

CSS – Bootstrap – Materialize

Treinamento: Angular, Node e Java – Capgemini

Instrutor: Ivan J. Borchardt

©2021

ProWay

CSS Cascading Style Sheets

- ❖ Documentos CSS são empregados para estilizar o conteúdo de um documento HTML, ou seja, controlar a apresentação de documentos de marcação.
- ❖ O HTML controla a estrutura de uma página, ao passo que o CSS determina sua aparência.
- ❖ Uma folha de estilos é uma coleção de regras de formatação aplicadas a um documento HTML.

CSS – Como e onde adicionar

- ◆ CSS Inline – estilos aplicados diretamente em um elemento HTML, dentro do documento.

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

CSS – Como e onde adicionar

- ◈ CSS Interno/Incorporado – O elemento `style` permite aplicar várias regras de uma vez no próprio documento HTML

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

CSS – Como e onde adicionar

- ◊ CSS Externo: um documento com extensão .css permite aplicar regras a um website inteiro de uma vez.

mystyle.css

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet" href="mystyle.css">  
  </head>  
  <body>  
  
    <h1>This is a heading</h1>  
    <p>This is a paragraph.</p>  
  
  </body>  
</html>
```

CSS – Ordem em Cascata

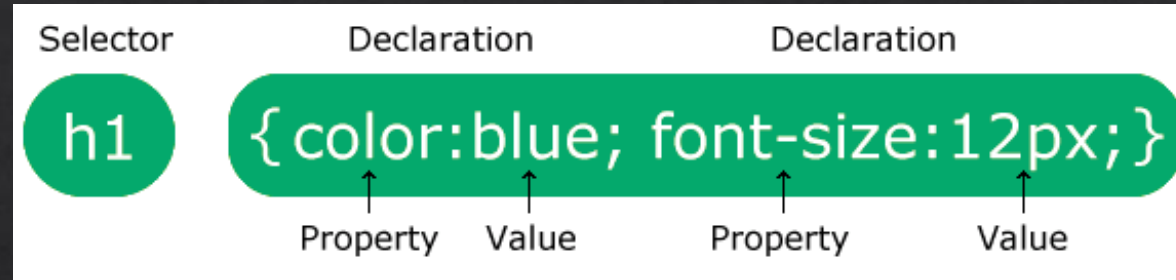
Qual estilo será usado quando houver mais de um estilo especificado para um elemento HTML?

Todos os estilos em uma página serão "cascadeados" em uma nova folha de estilo "virtual" pelas seguintes regras, onde o número um tem a prioridade mais alta:

- ◇ Estilo embutido (dentro de um elemento HTML)
- ◇ Folhas de estilo externas e internas (na seção principal)
- ◇ Padrão do navegador

Portanto, um estilo embutido tem a prioridade mais alta e substituirá os estilos externos e internos e os padrões do navegador.

CSS – Sintaxe



Exemplo:

```
p {  
  color: red;  
  text-align: center;  
}
```

CSS – Seletores

Os seletores CSS são usados para "encontrar" (ou selecionar) os elementos HTML que você deseja estilizar.

Podemos dividir os seletores CSS em cinco categorias:

- ◇ Seletores simples (selecione os elementos com base no nome, id, classe)
- ◇ Seletores combinadores (selecione os elementos com base em uma relação específica entre eles)
- ◇ Seletores de pseudoclasse (selecione os elementos com base em um determinado estado)
- ◇ Seletores de pseudoelementos (selecione e estilize uma parte de um elemento)
- ◇ Seletores de atributos (selecione os elementos com base em um atributo ou valor de atributo)

***Sempre declare seus seletores do mais abrangente ao mais específico de cima para baixo!!!.

CSS – Seletores Simples

```
/*----Seletor universal      */
*{
    margin: 0;
    padding: 0;
    border: 0;
}

/*----Seletor de elementos CSS*/
p{
    text-align: center;
    color: blue;
}
```

```
/*----Seletor de Classe      */
.titulos {
    text-align: center;
}

h1.destaque{
    background-color: aquamarine;
}

/*----Seletor de ID          */
#cabecalho {
    text-align: center;
}
```

CSS – Seletores combinadores

/*

Elementos de combinação de seletores são usados para separar dois ou mais seletores simples que compõem um seletor combinado. Os elementos de combinação disponíveis são: espaço em branco (qualquer quantidade de espaço, tabulação ou caracteres de espaçamento), o sinal de maior “>” e o sinal de adição “+” . A função de cada um destes elementos de combinação dos seletores será descrita adiante.

*/

CSS – Seletores combinadores

`/* Seletores descendentes`

Um seletor descendente é uma combinação de dois ou mais seletores simples separados por um espaço em branco. Casa com elementos que sejam descendentes do primeiro elemento simples declarado no seletor. Por exemplo, na regra a seguir o seletor `div p` casa com todos os elementos `p` que sejam descendentes do elemento `div`.

`*/`

