

Autoria Web

Aula 02 - HTML: Cabeçalhos, Títulos, Parágrafos e Links

Apresentação

Nesta aula, você irá começar a estudar HTML (*Hyper Text Markup Language*) e sua estrutura geral. Serão apresentados os conceitos de atributo e elemento, e será visto como se especificam informações sobre o documento e sobre o conteúdo. Você aprenderá a definir títulos, parágrafos e link, e também alguns tipos de formatação e estilos para texto. Você deve retornar à **Aula 1** da nossa disciplina para relembrar quais as ferramentas mínimas necessárias para desenvolver documentos web.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você deverá ser capaz de:

- Conceituar HTML.
- Definir e diferenciar atributos e elementos.
- Citar e definir os elementos do cabeçalho de documentos HTML.
- Criar um documento HTML com títulos, parágrafos e links.

HTML

HTML significa Linguagem de Marcação para HiperTexto. Essa linguagem é utilizada para escrever páginas web. A versão de HTML que você irá estudar neste curso é a 4.01. A seguir, você irá aprender a estrutura geral de um documento HTML, conhecerá alguns elementos e atributos que compõem essa estrutura, tais como as marcações para títulos e links, e ainda saberá especificar metadados que podem ser analisados por motores de busca.

Apesar de estarmos focando na versão 4.01 do HTML, vale a pena conhecer a versão mais nova.

A especificação para o HTML 5 está disponível em: <http://www.w3.org/TR/html5/> (em inglês).

Principais mudanças do HTML 4 para o HTML 5: <http://fabrica.ms.senac.br/2015/06/entenda-a-diferenca-entre-html-e-html5-veja-como-tudo-ficou-mais-facil/>

Atributos e elementos

Sendo uma linguagem de marcação, HTML consiste de tags (ou etiquetas) que indicam as partes do documento. Essas tags são **elementos** do hipertexto, por exemplo, a tag <p> é um elemento que indica um parágrafo (usamos os termos tag e elemento como sinônimos). Todos os elementos são representados entre os símbolos de menor e maior. E a maioria das tags ocorre em pares, sendo uma tag de abertura e uma de fechamento, como no exemplo genérico a seguir.

| | |
|---|----------------|
| 1 | <tag>...</tag> |
|---|----------------|

Isso é necessário porque essas tags marcam onde começa e termina o texto com a formatação especificada por ela. Há também elementos “vazios”, pois não marcam uma região de texto, apenas inserem alguma coisa no documento, por

exemplo, a tag `
` insere uma quebra de linha. Nesses casos, pode se escrever `
</br>`, uma tag junto da outra, ou simplesmente `
`, nestes casos tem-se genericamente:

`<tag/>`

Vimos que os elementos são os termos que aparecem entre os símbolos de menor e maior. No entanto, além destes, entre os símbolos de menor e maior pode haver também atributos. Assim, elementos podem ter atributos. Os atributos oferecem algumas informações adicionais ao elemento. Cada atributo possui um nome e um valor, representado da seguinte maneira:

```
1 <tag atributo1=valor1 atributo2=valor2>...</tag>
```

Em HTML, o valor dos atributos pode ser escrito entre aspas duplas ou sem aspas. Para padronizar, e ficar de acordo com o padrão XHTML, adotamos a descrição de atributos entre aspas, como no exemplo a seguir:

```
1 <tag atributo1="valor1" atributo2="valor2">...</tag>
```

A seguir, você verá como um documento HTML é estruturado e no decorrer desta aula você verá vários exemplos de elementos (tags) e atributos.

Estrutura de um documento HTML 5

No HTML5 a única declaração necessária é o:

`<!DOCTYPE html>`

Um exemplo mais básico que podemos ter em HTML5 é mostrado a seguir:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Título do documento</title>
7 </head>
8
9 <body>
10   Conteúdo
11 </body>
12
13 </html>
```

Estrutura de um documento HTML 4.01

Qualquer documento HTML é composto de 2 partes, sendo que a segunda parte ainda é subdividida por mais duas, como definido a seguir.

1ª) Informações sobre a versão do HTML. Essa parte é marcada por uma seção DOCTYPE, veja no exemplo da **Figura 1**. Doctype é uma DTD (Definição do Tipo de Documento), ou seja, um documento que especifica as regras sintáticas da linguagem. Observe que mesmo estando entre os símbolos de menor "<" e maior ">", o doctype não é uma tag, é uma instrução muito importante que diz ao browser qual o tipo e versão (HTML 3.2, HTML 4.01, XHTML etc.) em que a página HTML está escrita. Essa instrução sempre aparece no topo do código, antes da tag . Existem, basicamente, 3 DTD para HTML 4.01. Elas estão definidas no quadro a seguir.

- [Strict DTD](#) – inclui todos os elementos e atributos recomendados e que não estão em desuso (deprecated). Para esses documentos, use a seguinte declaração:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- Transitional DTD [<http://www.w3.org/TR/1999/REC-html401-19991224/sgml/loosedtd.html>] – inclui todos os elementos da strict DTD e adiciona os elementos e atributos deprecated. Essa DTD é usada em páginas que ainda usam práticas e códigos antigos que se aconselha não usar mais. Para esses documentos use a seguinte declaração:
- Frameset DTD [<http://www.w3.org/TR/1999/REC-html401-19991224/sgml/framesetdtd.html>] – inclui todos os elementos e atributos da transitional DTD e adiciona o elemento frame, possibilitando a estruturação de páginas em frames. Para esses documentos, a seguinte declaração é usada:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

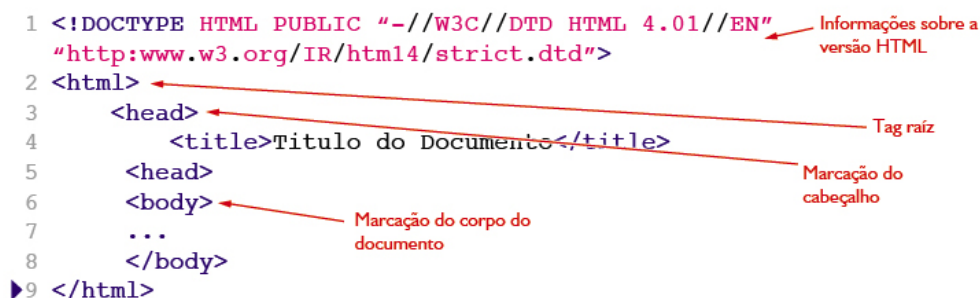
2ª) O documento propriamente dito, delimitado pelas tags <html>...</html>, que subdivide-se em:

- uma seção de cabeçalho, delimitado pelo elemento <head>...</head>;
- uma seção de corpo, delimitada pelo elemento <body> ...</body>, que contém o conteúdo do documento.

Na **Figura 1**, você pode ver a estrutura geral de um documento HTML. A primeira linha do código informa ao browser (navegador) qual é o tipo/versão de HTML que está sendo usado. As tags <html>... </html> demarcam, respectivamente, o início e o fim do documento. <html> é a raiz do documento, ou seja, qualquer outra tag deve estar dentro dela. Entre essas tags, deve haver apenas duas partes, a primeira demarcada pela tag <head>...</head>, que indica o cabeçalho da página e a segunda parte, demarcada pela tag <body>...</body>, que indica o corpo do documento. Vamos aos detalhes de cada uma dessas partes.

Figura 01 - Estrutura geral de um documento HTML

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     <title>Titulo do Documento</title>
5   </head>
6   <body>
7     ...
8   </body>
9 </html>
```



Atenção!

