



CSS, CLASSES, DIVS E SPANS.

Prof. Me. Hélio Esperidião

CORES EM CSS

- As cores em css podem ser representados por nomes.
 - Blue, gray, red, etc.
- As cores em css podem ser representadas por números Hexadecimais
- Estes números Hexadecimais são arranjados no padrão RGB



RGB

- RGB é a abreviatura de RED, GREEN e BLUE.
 - Vermelho, verde e azul.
- A mistura dessas cores pode gerar qualquer cor.
- Dois algarismos Hexadecimais são utilizados para representar cada cor
 - 2 para Vermelho
 - 2 para Verde
 - 2 para Azul



NÚMEROS HEXADECIMAIS

Decimal	Hexadecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F



DEFININDO UMA COR EM CSS

```
<style type="text/css">  
p {  
  background-color: 0000FF;  
}  
</style>
```

Quantidade de
Vermelho

Quantidade de
verde

Quantidade
máxima de
Azul



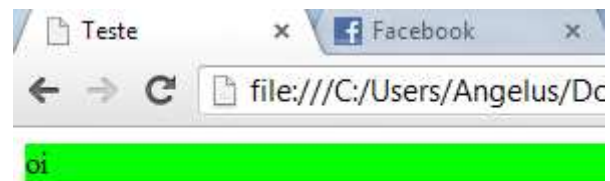
DEFININDO UMA COR EM CSS

```
<style type="text/css">  
p {  
    background-color: 00FF00;  
}  
</style>
```

Quantidade de
Vermelho

Quantidade de
verde

Quantidade
de Azul



DEFININDO UMA COR EM CSS

```
<style type="text/css">
p {
  background-color: FF0000;
}
</style>
```

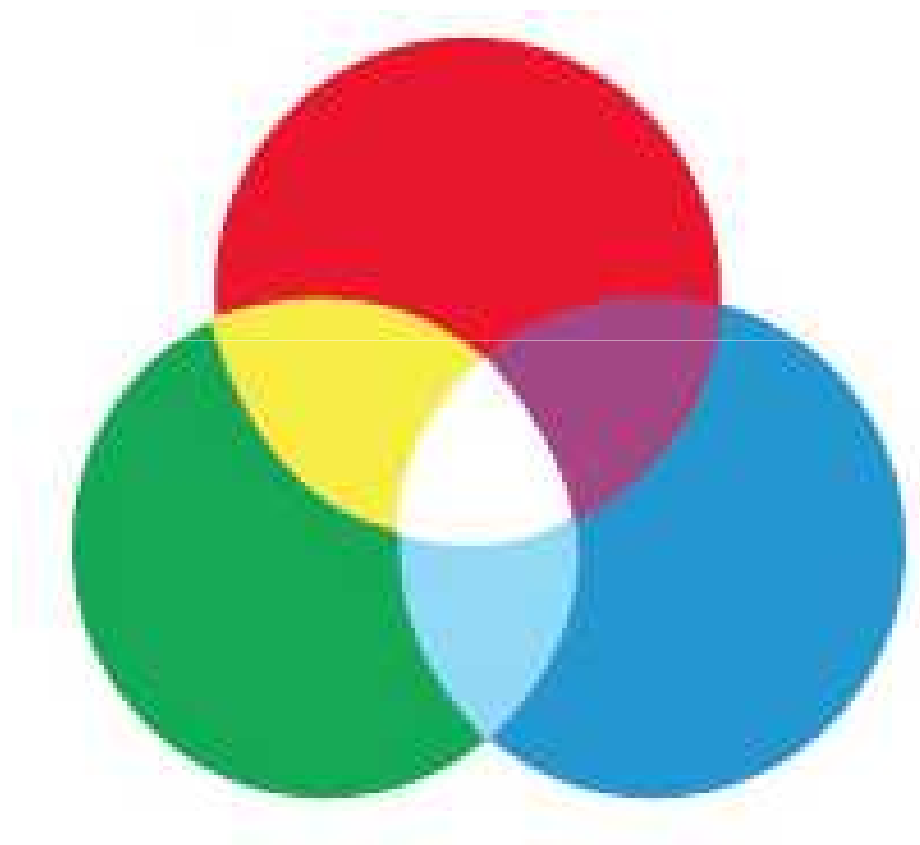
Quantidade de
Vermelho

Quantidade de
verde

Quantidade
de Azul



MISTURANDO AS CORES



DEFININDO UMA COR EM CSS

```
<style type="text/css">  
p {  
  background-color: FFFF00;  
}  
</style>
```

