style>
</head>
<body>

<h1>Texto visível </h1>
<h1 class="hidden">Texto oculto</h1>
<p>Observe que o elemento h1 com display: none; não ocupa espaço</p>

</body>
</html>
|
```



## Texto visível

Observe que o elemento h1 com display: none; não ocupa espaço

Vamos testar???





# Referências

- <https://pt-br.learnlayout.com/inline-block.html>
  - <https://www.w3schools.com/css/default.asp>
-