



**INSTITUTO FEDERAL**  
Sul de Minas Gerais

# Posicionamento de elementos HTML via CSS



**Disciplina: Tecnologia Web**



**Prof. Me. Fernando Roberto Proença**

2

## **Propriedades *Margin* e *Padding***

## CSS – Propriedade *Margin*

3

- Define as dimensões das ***margens externas*** de um elemento HTML, ou seja, a distância de um elemento para outro.
- Caso os elementos não tenham margens em torno deles, eles vão “encostar” uns nos outros.
- Alguns elementos têm margens por padrão, ou seja, mesmo não adicionando margem o elemento já é criado com margens. Exemplo: os elementos de cabeçalhos h1, h2, ..., h6.
- Podem ser definidas em porcentagem ou valores absolutos.

## CSS – Propriedade *Margin*

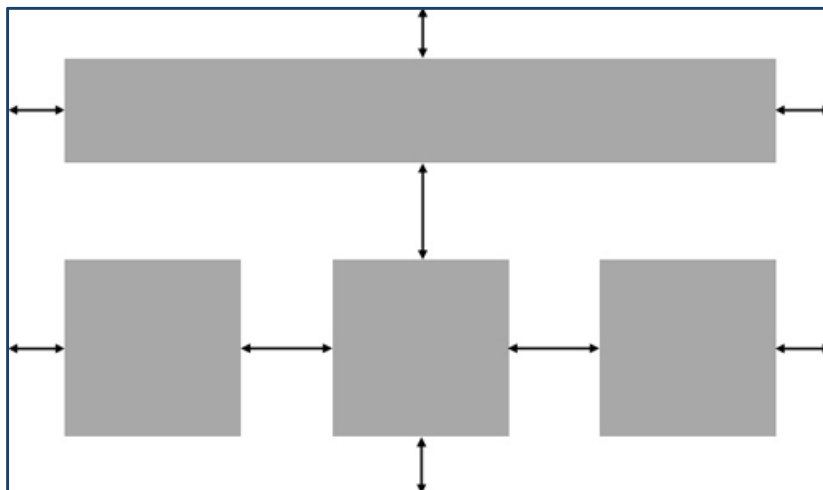
4

- Suas propriedades são:
  - ▣ ***margin***: propriedade de atalho para definição de todas as margens ao mesmo tempo;
  - ▣ ***margin-bottom***: definição da margem inferior do elemento;
  - ▣ ***margin-left***: definição da margem esquerda do elemento;
  - ▣ ***margin-right***: definição da margem direita do elemento;
  - ▣ ***margin-top***: definição da margem superior do elemento.

# CSS – Propriedade *Margin*

5

- Exemplo das margens externas em CSS



# CSS – Propriedade *Margin*

6

Propriedade	Descrição	Valores	Exemplo
margin (não herdado)	especifica a margem do elemento	tamanho (pt, in, cm, px)	{ margin: 8px; }
margin-top margin-left margin-bottom margin-right (não herdado)	especifica cada uma das margens	tamanho (pt, in, cm, px)	{ margin-top: 12pt; }

```
{ margin: 5px 10px 15px 20px; }
```

- *Margem superior de 5px. Margem direita de 10px. Margem inferior de 15px. Margem esquerda de 20px.*

## CSS – Propriedade *Padding*

7

- Define as ***margens internas***, ou seja, determina o **espaçamento** entre o conteúdo e a borda de um elemento HTML.
- Ao contrário da propriedade ***margin***, a propriedade ***padding*** define o espaçamento interno do elemento.
- Esta borda por padrão não aparece para o usuário.
- Podem ser definidos valores absolutos ou porcentagens.

## CSS – Propriedade *Padding*

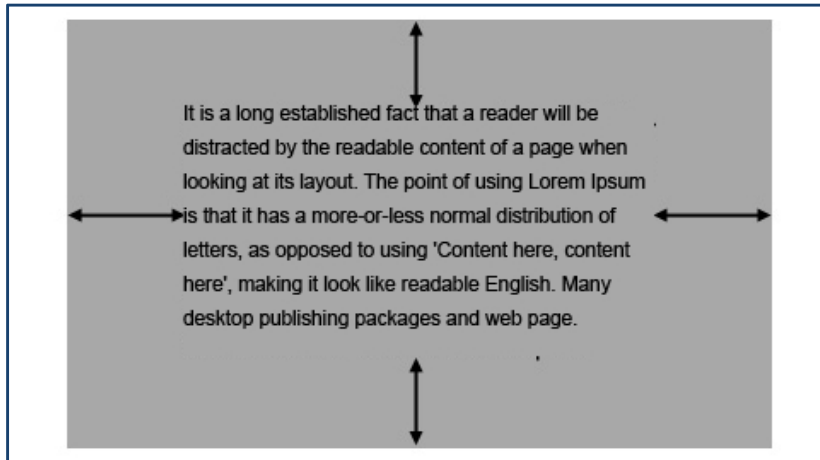
8

- Suas propriedades são:
  - ▣ ***padding***: propriedade de atalho para definição de todos os espaçamentos ao mesmo tempo;
  - ▣ ***padding-bottom***: define o espaçamento inferior do elemento;
  - ▣ ***padding-left***: define o espaçamento esquerdo do elemento;
  - ▣ ***padding-right***: define o espaçamento direito do elemento;
  - ▣ ***padding-top***: define o espaçamento superior do elemento.

# CSS – Propriedade *Padding*

9

- Exemplo das espaçamentos (margens internas) em CSS



# CSS – Propriedade *Padding*

10

Propriedade	Descrição	Valores	Exemplo
padding (herdado)	Espaço entre a borda do elemento e o conteúdo do elemento	tamanho (pt, in, cm, px)	{ padding: 20px; }
padding-top padding-left padding-bottom padding-right (herdado)	Em relação a cada um dos lados	tamanho (pt, in, cm, px)	{ padding-top: 0.5in; }

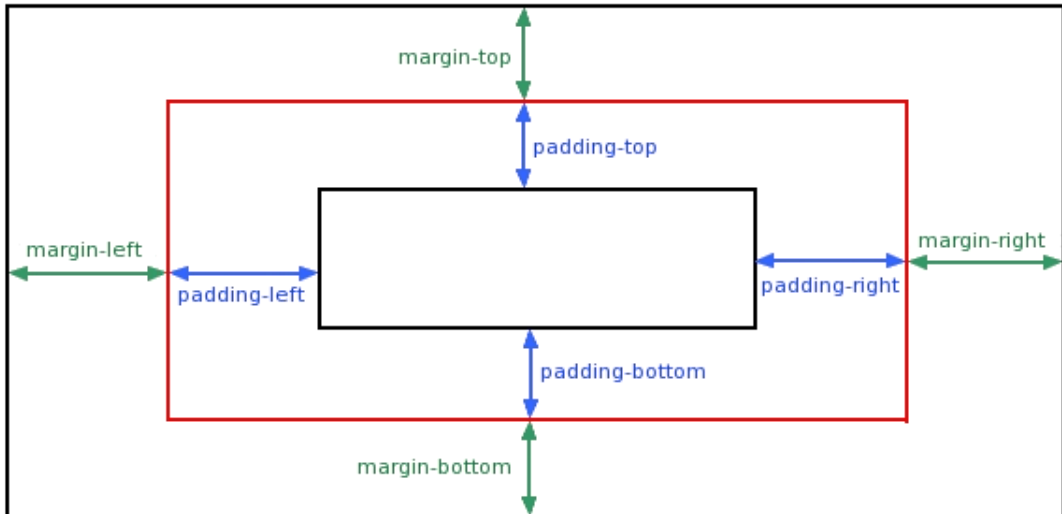
```
{ padding: 5px 10px 15px 20px; }
```

- *Margem superior interna de 5px. Margem direita interna de 10px. Margem inferior interna de 15px. Margem esquerda interna de 20px.*