Em HTML é permitido escrever as tags em minúsculo, ou em maiúsculo, ou misturando ambos (<HTML>, <Html>, <hTML>, <HtMl>, ...). Entretanto, é aconselhado que você se habitue, desde já, a escrever tudo em minúsculo, pois logo você estudará XHTML e essa linguagem exige a escrita em minúsculo.

A linguagem XHTML é *case sensitive*, isso significa sensível ao tamanho, ou sensível a maiúsculas e minúsculas, ele é usado para indicar que há diferenças entre letras com caixa alta e com caixa normal.

O cabeçalho –<head>...</head>

A seção <head> contém informações sobre o documento, por exemplo: tipo de dado contido no documento, autor, sua versão, em que janela os links devem abrir, dentre outros. As possíveis tags contidas em <head> são mostradas no **Quadro 1**, a seguir.

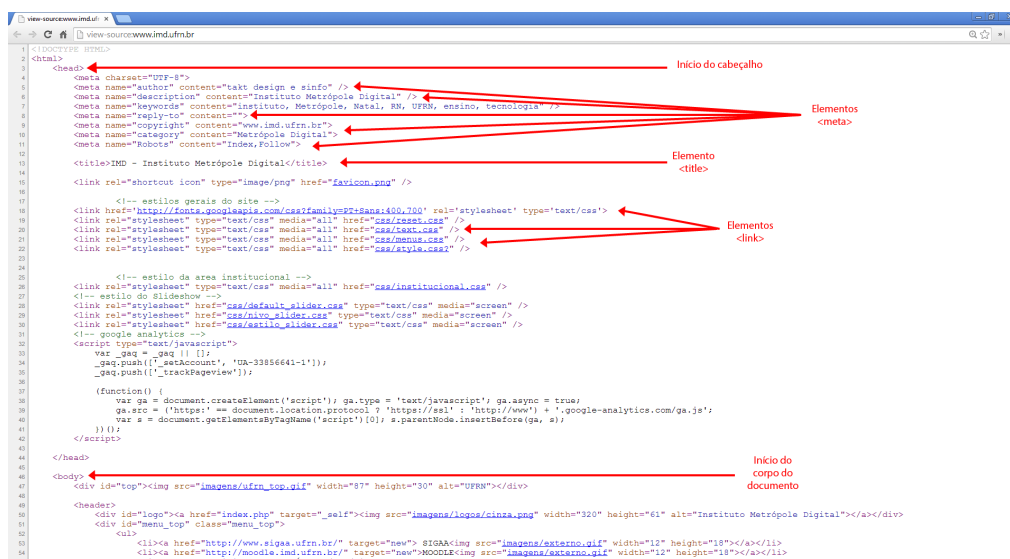
| Tag | Descrição |
|-------------|---|
| <title> | Especifica o título do documento. |
| <base /> | Especifica um endereço ou um alvo para os links do documento. |

| Tag | Descrição |
|-----------------------------|---|
| <code><link /></code> | Especifica a relação entre o documento e arquivos externos. |
| <code><meta /></code> | Especifica metadados. |
| <code><script></code> | Especifica <u>scripts</u> (São códigos de programas que efetuam determinadas tarefas, tal como verificar se os campos de um formulário foram preenchidos.). |
| <code><style></code> | Especifica o estilo dos elementos do documento. |

Quadro 1 – Elementos da tag

O elemento `<head>` pode ter um atributo chamado `profile`, que especifica o local de um ou mais profiles (perfis) de metadados. O valor desse atributo é um URI Uniform Resource Identifier (Cadeia de caracteres compacta usada para identificar ou denominar um recurso na internet.). A seguir, na **Figura 2**, é exibido um exemplo de cabeçalho contendo alguns de seus subelementos, os quais serão explicados posteriormente. Preste bem atenção nesta figura, pois ainda nesta aula voltaremos a ela algumas vezes.

Figura 02 - Exemplo de cabeçalho da página do IMD



Fonte: <http://www.imd.ufrn.br> Acesso em: 12 mar. 2012.

Elemento <title>...</title>

O elemento <title> define o título do documento e deve aparecer exatamente uma vez dentro do elemento <head>. O título é usado para identificar um documento Web. Na prática, esse título aparece no topo do browser quando a página está carregada e na lista de favoritos, caso você adicione-a. Veja na **Figura 3** onde aparece o texto definido pela tag <title> da **Figura 2**. Assim, é importante que o título de um documento seja sugestivo quanto ao seu conteúdo, pois ele será usado como um referencial de sua página.

Figura 03 - Interface do browser ilustrando o título de um documento HTML e esse mesmo título na lista de favoritos



Vídeo 02 - Notepad

Elemento <meta/>

Além do título, <head> pode conter outras informações utilizadas por motores de busca para identificar informações além do conteúdo do documento. [Motores de busca](#) são sistemas de software projetados para encontrar informações armazenadas em um sistema computacional (ou, nesse caso, na web) a partir de palavras-chave. Alguns exemplos são Google, Yahoo, Lycos, Cadê. Essas informações são chamadas de metadados e são marcadas por elementos **<meta>**. Por exemplo, é possível indicar qual o autor de um documento, da seguinte maneira:

| | |
|---|---|
| 1 | <meta name="author" content="Jorge Amado"/> |
|---|---|

A w3c não especifica quais metadados um documento deve possuir. Os desenvolvedores são livres para criar os metadados que quiserem, alguns exemplos comuns são: "autor", "copyright", "palavra-chave", "versão", "data de publicação", dentre outros.

Para especificar metadados, são necessários dois passos:

- 1 – Declarar uma propriedade e um valor para ela. No exemplo anterior, autor é a propriedade enquanto o valor é Jorge Amado. Isso pode ser feito de 2 maneiras.
- de dentro de um documento, por meio do elemento **<meta>**, como mostrado no exemplo anterior;
- de fora do documento, ligando-o aos metadados por meio do elemento **<link>** (explicado a seguir).
- 2 – Indicar um profile onde a propriedade e seus valores são definidos. Para designar um profile, use o atributo profile do elemento **<head>**. Se um profile for definido para o elemento head, o mesmo profile se aplica para todos os elementos **<meta>** e **<link>**.

Os atributos que o elemento <meta> pode conter são 4:

- 1 – Name: indica o nome de uma propriedade.

- 2 – Content: indica o valor para uma propriedade.

A seguir, apresentamos um exemplo de metadado especificado na linha 6 da **Figura 2**. Esse exemplo especifica, através do atributo `name="keywords"` e `content="..."`, palavras-chave do site da W3C-Brasil.

```
1 <meta name="keywords" content="W3C, World Wide Web, Web, WWW, Consortium, computer,  
2 access, accessibility, semantic, worldwide, W3, HTML, XML, standard, language, technology, link,  
3 CSS, RDF, XSL, Berners-Lee, Berners, Lee, style sheet, cascading, schema, XHTML, mobile, SVG,  
4 PNG, PICS, DOM, SMIL, MathML, markup, Amaya, Jigsaw, free, open source, software" />
```

- 3 – Scheme: provê o contexto para interpretação de um metadado. Por exemplo, na declaração seguinte, o atributo **scheme** pode ajudar a indicar que o identificador é um número ISBN.

```
1 <meta scheme="ISBN" name="identifier" content="0-8230-2355-9"/>
```

Elemento <meta/>

Essa contextualização dada pelo atributo **scheme** às vezes é essencial, principalmente quando um metadado pode ser especificado em diferentes formatos e assim ter diferentes interpretações. Por exemplo, um autor pode especificar a data "10-9-97"; essa data pode ser interpretada como 9 de outubro de 1997 ou 10 de Setembro de 1997. Um atributo [scheme](#) com valor "Month-Day-Year" resolve essa ambiguidade.

```
1 <meta scheme="Month-Day-Year" name="publish_date" content="10-9-97"/>
```

Os valores do atributo **scheme** dependem do nome da propriedade e do profile associado.