Quantidade de
Vermelho

Quantidade de
verde

Quantidade
de Azul



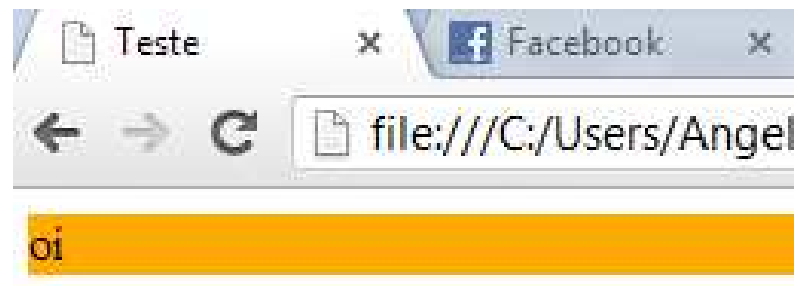
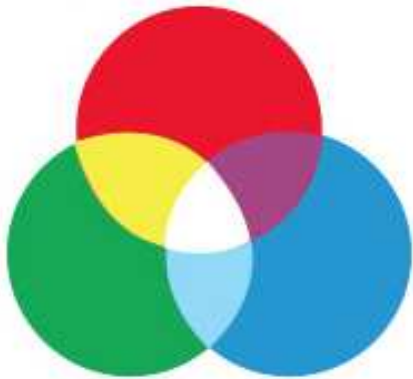
DEFININDO UMA COR EM CSS

```
<style type="text/css">  
p {  
  background-color: FF00FF;  
}  
</style>
```

Quantidade de
Vermelho

Quantidade de
verde

Quantidade
de Azul



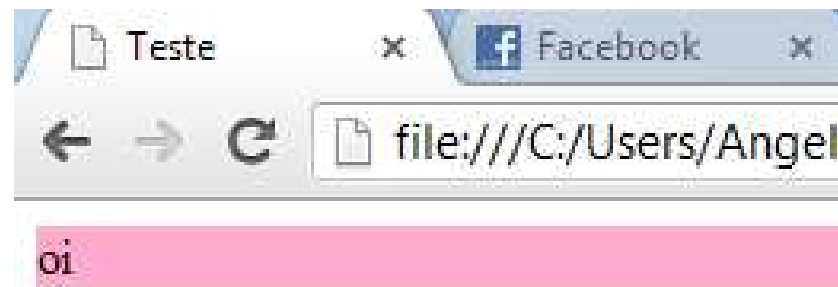
DEFININDO UMA COR EM CSS

```
<style type="text/css">
p {
  background-color: FFAACC;
}
</style>
```

Quantidade de
Vermelho

Quantidade de
verde

Quantidade
de Azul



REPRESENTAÇÃO DOBRADA

- Na notação abreviada cada um dos três dígitos é automaticamente dobrado conforme exemplos a seguir:

```
#FFF = #FFFFFF
```

```
#CF9 = #CCFF99
```

```
#cde = #ccddee
```

```
#49c = #4499cc
```



A PROPRIEDADE BORDER

- Define borda de componentes
- **border-width** - espessura da borda
 - **thin**: borda fina
 - **medium**: borda média
 - **thick**: borda grossa
 - **length**: uma medida reconhecida pelas CSS (px)



A PROPRIEDADE BORDER

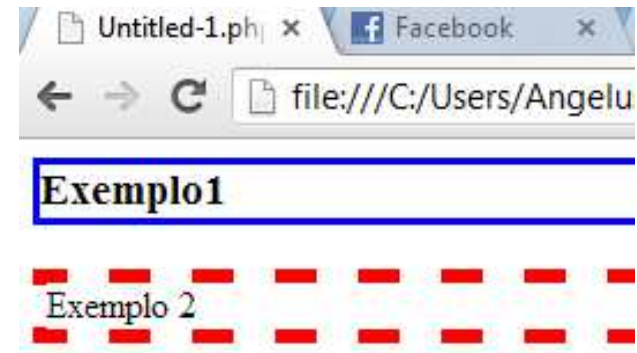
- **border-style**: estilo da borda
 - **none**: nenhuma borda
 - **hidden**: equivalente a none
 - **dotted**: borda pontilhada
 - **dashed**: borda tracejada
 - **solid**: borda contínua
 - **double**: borda dupla
 - **groove**: borda entalhada
 - **ridge**: borda em relevo
 - **inset**: borda em baixo relevo
 - **outset**: borda em alto relevo