# CSS – Margin e Padding

11

## □ Diferença entre *Margin* e *Padding*



## CSS – Margin e Padding – Exemplo - Arquivo "margin-e-padding.html"

12

```
<head>
  <title>Exemplo Margin e Padding</title>
  <link rel="stylesheet" href="style-margin-padding.css">
</head>
<body>
<header><h1>Exemplo Margin e Padding</h1></header>
<main>
  <div class="primeira">Primeira</div>
  <div class="segunda">Segunda</div>
  <div class="terceira">Terceira</div>
</main>
<footer><p>Todos os direitos reservados</p></footer>
</body>
```

# CSS – Margin e Padding – Exemplo - Arquivo "style-margin-padding.css"

13

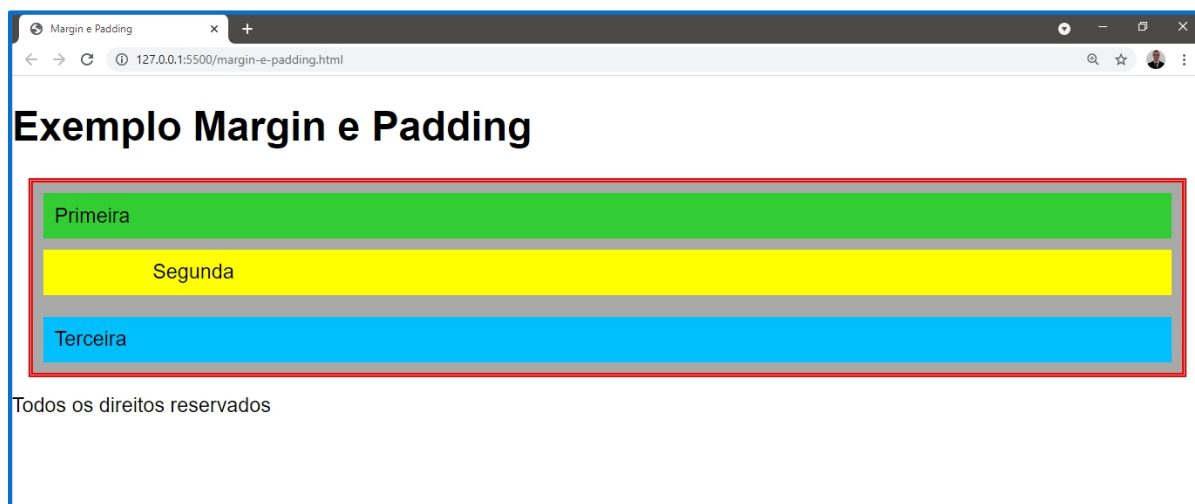
```
body, header, main, footer, div, p {  
    margin: 0; padding: 0;  
    font-family: Arial; font-size: 14pt; }  
main {  
    background-color: darkgrey;  
    border: 4px double red;  
    margin: 15px; padding: 10px;  
}  
.primeira {  
    background-color: limegreen;  
    margin-bottom: 10px; padding: 10px;  
}
```

```
.segunda {  
    background-color: yellow;  
    padding: 10px;  
    padding-left: 100px;  
}  
.terceira {  
    background-color: deepskyblue;  
    margin-top: 20px;  
    padding: 10px;  
}
```

## CSS – Margin e Padding – Exemplo

14

- Interface da página até o momento...



## CSS – Bordas (ou contornos)

- Definem o estilo, as cores e a espessura das bordas do elemento.
- Podem ser definidos valores absolutos ou porcentagens.
- Suas propriedades são:
  - ***border***: propriedade de atalho para definição das propriedades *border-top*, *border-right*, *border-bottom* e *border-left* ao mesmo tempo.
  - ***border-bottom***: propriedade de atalho para definição das propriedades *border-width*, *border-style* e *border-color* ao mesmo tempo.
  - ***border-bottom-color***: define a cor da borda inferior do elemento. É possível indicar o valor "*transparent*".
  - ***border-bottom-style***: define o estilo da borda inferior do elemento.



## CSS – Bordas (ou contornos)

17

- Suas propriedades podem ter os seguintes valores:
  - ▣ **none**: nenhuma estilo é definido
  - ▣ **dotted**: a borda é pontilhada
  - ▣ **dashed**: a borda é tracejada
  - ▣ **solid**: a borda é uma linha
  - ▣ **inset / outset**: a borda é uma linha em 3D
  - ▣ **double**: a borda é uma linha dupla
- Outra propriedade:
  - ▣ **border-bottom-width**: define a largura da borda inferior do elemento.

## CSS – Bordas (ou contornos)

18

Propriedade	Descrição	Valores	Exemplo
border (herdado)	Ajusta o contorno do elemento	grossura: em pixels. estilo: none, dotted, dashed, solid, double. cor: <i>valor da cor</i> .	<pre>{ border: 4px double green; }</pre>
border-top border-left border-bottom border-right (não herdado)	Ajusta cada um dos contornos	grossura: em pixels. estilo: none, dotted, dashed, solid, double. cor: <i>valor da cor</i> .	<pre>{ border-top: 2px dotted red; }</pre>

## CSS – Dimensionamento

- Define o **tamanho** dos elementos que devem ser exibidos na página. As propriedades são:
  - ▣ **height**: Define a **altura** de um elemento da página HTML. O tamanho pode ser fixo ou em percentual.
  - ▣ **max-height**: Define a **altura máxima** de um elemento, mesmo que a altura tenha um valor maior. Além de números e percentuais, pode-se usar *none*.
  - ▣ **min-height**: Define a **altura mínima** de um elemento, mesmo que a altura tenha um valor maior. Além de números e percentuais, pode-se usar *none*.

## CSS – Dimensionamento

21

- Define o **tamanho** dos elementos que devem ser exibidos na página. As propriedades são:
  - ▣ ***width***: Define a **largura** de um elemento. Seus valores podem ser números, percentuais e *auto*, onde ela é calculada automaticamente.
  - ▣ ***max-width***: Define a **largura máxima** do elemento, mesmo que a largura seja maior. Além de números e percentuais, pode-se usar *none*.
  - ▣ ***min-width***: Define a **largura mínima** do elemento, mesmo que a largura seja maior. Além de números e percentuais, pode-se usar *none*.

22

## CSS - propriedade “*display*”

## CSS – Propriedade *Display*

23

- A propriedade ***display***, define a maneira como determinado elemento HTML deve ser renderizado (exibido);
- Cada elemento HTML tem um valor padrão de exibição.

## CSS – Propriedade *Display*

24

- Valores possíveis para a propriedade ***display***:
  - **block**: O elemento se comporta como um bloco.
    - Algumas TAGs que possuem este display por padrão: <div>, <header>, <footer>, <p>, <h1>, <h2>, etc.
  - **inline**: O elemento se comporta como um elemento em linha.
    - Algumas TAGs que possuem este display por padrão: <span> e <a>.
  - **inline-block**: Semelhante ao *inline*, no entanto, é possível definir as propriedades de largura e altura.

