



# **Análise e Desenvolvimento de Sistemas**

## **Aula 4**

**Professor Ederson Cichaczewski**

## Conversa Inicial

Olá, seja bem-vindo à quarta aula da disciplina “Programação Visual”!

Neste encontro, serão abordados conceitos básicos sobre aplicações WEB desenvolvidas em linguagem HTML5. Conheceremos a linguagem de definição de estilos CSS. Entraremos mais a fundo, entendendo as diferenças entre a utilização de CSS externa e interna, assim como os aspectos mais avançados.

Veremos exemplos de aplicações em linguagem HTML5 com CSS. Serão abordados aspectos sobre a adaptação de uma página a diferentes dispositivos, característica conhecida como responsividade. Serão verificadas, também, algumas considerações sobre usabilidade. Bons estudos!

Confira no vídeo disponível no material *on-line* os comentários iniciais do professor Ederson.

## Contextualizando

HTML é a linguagem de marcação das páginas de internet, que foi evoluindo para novas versões, chegando ao HTML5 em 2014. Da mesma forma, os navegadores (ou browsers) foram evoluindo para suportar a visualização de páginas ao mesmo passo.

A linguagem HTML é desenvolvida pelo World Wide Web Consortium (W3C), organização que define as regras e os padrões para a WEB, mantendo um time de desenvolvedores dedicados.

Acessando o *link* a seguir você acessa o *site* da organização:

<https://www.w3.org/>

Para auxiliar e oferecer mais recursos de formatação e apresentação visual de uma página WEB, foi criado pelo W3C o CSS, que consiste em uma linguagem de estilos que possibilita o desenvolvimento de uma interface com o usuário mais amigável.

O CSS tem a função de separar o conteúdo da página da apresentação da página WEB, incluindo o *layout*, as cores e as fontes. Acesse o link a seguir e confira um *site* sobre CSS:

[www.w3schools.com/css/](http://www.w3schools.com/css/)

Veremos nesta aula exemplos de interface gráfica para WEB com HTML5 e CSS.

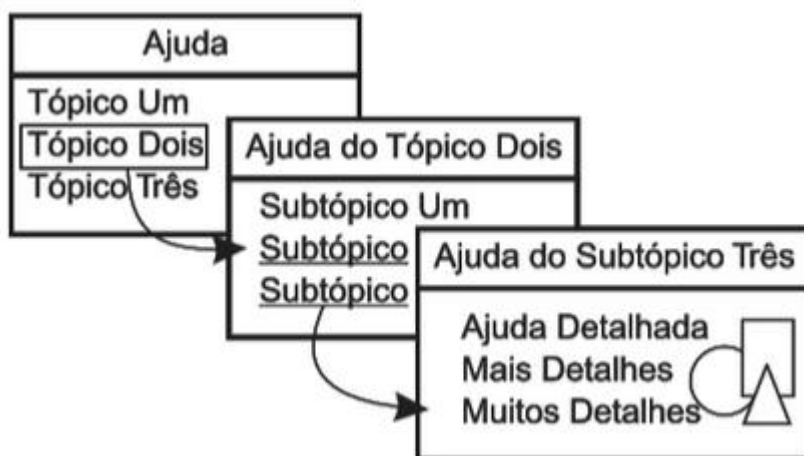
Confira no vídeo disponível no material *on-line* a contextualização feita pelo professor Ederson.

## Tema 01 - Conceitos de HTML e HTML5

A WEB é baseada em hipertexto

O hipertexto permite ler e navegar entre texto e informações visuais de uma forma não linear em uma página WEB.

Um *link* pode levar a uma outra área ou nova tela, como você pode ver a seguir:



Navegador é o aplicativo que serve para visualizar e navegar nas páginas WEB, ele cuida, portanto, da formatação e exibição de páginas HTML, em sua forma básica. Localiza a página por meio da sua URL (Uniform Resource Identifier), que consiste no seu endereço, o qual pode ser de um servidor da WEB ou de um arquivo local. Um *site* da WEB é um conjunto de páginas.