```
div p {  
  color:#f00;  
}
```

CSS – Seletores combinadores

```
/*
```

Cada um dos seletores que compõem um seletor descendente pode ser um seletor simples de qualquer natureza. Na regra a seguir o seletor casa com todo o elemento p da classe info contido em um elemento li que esteja contido em um elemento div cuja id seja myid.

```
*/
```

```
div#myid li p.info {  
    color:#f00;  
}
```

CSS – Seletores combinadores

`/* Seletores filho`

Um seletor filho tem como alvo um filho imediato de um elemento.

O seletor filho consiste de um ou mais seletores simples separados por um sinal de maior “>”. O elemento pai fica à esquerda do sinal “>”, e é permitido deixar espaço em branco entre o elemento de combinação e os seletores. A regra a seguir aplica-se a todos os elementos strong que sejam filhos de um elemento div.

`*/`

```
div > strong {  
    color:#f00;  
}
```


CSS – Seletores combinadores

`/*Seletores irmãos adjacentes (sibling selectors)`

Um seletor irmão adjacente consiste de um ou mais seletores simples separados por um sinal de adição, “+”. é permitido deixar espaço em branco entre o elemento de combinação e os seletores. O seletor tem como alvo um elemento que seja irmão e adjacente ao primeiro elemento. Os elementos devem ter o mesmo pai e o primeiro elemento deve ser imediatamente anterior ao segundo.

`*/`

```
p + p {  
    color:#f00;  
}
```

```
p ~ p {  
    color:#f00;  
}
```

CSS – Seletores combinadores

`/*Grupando`

Para aplicar uma mesma regra a diferentes elementos alvo casados por diferentes seletores você pode agrupar os seletores em uma lista e separando-os por uma vírgula no lugar de escrever repetidamente a mesma regra para cada um dos seletores.

`*/`

```
div#news h3,div#news ul {  
    margin:0 2em;  
}
```

CSS – Seletores de pseudo-classe

```
/*
```

```
Pseudo-classes casam elementos baseadas em características outras que não;  
seu nome, seus atributos ou seu conteúdo.
```

```
*/
```

```
/*:first-child
```

```
Esta pseudo-classe casa com o elemento que é o primeiro filho de um outro  
elemento. Suponha que você quer estilizar diferenciadamente o primeiro  
parágrafo de um artigo. Se tal artigo estiver contido dentro de um elemento  
div ao qual foi atribuído a classe “article”, a regra a seguir casa com o  
primeiro elemento parágrafo p no artigo
```

```
*/
```

```
div.article p:first-child {  
    font-style:italic;  
}
```

CSS – Seletores de pseudo-classe

```
/*  
    Para casar com todos os elementos p que sejam filhos de qualquer elemento  
    você poderia usar o seletor da regra a seguir  
*/  
p:first-child {  
    font-style:italic;  
}
```

CSS – Seletores de pseudo-classe

`/*:link e :visited`

As pseudo-classes `link` afetam o estado dos links visitados e não visitados. Estes dois estados são mutuamente exclusivos – um link não pode ser visitado e não visitado ao mesmo tempo.

Estas pseudo-classes aplicam-se somente a hyperlinks e âncoras definidas na linguagem de marcação do documento. Em HTML, isto é válido para elementos com o atributo `href`

`*/`

CSS – Seletores de pseudo-classe

`/*:hover, :active, e :focus`

As pseudo-classes dinâmicas podem ser usadas para controlar a apresentação de determinados elementos na dependência de ações do usuário.

`:hover` aplica-se para quando o usuário coloca um dispositivo apontador em um elemento mas não o ativa. O uso mais comum é quando da ação de usuário de apontar o cursor do mouse sobre o elemento.

`:active` aplica-se para quando o usuário ativa um elemento. Para ação de mouse, equivale a pressionar o botão e mantê-lo pressionado até soltar.

`:focus` aplica-se para quando um elemento recebe foco, ou seja, enquanto aceita eventos de teclado.

Um elemento pode ser casado a várias pseudo-classes ao mesmo tempo. Um elemento pode receber foco e ter o cursor do mouse sobre ele ao mesmo tempo.

`*/`

CSS – Seletores de pseudo-classe