# CSS – Propriedade *Display*

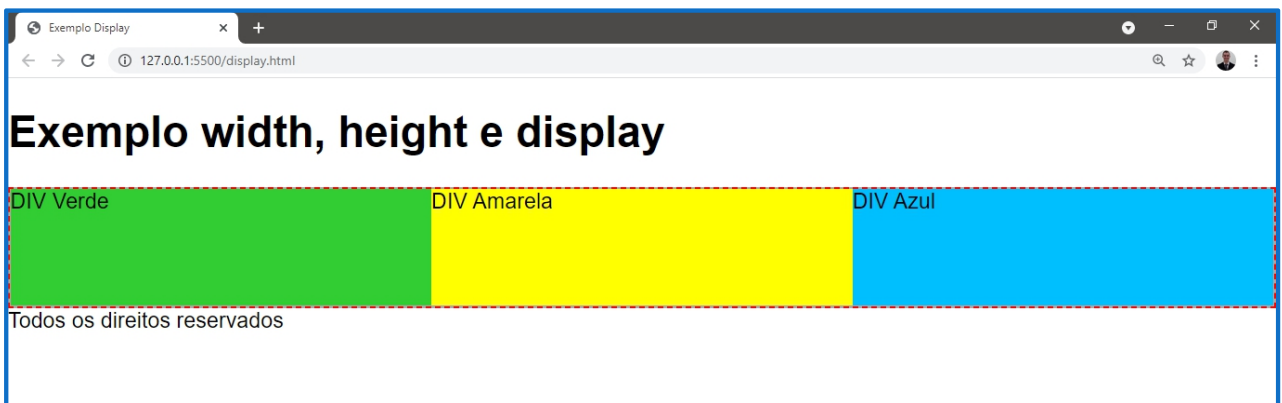
25

- Valores possíveis para a propriedade *display*:
  - ▣ **table**: O elemento se comporta como uma tabela.
  - ▣ **none**: Quando utiliza essa propriedade, o elemento HTML e todos seus elementos filhos não são renderizados (exibidos) na página.
  - ▣ **flex**: Implementado no CSS3, torna o elemento HTML um "*flex container*", transformando automaticamente todos os seus filhos diretos em itens "*flex*".

## CSS – Propriedade *Display* – Exemplo

26

- Página "*display.html*" desejada...



## CSS – Propriedade *Display* – Exemplo - Arquivo "*display.html*"

27

```
<head>
  <title>Exemplo Display</title>
  <link rel="stylesheet" href="style-display.css">
</head>
<body>
<header><h1>Exemplo width, height e display</h1></header>
<main>
  <div class="verde">DIV Verde</div>
  <div class="amarela">DIV Amarela</div>
  <div class="azul">DIV Azul</div>
</main>
<footer><p>Todos os direitos reservados</p></footer>
</body>
```

## CSS – Propriedade *Display* – Exemplo - Arquivo "*style-display.css*"

28

```
body, header, main, footer, div, p {
  margin: 0; padding: 0;
  font-family: Arial; font-size: 14pt; }
main {
  background-color: darkgrey;
  border: 4px dashed red;
}
.verde {
  background-color: limegreen;
  width: 33.33%; height: 100px;
}
```

```
.amarela {
  background-color: yellow;
  width: 33.33%;
  height: 100px;
}
.azul {
  background-color:
  deepskyblue;
  width: 33.33%;
  height: 100px;
}
```

# CSS – Propriedade *Display* – Exemplo

29

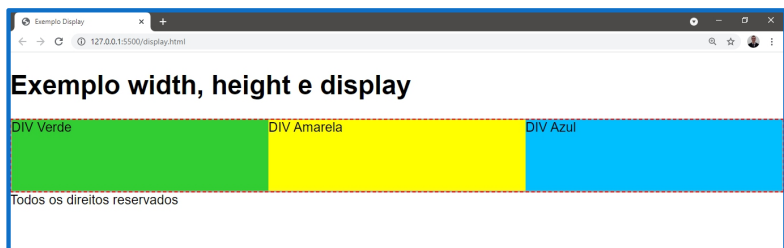
- Interface da página até o momento (sem definir o *display* dos parágrafos)...



# CSS – Propriedade *Display* – Exemplo

30

- Como alinhar as DIVs na horizontal?



# CSS – Propriedade *Display* – Exemplo

31

- Arquivo "css-display.css"

```
...  
.verde {  
  width: 33.33%; height: 150px;  
  background-color: limegreen;  
  display: inline;  
}
```

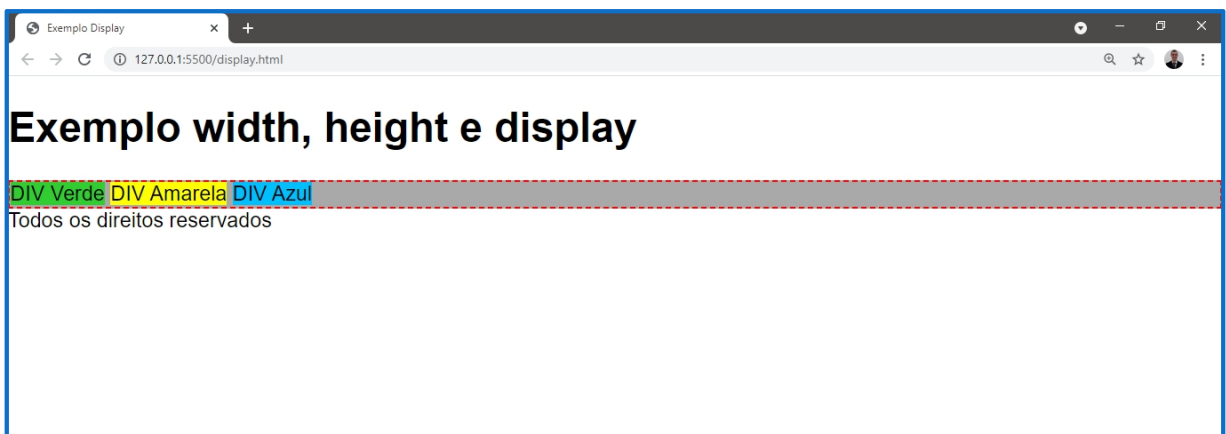
Vamos alterar o valor do *display* dos parágrafos para "inline"...

```
.amarela {  
  width: 33.33%; height: 150px;  
  background-color: yellow;  
  display: inline;  
}  
.azul{  
  width: 33.33%; height: 150px;  
  background-color: deepskyblue;  
  display: inline;  
}
```

# CSS – Propriedade *Display* – Exemplo

32

- Interface da página até o momento (*display: inline*)...





# CSS – Propriedade *Display* – Exemplo

33

- Arquivo "css-display.css"

```
...  
.verde {  
  width: 33.33%; height: 150px;  
  background-color: limegreen;  
  display: block;  
}
```

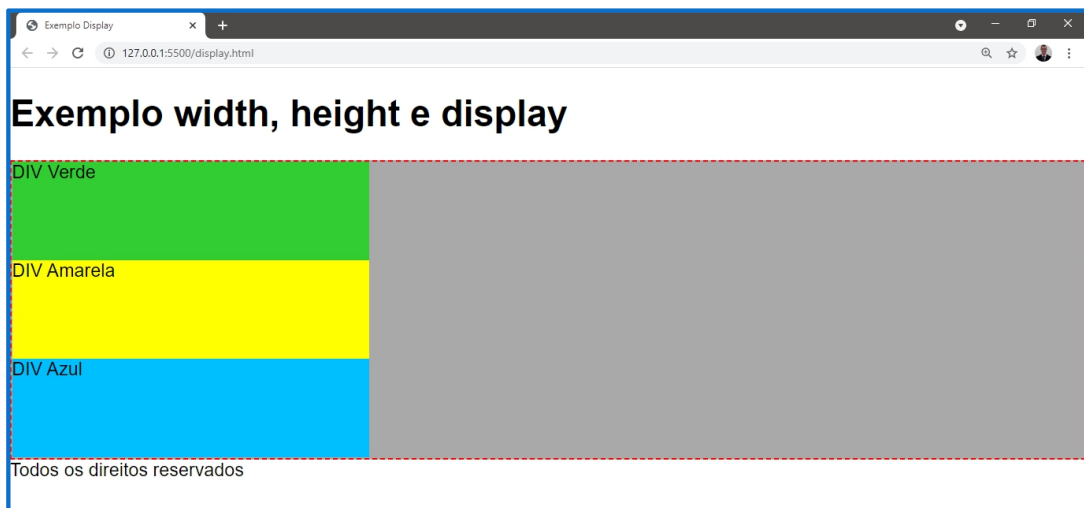
Vamos alterar o valor do *display* dos parágrafos para "block"...

```
.amarela {  
  width: 33.33%; height: 150px;  
  background-color: yellow;  
  display: block;  
}  
.azul {  
  width: 33.33%; height: 150px;  
  background-color: deepskyblue;  
  display: block;  
}
```

# CSS – Propriedade *Display* – Exemplo

34

- Interface da página até o momento (*display: block*)...



