



# Desenvolvimento Web

**Prof. M. Marcelo Petri**

# Olá! Sejam bem vindos



DISCIPLINA/UNIDADE CURRICULAR	Desenvolvimento Web
TEMA DA AULA	“CSS”
INDICADOR	2. Codifica aplicações utilizando linguagens para front-end
CRONOGRAMA	2.1 Folhas de estilos (CSS)
ATIVIDADE	Atividades de desenvolvimento
HABILIDADES	Codificar páginas utilizando as ferramentas de programação web.
CRITÉRIOS	Codificação

# Estilos

- inline

```
<p style="background-color: yellow; color: seagreen; ">Olá mundo!</p>
```

- Interno

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

- Externo:

```
<link href="css/estilo.css" rel="stylesheet">
```

# Seletores

- Elemento (<tag>)
- Id (id)
- Classe (class)
- Universal (\*)

## Especificidade de Seletores

- Se houver duas ou mais regras CSS que apontam para o mesmo elemento, o seletor com o maior valor de especificidade "vencerá" e sua declaração de estilo será aplicada a esse elemento HTML.

# Especificidade de Seletores

- Exemplo

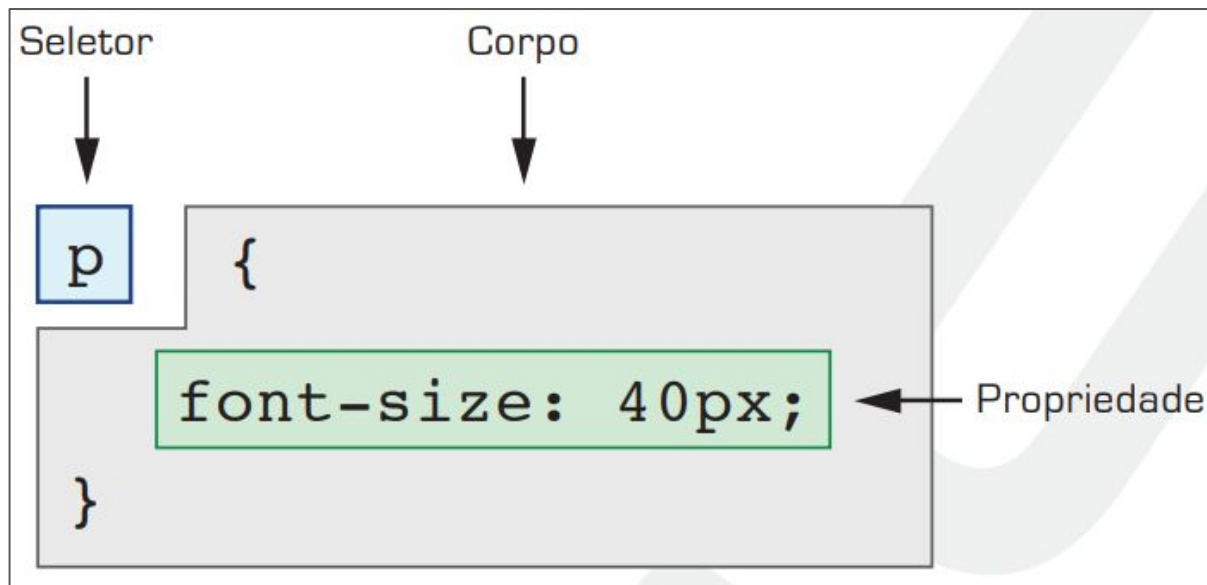
```
<html>
<head>
  <style>
    #demo {color: blue;}
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p id="demo" class="test" style="color: pink;">Olá Mundo!</p>

</body>
</html>
```

# Regras

- Podemos aplicar regras de formatação para cada elemento HTML



# CSS - AulaWeb

- **Comentários**


- Os comentários são usados para explicar o código e podem ajudar quando você editar o código-fonte posteriormente.
- 
- Os comentários são ignorados pelos navegadores.
- 
- Um comentário CSS é colocado dentro do elemento `<style>` e começa `/*` e termina com `*/`



# CSS - AulaWeb

- Cores

código hexadecimal	nome da cor	cor
#00FFFF	aqua	
#000000	black	
#0000FF	blue	
#FF00FF	fuchsia	
#808080	gray	
#008000	green	
#00FF00	lime	
#800000	marron	

código hexadecimal	nome da cor	cor
#000080	navy	
#808000	olive	
#800080	purple	
#FF0000	red	
#C0C0C0	silver	
#008080	teal	
#FFFFFF	white	
#FFFF00	yellow	

# CSS - AulaWeb

- **Cores**

- <https://materializecss.com/>
- <https://m3.material.io/>
- <https://color.adobe.com/pt/create/color-wheel>
- [https://developer.mozilla.org/pt-BR/docs/Web/CSS/color\\_value](https://developer.mozilla.org/pt-BR/docs/Web/CSS/color_value)
- [https://www.w3schools.com/css/css3\\_colors.asp](https://www.w3schools.com/css/css3_colors.asp)

# CSS - AulaWeb

- Cores e Backgrounds

```
p {  
  color: red; /*rgb(255,0,0) #fff*/  
  background-color: rgba(100,100,100,0.6); /*opacidade*/  
}
```

# Propriedades CSS

## ● Background

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background

```
body {  
  background-color: lightblue;  
}
```

```
body {  
  background-image: url("paper.gif");  
}
```

```
body {  
  background-image: url("gradient bg.png");  
  background-repeat: repeat-y; | repeat-x; | no-repeat;  
}
```

```
body {  
  background-image: url("img tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
}
```

```
body {  
  background-image: url("img tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: fixed;  
}
```

```
body {  
  background: #ffffff url("img_tree.png") no-repeat  
  right top;  
}
```

# CSS - AulaWeb

- **Tipografia**

```
font-family: 'Courier New';  
font-size: 50px;  
font-weight: lighter;  
font-style: italic;
```

**Fontes Google:** <https://fonts.google.com/>  
**CSS:** <https://uiverse.io>

# Propriedades CSS

- Formatação de texto

## FORMATAÇÃO DE TEXTO

Este texto é estilizado com algumas das propriedades de formatação de texto. O título usa as propriedades `text-align`, `text-transform` e `color`. O parágrafo é recuado, alinhado e o espaço entre os caracteres é especificado. O sublinhado é removido deste link colorido "Experimente você mesmo" .

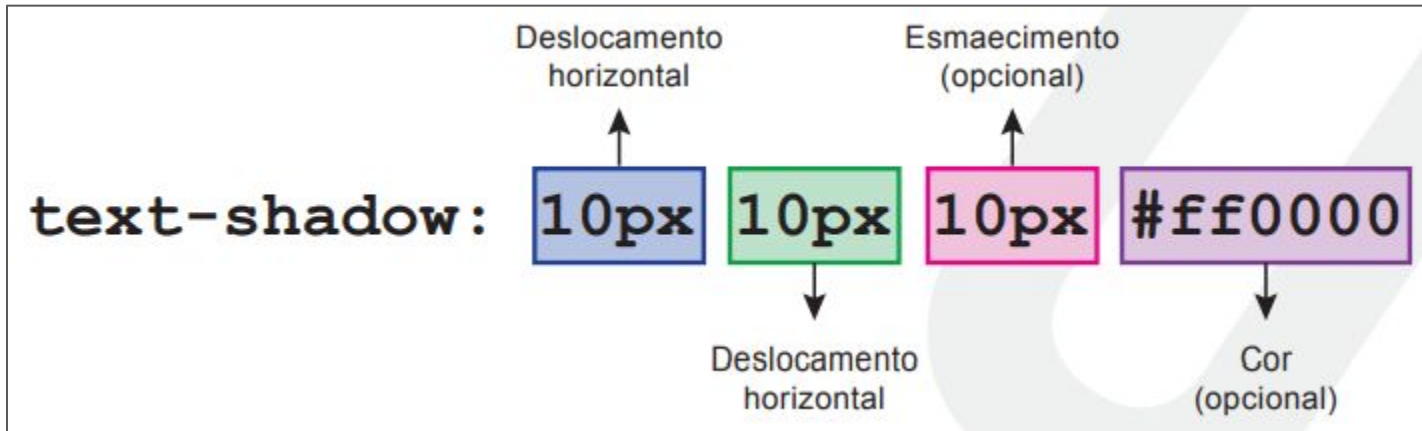
# Propriedades CSS

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <style>
5     * {
6       font-family: "Segoe UI",Arial,sans-serif;
7       font-weight: 400;
8     }
9     div {
10      border: 1px solid gray;
11      padding: 8px;
12    }
13    h1 {
14      text-align: center;
15      text-transform: uppercase;
16      color: #4CAF50;
17    }
18    p {
19      text-indent: 50px;
20      text-align: justify;
21      letter-spacing: 3px;
22    }
23    a {
24      text-decoration: none;
25      color: #008CBA;
26    }
27  </style>
28 </head>
```