```
input[type=text]:focus {  
    color:#000;  
    background:#ffe;  
}
```

```
input[type=text]:focus:hover {  
    background:#fff;  
}
```

CSS – Seletores de pseudo-classe

```
input[type=text]:focus {  
    color:#000;  
    background:#ffe;  
}
```

```
input[type=text]:focus:hover {  
    background:#fff;  
}
```

/*

A primeira regra casa com o elemento input e tem o foco, a segunda regra casa com o mesmo elemento quando tem o ponteiro do mouse sobre ele.

*/

CSS – Seletores de pseudo-classe

```
/*:lang
```

A pseudo-class para linguagem (idioma) pode ser usada para estilizar elementos cujo conteúdo está escrito em uma determinada linguagem (idioma - uma língua para humanos e não uma linguagem de marcação). A regra a seguir define que tipo de aspas usar para textos inline que estão escritos no idioma da Suécia.

```
*/
```

```
q:lang(sv) {  
    quotes: "\201D" "\201D" "\2019" "\2019";  
}
```

```
/*
```

A linguagem para humanos (idioma) de um documento, normalmente é especificada pelo atributo lang em HTML e pelo atributo xml:lang em XHTML.

```
*/
```

CSS – Seletores de pseudo-elementos

```
/*  
    Os pseudo-elementos permitem acessar e formatar partes do documento que não estão  
    disponíveis como nós da árvore do documento.  
*/  
  
/*:first-line  
    O pseudo-elemento :first-line afeta a primeira linha de texto de um parágrafo.  
    Aplica-se somente a elementos nível de bloco, blocos inline, table-caption ou table-cell.  
    O comprimento da primeira linha depende obviamente de uma série de fatores, aí incluído  
    o tamanho da fonte e a largura do elemento container do texto.  
  
    A regra a seguir aplica-se à primeira linha do texto de um parágrafo  
*/  
p:first-line {  
    font-weight:bold;  
    color: #600;  
}
```


CSS – Seletores de pseudo-elementos

`/*:first-letter`

Este pseudo-elemento permite casar a primeira letra ou primeiro caractere de um elemento e aplica-se a elementos nível de bloco, list-item, table-cell, table-caption e bloco inline.

A regra a seguir aplica-se ao primeiro caractere de um elemento cuja classe denomina-se “preamble”.

`*/`

```
.preamble:first-letter {  
    font-size:1.5em;  
    font-weight:bold;  
}
```

CSS – Seletores de pseudo-elementos

`/*:before e :after`

Entre uma das mais discutidas funcionalidades das CSS os pseudo-elementos `:before` e `:after` podem ser usados para gerar conteúdos antes e depois do conteúdo de um elemento

`*/`

```
.cbb:before {  
  content:"";  
  display:block;  
  height:17px;  
  width:18px;  
  background:url(top.png) no-repeat 0 0;  
  margin:0 0 0 -18px;  
}
```

CSS – Seletores de pseudo-elementos

```
/*  
  Um exemplo do uso de :after para inserir a URL logo após o texto de um link.  
*/  
a:link:after {  
  content: " (" attr(href) ") ";  
}
```

CSS – Seletores de atributos

```
/* [att]
   O seletor na regra a seguir casa com todos os elementos p que
   tenham o atributo title, independentemente do valor do atributo.
*/
p[title] {
    color: blueviolet;
}

/* [att=val]
   No próximo exemplo o seletor casa com todos os elementos div
   que tem um valor para o atributo class igual a error
*/
div[class=error] {
    color: #f00;
}
```

CSS – Seletores de atributos

```
/* [att~=val]
    Para atingir todos os elementos td cujo atributo headers
    contenha o valor "col1", podemos usar o seguinte seletor.
*/
td[headers~=col1] {
    color:#f00;
}

/* [att|=val]
    O seletor seguinte atinge todo elemento p cujo atributo
    lang comece com en.
*/
p[lang|=en] {
    color:#f00;
}
```


CSS – Seletores de atributos

```
/*
```

Múltiplos seletores de atributos podem ser usados em um mesmo seletor. Isto possibilita atingir vários diferentes atributos para o mesmo elemento. a regra a seguir aplica-se a todos os elementos `blockquote` que tenham o atributo `class` de valor igual a “quote”, e mais o atributo `cite` (independentemente do seu valor)

```
*/
```

```
blockquote[class=quote][cite] {  
    color:#f00;  
}
```

CSS – Reset

```
/*===== CSS Reset =====*/
*,
*:after,
*:before {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    text-decoration: none;
}
body{
    font-size: 100%;
    list-style-type: none;
}
```

Outros modelos de CSS Reset:

<http://meyerweb.com/eric/tools/css/reset/>

CSS – Cores

RGB – Red; Green; Blue

- ✓ Cada canal é representado por um valor numérico que pode variar de 0 à 255.
- ✓ É possível formar 16.777.216 cores.