# CSS – Propriedade *Display* – Exemplo

35

- Arquivo "css-display.css"

```
...  
.verde {  
  width: 33.33%; height: 150px;  
  background-color: limegreen;  
  display: inline-block;  
}
```

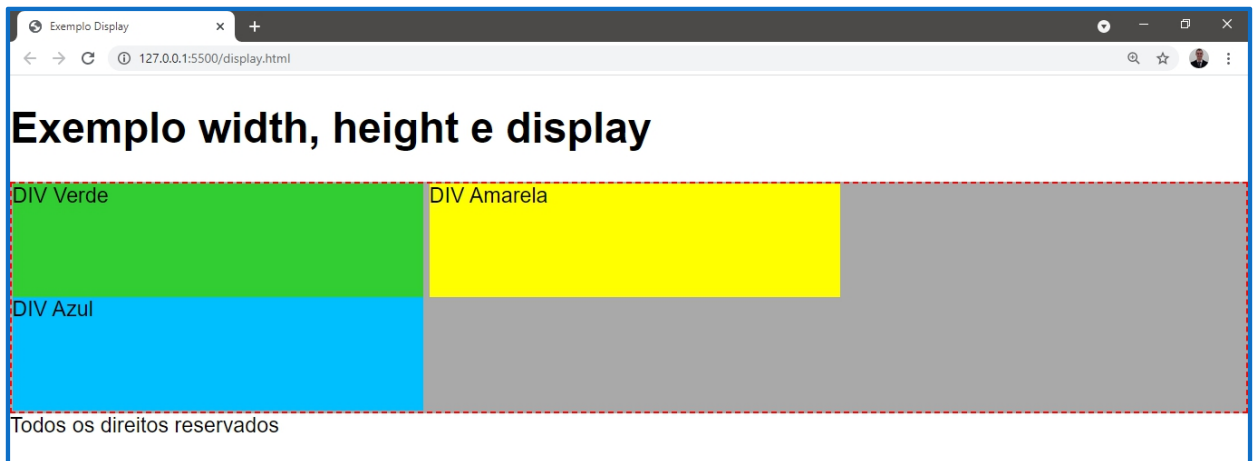
Vamos alterar o valor do *display* dos parágrafos para "inline-block"...

```
.amarela {  
  width: 33.33%; height: 150px;  
  background-color: yellow;  
  display: inline-block;  
}  
.azul {  
  width: 33.33%; height: 150px;  
  background-color: deepskyblue;  
  display: inline-block;  
}
```

# CSS – Propriedade *Display* – Exemplo

36

- Interface da página até o momento (*display: inline-block*)...



# CSS – Propriedade *Display* – Exemplo

37

- Arquivo "css-display.css"

```
...  
.verde {  
  width: 33.33%; height: 150px;  
  background-color: limegreen;  
  display: inline-block;  
}
```

Vamos alterar o valor do *display* do 2º parágrafo para "none"...

```
.amarela {  
  width: 33.33%; height: 150px;  
  background-color: yellow;  
  display: none;  
}  
.azul {  
  width: 33.33%; height: 150px;  
  background-color: deepskyblue;  
  display: inline-block;  
}
```

# CSS – Propriedade *Display* – Exemplo

38

- Interface da página até o momento (2º parágrafo com *display: none*)...



# CSS – Propriedade *Display* – Exemplo

39

- Arquivo "css-display.css"

```
body, header, main, footer, div, p{  
  margin: 0; padding: 0;  
  font-family: Arial;  
  font-size: 14pt; }  
main {  
  display: flex;  
}
```

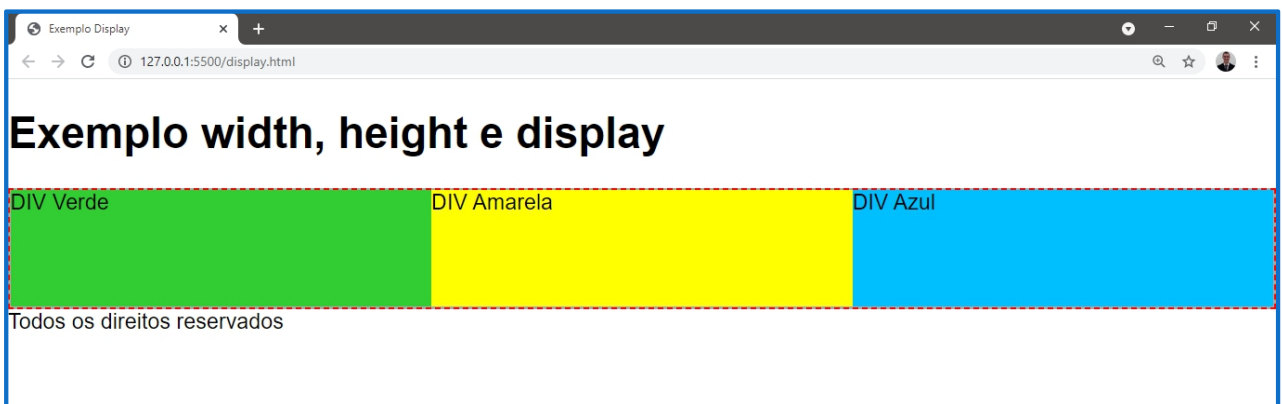
Vamos alterar o valor do *display* da DIV para "flex" ...

```
.verde {  
  width: 33.33%; height: 150px;  
  background-color: limegreen;  
}  
.amarela {  
  width: 33.33%; height: 150px;  
  background-color: yellow; }  
.azul {  
  width: 33.33%; height: 150px;  
  background-color: deepskyblue;  
}
```

# CSS – Propriedade *Display* – Exemplo

40

- Interface da página até o momento (*display: flex*)...



## CSS – FlexBox

- As propriedades de um contêiner (bloco) flexível (*display: flex*) são:
  - ▣ Direção: ***flex-direction***
  - ▣ Quebra linha: ***flex-wrap***
  - ▣ Fluxo: ***flex-flow***
  - ▣ Justificar conteúdo: ***justify-content***
  - ▣ Alinhar itens: ***align-items***
  - ▣ Alinhar caixas: ***align-content***

# CSS – FlexBox

43

## □ Propriedade ***flex-direction***

- ▣ Define em qual direção o contêiner deseja “empilhar” os elementos flexíveis.
- ▣ Valores possíveis: ***flex-direction: [ row (valor padrão), column, row-reverse, column-reverse ];***
- ▣ Exemplo:

```
main {  
  display: flex;  
  flex-direction: column;  
}
```

Os elementos “flex” serão exibidos em coluna...

# CSS – FlexBox

44

## □ Propriedade ***flex-wrap***

- ▣ Especifica se os elementos flexíveis serão quebrados ou não, caso “estourar” o tamanho da tela.
- ▣ Valores possíveis: ***flex-wrap: [ wrap, nowrap (valor padrão) ];***
- ▣ Exemplo:

```
main {  
  display: flex;  
  flex-wrap: wrap;  
}
```

Os elementos “flex” serão quebrados...

# CSS – FlexBox

45

- Propriedade ***flex-flow***
  - ▣ É uma junção das propriedades *flex-direction* e *flex-wrap*.
  - ▣ Exemplo:

```
main {  
    display: flex;  
    flex-flow: row wrap;  
}
```

Os elementos “*flex*”  
serão exibidos em  
coluna e quebrados...