- 4 –Http-equiv:o atributo [http-equiv](#) pode ser usado no lugar do atributo nome e tem um significado especial quando os documentos são recuperados através do protocolo HTTP. Servidores HTTP podem usar o nome da propriedade especificada por esse atributo para ocasionar algum comportamento especial. Você pode consultar a especificação HTTP ([RFC2616]) para obter detalhes sobre cabeçalhos válidos. No exemplo a seguir "content-type" indica o tipo

de mídia do documento, essa informação ajuda o browser a exibir corretamente os caracteres especiais que estiverem presentes no texto.

```
1 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Um outro exemplo no documento da **Figura 2** (linha 8) é o seguinte:

```
1 <meta http-equiv="Pragma" content="no-cache" />
```

Nesse exemplo, a propriedade “Pragma” com valor “no-cache” indica que o browser não deverá armazenar a página em cache , ou seja, sempre que essa for requisitada será procurada no servidor, permitindo que eventuais atualizações sejam carregadas.

O atributo **scheme** não é suportado no HTML 5.

No HTML5, a forma de definir o codificação de caracteres foi simplificada com a adição de um novo atributo, o **charset**: `<meta charset="UTF-8">`

Elemento <link/>

O elemento <link> define um relacionamento entre o documento e algum recurso externo.

```
1 <link rel="help" href="/help/">
```

No HTML5 foi adicionado um novo atributo na tag , o **sizes**. Este atributo é usado para definir o tamanho quando de um ícone: `<link rel="icon" href="site_icone.gif" type="image/gif" sizes="16x16">`

O elemento `<link/>` é comumente utilizado para ligar um documento a um CSS. No exemplo que segue, o documento é ligado a um CSS (indicado pelo atributo `rel` e `type`), cujo endereço é `"css/w3c.css"`.

```
1 <link rel="stylesheet" type="text/css" media="screen" href="css/w3c.css" />
```



Vídeo 03 - Ligação de um recurso externo

Elemento `<base>`

O elemento `<base>` especifica um endereço (através do atributo `href`) ou um alvo (através do atributo `target`) que deve ser default para todos os links contidos naquele documento. Por exemplo, na **Figura 4**, a primeira tag `<base href=.../>` indica que todos os links devem ser direcionados para o endereço `<http://www.w3c.org/>`, enquanto a segunda tag `<base target="_blank"/>` indica que todos os links devem ser abertos em uma nova janela do navegador. Sendo assim, ao clicar no link mostrado no lado direito da **Figura 4**, o navegador abrirá uma nova janela mostrando o conteúdo do endereço `<http://www.w3c.org/standards/webdesign/htmlcss>`, como mostrado na **Figura 5**.

Figura 04 - Exemplo do uso da tag e os atributos `href` e `target`

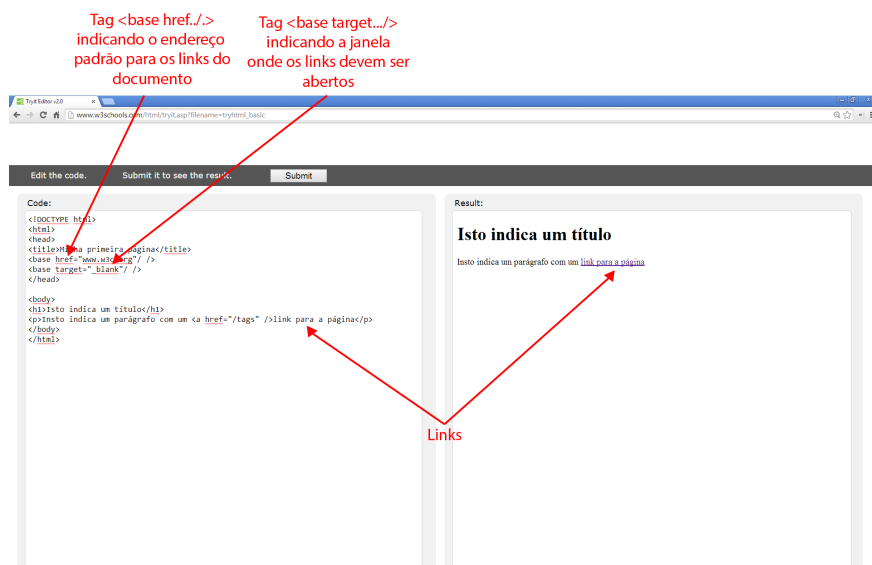
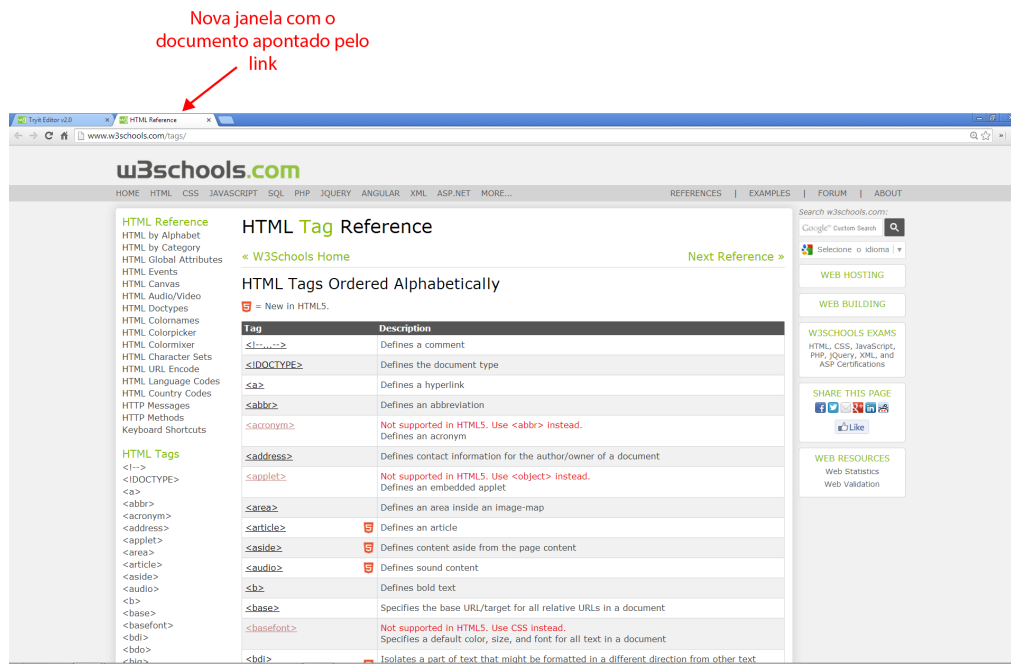


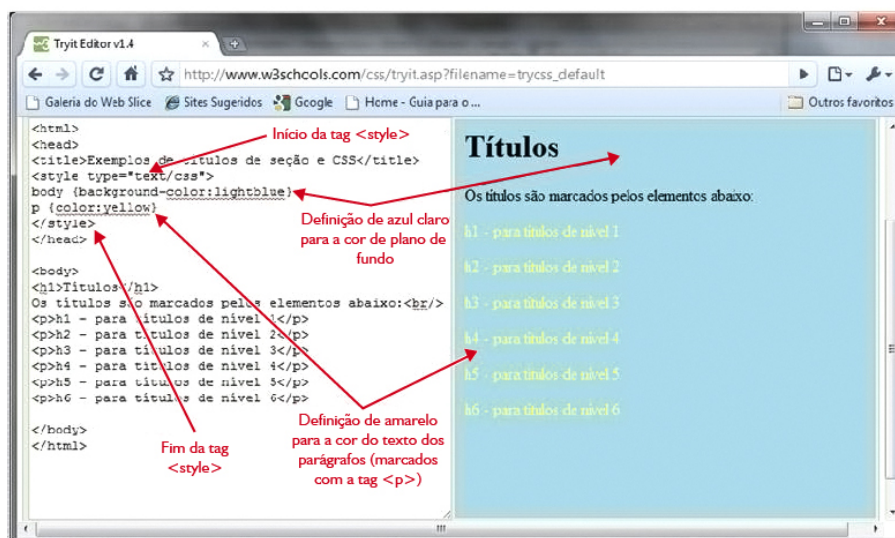
Figura 05 - Interface do browser mostrando a nova janela aberta, com o endereço da W3C



Elemento <style> ...</style>

O elemento **<style>** é usado para especificar o estilo de informação do documento HTML. Dentro do elemento style, você deve indicar como os elementos devem ser dispostos e organizados no navegador. Isso significa que se você não definir um arquivo separado para o seu documento CSS e ligá-lo com a tag **<link/>**, então o código CSS deve vir dentro da tag **<style>**, como no exemplo a seguir.

Figura 06 - Interface do browser mostrando um exemplo da tag <style>



Nesse exemplo da **Figura 6**, são definidas as cores do plano de fundo e do texto dos parágrafos, respectivamente, azul e amarela. Não se preocupe em entender todos os detalhes sintáticos de CSS, no momento você só precisa saber qual a utilidade da tag **<style>**. Você verá muitos exemplos de código CSS a partir da Aula 4.



Vídeo 04 - Estilizando o HTML

Elemento <script>...</script>

O elemento <script> é usado para especificar os scripts (javascript) que atuam no documento. Assim, essa tag pode conter o script diretamente ou pode, através do atributo src, indicar um arquivo externo enquanto o atributo type indica o tipo de script. Javascript é usado para manipulação de imagens, validação de formulários e mudanças dinâmicas de conteúdo.

Como nesta disciplina você se deterá a estudar somente o desenvolvimento de páginas estáticas, então o estudo de javascript não será abordado em nossas aulas. Contudo, você verá alguns pequenos exemplos quando estiver estudando formulários. A seguir, você verá como se representa um javascript que escreve no documento a frase Hello World.