```
29 <body>
30   <div>
31     <h1>Formatação de Texto</h1>
32     <p>Este texto é estilizado com algumas das
33       propriedades de formatação de texto.
34       O título usa as propriedades text-align,
35       text-transform e color. O parágrafo é recuado,
36       alinhado e o espaço entre os caracteres é especificado.
37       O sublinhado foi removido deste link colorido
38       <a target="_blank" href="">"Experimente você mesmo"</a> link.
39     </p>
40   </div>
41 </body>
42 </html>
```

# Propriedades CSS

- Text Shadow
  - Para adicionar uma sombra no texto de um elemento HTML





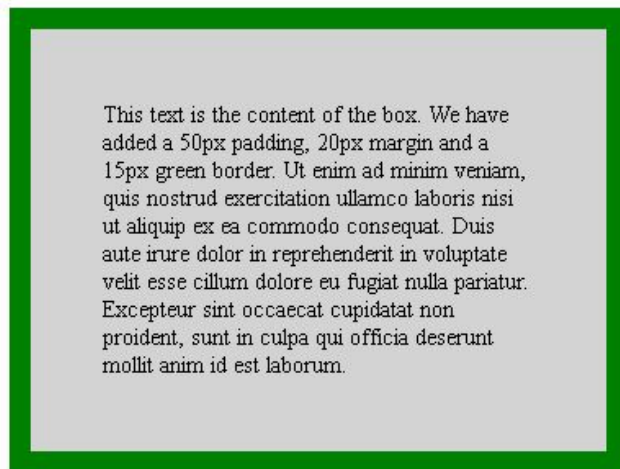
# Propriedades CSS

- Text Shadow

<p>Texto</p> <pre>text-shadow: 10px 10px #ff0000</pre>	<p>Texto</p> <pre>text-shadow: 10px 10px 10px #ff0000</pre>
<p>Texto</p> <pre>text-shadow: -10px -10px #ff0000</pre>	<p>Texto</p> <pre>text-shadow: -10px -10px 10px #ff0000</pre>
<p>Texto</p> <pre>text-shadow: 0 0 10px #ff0000</pre>	<p>Texto</p> <pre>text-shadow: 10px 10px 10px #ff0000, -10px -10px 10px #0000ff</pre>

## Box Model

- O modelo de caixa CSS é essencialmente uma caixa que envolve cada elemento HTML. Ele consiste em: margens, bordas, preenchimento e o conteúdo real.



# Box Model

Todo elemento HTML, no corpo de uma página web, está contido em um box

## Desenvolvimento Web

### HTML

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum enim est, ultrices ac vehicula vitae, pharetra id mauris. Quisque sit amet nisl mollis sem fermentum **accumsan**.

[Mais sobre](#)

### CSS

Lorem ipsum dolor sit amet, **consectetur** adipiscing elit. Vestibulum enim est, ultrices ac vehicula vitae, pharetra id mauris. Quisque sit amet nisl mollis sem fermentum accumsan.

[Mais sobre](#)

# Propriedades CSS

## ● Border

A propriedade `border-style` especifica que tipo de borda exibir.

Ex: `border: 1px solid green` => cria uma borda de 1 pixel, estilo sólida e cor green

Os seguintes valores são permitidos:

- `dotted`- Define uma borda pontilhada
- `dashed`- Define uma borda tracejada
- `solid`- Define uma borda sólida
- `double`- Define uma borda dupla
- `groove`- Define uma borda ranhurada 3D. O efeito depende do valor da cor da borda
- `ridge`- Define uma borda 3D estriada. O efeito depende do valor da cor da borda
- `inset`- Define uma borda de inserção 3D. O efeito depende do valor da cor da borda
- `outset`- Define uma borda inicial 3D. O efeito depende do valor da cor da borda
- `none`- Não define borda
- `hidden`- Define uma borda oculta

[https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)

# Propriedades CSS

- Border

```
p {  
  border: 5px solid red;  
}
```

```
p {  
  border-left: 6px solid red;  
}
```

```
p {  
  border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}
```

```
p {  
  border-style: solid;  
  border-color: red;  
  border-width: 5px;  
}
```

```
p {  
  border: 2px solid red;  
  border-radius: 5px;  
}
```

# Outros estilos

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

# Border-radius

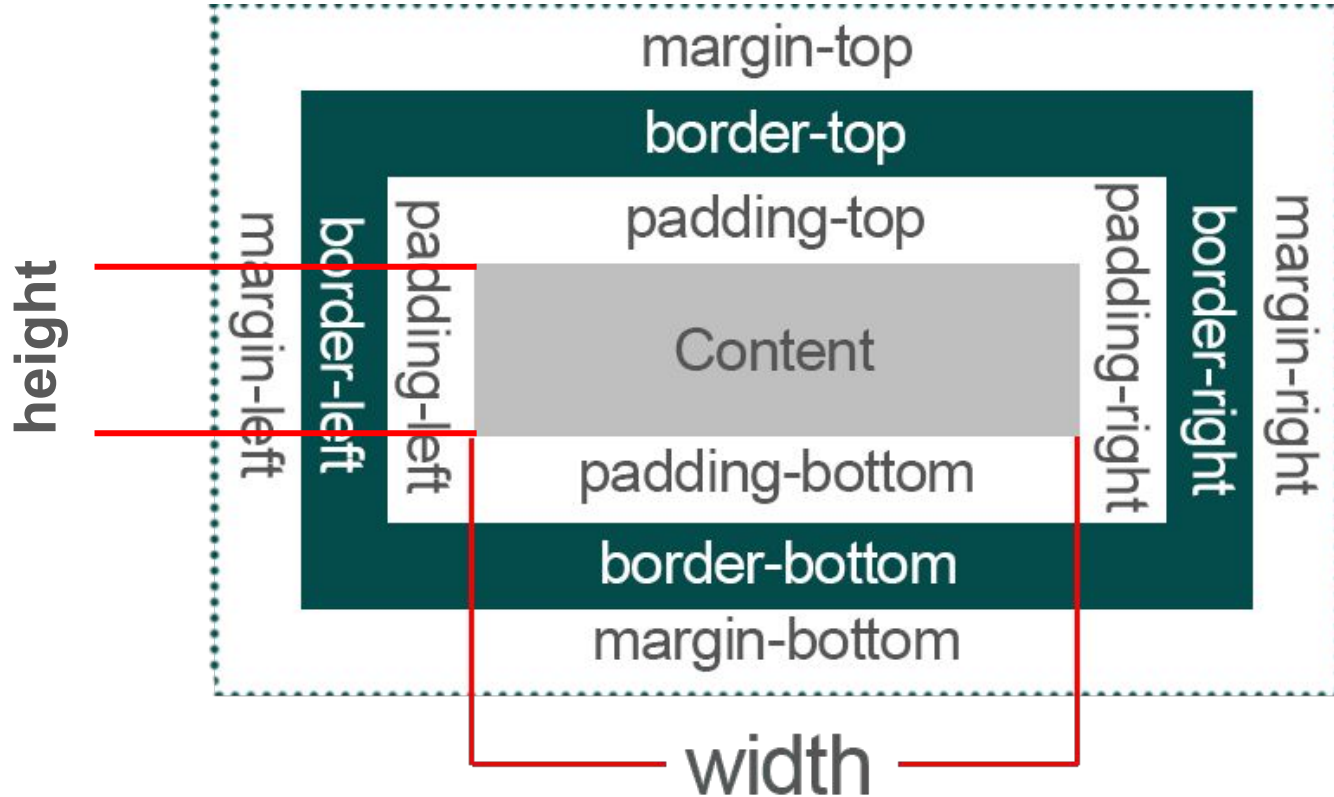
Normal border

Round border

Rounder border

Roudest border

# Propriedades CSS





# Propriedades CSS

- Margin
  - As margens são usadas para criar espaço ao redor dos elementos, fora de quaisquer bordas definidas.

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

