



História

HTML

HTML é uma abreviação de Hypertext Markup Language - Linguagem de Marcação de Hipertexto. É uma linguagem de marcação destinada a estruturar conteúdos de diversos tipos, como texto, imagens, vídeos e sons. É a linguagem presente sempre que visualizamos conteúdos na internet.

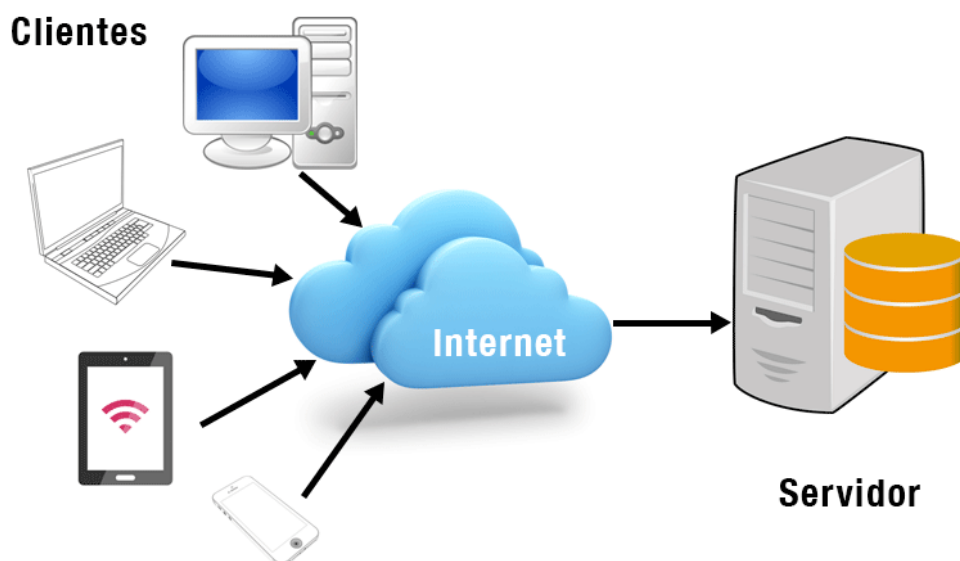
O HTML é baseado no conceito de Hipertexto. Hipertexto são conjuntos de elementos ou nós ligados por conexões. Estes elementos podem ser palavras, imagens, vídeos, áudio, documentos etc. Estes elementos conectados formam uma grande rede de informação. Eles não estão conectados linearmente como se fossem textos de um livro, onde um assunto é ligado ao outro seguidamente. A conexão feita em um hipertexto é algo imprevisível que permite a comunicação de dados, organizando conhecimentos e guardando informações relacionadas.

Uma página de internet é um arquivo de texto, em geral, escrito na linguagem HTML. Ele deve ficar armazenado em um computador que deve estar ligado o tempo todo e conectado à internet. A esse computador damos o nome de "servidor".

Outros computadores (chamados de clientes) podem acessar uma determinada página através de um navegador, inserindo um endereço (ou domínio). Esse endereço indicará na internet o servidor que armazena a página. Assim, o servidor enviará a página para o cliente.

Para distribuir informação de uma maneira global, é necessário haver uma linguagem que seja entendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem.

Desenvolvido originalmente por Tim Berners-Lee o HTML ganhou popularidade quando o Mosaic - browser desenvolvido por Marc Andreessen na década de 1990 - ganhou força. A partir daí, desenvolvedores e fabricantes de browsers utilizaram o HTML como base, compartilhando as mesmas convenções.



Protocolo HTTP

HTTP é a sigla em língua inglesa de HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto), um protocolo da camada de Aplicação do modelo OSI utilizado para transferência de dados na rede mundial de computadores, a World Wide Web. Normalmente, este protocolo utiliza a porta 80 e é usado para a comunicação de "sites".