- **border-color**
 - cor da borda em hexadecimal



EXEMPLO: BORDER

```
<html>
<head>
<style type="text/css">
  h3{
    border-width: medium;
    border-style: solid;
    border-color: #00f;
  }
  p {
    border-width: 6px;
    border-style: dashed;
    border-color: #f00;
  }
</style>
</head>
<body>
<h3>Exemplo1</h3>
<p>Exemplo 2</p>
</body>
</html>
```



A PROPRIEDADE PADDING

- A propriedade para espaçamentos (alguns traduzem como "enchimento"), define um valor para os espaçamentos entre o conteúdo e as bordas dos elementos HTML.
- **Padding** é responsável por definir o espaço entre o conteúdo do elemento HTML e sua borda,
- **padding-top**
 - define a espaçamento superior;
- **padding-right**
 - define a espaçamento direita;
- **padding-bottom**
 - define a espaçamento inferior;
- **padding-left**
 - define a espaçamento esquerda;
- **Padding**
 - maneira abreviada para todas os espaçamentos



EXEMPLO: PADDING

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

input {
  padding-left: 50px;
  color:F00;

}

</style>
</head>
<body>
<form action="teste.php" method="get">
<input name="" type="text">
</form>

</body>
</html>
```

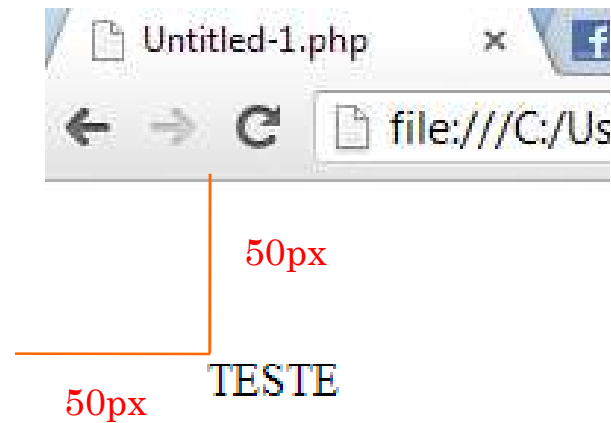


EXEMPLO PADDING

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

p {
    padding-left: 50px;
    padding-top: 50px;
}

</style>
</head>
<body>
<p>TESTE</p>
</body>
</html>
```



A PROPRIEDADE MARGIN

- **Margin** é responsável por definir o espaço entre os elementos HTML. Para que fique claro as diferenças, dê uma olhada na imagem abaixo, ela representa
- **margin-top**
 - define a margem superior;
- **margin-right**
 - define a margem direita;
- **margin-bottom**
 - define a margem inferior;
- **margin-left**
 - define a margem esquerda;
- **Margin**
 - maneira abreviada para todas as margens
-

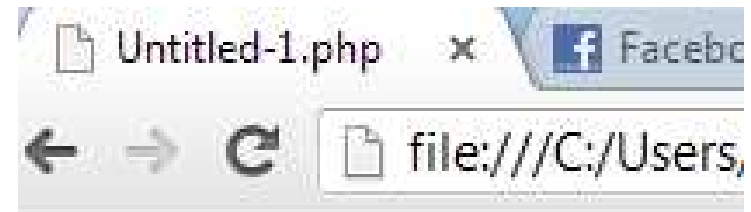


EXEMPLO MARGIN

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

p {margin-top: 65px;}

</style>
</head>
<body>
<p>Uma margem superior de 2 cm</p>
</body>
</html>
```



Uma margem superior de 2 cm



EXEMPLO MARGIN

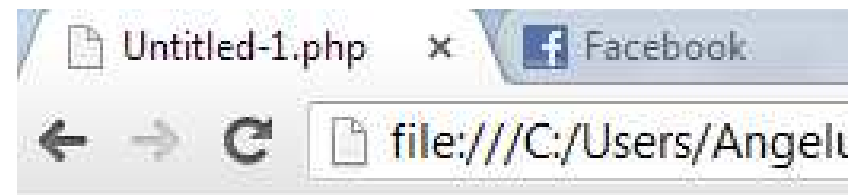
```
<html>
<head>
<style type="text/css">

input {
    margin-left: 50px;
    color:F00;