```
p {  
  margin: 25px 50px 75px 100px;  
}
```

Margem Superior: 25px  
Margem Direita: 50px  
Margem Inferior: 75px  
Margem Esquerda: 100px

# Propriedades CSS

- Padding
  - O preenchimento é usado para criar espaço ao redor do conteúdo de um elemento, dentro de qualquer borda definida.

```
div {  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```

```
div {  
  padding: 25px 50px 75px 100px;  
}
```

# Propriedades de tamanho abreviadas

**p { padding: 4px }**

*Abreviação para:* padding-top: 4px; padding-right: 4px; padding-bottom: 4px; padding-left: 4px;

**p { padding: 2px 5px }**

*define os valores para:* (top/bottom) (left/right)

**p { padding: 2px 5px 3px }**

*define os valores para:* (top) (left/right) (bottom)

**p { padding: 2px 5px 3px 4px }**

*define os valores para:* (top) (right) (bottom) (left)

# Propriedades CSS

- Height/Width
  - As propriedades são usados para definir a altura e a largura de um elemento.

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: powderblue;  
}
```

# Propriedades CSS

- Max-width / Min-width
  - max-width define a largura máxima de um elemento.
  - min-width define a largura mínima de um elemento.

```
div {  
  max-width: 500px;  
  height: 100px;  
  background-color: powderblue;  
}
```

```
div {  
  min-width: 500px;  
  height: 100px;  
  background-color: powderblue;  
}
```

# Display

- A linguagem CSS possui algumas propriedades para controlar a forma de exibição e a visibilidade dos elementos HTML.

Ex: [https://www.w3schools.com/CSSref/pr\\_class\\_display.asp](https://www.w3schools.com/CSSref/pr_class_display.asp)

- Valores básicos:
  - None: Desabilita a exibição do elemento.
  - Inline: geralmente elementos de texto e possuem algumas propriedades específicas, dois elementos em linha serão exibidos um após o outro na horizontal, largura e altura não serão aplicados neste tipo de elemento.
  - Block: inicia uma nova linha, ocupando todo o espaço horizontal e pode ter largura e altura definidas.
  - inline-block: tem comportamento semelhante ao inline, porém pode ter altura e largura definidos.

# Display inline

```
<style>
.bloco1,.bloco2,.bloco3{
  display:inline;
  width:400px;
  height: 500px;;
}

.bloco1{
  background-color: ■ #ff00ff;
}

.bloco2{
  background-color: ■ #0044cc;
}

.bloco3{
  background-color: ■ #ffdd00;
}

</style>
```

```
<div class="bloco1">
  Bloco 1
</div>
<div class="bloco2">
  Bloco 2
</div>
<div class="bloco3">
  Bloco 3
</div>
```

Bloco 1 Bloco 2 Bloco 3

# Display block

```
.bloco1,.bloco2,.bloco3{  
  display:block;  
  width:70px;  
  height: 50px;;  
}
```

Bloco 1

Bloco 2

Bloco 3



# Display inline-block

```
.bloco1,.bloco2,.bloco3{  
  display:inline-block;  
  width:70px;  
  height: 50px;;  
}
```

Bloco 1

Bloco 2

Bloco 3

# Display inline-block

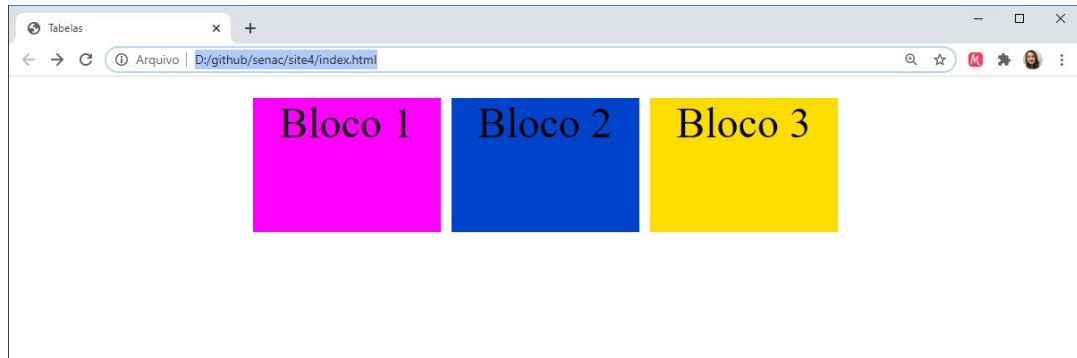
- Como o inline-block herda algumas características dos elementos inline, ele também herda a capacidade de se comportar como um texto.
- Uma propriedade que podemos utilizar é a text-align
- Text-align possui alguns possíveis valores:

```
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;  
text-align: justify-all;  
text-align: start;  
text-align: end;  
text-align: match-parent;
```

# Display inline-block

- Utilizando text-align no parent dos elementos inline-block:

```
body{  
  text-align:center;  
}  
  
.bloco1,.bloco2,.bloco3{  
  display:inline-block;  
  width:70px;  
  height: 50px;;  
}
```



# Propriedades CSS

- **Opacity**

- Define o nível de opacidade de um elemento.
- O nível de opacidade descreve o nível de transparência, em que 1 não é transparente, 0,5 é 50% transparente e 0 é completamente transparente.
- Ex: [https://www.w3schools.com/CSSref/css3\\_pr\\_opacity.asp](https://www.w3schools.com/CSSref/css3_pr_opacity.asp)



# Propriedades CSS

- **Filter**

- Define efeitos visuais (como desfoque e saturação) para um elemento (geralmente <img>).
- Ex: [https://www.w3schools.com/CSSref/css3\\_pr\\_filter.asp](https://www.w3schools.com/CSSref/css3_pr_filter.asp)

# Propriedades CSS

- **Formatação de tabelas**

- A aparência de uma tabela HTML pode ser muito melhorada com CSS
- Ex: [https://www.w3schools.com/css/css\\_table.asp](https://www.w3schools.com/css/css_table.asp)

# Position

- Define a posição do elemento, pode ao menos 4 valores:

## STATIC



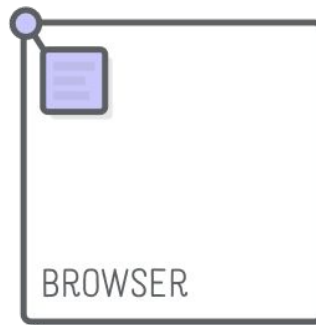
`position: static`  
Valor padrão

## RELATIVE



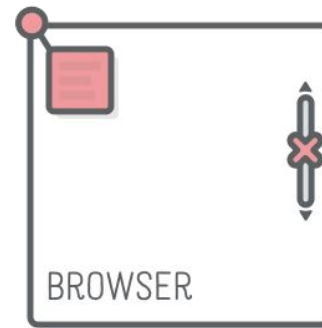
`position: relative`  
posição relativa a sua  
posição natural

## ABSOLUTE



`position: absolute`  
posição relativa à  
posição do  
box do elemento HTML  
ancestral mais próximo  
com position diferente  
de static ou a página

## FIXED



`position: fixed`  
posição relativa à  
página e não  
acompanha a rolagem  
do conteúdo

# Position - coordenadas

- Para posicionar seus elementos, você precisa inserir uma coordenada. Essas coordenadas são comandadas pelas propriedades: top, left, right ou bottom. Todos os valores de positions só trabalham com essas coordenadas.
- Se definir um left para o seu elemento, não faz sentido definir um right.
- Exemplo:

```
div {  
  position: absolute;  
  top: 150px;  
  left: 150px;  
}
```



# Position Static

- Esse é o valor default de todo elemento HTML, ou seja, ele vai seguir o fluxo comum da sua página.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan vehicula diam sit amet bibendum. Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin.

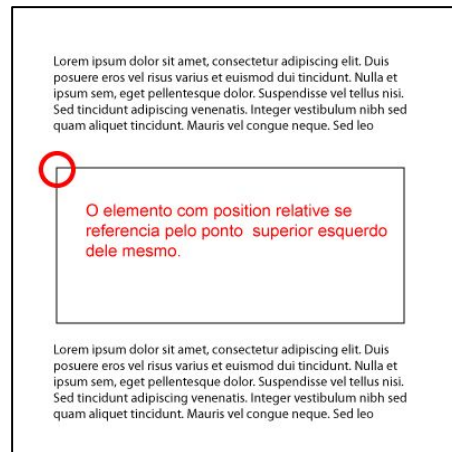
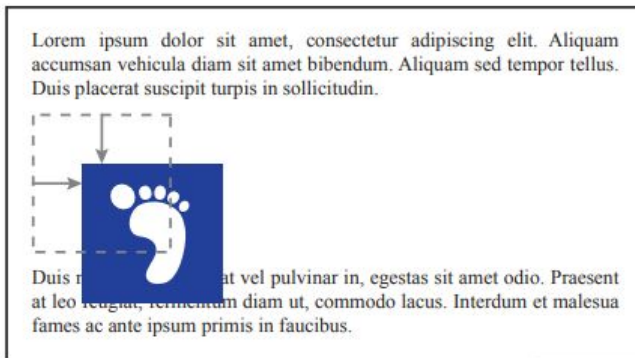


Duis nunc est, consequat vel pulvinar in, egestas sit amet odio. Praesent at leo feugiat, fermentum diam ut, commodo lacus. Interdum et malesuada fames ac ante ipsum primis in faucibus.

`position: static`

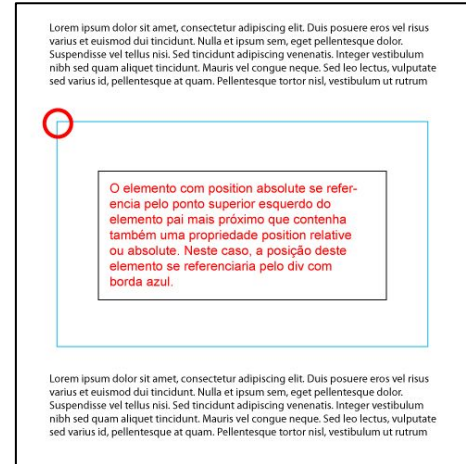
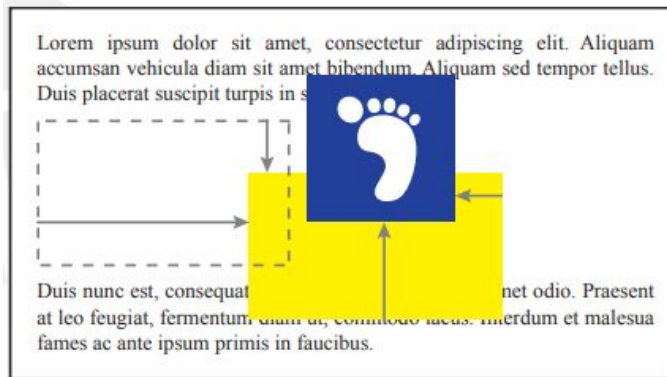
# Position relative

- Todos os positions precisam de um ponto para iniciar o cálculo da coordenada para assim posicionar o elemento na tela.
- O ponto base é o canto superior esquerdo do elemento. A partir deste canto, o browser irá calcular a coordenada que você definiu e irá posicionar o elemento no viewport.
- O position relative posiciona o elemento em relação a si mesmo. Ou seja, o ponto zero será o canto superior esquerdo, e ele começará a contar a partir dali.



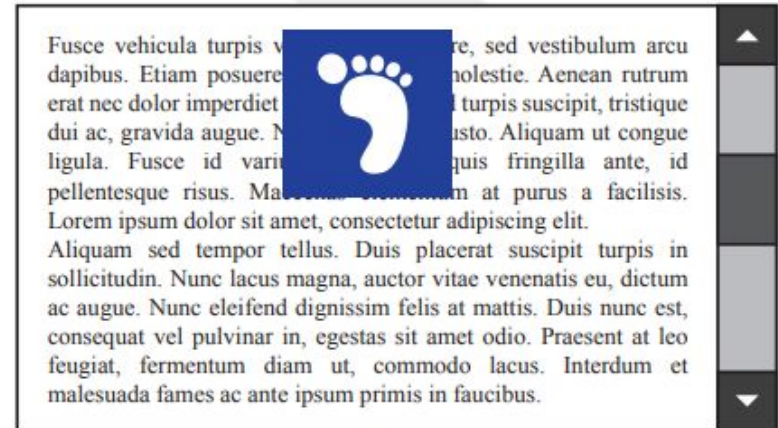
# Position absolute

- Enquanto o elemento com Position Relative utiliza seu próprio canto para referenciar sua posição, o elemento com Position Absolute se utiliza do ponto superior esquerdo de outros elementos. Estes elementos são os parentes do elemento com position absolute (o parente mais próximo que possuir um position definido).



# Position fixed

- O position: fixed; irá fixar a posição do elemento na coordenada que você definir. A medida que a página é rolada, o elemento continua fixo na posição que você definiu e o conteúdo da página rola normalmente.




position: fixed

# Float

- Podemos fazer o box de um elemento HTML flutuar ao redor dos demais elementos através da propriedade float.


Valor	Descrição
left	Flutuar à esquerda.
right	Flitar à direita.
none	Sem flutuação (padrão).
inherit	Assume o valor da mesma propriedade no elemento HTML pai.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan vehicula diam sit amet bibendum. Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin.

`float: left`

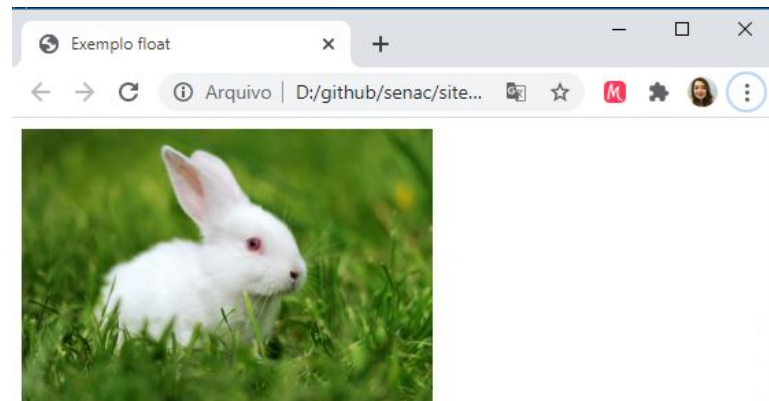
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan vehicula diam sit amet bibendum. Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin.



`float: right`

# Float – exemplo none

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemplo float</title>
5     <meta charset="UTF-8">
6     <style>
7       img{
8         width:300px;
9       }
10    </style>
11  </head>
12
13  <body>
14    <div>
15      
16      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
17      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
18      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
19    </div>
20  </body>
21 </html>
```