```
1 <script type="text/javascript">
2   document.write("Hello World!")
3 </script>
```

Entretanto, nem sempre os navegadores dão suporte à execução de javascript. Assim, existe um elemento, chamado **<noscript>**, que tem como objetivo ser uma alternativa para essas situações. Ou seja, ele só será considerado/mostrado caso o navegador não dê suporte a javascript ou à opção de javascript esteja desabilitada. O elemento **<noscript>** pode conter qualquer dos elementos que são definidos

dentro do elemento **<body>**(que você estudará mais adiante, nesta mesma aula). O exemplo a seguir especifica uma mensagem indicando que não há suporte para javascript.

```
1 <script type="text/javascript">
2   document.write("Hello World!")
3 </script>
4 <noscript> Sorry, your browser does not support JavaScript! </noscript>
```

A tag <noscript> no HTML 5 pode ser adicionado dentro das tags <head> ou <body>, no HTML4 esta tag deve adicionada dentro de <body>.

Atenção!

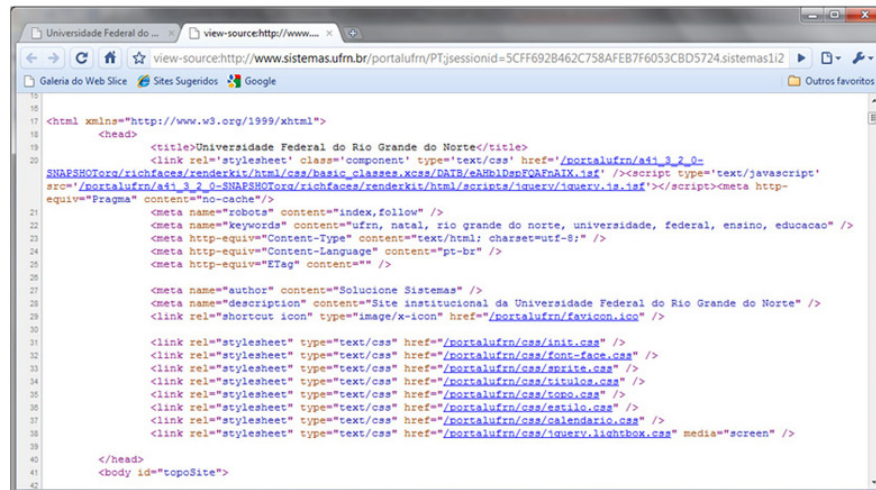
Ao testar os dois exemplos anteriores, sobre a tag <script> e <noscript>, perceba que seu browser mostrará apenas uma das seguintes mensagens: "Hello world" – caso o seu browser dê suporte a javascript – ou "Sorry, your browser does not support Java Script!", caso o seu browser não seja capaz de executar javascript.

Atividade 01

1. Veja a figura a seguir com a especificação do cabeçalho da página **<http://www.ufrn.br/>** e identifique as seguintes informações:

- o título da página
- palavras-chave
- idioma
- o nome do arquivo de script o nome de pelo menos um arquivo CSS usado nesse documento

Figura 07 - Atividade - Código HTML da página da UFRN



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Universidade Federal do Rio Grande do Norte</title>
<link rel="stylesheet" class="component" type="text/css" href="/portalufrn/s41_3_2_0-
SNAFSHOTorg/richfaces/renderkit/html/css/basic_classes.xcss/DATB/sAbldspFOAFnALK.jsf" /><script type="text/javascript"
src="/portalufrn/s41_3_2_0-SNAFSHOTorg/richfaces/renderkit/html/scripts/igwerv/igwerv.jsf"></script><meta http-
equiv="Pragma" content="no-cache"/>
<meta name="robots" content="index,follow" />
<meta name="keywords" content="ufrn, natal, rio grande do norte, universidade, federal, ensino, educacao" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8;" />
<meta http-equiv="Content-Language" content="pt-br" />
<meta http-equiv="ETag" content="" />
<meta name="author" content="Soluçõe Sistemas" />
<meta name="description" content="Site institucional da Universidade Federal do Rio Grande do Norte" />
<link rel="shortcut icon" type="image/x-icon" href="/portalufrn/favicon.ico" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/init.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/font-face.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/serif.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/titulos.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/sons.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/estilo.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/calendario.css" />
<link rel="stylesheet" type="text/css" href="/portalufrn/css/igwerv.lightbox.css" media="screen" />
</head>
<body id="topoSite">
```

2. Defina um cabeçalho contendo os seguintes elementos para o documento sobre você que começou a ser desenvolvido na aula passada.

- a. título
- b. autor
- c. palavras-chave

Agora, vamos começar a aprender os elementos que são definidos na segunda parte da tag <html>.