O protocolo HTTP é baseado em requisições e respostas entre clientes e servidores. O cliente — navegador ou dispositivo que fará a requisição; também é conhecido como *user agent* — solicita um determinado recurso (resource), enviando um pacote de informações contendo alguns cabeçalhos (headers) a um URI ou, mais especificamente, URL. O servidor recebe estas informações e envia uma resposta, que pode ser um recurso ou um simplesmente um outro cabeçalho.

O HTML5 é a mais recente versão da linguagem HTML. Ela trouxe novas funcionalidades e perspectivas em relação ao papel exercido pelo HTML na Web. O HTML5 traz novas tags, como `<header>` para cabeçalhos, `<nav>` para menus, `<video>` e `<audio>` para respectivos elementos, além de apresentar uma sintaxe mais simples como, por exemplo, para declaração do DOCTYPE.

Uma página HTML é composta por alguns elementos básicos. São eles:



- I. a declaração do tipo do documento;
- II. a tag <HTML>, que armazenará todo conteúdo da página;
- III. o cabeçalho <HEAD>, que armazena informações estruturais da página; e
- IV. o corpo <BODY>, que armazenará os conteúdos e formatações da página.

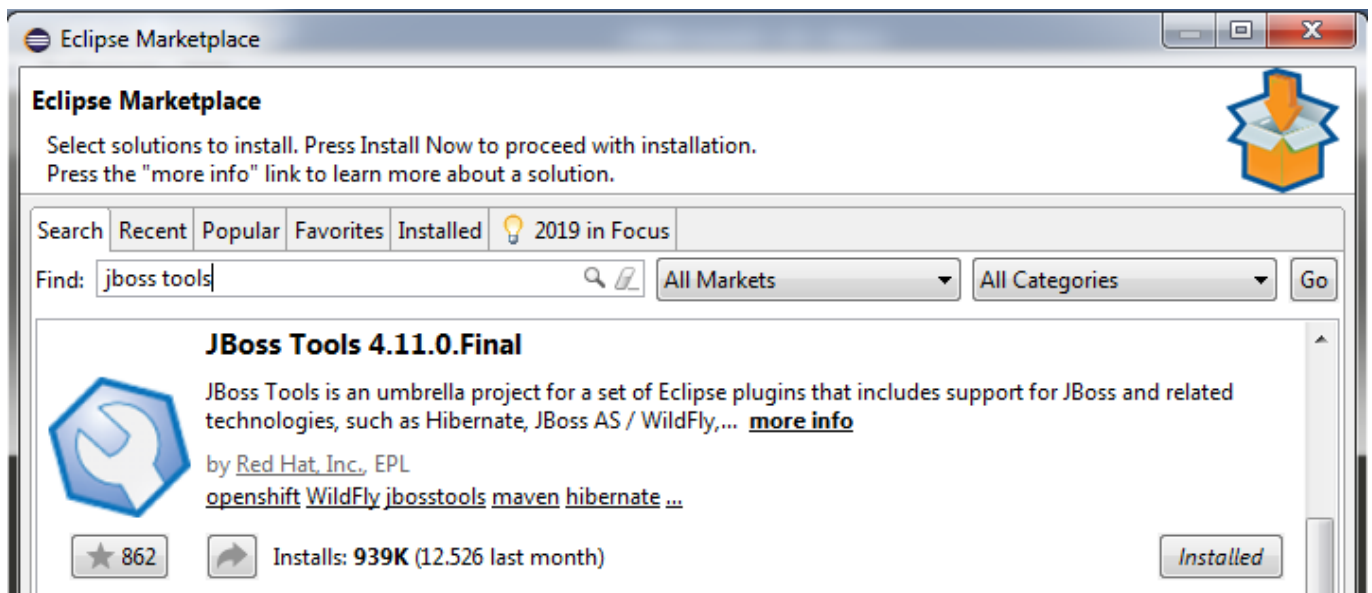
Para as nossas aulas, vamos padronizar o uso dos seguintes programas.