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



# Float – exemplo float left imagem

```
<style>  
  img{  
    width:300px;  
    float:left;  
  }  
</style>
```



# Float – exemplo float right imagem

```
<style>
  img{
    width:300px;
    float:right;
  }
</style>
```





# Float – exemplo float right imagem

```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  exemploFloat.html - htdocs - Visual Studio Code

exemploFloat.html X
ADS > <> exemploFloat.html > head > style > div
1  <!DOCTYPE html>
2  <head>
3      <meta charset="UTF-8">
4      <title>Exemplo Float</title>
5
6      <style>
7          img{
8              width: 300px;
9              float:left;
10         }
11         .paragrafo-baixo{
12             clear:left;
13         }
14         div{
15             position: absolute;
16             top:150px;
17             left: 150px;
18         }
19     </style>
20 </head>
21 <body>
22     <div>
23         
24         <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
25         <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
26         <p class="paragrafo-baixo">Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Dui
27     </div>
28 </body>
29 </html>
```

# Float – exemplo float right imagem

```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda exemploFloat.html - htdocs - Visual Studio Code
exemploFloat.html X
ADS > exemploFloat.html > head > style > .container
7      img{
8          width: 300px;
9          float:left;
10     }
11     .paragrafo-baixo{
12         clear:left;
13     }
14     div{
15         position: absolute;
16         top:150px;
17         left: 150px;
18     }
19     .container{
20         width: 800px;
21         margin: 0 auto;
22     }
23 </style>
24 </head>
25 <body>
26     <div class="container">
27         
28         <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
29         <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
30         <p class="paragrafo-baixo">Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Dui
31     </div>
32 </body>
33 </html>
```

# Clear

- Tem a função de controlar o comportamento de elementos que apareçam no fluxo do documento após determinado elemento que possui a propriedade *float*.
- Especifica se um elemento deve ser posicionado ao lado de elementos com *float* ou se devem ser colocados abaixo deles. A propriedade aceita 4 valores:

Valor	Descrição
left	Não permite boxes flutuando à esquerda dele.
right	Não permite boxes flutuando à direita dele.
both	Não permite boxes flutuando nem à esquerda nem à direita dele.
none	Permite boxes flutuando à esquerda e à direita (padrão) dele.
inherit	Assume o valor da mesma propriedade no elemento HTML pai.

# Clear



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

`clear: none`

Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus magna, auctor vitae venenatis eu, dictum ac augue.

`clear: none`



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

`clear: left`

Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus

`clear: left`



Lorem ipsum dolor sit amet, consectetur adipiscing elit.

`clear: none`

Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus magna, auctor vitae venenatis eu, dictum ac augue.

`clear: none`

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



`clear: right`

Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus

`clear: right`



Lorem ipsum dolor sit amet.

`clear: none`



Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus magna, auctor vitae venenatis eu, dictum ac augue.

`clear: none`



Lorem ipsum dolor sit amet.

`clear: both`

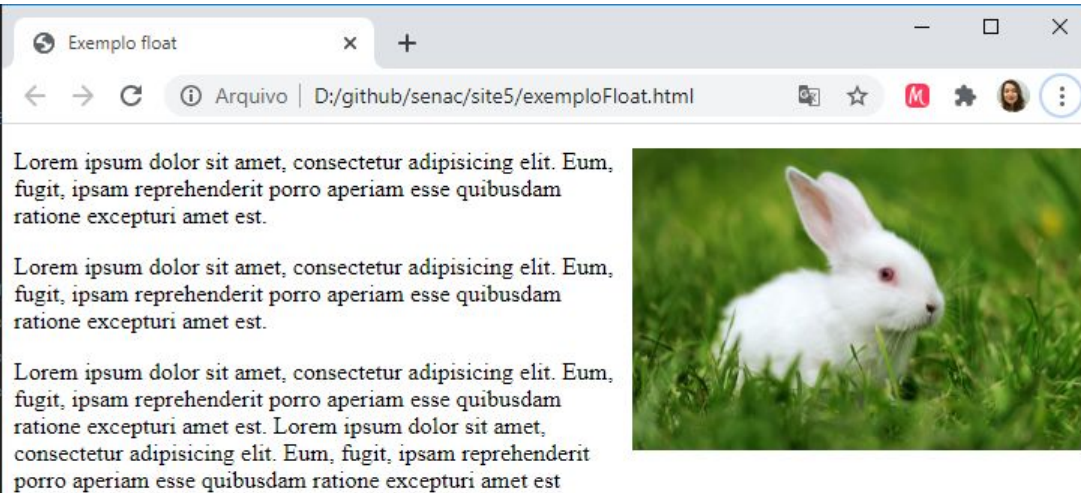
Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus

`clear: both`

# Float – exemplo sem clear

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo float</title>
    <meta charset="UTF-8">
    <style>
      img{
        width:300px;
        float:right;
      }
    </style>
  </head>

  <body>
    <div>
      
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
      <p class="paragrafo-abaxio">Lorem ipsum dolor sit amet, consectetur
    </div>
  </body>
</html>
```





# Float – exemplo com clear right

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo float</title>
    <meta charset="UTF-8">
    <style>
      img{
        width:300px;
        float:right;
      }
    </style>
  </head>

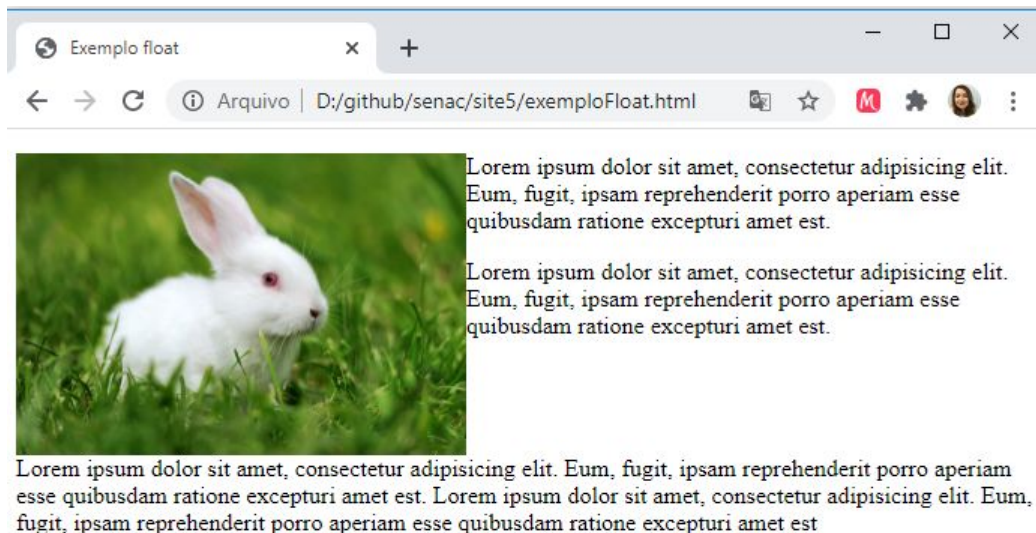
  <body>
    <div>
      
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eum, fugit, ipsam reprehenderit porro aperiam esse quibusdam ratione excepturi amet est.
      <p class="paragrafo-abixo">Lorem ipsum dolor sit amet, consectetur
    </div>
  </body>
</html>
```



# Float – exemplo com clear left

```
<style>
  img{
    width:300px;
    float:left;
  }

  .paragrafo-abaixo{
    clear:left;
  }
</style>
```



## Z-Index

- A sobreposição dos boxes dos elementos HTML pode ser controlada através da propriedade z-index. Basicamente, essa propriedade permite determinar em qual plano um box deve ser exibido.