### **HTML (Linguagem de Marcação de Hipertexto)**

É uma linguagem para descrição da estrutura de uma página, que compreende alguns elementos, como título, parágrafo, listas etc. Cada elemento tem um nome, que corresponde a uma tag, entre os sinais de menor e maior <>. A seguir, você confere as principais tags do HTML:

- Estrutura: <html>, <head>, <body>;
- Títulos, cabeçalhos e parágrafos: <title>, <h1> a <h6> e <p>.

A tag deve ser fechada com /, exemplo:

```
<html>
<head>
<title>Título da Página</title>
</head>
<body>
...conteúdo da sua página...
</body>
</html>
```

XHTML: reformulação do HTML em sua versão 4, uma versão extensível com base em XML, cujo objetivo foi padronizar o HTML para exibição em diversos dispositivos, inclusive portáteis.

HTML 5: é a versão mais atual, que traz novos recursos e também é compatível com elementos de código das versões anteriores. A tag html que identifica um código HTML5 é <!DOCTYPE HTML>.

CSS 3 - folhas de estilo em cascata: consiste na formatação da página. Com CSS, especificam-se as fontes de texto, as cores, os planos de fundo, as margens etc.

### **Design da página WEB**

Uma página WEB deve considerar alguns pontos, como:

- Comunicação visual: *layout*, estrutura, usabilidade, criatividade, interatividade etc.;
- Variáveis: cor, textura, tamanho, posição etc.
- Atendimento às necessidades dos usuários: persuasão.

Na figura abaixo, temos uma pirâmide das necessidades de uma página WEB, fazendo uma analogia à pirâmide das necessidades de Maslow, em que na base temos as necessidades essenciais, que todas devem ter, e mais ao topo as características que são diferenciais e buscam a persuasão:



Recomendações para um *design* eficiente:

- Não esconda suas seções principais;
- Use cabeçalhos para resumir os tópicos (títulos descritivos);
- Tenha menus de *links* para navegação;
- Tenha um *link* para a página inicial (normalmente no logotipo);
- Parágrafos devem ser curtos e claros;
- Informações importantes devem ficar no topo da página;
- Agrupar informações relacionadas visualmente.

Para mais informações sobre os conceitos principais a respeito de HTML e HTML5, confira a seguir a videoaula do professor Ederson:

## **Tema 2- Conhecendo o CSS**

### CSS3 e HTML5

Vamos ver os elementos de programação visual para WEB, utilizando os recursos destas duas linguagens, visto que andam juntas em muitos aspectos. Existem 2 formas de adicionar folhas de estilo:

- Interna: usar a tag <style> incorporada no HTML;
- Externa: usar uma folha de estilo externa.

## Uso do CSS

Ideia principal: separar o conteúdo do documento (escrito em HTML) da apresentação do documento (escrita em CSS).

Como isto é feito: através de um conjunto de regras (estilos) que são aplicadas aos elementos das páginas web (tags).

Regra de ouro do design CSS: “Tudo que deve sobrar quando a folha de estilo é removida é o texto que provê a informação e os elementos de HTML que descrevem a estrutura do documento”.

Folha de estilo interna (inline):

```
<html>
<head></head>
<body>
<p style="color: green;
font-weight: bold;
text-align: center"> Folha de estilo inline
</p>
</body>
</html>
```

Dentro da tag <body>, definido como atributos das tags html normais. Escopo: estilo vale somente para um elemento da página.

## Folha de estilo interna (incorporada)



Dentro da tag `<head>` através de uma tag `<style>` com atributo `type`.

Escopo: estilo vale para todas as tags de um dado tipo, dentro de um único documento html.

## Folha de estilo externa

Usa-se a tag `<link>` para vincular a folha de estilos externa à página WEB:

```
<link rel="stylesheet" type="text/css" ref="styles.css" />
```

É definida em um arquivo `.css` separado do documento html. Escopo: pode ser compartilhada por múltiplos documentos html. Tem prioridade 3, mais baixa.

A tag `<link>` é colocada no container `<head>`.