}

</style>
</head>
<body>
<form action="teste.php" method="get">
<input name="" type="text">
</form>

</body>
</html>
```



teste margin

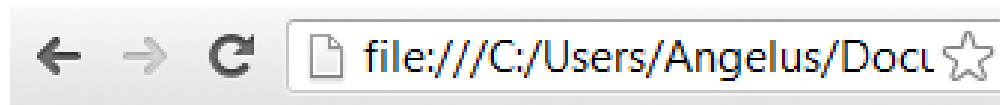


EXEMPLO MARGIN

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

p {margin: 20px 40px 80px 5px;}

</style>
</head>
<body>
<p>Uma margem superior de 20px, uma margem direita de 40px,
uma margem inferior de 80px e uma margem esquerda de 5px</p>
</body>
</html>
```



Uma margem superior de 20px, uma margem direita de 40px, uma margem inferior de 80px e uma margem esquerda de 5px



CLASSES CSS

- Consiste em definir um conjunto de estilos que poderão ser utilizados por objetos distintos.

```
<html>
<head>
<title>Teste</title>

<style type="text/css">
.classeEstico {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
    font-style: normal;
    font-weight: bold;
    color: red;
    background-color: blue;
}
</style>
</head>
<body>
<p class="classeEstico">oi</p>
<b class="classeEstico"> oi</b>

</body>
</html>
```



TAG DIV

- A tag `<div>` define um elemento do tipo bloco.
- É um mecanismo genérico para agrupar componentes em um documento html

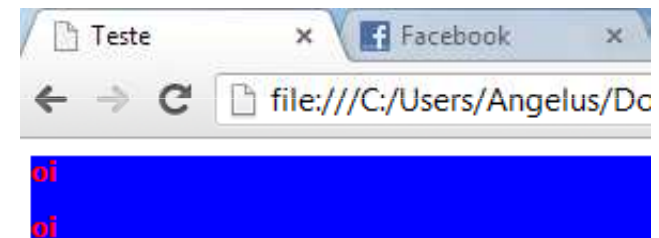


EXEMPLO DE DIV

```
<html>
<head>
<title>Teste</title>

<style type="text/css">
.classeEstico {
  font-family: Verdana, Geneva, sans-serif;
  font-size: 12px;
  font-style: normal;
  font-weight: bold;
  color: red;
  background-color: blue;
}
</style>
</head>
  <body>
    <div class="classeEstico">
      <p>oi</p>
      <b >oi</b>
    </div>

  </body>
</html>
```



IDENTIFICANDO COMPONENTES DE FORMA ÚNICA

```
<html>
<head>
<title>Teste</title>
<style type="text/css">
#MinhaDiv {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
    font-style: normal;
    font-weight: bold;
    color: red;
    background-color: blue;
}
#MinhaDiv2 {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
    font-style: normal;
    color: white;
    background-color: red;
}
</style>
</head>
<body>
    <div id="MinhaDiv">
        oi
    </div>
    <div id="MinhaDiv2">
        oi 2
    </div>

</body>
</html>
```



IDENTIFICANDO COMPONENTES DE FORMA ÚNICA

```
<html>
<head>
<title>Teste</title>
<style type="text/css">
#MinhaDiv {
  font-family: Verdana, Geneva, sans-serif;
  font-size: 12px;
  font-style: normal;
  font-weight: bold;
  color: red;
  background-color: blue;
}
#MinhaDiv2 {
  font-family: Verdana, Geneva, sans-serif;
  font-size: 12px;
  font-style: normal;
  color: white;
  background-color: red;
}
</style>
</head>
<body>
  <h1 id="MinhaDiv">
    oi
  </h1>
  <h1 id="MinhaDiv2">
    oi 2
  </h1>

</body>
</html>
```



SPANS

- Spans também são utilizados para agrupamento de componentes.
- Não funcionam como blocos como os divs
- Funciona para agrupar informações em uma mesma linha ou região



SPANS EXEMPLO

```
<html>
<head>
<title>Teste</title>
<style type="text/css">
#MeuSpan1 {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
    font-style: normal;
    font-weight: bold;
    color: red;
    background-color: blue;
}
#MeuSpan2 {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
    font-style: normal;
    color: white;
    background-color: red;
}
</style>
</head>
<body>
    <span id="MeuSpan1">oi</span>

    </h1>
    <span id="MeuSpan2">oi</span>

    </h1>

    </body>
</html>
```