Valor	Descrição
auto	Mesmo plano do box do elemento HTML pai (padrão).
<i>x</i>	O valor <i>x</i> é o plano desejado. Esse valor pode ser negativo.
inherit	Assume o valor da mesma propriedade no elemento HTML pai.



# Z-Index

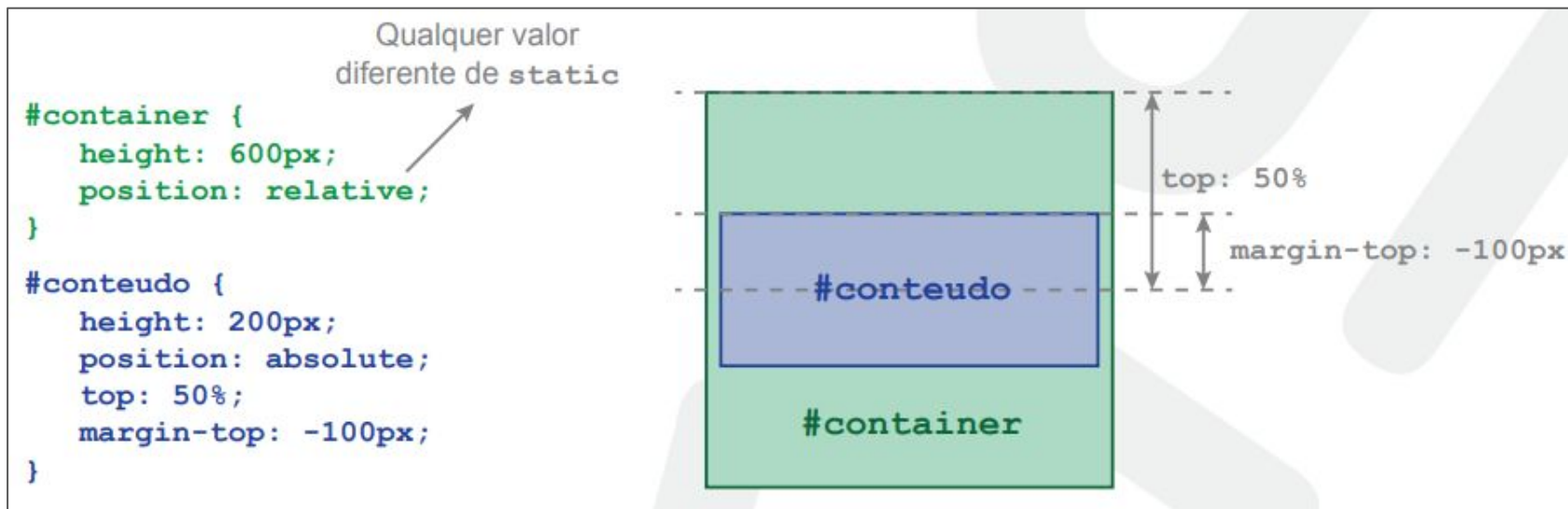
z-index: 1

z-index: 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan vehicula diam sit amet bibendum. Aliquam sed tellus. Duis placerat suscipit turpis in sollicitudin. Nunc lacus magna, auctor vitae vestibulum. Nunc ac augue. Nunc eleifend dignissim felis at mattis. Duis nunc est, consequat egestas sit amet odio. Praesent at leo feugiat, fermentum diam ut, commodo

## Alinhamento vertical

- Utilizando as propriedades de margem e posicionamento podemos alinhar um elemento de forma centralizada na vertical.



# CSS Grid

- [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)

# Elementos de bloco x Elementos de linha

- Elementos em Bloco

- div (divisão, tag bloco genérica)
- h1 até h6 (títulos)
- p (parágrafo)
- ul (lista não ordenada)
- ol (lista ordenada)
- form (formularios)

- Elementos em linha

- span (tag em linha genérica)
- strong (destaca importância)
- em (ênfase)
- a (âncora, usada para links)
- img (imagem)
-

# CSS Reset e CSS Normalize

- CSS Reset

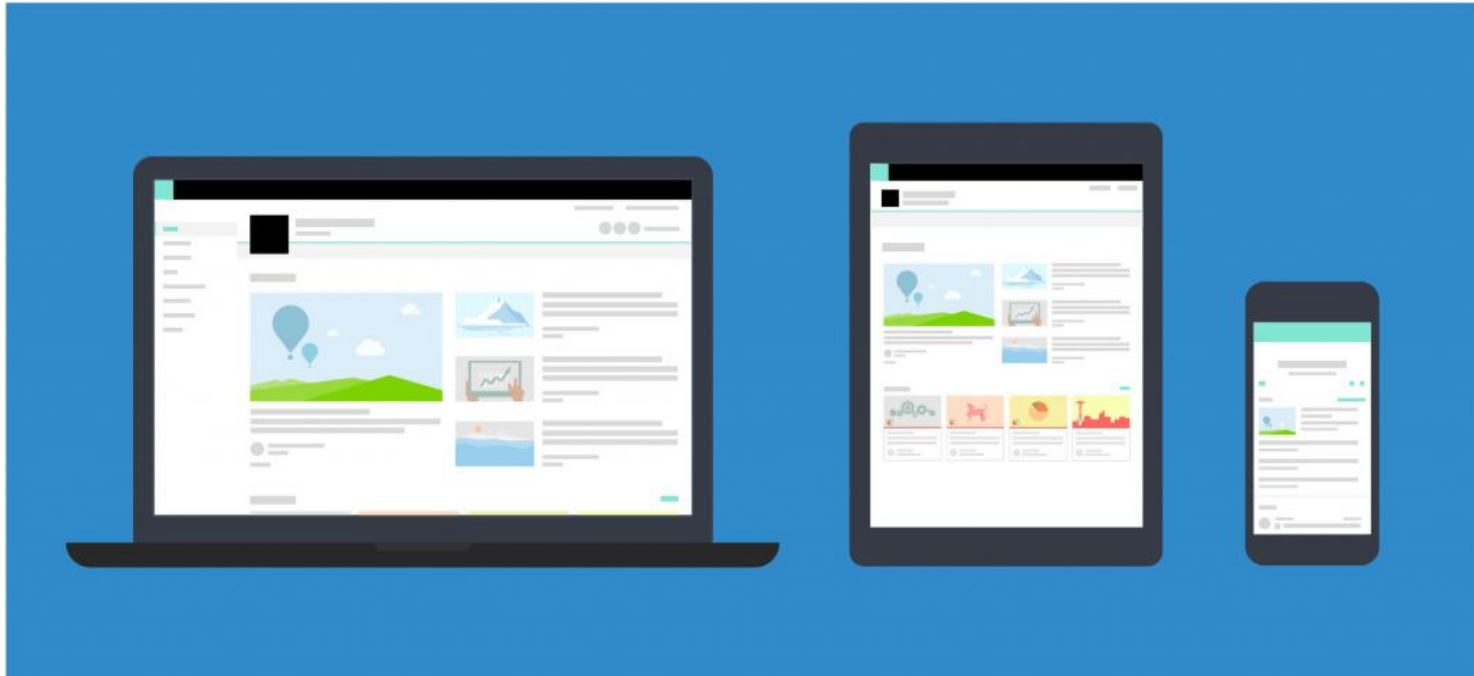
```
* {  
    vertical-align: baseline;  
    font-weight: inherit;  
    font-family: inherit;  
    font-style: inherit;  
    font-size: 100%;  
    border: 0 none;  
    outline: 0;  
    padding: 0;  
    margin: 0;  
}
```

- CSS Normalize

```
section {  
    display: block;  
}
```

# Responsividade

- [https://www.w3schools.com/css/css\\_rwd\\_intro.asp](https://www.w3schools.com/css/css_rwd_intro.asp)