Seletor: o nome do estilo. Possui duas formas:

- Nome de um elemento HTML. Ex.: h1 (aplicável a todos h1).
- Variável: associada manualmente às tags no arquivo HTML através do atributo class. No arquivo CSS, usa-se um ponto (.) na frente da variável.
- ID: semelhante à variável, mas definida pelo caractere #.

### **Arquivo CSS**

Inicia com a definição do conjunto de caracteres UTF-8. Depois vêm os nomes das tags ou das variáveis criadas com class.

```
@charset "UTF-8";  
body {  
font-family: "Arial", sans-serif;  
margin-left: 15%;  
margin-right: 15%; }  
.titulo1 {  
color: red;  
font-family: "Arial";  
font-size: 25px; }  
#corpo{  
color: blue;}
```

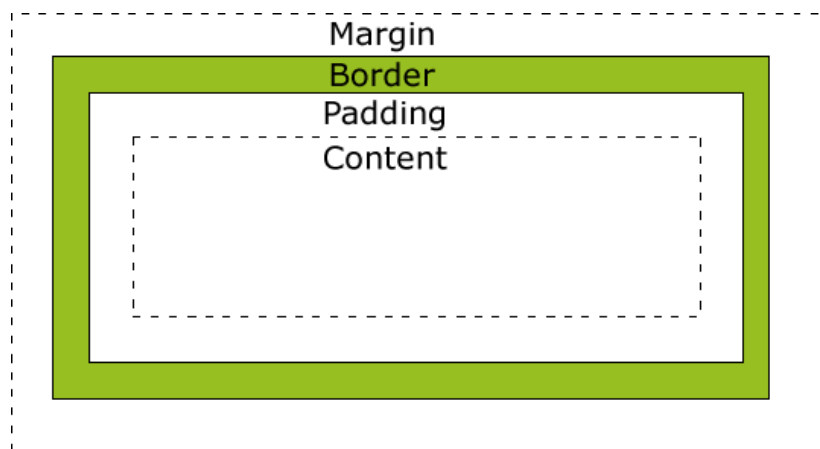
### **Box Model (Modelo de Caixas)**

Cada caixa é composta por 4 elementos:

- Conteúdo (*content*) do elemento, que pode ser um texto do parágrafo, uma imagem etc.;

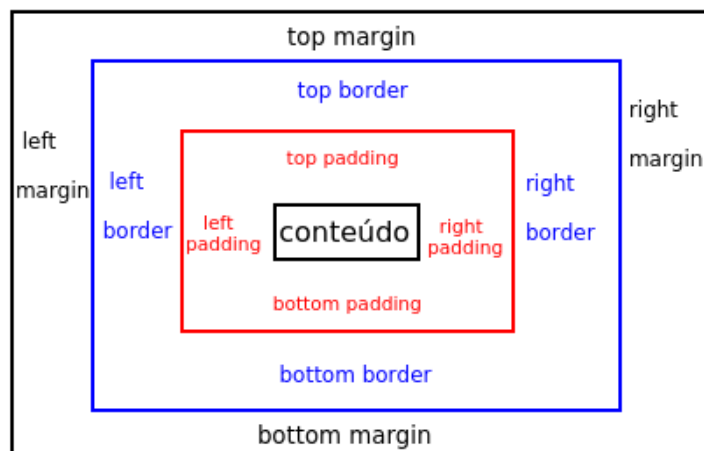
- *Padding* (acolchoamento ou enchimento, em inglês), fica imediatamente ao redor do conteúdo;
- *Border* (bordas) são linhas que delimitam o elemento, podendo aparecer em uma ou mais direções (em cima, em baixo e nas laterais);
- *Margin* (margens) é o espaçamento entre os elementos, ou seja, entre as caixas ou tags (boxes).

Veja a seguir um exemplo:



Podemos formatar a região superior (top), inferior (bottom), esquerda (left) e direita (right) de cada um desses elementos.

Veja na figura como é a ordem das caixas que envolvem qualquer elemento de conteúdo:



Para mais informações sobre o CSS, confira no material *on-line* a videoaula do professor Ederson.