O corpo do documento –<body>...</body>

Como você viu anteriormente, os elementos definidos no cabeçalho, elemento <head>, indicam informações sobre o documento como um todo, mas não especificam o conteúdo do documento. Esse conteúdo deve ser especificado pelo elemento <body>.

Assim, o elemento <body> pode conter títulos de seção, parágrafos, listas, tabelas, links para outros documentos, imagens, formulários, animações, vídeos, sons e scripts embutidos. A seguir, começaremos pelos títulos de seção.

Atenção!

Em versões anteriores à HTML 4.01, a maioria dos elementos possuía atributos relacionados à forma como eles deveriam ser apresentados no browser. Então, existiam atributos tais como background, text e link que especificam, respectivamente, a cor do background, do texto e dos links, dentre outros. Tal como no exemplo a seguir.

```
1 <body bgcolor="#rrggb" text="#rrggb" link="#rrggb" alink="#rrggb" vlink="#rrggb"  
2 background="URL">
```

Nesse exemplo, BGCOLOR indica a cor de fundo (quando não é indicada, o browser irá mostrar uma cor padrão, geralmente o cinza ou o branco), TEXT especifica a cor dos textos da página (o padrão é preto), LINK indica a cor dos links (o padrão é azul), ALINK aponta a cor dos links quando acionados (o padrão é vermelho), e VLINK, a cor dos links depois de visitados (o padrão é azul escuro ou roxo).

Entretanto, com a necessidade de separar conteúdo de formatação (releia a aula anterior para relembrar os motivos), alguns desses atributos estão em desuso, chamados de Deprecated. Então, quando você vir nas especificações de HTML que um elemento ou atributo é Deprecated, trata-se de um elemento ou atributo defasado e substituído por novos elementos nas versões mais novas daquela especificação.

Sendo assim, na medida do possível, elementos e atributos Deprecated não serão estudados nesta disciplina, exceto a título de comparação.

Nesse curso, aprenderemos primeiramente quais os elementos e atributos principais de HTML para escrever conteúdo e somente na Aula 5 começaremos a estudar como formatar esses elementos para que tenham uma boa aparência no browser, através de CSS.

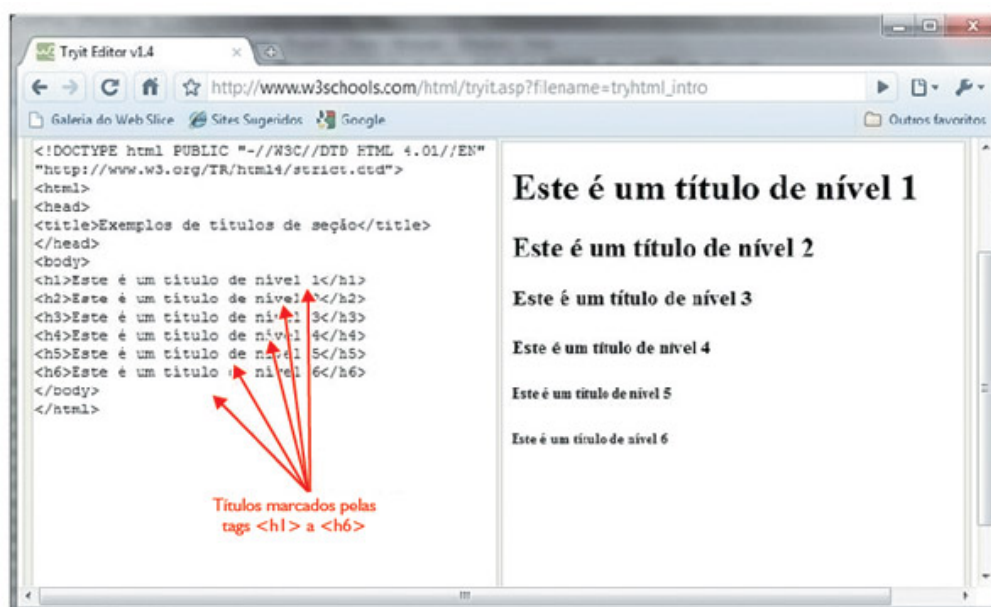
Os atributos de formatação no exemplo acima também foram removidos no HTML 5. A recomendação pela W3C é usar ao CSS para definir a apresentação de documentos escritos em uma linguagem de marcação.

Títulos: Elementos <h1>, <h2>, <h3>, <h4>, <h5> e <h6>

Há seis níveis de títulos em HTML, de <h1> (o mais importante) até <h6> (o menos importante). Um título descreve brevemente um tópico do conteúdo, ou seja, nomeia uma seção do conteúdo. No exemplo da **Figura 8**, lado esquerdo, você pode observar a especificação de 6 títulos, de h1 a h6, e na mesma figura, lado direito, você pode visualizar como esses títulos são mostrados no browser.

As tags de título <h1> - <h6>, no HTML4, possuem o atributo align. No entanto, este atributo de formatação foi removido no HTML 5.

Figura 08 - Exemplos de títulos de seção



Esses elementos não devem ser usados simplesmente para indicar que um texto deve aparecer grande ou em negrito. Eles devem ser usados para indicar o título e a estruturação de seu documento em seções. Detalhes sobre a formatação desses títulos você aprenderá quando estiver estudando CSS.

As tags de título <h1> - <h6>, no HTML4, possuem o atributo align. No entanto, este atributo de formatação foi removido no HTML 5.

Atividade 02

1. Crie duas seções no documento HTML sobre você, Dados pessoais e Dados profissionais, e em Dados pessoais crie subseções para Documentos e Endereço.

Separadores

Você já deve ter notado que as quebras de linha no texto HTML não são significativas na apresentação do documento no browser. Por exemplo, o texto da **Figura 9**, mesmo estando com várias quebras de linha do lado esquerdo, aparece sem nenhuma quebra de linha do lado direito.

Figura 09 - Exemplo do não uso de separadores





Vídeo 04 - Elementos de título

Mas então como realizar essa quebra de linha? Afinal de contas, um texto muito grande e contínuo é cansativo e pouco atrativo para os visitantes. Para organizar os textos, precisamos de separadores tais como a quebra de linha ou linhas horizontais.

Quebra de linha

Quando se deseja forçar uma quebra de linha em um determinado ponto do texto, deve-se usar o elemento `
` ou ainda `
</br>` (juntos). Isso só é necessário quando se deseja uma quebra de linha em determinado ponto, pois os browsers já quebram as linhas automaticamente (de acordo com o tamanho da janela) para apresentar os textos. Assim, se incluirmos o elemento `
` no exemplo anterior, teremos o seguinte resultado:

Figura 10 - Exemplo do uso da quebra de linha



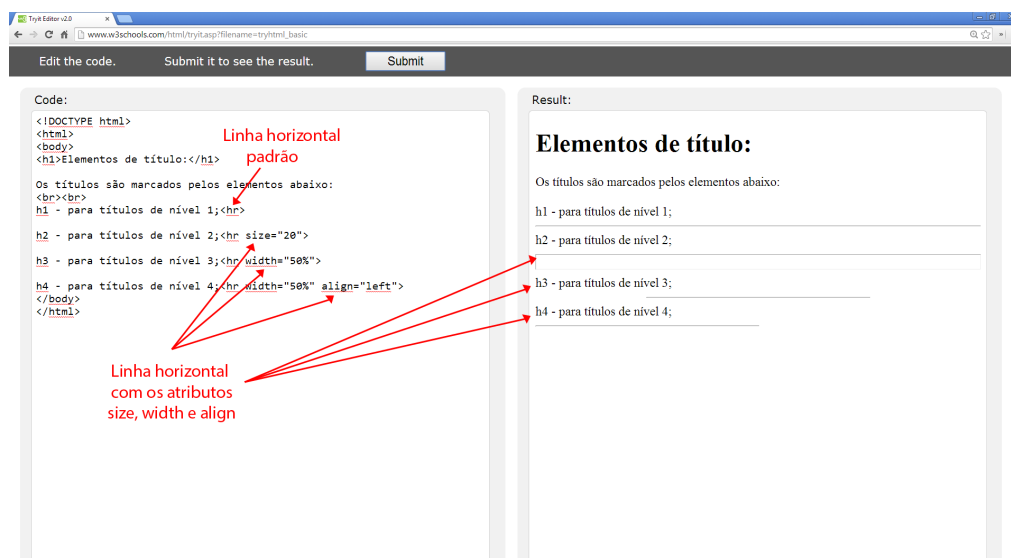
Note na **Figura 10** (lado esquerdo) que são usadas duas quebras de linha (tags `
`) seguidas para separar os textos. Isso tem o efeito ilustrado do lado direito.

Linha Horizontal

Outro tipo de separador é a linha horizontal, especificada pelo elemento vazio `<hr/>`. Esse elemento pode possuir alguns atributos, tais como `size` (indica a altura em [pixels](#) (É o menor ponto que forma uma imagem digital (como uma unidade de medida),)), `width` (indica a largura em percentual) e `align` (alinhamento horizontal, se centralizado, à direita ou à esquerda).

Na **Figura 11**, são ilustrados quatro exemplos de linhas horizontais, veja do lado direito a especificação de cada um dos elementos `<hr/>` e do lado esquerdo a saída no browser.

Figura 11 - Exemplo de linhas horizontais



Os atributos de formatação do `<hr>` foram removidos no HTML5, use CSS para definir tais propriedades. Além disso, a tag `<hr>` possui um significado semântico, ou seja, ela representa o fim de um tema abordado e o início de outro tema.

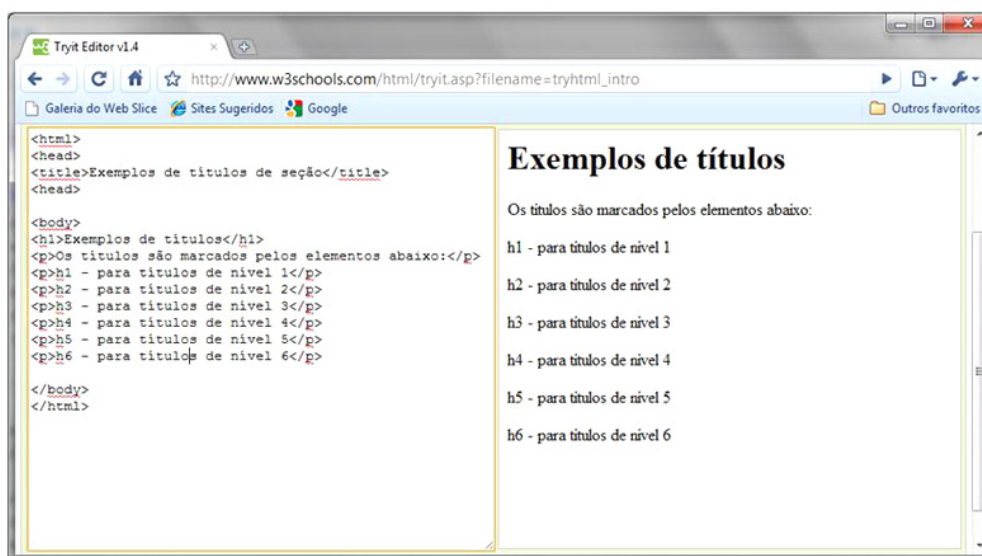
Atividade 03

1. No documento HTML sobre você, escolha um lugar apropriado e inclua uma linha horizontal, com os atributos `size="15"`, `width="70"` e `align="right"`.

Parágrafos

Para separar blocos de texto, deve ser utilizada a tag `<p>`. Os browsers automaticamente adicionam uma quebra de linha antes e depois de cada parágrafo. Assim, o exemplo da **Figura 10** poderia ter o mesmo efeito se escrito como mostrado na **Figura 12**.

Figura 12 - Exemplo de linhas horizontais



A importância de usar o elemento `<p>` ao invés de deixar o texto sem marcações está relacionada ao fato de que ele possibilita a definição dos atributos *id* e *class* para cada parágrafo e assim é possível formatá-los uniformemente com CSS. Você verá a definição desses atributos ainda nesta aula, mas veja primeiramente alguns elementos de formatação de [texto](#), tais como negrito, itálico, dentre outros.

| Tag | Descrição |
|---------|--|
| | Define texto em negrito |
| <u> | Define texto sublinhado |
| <s> | Define texto riscado |
| <i> | Define texto em itálico |
| <small> | Define texto em uma fonte de tamanho menor |
| <sub> | Define texto subescrito |
| <sup> | Define texto superescrito |

Quadro 2 -Tags de formatação de texto

É importante ressaltar que embora essas tags estejam, de certa forma, ligadas à aparência do texto, elas não são consideradas tags puramente de formatação, tais como aquelas que o padrão XHTML tenta abolir. Essas tags devem ser utilizadas para marcar o texto indicando sua importância no documento, por exemplo, o uso de negrito indica que se deseja chamar atenção para algumas palavras, o uso de itálico pode ser usado para indicar que o texto está em outro idioma ou que é uma citação para alguma referência bibliográfica, o uso de superescrito pode ser utilizado em fórmulas matemáticas indicando potenciação, dentre outros.

Além disso, perceba que essas tags não informam tudo sobre a aparência, ainda é possível formatar o texto quanto ao tipo de fonte, tamanho, cor, alinhamento, dentre outros. São essas as características que você aprenderá a formatar por meio do CSS. Como dissemos anteriormente, a definição dos atributos id e class serão essenciais para isso. Mas que atributos são esses? Id e class são chamados de identificadores de elemento, veja a seguir.

No HTML 5 o atributo **align** da tag <p> foi removida. Use CSS para formatar o alinhamento.

Identificadores de elemento: os atributos id e class

O atributo id define um nome para o elemento que o contém, esse nome deve ser único em um documento, ou seja, não pode haver dois elementos com nomes iguais num mesmo documento. Já o atributo class especifica o nome de uma classe para um elemento.

Assim, vários elementos podem possuir a mesma classe e um mesmo elemento pode possuir várias classes, separadas por espaço.

O principal objetivo desses atributos é funcionar como seletor no CSS, ou seja, identificar quais elementos devem ser estilizados por uma regra CSS. Veja um exemplo bem simples a seguir [w3c]. Não se preocupe com as tags que você ainda não estudou.

```
1 <p><span id="msg1" class="info" lang="en">Variable declared twice</span></p>
2 <p><span id="msg2" class="warning" lang="en">Undeclared variable</span></p>
3 <p><span id="msg3" class="error" lang="en">Bad syntax for variable name</span></p>
```

Nesse exemplo, foram definidas três classes, nomeadas de info, warning e error. O seguinte fragmento CSS permite que essas mensagens sejam exibidas em três cores diferentes: verde, se for da classe info, amarela, se for da classe warning e vermelha se for da classe error.

```
1 span.info { color: green }
2 span.warning { color: yellow }
3 span.error { color: red }
```

Os elementos <div> e juntamente com os atributos id e class oferecem mecanismos para adicionar estrutura ao documento HTML. Elas são tags de agrupamento, usadas para agrupar informações que terão uma mesma configuração. O elemento é utilizado para agrupar informações dentro de um mesmo grupo e <div> para informações de blocos diferentes. Não se preocupe, esse assunto será revisto em detalhes quando você estiver aprendendo CSS.