# Responsividade

- Media Query
  - A @media, introduzida no CSS2, possibilitou definir diferentes regras de estilo para diferentes tipos de mídia.
  - As consultas de mídia podem ser usadas para verificar muitas coisas, como:
    - largura e altura da janela de visualização
    - largura e altura do dispositivo
    - orientação (o tablet/telefone no modo paisagem ou retrato)
    - resolução
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_media\\_queries2](https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries2)

# Responsividade

- Media Query

- Ex:

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
  body {  
    background-color: pink;  
  }  
  
  @media screen and (min-width: 480px) {  
    body {  
      background-color: lightgreen;  
    }  
  }  
</style>  
</head>  
<body>  
  
  <h1>Resize the browser window to see the effect!</h1>  
  <p>The media query will only apply if the media type is screen and the  
  viewport is 480px wide or wider.</p>  
  
</body>  
</html>
```



# Responsividade

- Viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

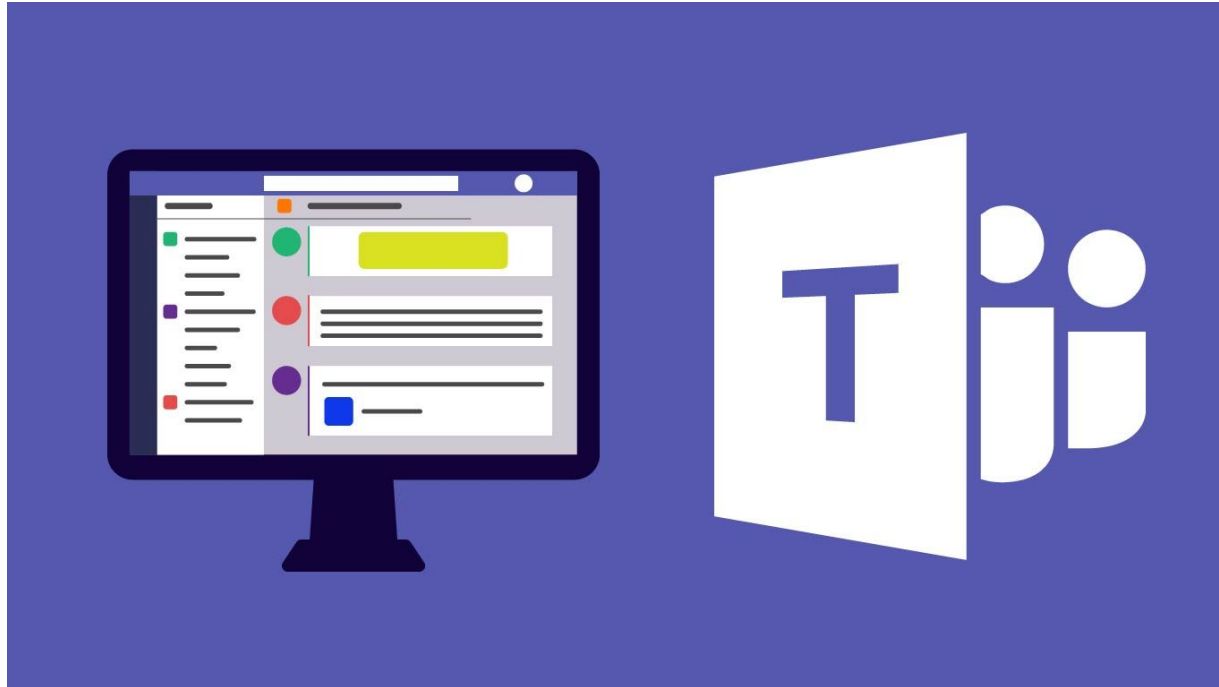
- Fornece ao navegador instruções sobre como controlar as dimensões e o dimensionamento da página.
- A ***width=device-width*** define a largura da página para seguir a largura da tela do dispositivo (que varia de acordo com o dispositivo).
- A ***initial-scale=1.0*** define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

# Exemplo navbar

- Desenvolver navbar com a turma.

# Atividades

- Entregar Atividades no TEAMS



# CSS - AulaWeb

**Fale conosco**

**Nome:**

**E-mail:**

**Mensagem:**

# CSS - AulaWeb

## Clientes cadastrados

Cliente	Idade	Cidade	Estado	E-mail
Ana Regina	29	Curitiba	Paraná	ana@gmail.com
Fernanda Silva	42	Joinville	Santa Catarina	fernanda@gmail.com
Rodrigo Ferreira	70	Curitiba	Paraná	rodrigo@gmail.com

### Cadastrar novo cliente

Nome:

Idade:

Cidade:

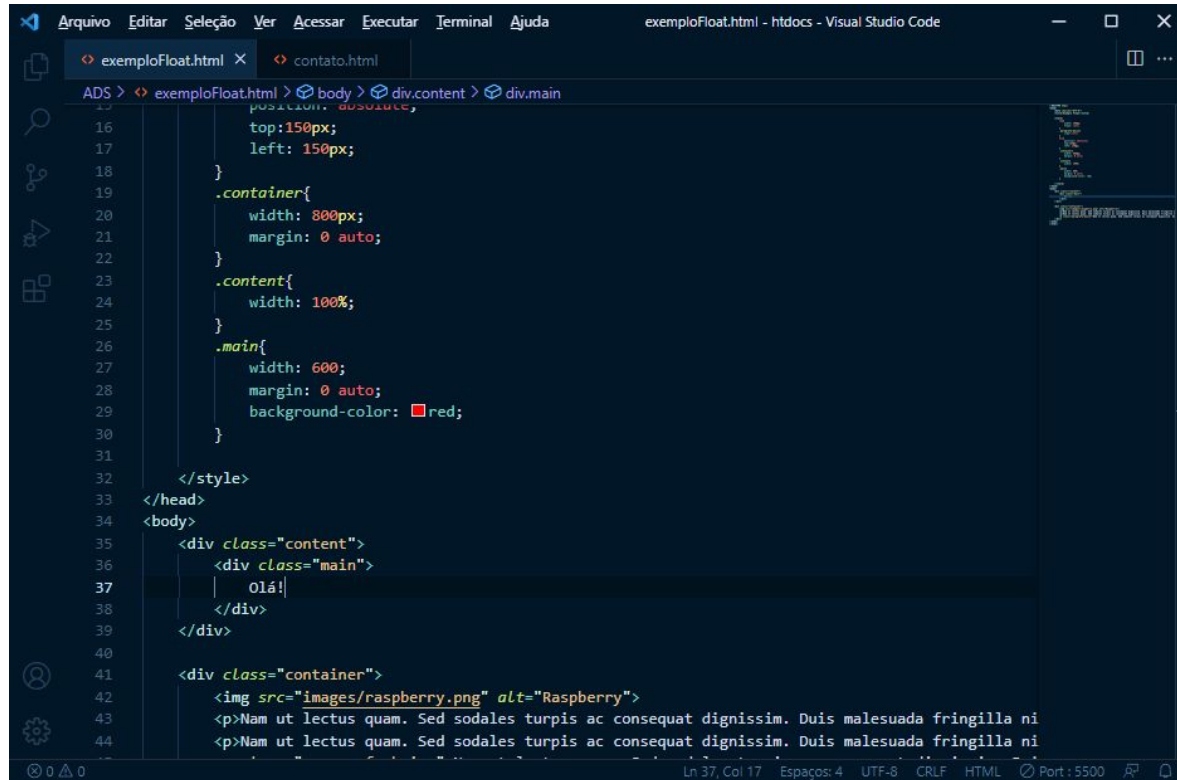
Estado:  ▼

E-mail:

Cancelar

Salvar

# Exemplo com Main



The screenshot shows the Visual Studio Code editor with two files open: `exemploFloat.html` and `contato.html`. The `exemploFloat.html` file is active, displaying CSS and HTML code. The CSS code defines a `.main` class with a red background and a `.container` class with a width of 800px. The HTML code uses these classes to structure the page content, including a main heading and a container for a Raspberry image and text.