# CSS – FlexBox

46

- Propriedade ***justify-content***
  - ▣ Usada para alinhar os elementos flexíveis horizontalmente.
  - ▣ Valores possíveis: ***justify-content: [center, flex-start, flex-end, space-between (justificado), space-around ]***;
  - ▣ Exemplo:

```
main {  
    display: flex;  
    justify-content: space-around;  
}
```

Exibe os elementos “*flex*”  
com espaços iguais antes,  
entre e depois.

# CSS – FlexBox

47

- Propriedade ***align-items***
  - ▣ Usada para alinhar os elementos flexíveis verticalmente.
  - ▣ Valores possíveis: ***align-items: [center, flex-start, flex-end, stretch (padrão), baseline ];***
  - ▣ Exemplo:

```
main {  
  display: flex;  
  justify-content: center;  
}
```

Exibe os elementos “flex”  
com alinhamento vertical  
centralizado.

48

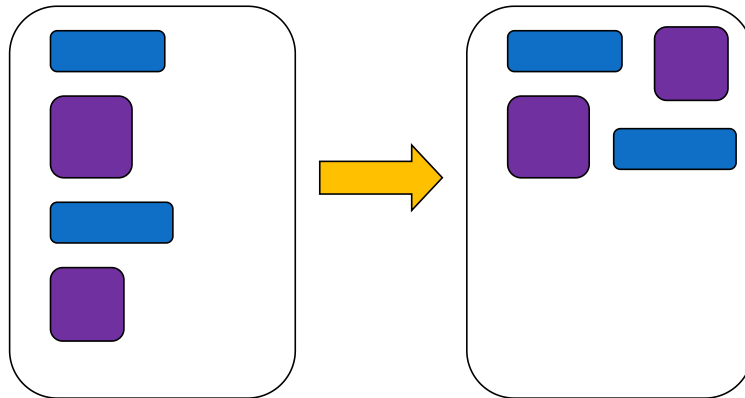
## CSS - Posicionamento



# CSS – Posicionamento

49

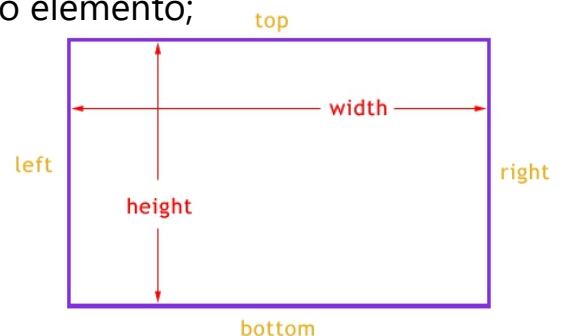
- Com CSS é possível alterar o **fluxo normal** de uma página e definir o posicionamento de cada elemento.



# CSS – Posicionamento

50

- Algumas propriedades importantes:
  - ▣ **top**: deslocamento do lado superior do elemento;
  - ▣ **left**: deslocamento do lado esquerdo do elemento;
  - ▣ **bottom**: deslocamento do lado inferior do elemento;
  - ▣ **right**: deslocamento do lado direito do elemento;
  - ▣ **width**: largura do elemento;
  - ▣ **height**: altura do elemento.



## CSS – Propriedade *Position*

- A propriedade ***position*** é utilizada para definir a posição (localização) de um elemento HTML na página HTML.
- Todos os elementos HTML têm a propriedade ***position*** que estabelece a forma de "cálculo" da posição de um elemento.
- Valores mais comuns para a propriedade ***position***:
  - ▣ ***static***
  - ▣ ***absolute***
  - ▣ ***relative***
  - ▣ ***fixed***

# CSS – Propriedade *Position*

53

## □ Posicionamento Estático (*static*):

- O elemento é posicionado segundo o fluxo normal;
- Este é o valor padrão do posicionamento.

# CSS – Propriedade *Position*

54

## □ Arquivo "*posicionamento.html*"

```
<head>
  <title>Exemplo Position</title>
  <link rel="stylesheet"
    href="style-posicionamento.css">
</head>
<body>
  <div id="azul"></div>
  <div id="vermelha"></div>
  <div id="amarela"></div>
</body>
```

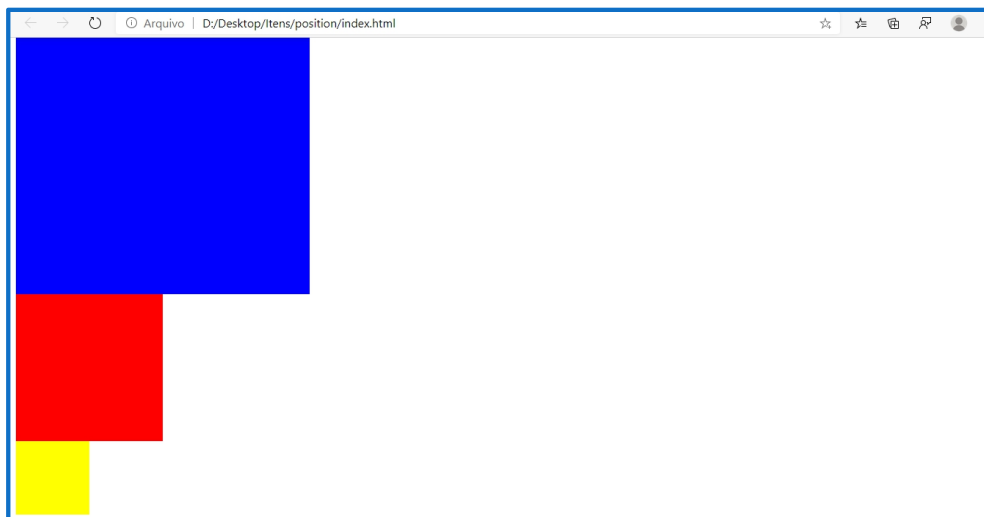
## □ Arquivo "*style-posicionamento.css*"

```
#azul {
  width: 300px; height: 300px;
  background-color: blue; }
#vermelha {
  width: 150px; height: 150px;
  background-color: red; }
#amarela {
  width: 80px; height: 80px;
  background-color: yellow; }
```

# CSS – Propriedade *Position*

55

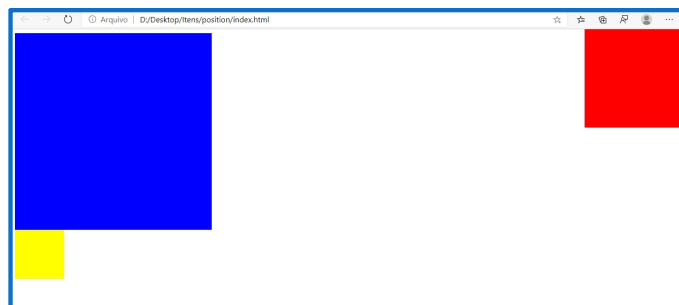
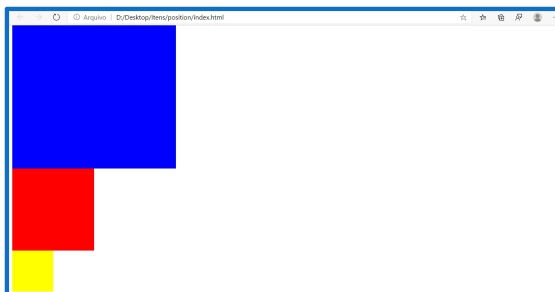
- Interface da página até o momento...



# CSS – Propriedade *Position*

56

- Como fazer para a DIV vermelha ficar no canto superior direito da página?



# CSS – Propriedade *Position*

57

□ Arquivo "posicionamento.html"

```
<head>
  <title>Exemplo Position</title>
  <link rel="stylesheet"
        href="style-posicionamento.css">
</head>
<body>
  <div id="azul"></div>
  <div id="vermelha"></div>
  <div id="amarela"></div>
</body>
```

□ Arquivo "style-posicionamento.css"

```
...
#vermelha {
  width: 200px; height: 200px;
  background-color: red;
  top: 0;
  right: 0;
}
...
```

Vamos posicionar a DIV no topo direito da página...