```
div[class=error] {  
  color: ■rgb(255, 0, 0);  
}
```

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)


rgb(255, 165, 0)


rgb(106, 90, 205)

CSS – Cores

HEX – Hexadecimal

É idêntico ao sistema RGB, porém em hexadecimal.

```
td[headers~=col1] {  
  color:  #f00;  
}
```

```
td[headers~=col1] {  
  color:  #ff0000;  
}
```

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

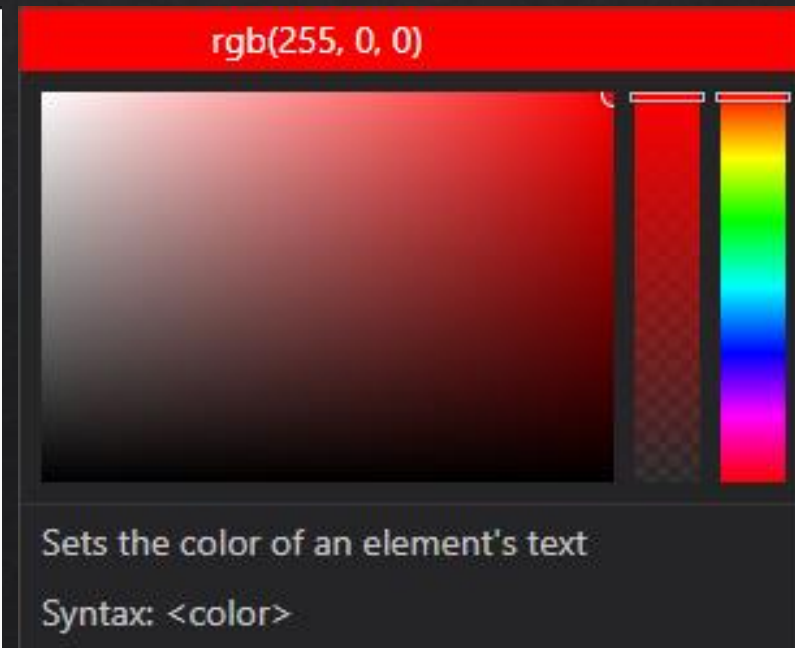
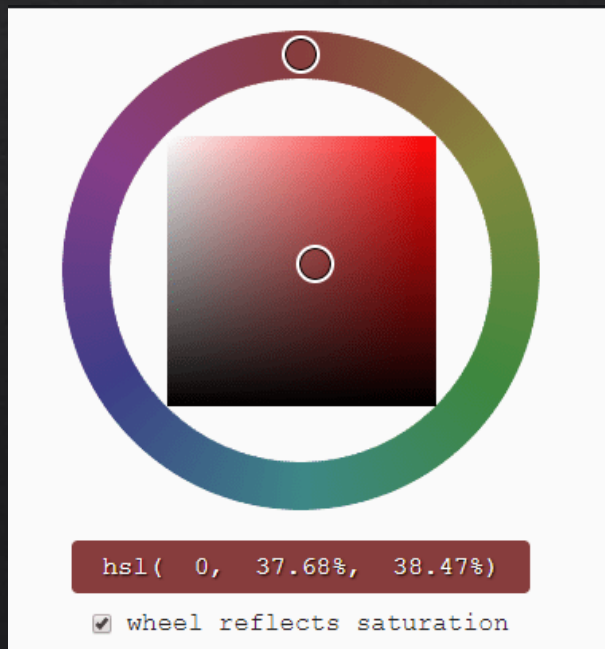
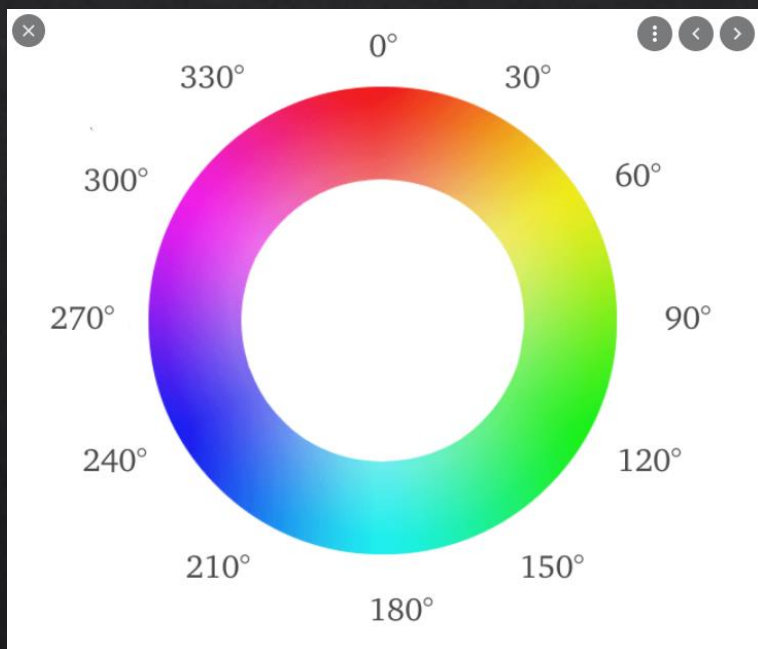
CSS – Cores

HSL – Hue; Saturation; Lightness

Matiz é um grau na roda de cores de 0 a 360. 0 é vermelho, 120 é verde e 240 é azul.

A saturação é um valor percentual, 0% significa um tom de cinza e 100% é a cor total.