```
15 position: absolute;
16 top: 150px;
17 left: 150px;
18 }
19 .container{
20 width: 800px;
21 margin: 0 auto;
22 }
23 .content{
24 width: 100%;
25 }
26 .main{
27 width: 600;
28 margin: 0 auto;
29 background-color: red;
30 }
31
32 </style>
33 </head>
34 <body>
35 <div class="content">
36 <div class="main">
37 | Olá!
38 </div>
39 </div>
40
41 <div class="container">
42 
43 <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
44 <p>Nam ut lectus quam. Sed sodales turpis ac consequat dignissim. Duis malesuada fringilla ni
```

# Atividade

## Login

Usuário

---

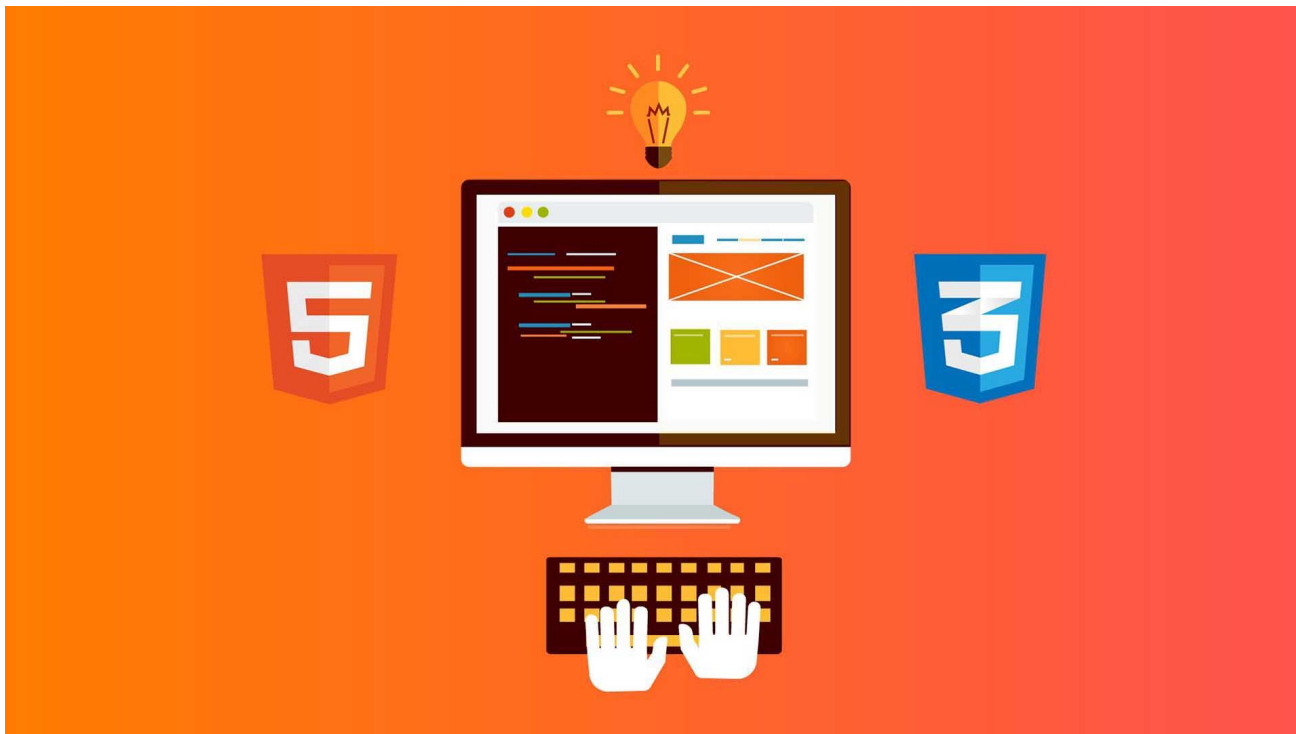
Senha

---

login

[Esqueceu a senha?](#)

# Projeto Final





# Trabalho

---

- **Desenvolva uma página pessoal, contando um pouco sobre você, hobby, atividades extracurriculares, trabalho, formação acadêmica – Utilize “Ancoras”**
- **Seja criativo insira figuras, altere cores de fundo e das fontes.**
- **Aplique todas os recursos ensinados.**
- **Busque novas funcionalidades (semânticas) na internet.**
- **Procure por modelos na internet**



# Trabalho Dicas

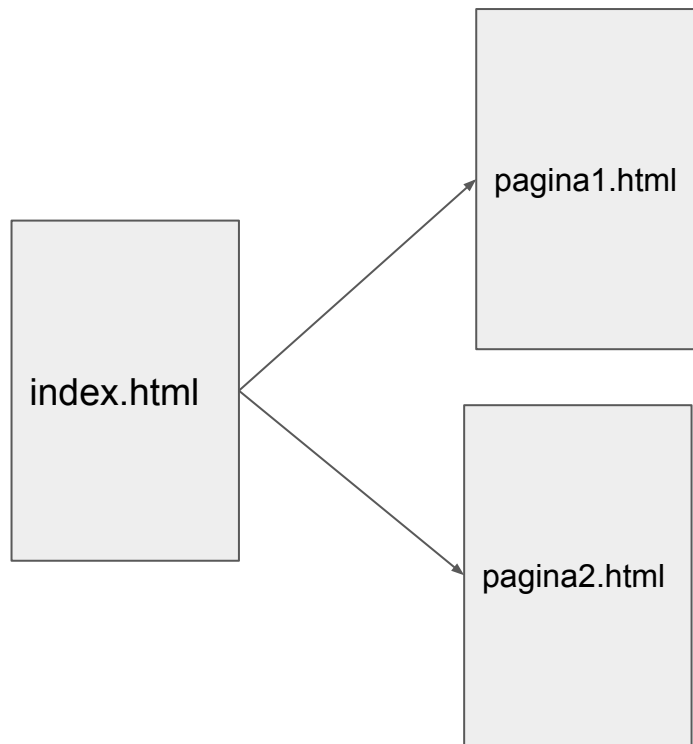
---

- **Página Pessoal contendo**

- **Título**
- **Fale sobre você**
- **Hobbies**
- **Disciplinas frequentadas**
- **Esportes praticados**
- **Leituras**

- **Utilizar**

- **html5**
- **imagens**
- **tabelas**
- **âncoras**
- **tags (pesquisar +3)**
- **diretório contendo as imagens**



# Referências

- Básicas: [**OBRIGATORIAMENTE 3 BIBLIOGRAFIAS**]
- ARAÚJO, Everton Coimbra de. Desenvolvimento para web com Java. Florianópolis: Visual Books, 2010.
- QIAN et al. Desenvolvimento Web Java. LTC, 2010.
- GOMES, Yuri Marx Pereira. Java na web com JSF, Spring, Hibernate e Netbeans 6. Rio de Janeiro, RJ: Ciência Moderna, 2008.
  
- Complementar: [**OBRIGATORIAMENTE 5 BIBLIOGRAFIAS**]
- GONÇALVES, Edson. Desenvolvendo aplicações web com netbeans ide 6. Rio de Janeiro, RJ: Ciência Moderna, 2008.
- LUCKOW, Décio Heinzelmänn; MELO, Alexandre Altair de. Programação Java para Web. 2ª Ed. São Paulo: Novatec, 2015.
- SOUZA, Thiago Hernandez de. Java mais primefaces mais ireport: desenvolvendo um CRUD para web. Rio de Janeiro, RJ: Ciência Moderna, 2013.
- KALIN, Martin. Java Web Services: implementando. Rio de Janeiro, RJ: Alta Books, 2010.
- EIS, Diego. Guia Front-End: o caminho das pedras para ser um Dev Front-End. São Paulo: Casa do Código, 2015.
- SOUZA, Alberto. Spring MVC: domine o principal framework web Java. São Paulo: Casa do Código, 2015.