### **Tema 3- Utilização do CSS**

Vamos ver alguns recursos de programação visual que o CSS 3 e o HTML 5 proporcionam para uma página WEB.

#### **Cores**

A definição X11 estabelece 131 nomes de cores. Mas é possível configurar a cor pelo seu valor RGB em porcentagem, para as cores vermelha, verde e azul respectivamente.

Ex.: `rgb(42.8%, 50.3%, 57.5%)`

Também pode-se usar um valor inteiro de 0 a 255.

Ex.: `rgb(255, 127, 0)`

Clicando no botão a seguir, você confere uma lista completa das cores:

<https://www.w3.org/TR/css3-iccprof#x11-color>

Outra forma é usar uma representação hexadecimal do RGB, sendo agrupados os 3 valores inteiros convertidos para hexa com 2 dígitos, resultado em 6 dígitos.

Ex.: `#FF7F00`

Outra forma é atribuir a cor pelo modelo de matiz-saturação-luminância HSL (*Hue-Saturation-Luminance*). O primeiro valor não é uma porcentagem, mas um valor entre 0 e 359 (graus de um círculo).

Ex.: `hsl(0, 100%, 50%)`

## Transparência

Adiciona-se um 4º parâmetro com valor entre 0 e 1, que corresponde à propriedade alfa. Usa-se rgba e hsla.

Ex.: rgba(255, 127, 31, 0.5)

Ex.: hsla(120, 90%, 50%, 0.3)

## Layout

Seções do HTML5:

- Body – começo de um elemento do corpo;
- Article – artigo com começo, meio e fim dentro do documento;
- Header – *container* para o cabeçalho de uma seção ou página;
- h1 a h6 – títulos de texto 1 a 6;
- Nav – representação de uma seção de um documento que será usada para navegação;
- Section – identificador de tema para agrupamento de conteúdo;
- Aside – conteúdo tangencialmente relacionado ao conteúdo do artigo;
- Footer – *container* para o rodapé de uma seção ou página;
- Address – *container* para o endereço.



### **Background (plano de fundo)**

Propriedades:

`background-image` → `url(arquivo)`

`background-color` → cor

`background-position` → top, center, bottom, left, center, right

`background-repeat` → repeat-x, repeat-y, repeat, no-repeat

`background-attachment` → scroll, fixed

```
body { background-image: url(logo.gif);  
background-position: bottom right;  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-color: #eeeeee }
```

## Fontes

Propriedades:

Font-size →

- tam-abs: xx-small, x-small, small, medium, large, x-large, xx-large;
- tam-relat: larger, smaller;
- tamanho: valor em pontos (pt);
- percent: valor percentual em relação ao padrão.

Font-family → fonte, genérica (famílias genéricas: serif, sans-serif, cursive, fantasy, monospace);

Font-weight → normal, bold, bolder, lighter, 100-900;

Font-style → normal, italic, oblique.

```
<style type = "text/css">  
  em      { font-weight: bold;  
            color: black }  
  h1      { font-family: tahoma, helvetica, sans-serif }  
  p       { font-size: 12pt;  
            font-family: arial, sans-serif }  
  .special { color: #6666ff }  
</style>
```

---

## Textos

Propriedades:

Letter-spacing → normal, valor em px ou pt;

Line-height → normal, valor em px ou pt, percentual;

Text-align → left, right, center, justify;

Vertical-align → top, middle, bottom;

Text-decoration → none, underline, overline, line-through, blink;

Text-indent → valor, percentual;

Text-transform → none, capitalize, uppercase, lowercase;

Word-spacing → normal, valor em px ou pt.

```
<span style="letter-spacing: normal">letter-spacing: normal<br />
</span>
```

```
<span style="letter-spacing: 10px">letter-spacing: 10px</span>
```

```
<span style="text-align: right">text-align: right<br />
</span>
```

```
<hr />
```

```
<span style="text-decoration: underline">text-decoration:
underline</span><br />
```

```
<span style="text-decoration: overline">text-decoration:
overline</span><br />
```