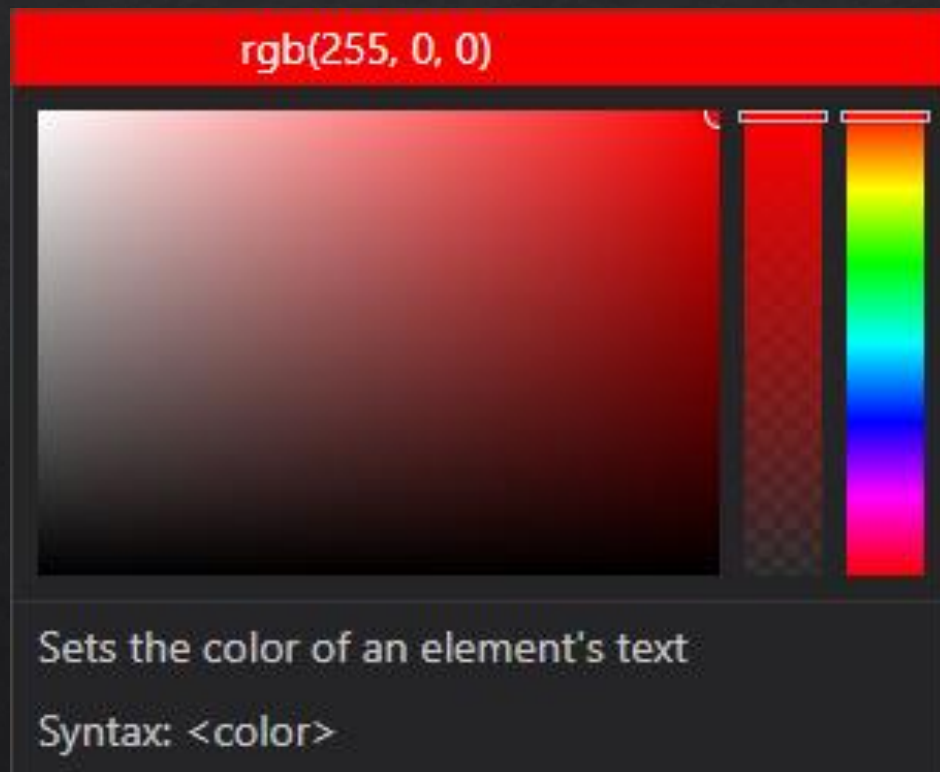
A luminosidade também é uma porcentagem, 0% é preto, 50% não é nem claro nem escuro, 100% é branco.



CSS – Cores

RGBA - Red; Green; Blue; Alpha

HSLA - Hue; Saturation; Lightness; Alpha



HWB – Hue; Whiteness Blackness

CMYK – Cyan; Magenta; Yellow; black

Ncol – Natural Colors

CSS Color Names:

<https://www.w3.org/wiki/CSS/Properties/color/keywords>

https://www.w3schools.com/colors/colors_names.asp

CSS – Backgrounds

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background (shorthand property)

```
div {  
    background-color: green;  
    opacity: 0.3;  
}
```

```
div {  
    background: rgba(0, 255, 0, 0.3) /* Green background with 30% opacity */  
}
```