Não funcionou, pois por padrão a posição de uma DIV é estático...

# CSS – Propriedade *Position*

58

□ Arquivo "posicionamento.html"

```
<head>
  <title>Exemplo Position</title>
  <link rel="stylesheet"
        href="style-posicionamento.css">
</head>
<body>
  <div id="azul"></div>
  <div id="vermelha"></div>
  <div id="amarela"></div>
</body>
```

□ Arquivo "style-posicionamento.css"

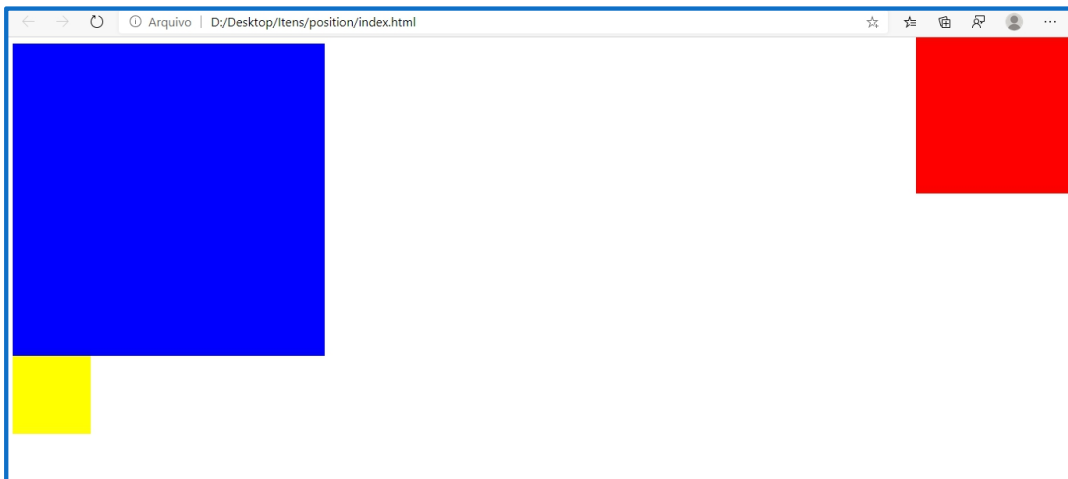
```
...
#vermelha {
  width: 200px; height: 200px;
  background-color: red;
  top: 0;
  right: 0;
  position: absolute;
}
...
```

Vamos alterar o valor da posição da DIV para "absolute"...

## CSS – Propriedade *Position*

59

- Interface da página após alterar o valor da posição da DIV vermelha para "*absolute*"...



## CSS – Propriedade *Position*

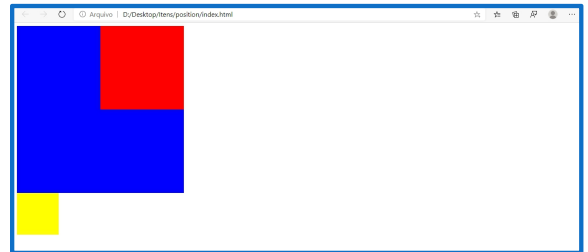
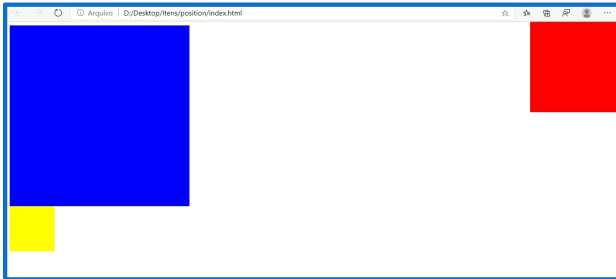
60

- **Posicionamento Absoluto (*absolute*):**
  - ▣ O elemento é posicionado relativamente em relação ao primeiro elemento pai.

# CSS – Propriedade *Position*

61

- Como fazer para a DIV vermelha ficar no canto superior direito da DIV azul?



# CSS – Propriedade *Position*

62

- Arquivo "*posicionamento.html*"

```
...  
<body>  
  <div id="azul">  
    <div id="vermelha"></div>  
  </div>  
  <div id="amarela"></div>  
</body>
```

Vamos colocar a DIV vermelha dentro da DIV azul...

Não funcionou, pois por padrão a posição de uma DIV é estática...

# CSS – Propriedade *Position*

63

□ Arquivo "posicionamento.html"

```
...  
<body>  
  <div id="azul">  
    <div id="vermelha"></div>  
  </div>  
  <div id="amarela"></div>  
</body>
```

□ Arquivo "style-posicionamento.css"

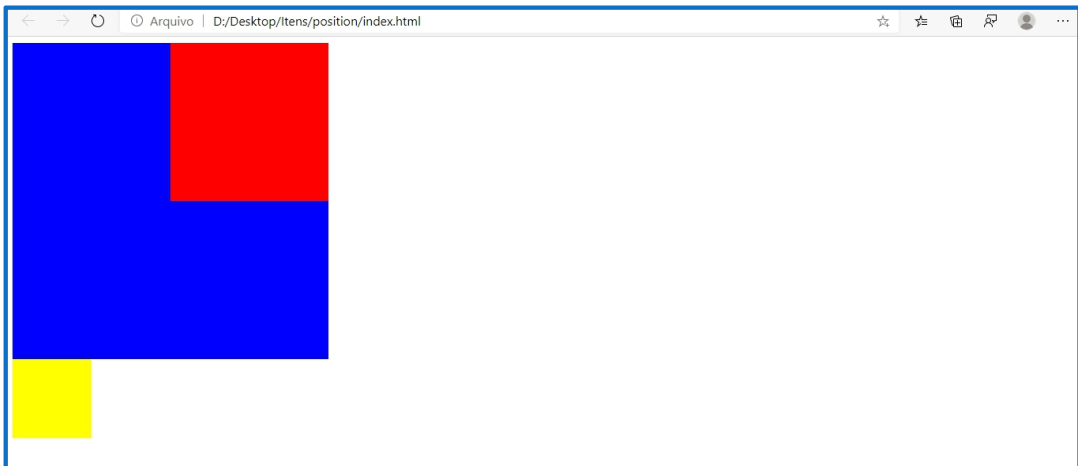
```
...  
#azul {  
  width: 400px; height: 400px;  
  background-color: blue;  
  position: relative;  
}
```

Além de colocar a DIV vermelha dentro da DIV azul, vamos definir a posição da DIV azul (DIV pai) como "relative"...

# CSS – Propriedade *Position*

64

□ Interface da página após colocar a DIV vermelha dentro da DIV azul e definir a posição da DIV azul (DIV pai) como "relative"...





## CSS – Propriedade *Position*

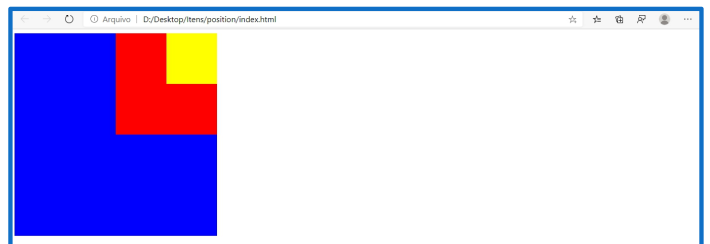
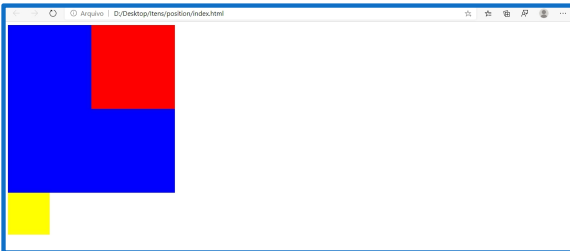
65

- **Posicionamento Relativo (*relative*):**
  - ▣ Estabelece uma relação do elemento que possui a propriedade "***position = relative***" com seus elementos filhos.