### Modelo de caixa

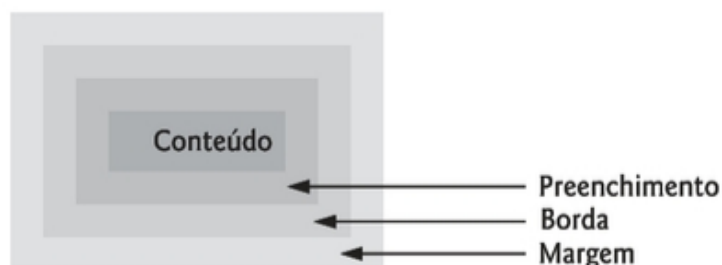
Todos os elementos de bloco possuem uma caixa virtual ao seu redor.

Propriedades:

Border-width → border-bottom-width, border-left-width, border-right-width, border-top-width.

Valores: thin, medium, thick, valor em px.

Border-color → border-bottom-color, border-left-color, border-right-color, border-top-color.



Border-style → border-bottom-style, border-left-style, border-right-style, border-top-style.

Valores: none, hidden, dotted, solid, double, groove, inset, outset, dashed, ridge.

Padding → padding-bottom, padding-left, padding-right, padding-top. Valores em px.

<b>.thin</b>	<b>{ border-width: thin }</b>
<b>.solid</b>	<b>{ border-style: solid }</b>
<b>.dashed</b>	<b>{ border-style: dashed }</b>
<b>.red</b>	<b>{ border-color: red }</b>

## Dimensões

Propriedades:

Height → auto, valor em px, valor percentual.

Width → auto, valor em px, valor percentual.

```
<div style = "width: 20%; height: 150px; overflow: scroll">
```

## Posicionamento

Propriedades:

Position → absolute, relative, static;

Left → auto, valor em px, valor percentual;

Right → auto, valor em px, valor percentual;

Top → auto, valor em px, valor percentual;

Bottom → auto, valor em px, valor percentual;

Visibility → inherit, visible, hidden;

Z-index → número da camada;

No vídeo disponível no material *on-line*, o professor Ederson fala mais sobre a utilização do CSS.



## Tema 4 - Aspectos avançados com CSS

### Posicionamento relativo

É relativo a outros elementos. Em muitos casos, é melhor usar valores relativos, por exemplo 1 cm de distância em um computador parece pouco, mas em um *smartphone* é bastante.

Valores em “em” significa que é múltiplo do tamanho da fonte padrão, normalmente 16px (pixels). Por exemplo, uma fonte de 24px pode ser especificada como:

**font-size: 1.5em**

O mesmo pode ser feito para outras dimensões ou distâncias, usadas para o posicionamento. Por exemplo:

```
#main .info {float: left; width: 7.308em;}  
#main .text {float: right; width: 44.231em;}
```

Valores dados em “ex” correspondem à altura x de uma fonte, normalmente igual à altura de um x minúsculo:

```
.super    { position: relative;  
             top: -1ex }  
.sub      { position: relative;  
             bottom: -1ex }  
.shiftright { position: relative;  
               left: -1ex }  
.shiftright { position: relative;  
               right: -1ex }
```

A propriedade float faz elementos posicionados de forma relativa flutuar para uma direção.

Float → none, left, right.

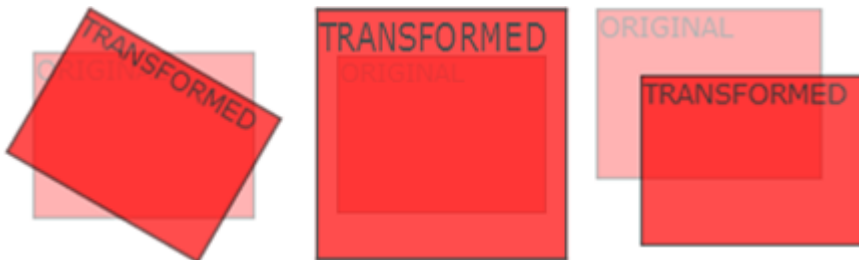
### Propriedades adicionais:

- Display → especifica como um elemento será criado e disposto na tela. Valores: inline, block, none.
- Overflow → especifica se o elemento será cortando quando ultrapassar a caixa do elemento. Valores: visible, hidden, scroll, auto, inherit.
- clear → indica o lado de uma caixa que não pode ser adjacente à outra. Valores: none, left, right, both.