Estes dois atributos, id e class, são chamados de atributos globais, eles podem ser utilizados em qualquer elemento HTML. Outros exemplos de atributos globais são: *accesskey*, *style*, *tabindex* e *title*.

Caracteres especiais

Em HTML, para escrever caracteres especiais no conteúdo do documento, tais como os símbolos de maior e menor, é necessário usar sequências de escape. As sequências de escape são compostas de três partes: um "&" inicial, um número ou cadeia de caracteres correspondente ao caractere desejado, e um ";" final, veja o exemplo a seguir.

Figura 13 - Sequência de escape para representar o símbolo &

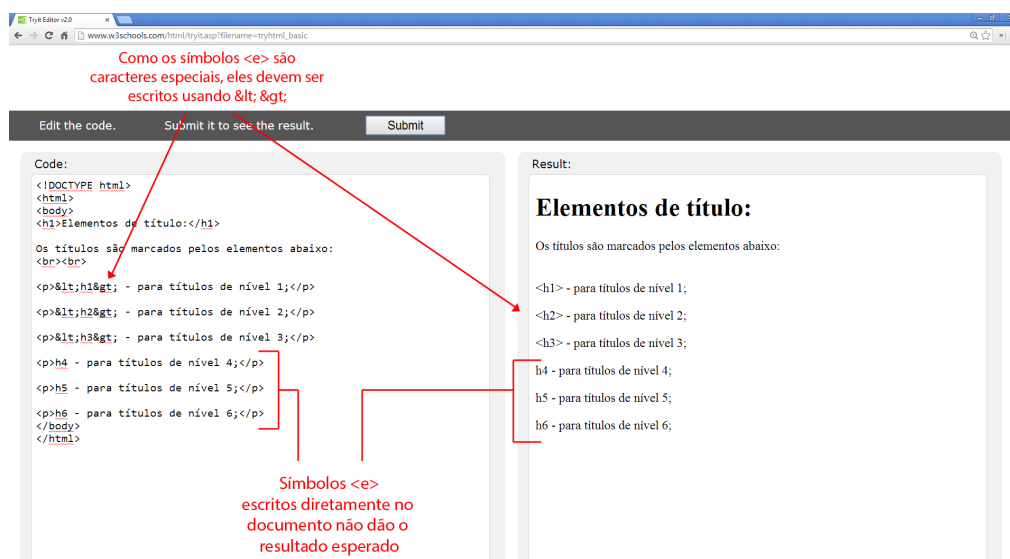


Por exemplo, os caracteres "<", ">", e "&" possuem um significado especial em HTML, então, se for necessário escrevê-los no conteúdo de um documento HTML, é necessário utilizar essa forma de escrita. Veja na tabela a seguir quais são os códigos correspondentes a esses [símbolos](#).

| Entidade | Caractere |
|----------|-----------|
| < | < |
| > | > |
| & | & |

Assim, para escrever esses símbolos, você precisa usar as respectivas entidades. Na **Figura 14**, há um exemplo ilustrando a necessidade do uso desses caracteres especiais.

Figura 14 - Exemplo de uso de caracteres especiais



Links

Os links constituem o elemento que possibilita uma das melhores características do hipertexto e da web: **a navegação**. Os links permitem a ligação de uma região de texto (ou imagem) a outro documento. Para especificar links, é necessário usar o elemento <a>. A tag <a> pode ser usada de duas maneiras:

1. usando o atributo **href** para criar um link para algum (outro) documento;
2. usando o atributo **name** para criar uma marcação dentro de um documento para o qual algum link apontará.

Um exemplo simples do primeiro caso é ilustrado a seguir. Nesse exemplo, a tag <a> com o atributo href está indicando que a palavra W3C é um link para o endereço <<http://www.w3c.org>>. O conteúdo entre as tags <a> e não tem que ser texto, pode ser uma imagem ou outros elementos HTML.

| | |
|---|--------------------------------------|
| 1 | W3C |
|---|--------------------------------------|

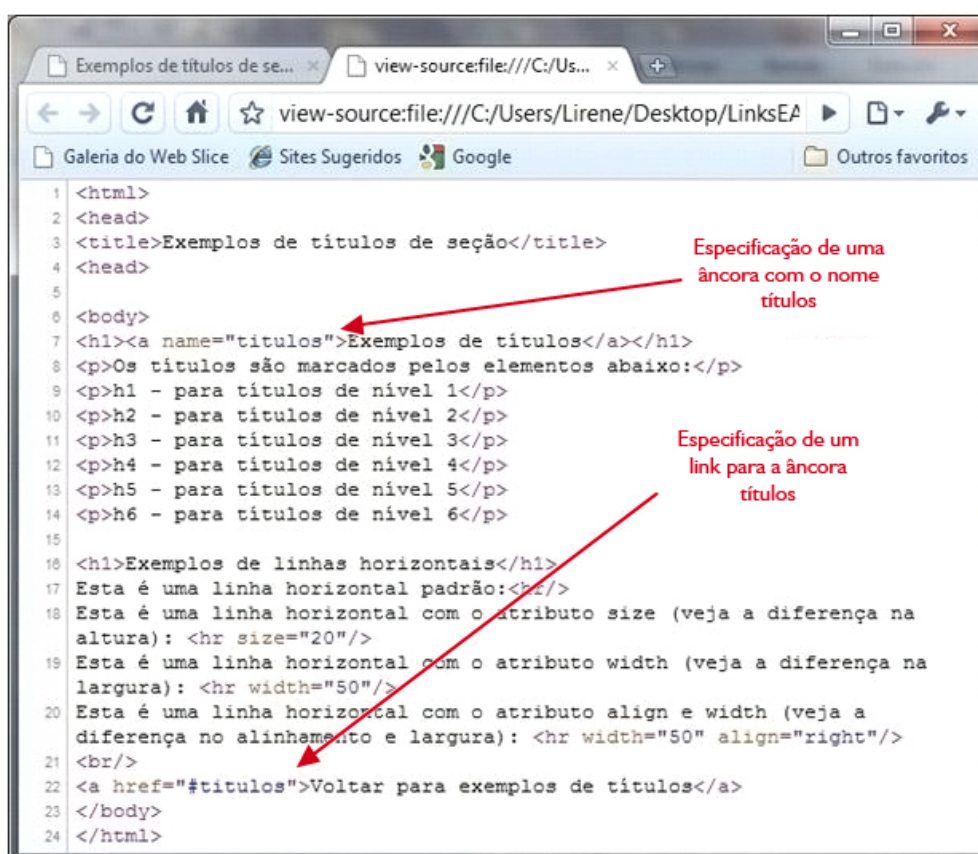
Juntamente com o atributo href pode-se usar o atributo target. O atributo **target** indica onde o documento especificado em href deve ser aberto, se na mesma janela do documento corrente (esse é o padrão) ou se em outra janela, por meio do valor _blank.

```
1 <a href="http://www.w3c.org" target="_blank">W3C</a>
```

Por outro lado, o atributo **name** especifica o que se chama de âncora, ou seja, um ponto dentro de um documento para onde um certo link deve apontar (ancorar). Diferentemente dos links que são geralmente mostrados sublinhados, as âncoras não possuem uma aparência diferenciada. Elas são invisíveis para o usuário. Na **Figura 15**, há uma ilustração de como uma âncora é especificada.