## CSS – Propriedade *Position*

66

- Como fazer para a DIV amarela ficar no canto superior direito da DIV vermelha?



# CSS – Propriedade *Position*

67

□ Arquivo "posicionamento.html"

```
...  
<body>  
  <div id="azul">  
    <div id="vermelha">  
      <div id="amarela">  
      </div>  
    </div>  
  </div>  
</body>
```

Vamos colocar a DIV amarela dentro da DIV vermelha...

Além de colocar a DIV amarela dentro da DIV vermelha, vamos posicionar a DIV amarela no topo direito...

□ Arquivo "style-posicionamento.css"

```
#amarela {  
  width: 100px; height: 100px;  
  background-color: yellow;  
  top: 0;  
  right: 0;  
}
```

Não funcionou, pois por padrão a posição da DIV amarela é estática...

# CSS – Propriedade *Position*

68

□ Arquivo "posicionamento.html"

```
...  
<body>  
  <div id="azul">  
    <div id="vermelha">  
      <div id="amarela">  
      </div>  
    </div>  
  </div>  
</body>
```

□ Arquivo "style-posicionamento.css"

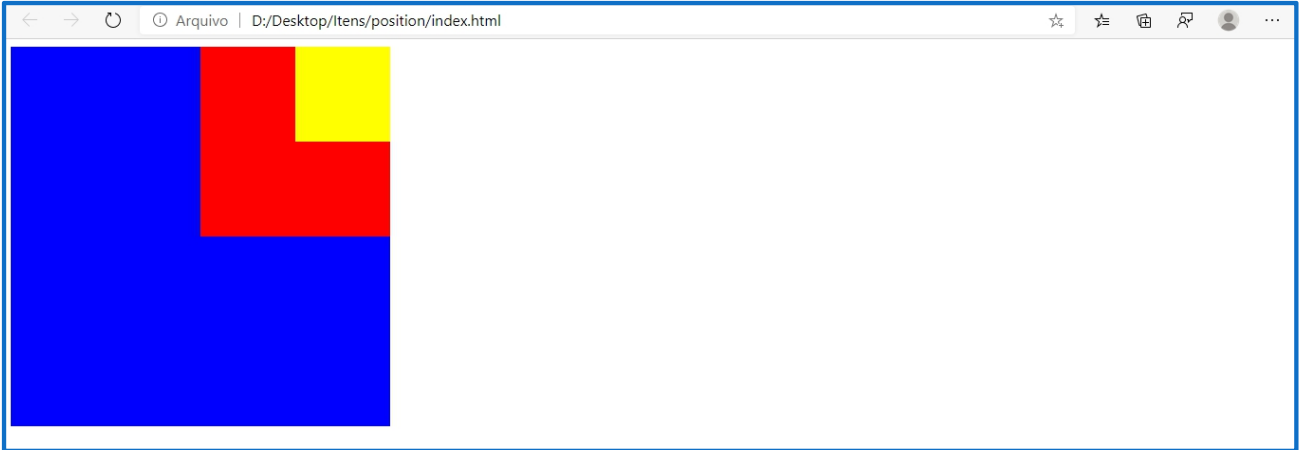
```
#amarela {  
  width: 100px; height: 100px;  
  background-color: yellow;  
  top: 0;  
  right: 0;  
  position: absolute;  
}
```

Vamos alterar o valor da posição da DIV amarela para "absolute"...

## CSS – Propriedade *Position*

69

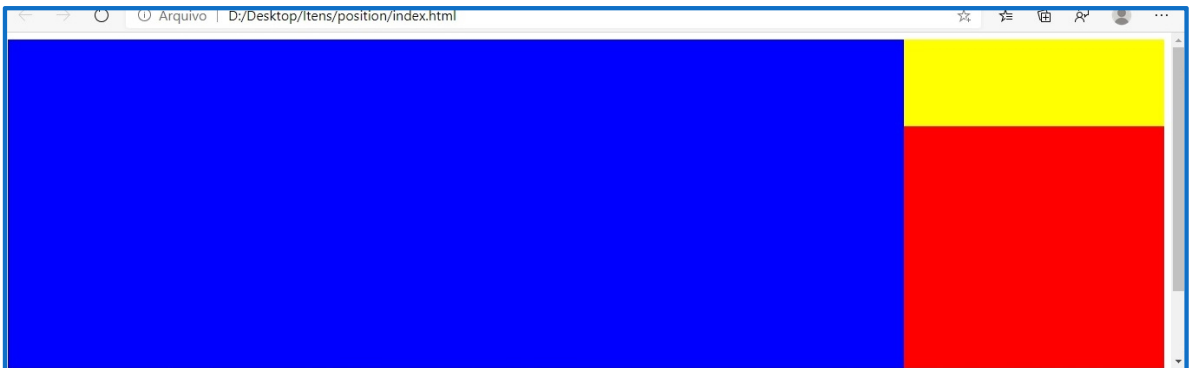
- Interface da página após colocar a DIV amarela dentro da DIV vermelha e definir a posição da DIV amarela como "*absolute*"...



## CSS – Propriedade *Position*

70

- Como transformar a DIV azul no fundo do nosso site?
- Como transformar a DIV vermelha em um SIDEBAR (lateral) do site?
- Como transformar a DIV amarela em um módulo (*container*) fixo dentro do SIDEBAR?



# CSS – Propriedade *Position*

71

- Arquivo "style-posicionamento.css"

```
#azul {  
    width: 100%; height: 1400px;  
    background-color: blue;  
    position: relative; }  
#vermelha {  
    width: 300px; height: 1400px;  
    background-color: red;  
    top: 0;  
    right: 0;  
    position: absolute; }
```

```
#amarela {  
    width: 300px; height: 100px;  
    background-color: yellow;  
    top: 0;  
    right: 0;  
    position: absolute;  
}
```

DIV amarela não está  
fixa na página...

# CSS – Propriedade *Position*

72

- Arquivo "style-posicionamento.css"

```
#azul {  
    width: 100%; height: 1400px;  
    background-color: blue;  
    position: relative; }  
#vermelha {  
    width: 300px; height: 1400px;  
    background-color: red;  
    top: 0;  
    right: 0;  
    position: absolute; }
```

```
#amarela {  
    width: 300px; height: 100px;  
    background-color: yellow;  
    top: 0;  
    right: 0;  
    position: fixed;  
}
```

Vamos alterar o valor da posição  
da DIV amarela para "fixed"...

# CSS – Propriedade *Position*

73

- Interface da página após definir a posição da DIV amarela como "*fixed*"...



# CSS – Propriedade *Position*

74

- Arquivo "*style-posicionamento.css*"

Vamos colocar a margem do corpo da página com o valor zero...

```
body {  
    margin: 0;  
}
```



## CSS – Profundidade

- A propriedade ***z-index*** define a profundidade do elemento HTML.
- Quanto maior o valor desta propriedade, mais para frente fica o elemento.
- Quando o valor desta propriedade dos elementos é "***auto***", o elemento filho se sobrepõe ao elemento pai.

## CSS – Posicionamento

77

- A propriedade **overflow** define como o navegador deve proceder quando as dimensões de um elemento são maiores do que o espaço reservado para ele na página. Variações:
  - ▣ **overflow-x**: Semelhante à overflow mas, atua somente quando a área horizontal reservada é menor do que o conteúdo. Os valores são os mesmos.
  - ▣ **overflow-y**: Semelhante à overflow, mas, atua somente quando a área vertical reservada é menor do que o conteúdo. Os valores são os mesmos.