### Transformações geométricas 2D

Propriedades:

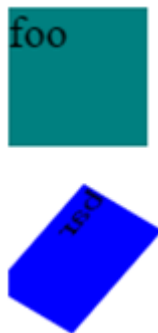
- Transform: rotate. Ex.: transform: rotate(20deg);
- Scale, scaleX, scaleY. Ex.: transform: scale(2, 3);
- Translate, translateX, translateY. Ex.: transform: translate(50px, 100px);
- Skew, skewX, skewY. Ex.: transform: skew(20deg, 10deg);
- Matrix. Ex.: transform: matrix(1, -0.3, 0, 1, 0, 0);
- Transform-origin. Ex.: transform-origin: 20% 40%.



## Transformações geométricas 3D

Propriedades:

- Transform: rotate3d. Ex.: transform: rotate3d(1,0,0,60deg);
- Scale3d, scaleZ. Ex.: transform: scale3d(0.8, 2, 0.2);
- Translate3d, translate Z. Ex.: transform: translate3d(10px,0px,0px);
- Matrix3d. Ex.: transform: matrix3d(2,0,0,0,0,0,0,0.98,0.17,0,0,-0.17,0.98,0,3,4,0,1);
- Perspective. Ex.: perspective: 500px.



Acesse o *link* a seguir e confira uma referência *on-line* sobre CSS:

<http://www.htmlprogressivo.net/p/curso-de-css-cs-cascading-style-sheets.html>

Para mais informações sobre os aspectos avançados com CSS, confira no material *on-line* videoaula do professor Ederson.

## Tema 5 - Aplicações com CSS

Criando uma página com CSS

Página sobre a minha biblioteca WEB.

- Ferramenta: Notepad++.
- Estrutura básica: arquivo csshtml.html.

1º Passo: criar uma estrutura básica HTML

```
<!DOCTYPE html>

<html>

<head> <title> A Biblioteca </title> </head>

<body>

<h1> Minha Biblioteca WEB </h1>

<p> Aqui estão os livros da minha coleção. </p>

</body>

</html>
```

2º passo: adicionando seções do HTML5 (\*Não altera nada na visualização da página.)

```
<!DOCTYPE html>
<html>
<head>
<title> A Biblioteca </title>
</head>
<body>
<header> <h1> Minha Biblioteca WEB </h1> </header>
<section id="main">
<p> Aqui estão os livros da minha coleção. </p>
</section>
<footer> </footer>
</body>
</html>
```

3º passo: adicionando um arquivo CSS no HTML

```
<!DOCTYPE html>
<html>
<head>
<title> A Biblioteca </title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <header> <h1> Minha Biblioteca WEB </h1> </header>
  <section id="main">
    <p> Aqui estão os livros da minha coleção. </p>
  </section>
  <footer> </footer>
</body>
</html>
```