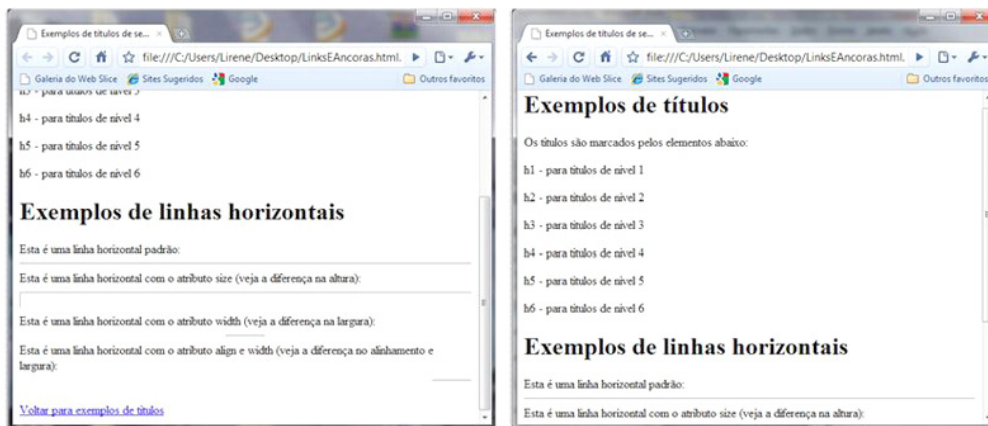
A utilidade de se definir âncoras é que ao clicar nos links o navegador posiciona o documento exatamente no ponto em que a âncora determina. Como exemplo, veja a **Figura 16**, nela o navegador mostra uma parte do documento onde aparece um link. Ao clicar nesse link, o navegador posiciona o documento no ponto marcado pela âncora correspondente, que nesse caso é no início do mesmo documento.

Figura 15 - Exemplo de link e âncora



No HTML5 o atributo **name** não é mais suportado. Use o atributo global **id** em seu lugar.

Figura 16 - Browser mostrando a utilidade de se especificar âncoras: lado esquerdo – ilustração de um link, e lado direito – ilustração do comportamento do browser quando o link foi clicado



Se o link apontar para outro documento, então é necessário especificar o URI completo adicionado do símbolo # e o nome da âncora.

Como foi dito anteriormente, é a utilização de links que possibilita a navegação, um dos recursos mais importantes da web. Agora, você está pronto para construir seu primeiro documento HTML contendo navegação. Por enquanto, ele conterá apenas textos e links, mas, como foi visto, você já pode organizar esse texto em diferentes níveis de seções, bem como enfatizá-lo de diferentes maneiras através das tags de formatação para negrito e itálico, dentre outras.

Na próxima aula, você aprenderá a torná-lo mais interessante, através da inclusão de figuras, listas, tabelas e formulários, bem como aprenderá a definir frames.

Atividade 04

1. Inclua na página sobre você uma seção de sites que você acha interessante. Cada item dessa seção deve ter um link para tal documento.
2. Escreva um documento HTML correspondente a seguinte saída no browser.

Figura 17 - Atividade – A arte da guerra

A Arte da Guerra (Sun Tzu) x

file:///C:/a_arte_da_guerra.html

🔍 ⚙ 📄

A Arte da Guerra (Sun Tzu)

Definição

A Arte da Guerra é um tratado militar escrito durante o século IV a.C. pelo estrategista conhecido como Sun Tzu. O tratado é composto por [seze capítulos](#), cada qual abordando um aspecto da estratégia de guerra, de modo a compor um panorama de todos os eventos e estratégias que devem ser abordados em um combate racional. Acredita-se que o livro tenha sido usado por diversos estrategistas militares através da história como Napoleão, Zhuge Liang, Cao Cao, Takeda Shingen, Vo Nguyen Giap e Mao Tse Tung.

Hoje, A Arte da Guerra parece destinado a secundar outra guerra: a das empresas no mundo dos negócios. Leia no [Wikipédia](#) mais informações.

Alguns pensamentos

"A suprema arte da guerra é derrotar o inimigo sem lutar."

"A suprema arte da guerra consiste em vencer o inimigo sem ter que enfrentá-lo."

"As oportunidades multiplicam-se à medida que são agarradas."

"Diante de uma larga frente de batalha, procure o ponto mais fraco e, ali, ataque com a sua maior força."

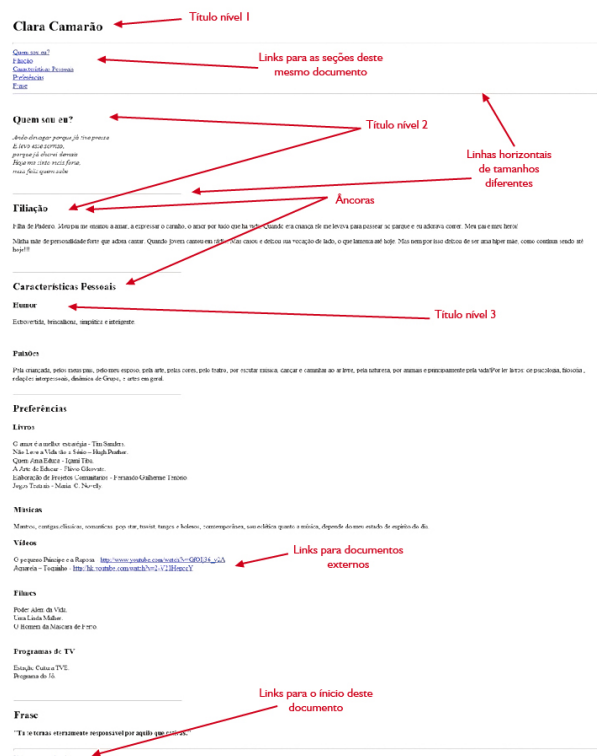
Capítulos

1. Planejamento Inicial
2. Guerreando
3. Estratégia ofensiva
4. Disposições
5. Energia
6. Fraquezas e forças
7. Manobras
8. As nove variáveis
9. Movimentações
10. Terreno
11. As nove variáveis de terreno
12. Ataques com o emprego de fogo
13. Utilização de agentes secretos

Nesta aula, você começou a aprender HTML 4.01. Você conheceu a estrutura básica de um documento HTML e aprendeu como incluir cabeçalhos, títulos, links, parágrafos e separadores em um documento, bem como alguns elementos de formatação, tais como negrito, itálico, sublinhado, dentre outros.

1. Escreva um documento HTML correspondente à seguinte saída mostrada no browser. O exercício consiste na elaboração de um perfil pessoal. Você deverá substituir as informações pessoais mostradas na página por suas informações, porém, será necessário seguir a estrutura (títulos, links, formatação etc.) mostrada no documento a seguir. Os comentários na cor vermelha são apenas para lembrá-los de alguns itens.

Figura 18 - Atividade – Clara Camarão



2. Para que servem os seguintes elementos? Mostre um exemplo de cada um deles no documento que você acabou de escrever na questão anterior.

1. Head

2. Body

3. Title

4. Meta

5. P

6. Br

7. Hr

Referências

W3C. Disponível em: <<http://www.w3.org/>>. Acesso em: 5 set. 2014.

W3SCHOOL. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 5 set. 2014.