## CSS – Posicionamento

78

- Os valores da propriedade **overflow** podem ser:
  - ▣ **visible**: Todo o conteúdo é exibido;
  - ▣ **hidden**: O conteúdo não é truncado mas o que excede a área reservada ao elemento não é exibido;
  - ▣ **scroll**: O conteúdo é exibido na área a ele reservado e barras de rolagem são exibidas;
  - ▣ **auto**: O navegador decide que método utilizar.

### Centralizando elementos HTML via CSS

- Uma tarefa comum em CSS é a centralização de textos e imagens.
- Basicamente, existem três tipos de tarefas relacionadas à centralização:
  - ▣ Centralizar na horizontal linhas de textos
  - ▣ Centralizar na horizontal um bloco
  - ▣ Centralizar na horizontal uma imagem
  - ▣ Centralizar na vertical um bloco ou uma imagem



# Centralizar linhas de textos

81

- Defina o valor '**center**' a propriedade CSS '**text-align**', conforme exemplo:

```
h1 { text-align: center; }  
p { text-align: center; }
```

- Esta propriedade funcionará para elementos *inline* (textos, títulos e links). Exemplo:

# Centralizar um bloco

82

- Defina o valor '**auto**' para a propriedade '**margin**' da esquerda e da direita;
- Defina uma largura fixa para o bloco.

```
footer {  
    width: 50%;  
    margin-left: auto;  
    margin-right: auto; }
```

- Esta propriedade funcionará para elementos *block* (div, header, main, footer, section, etc.)

# Centralizar uma imagem

83

- Defina o valor '**auto**' para a propriedade '**margin**' da esquerda e da direita;
- Defina o valor '**block**' para a propriedade '**display**'.

```
img {  
    display: block;  
    margin-left: auto;  
    margin-right: auto;  
}
```

# Centralizar mais de um bloco

84

- Se você tem dois ou mais blocos que precisam ser centralizados horizontalmente em uma linha, o melhor a fazer é usar um **display inline-block** ou **flex**.

```
.em-linha {  
    display: inline-block;  
}
```

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados. Eu tenho mais conteúdo do que o meus irmãos.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

```
.div-pai {  
    display: flex;  
    justify-content: center;  
}
```

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados. Eu tenho mais conteúdo do que o meus irmãos.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

# Centralizar mais de um bloco

85

- Se você tem dois ou mais blocos que precisam ser centralizados horizontalmente, o melhor a fazer é usar a classe **display: flex**.

Coloque esta classe nos blocos que devem ficar alinhados em linha.

Coloque esta classe no bloco "pai" dos blocos que devem ficar alinhados em linha.

```
.em-linha {  
  display: inline-block;  
}
```

```
.div-pai {  
  display: flex;  
  justify-content: center;  
}
```

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados. Eu tenho mais conteúdo do que o meus irmãos.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados. Eu tenho mais conteúdo do que o meus irmãos.

Eu sou um elemento block assim com meus irmãos e nós estamos centralizados.

# Centralizar Elementos na Horizontal

86

- Exemplo



## Centralizando elementos HTML via CSS

### – Exemplo - Arquivo “*centralizar.html*”

87

```
<head>
  <title>Exemplo Centralizar Elementos</title>
  <link rel="stylesheet" href="style-centralizar.css">
</head>
<body>
  <header><h1>Título</h1><p>Parágrafo centralizado</p></header>
  
  <main> <div class="div-pai">
    <div class="em-linha">Texto 1 alinhado.</div>
    <div class="em-linha">Texto 2 alinhado.</div>
    <div class="em-linha">Texto 3 alinhado.</div>
  </div> </main>
</body>
```

88

## Centralizar na vertical

## Centralizar na vertical um bloco

89

- Se você tem dois ou mais blocos que precisam ser centralizados verticalmente em uma linha, segue uma solução.

```
.div-pai {  
    display: flex;  
    align-items: center;  
}
```

Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados.

Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados. Eu tenho  
mais conteúdo do que o  
meus irmãos.

Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados.

## Centralizar na vertical e na horizontal um bloco

90

- No bloco "pai" defina as propriedades e respectivos valores seguintes:

```
.div-pai {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados.

Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados. Eu tenho  
mais conteúdo do que o  
meus irmãos.

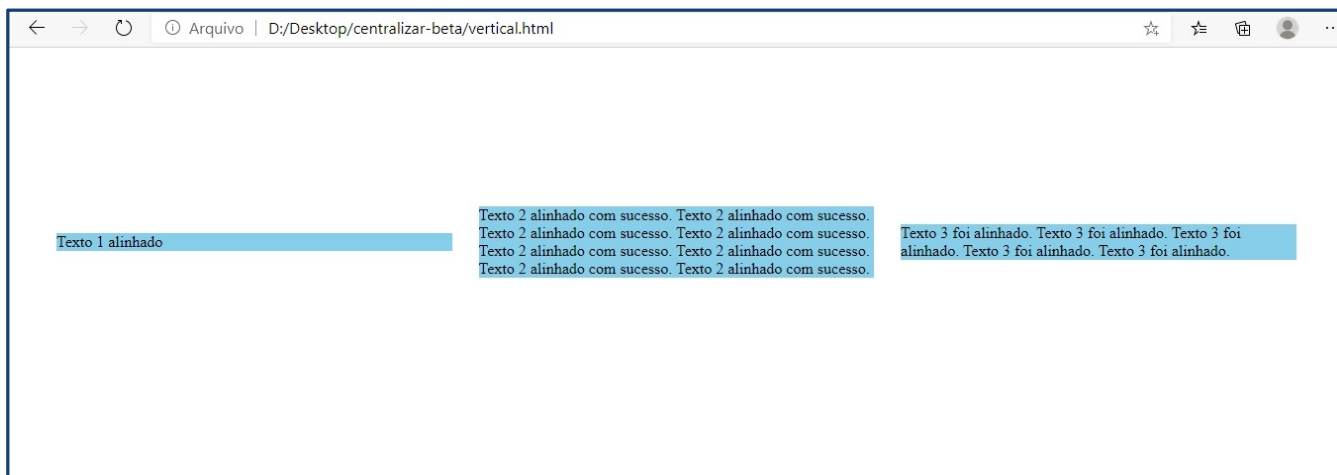
Eu sou um elemento  
block assim com meus  
irmãos e nós estamos  
centralizados.

- A propriedade '**justify-content**' controla o **alinhamento horizontal** e a propriedade '**align-items**' controla o **alinhamento vertical** dos conteúdos de um bloco '**flex**'.

# Centralizar Elementos na Vertical

91

## □ Exemplo



## Dúvidas?

92



**Prof. Me. Fernando Roberto Proença**

***[fernandorroberto@gmail.com](mailto:fernandorroberto@gmail.com)***

# Referências

93

- [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)
- <https://origamid.com/projetos/flexbox-guia-completo/>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Conceitos\\_Basicos\\_do\\_Flexbox](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceitos_Basicos_do_Flexbox)
- [https://www.w3.org/Style/Examples/007/center.pt\\_BR.html](https://www.w3.org/Style/Examples/007/center.pt_BR.html)
- <http://www.maujor.com/blog/2006/07/17/propriedade-vertical-align/>
- <https://www.webcis.com.br/centralizar-divs-na-horizontal-e-vertical-com-css.html>
- <https://tableless.com.br/centralizando-conteudo-na-vertical-e-horizontal-com-css-flexbox/>
- [https://www.youtube.com/playlist?list=PLirko8T4cEmx5eBb1-9j6T6Gl4aBtZ\\_5x](https://www.youtube.com/playlist?list=PLirko8T4cEmx5eBb1-9j6T6Gl4aBtZ_5x)
- <https://www.youtube.com/watch?v=wGbru80sdOo>
- [https://www.youtube.com/watch?v=x-4z\\_u8LcGc](https://www.youtube.com/watch?v=x-4z_u8LcGc)