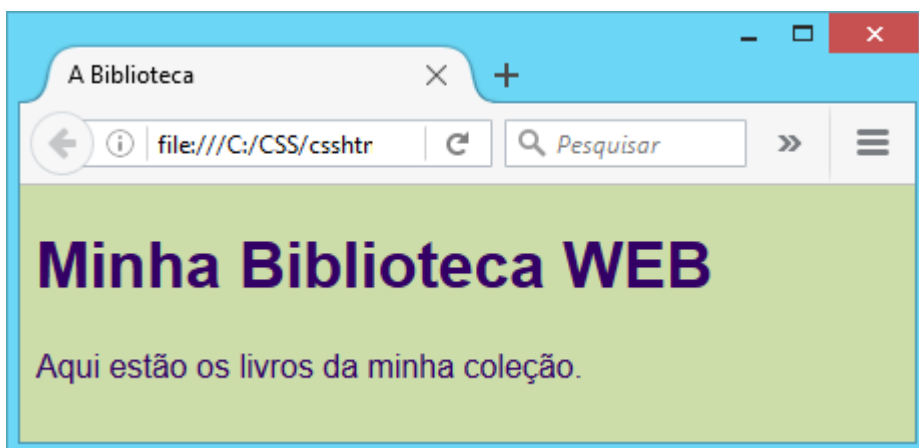
Editando o arquivo style.css

Configurando a cor do fundo, fonte e cor da fonte da seção body.

```
body{
background-color: #ccddaa;

font-family: Arial, Helvetica, sans-serif;

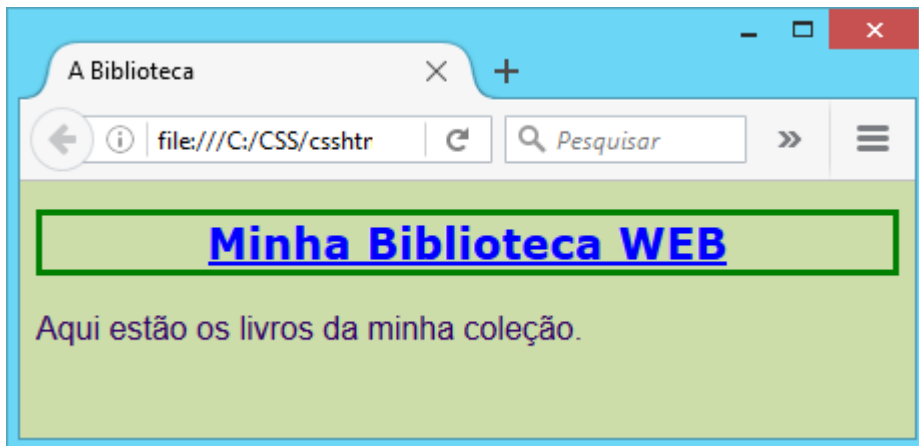
color: #330066}
```



## CSS: configurando o título

Parâmetros: fonte, cor da fonte, alinhamento do texto, sublinhado e borda.

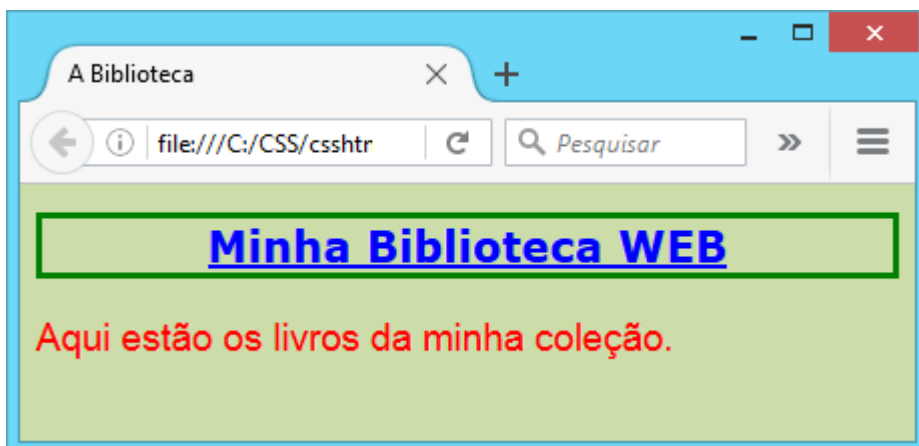
```
h1{  
  
font-family: Verdana, sans-serif;  
  
font-size: 16pt;  
  
color: blue;  
  
text-align: center;  
  
text-decoration: underline;  
  
border-style: solid;  
  
border-color: green;  
  
border-width: medium; }
```



## CSS: configurando o texto

Parâmetros: fonte e cor.

```
#main{  
  
font-family: Arial, sans-serif;  
  
font-size: 14pt;  
  
color: red; }
```



### Dicas para página responsiva

Algumas recomendações para que a página fique adaptável a diferentes dispositivos, desde um smartphone até uma TV de 46 polegadas:

- Usar posicionamento relativo;
- Usar dimensões em porcentagem;
- Dividir o *layout* horizontal (largura) com 73% para o conteúdo e 27% para o menu lateral: `width: 27%;`
- Imagens com redimensionamento automático: `img{ max-width: 100%; height: auto;}`;
- Página centralizada e largura máxima: `#main { max-width: 960px; margin: 0 auto; }.`

### Dicas de usabilidade

Algumas recomendações para que a página seja consistente:

- Logotipo sempre aponta para a página principal;
- Destacar os *links* já visitados;
- Destacar campos ativos;
- Colocar uma cor de fundo em imagens de fundo;
- Evitar textos longos no conteúdo;
- Evitar sublinhar textos que não são *links*;

- Usar textos alinhados à esquerda ao invés de alinhamento justificado.
- Usar espaçamento entre linhas igual a 1.5 vezes o tamanho da fonte.

Para mais informações sobre as aplicações com CSS, confira no material *on-line* a videoaula do professor Ederson.

## **Trocando ideias**

Nesta aula, iniciamos com uma introdução à linguagem HTML, seguindo com as novidades da versão HTML 5, linguagem utilizada como base para todas as aplicações da WEB. Vale lembrar também que a linguagem HTML é interpretada pelo navegador de internet, portanto, não é necessário compilá-la. Ou seja, para desenvolver o código fonte não é necessária nenhuma ferramenta de *software* específica, apenas um editor de texto já é o suficiente.

É importante considerarmos que hoje em dia temos diversos dispositivos, de diferentes tamanhos, que exibem páginas WEB, como computadores, *tablets* e smartphones. A tela de cada um desses dispositivos tem tamanhos diferenciados, portanto é importante nos preocuparmos em desenvolver páginas que sejam adaptáveis, levando em conta as questões de responsividade.

De forma geral, a página deve ser atrativa ao usuário, e para os elementos de apresentação aprendemos a linguagem de estilos CSS, que possui uma infinidade de recursos a serem explorados. O conhecimento adquirido nesta aula é fundamental para o aprendizado do aluno, de forma a ter em suas mãos os conhecimentos de desenvolvimento WEB que serão utilizados ao longo do curso.



Não fique com dúvidas sobre o assunto desta aula, estude o tema consultando as referências bibliográficas desta aula e outras fontes de pesquisa.

Entre também no fórum e discuta as questões que achar mais relevantes.

## **Na Prática**

Como exercício para consolidar os conceitos vistos nesta aula, vamos desenvolver uma página WEB com *layout* responsivo.

O ideal para a largura máxima de *layout* de um *site* da atualidade seria 960px, em função da resolução da maioria dos monitores ser 1024x768. O *layout* será dividido em 710px para o conteúdo e 250px para a barra lateral.

Para isto será necessário criar 3 divs: o container (que irá definir a largura máxima da página), o content (conteúdo) e a sidebar (barra lateral).

Acesse o material *on-line* e confira a resolução.

## **Síntese**

Nesta aula vimos os conceitos das linguagens de marcação HTML e HTML 5. Vimos também uma introdução sobre a linguagem de estilos CSS, seus recursos básicos e avançados. Por fim, trabalhamos em exemplos de todos os principais conceitos de HTML, HTML 5 e CSS abordados ao longo da aula.

O bom entendimento desta aula é fundamental, visto que trata dos recursos de interface visual WEB, utilizada na criação de *sítes* da internet.

Confira no material *on-line* a videoaula de síntese do professor Ederson.

## Referências

LEMAY, Laura. Aprenda a Criar Páginas Web com HTML e XHTML em 21 dias. São Paulo: Pearson, 2002.

DEITEL, P. J. Ajax, Rich. Internet Applications e desenvolvimento WEB para programadores. São Paulo: Pearson, 2008.

MEYER, Eric A. Smashing CSS: Técnicas Profissionais para um Layout Moderno. São Paulo: Bookman, 2011.

SANDERS, Bill. Smashing HTML 5: Técnicas para a Nova Geração da WEB. São Paulo: Bookman, 2012.