- Editor de HTML: **Visual Studio Code**, Bracket ou Suplme text3
- Navegador de Internet: Firefox ou Mozilla
- Servidor de WEB (opcional – pode usar o XAMPP ou qualquer servidor WEB)

Também podemos utilizar os seguintes programas:

- NotePad++: Editor de código
- Eclipse com Jboss Tools (versão): Editor de HTML

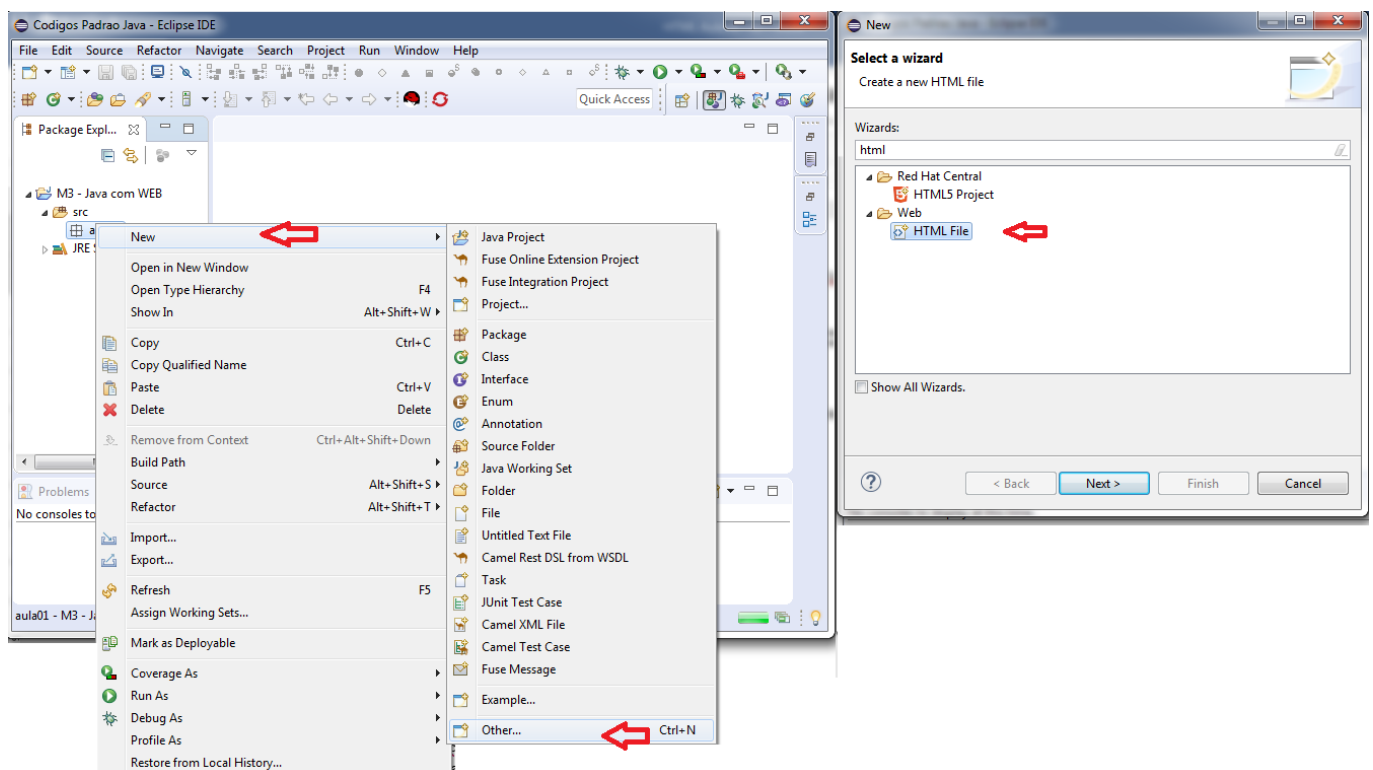
Instalando JBoss no Eclipse.





Trabalhando no Eclipse. (Não necessário por enquanto)

- Instale o JDK 1.11;
- Baixe e Execute o Eclipse mais atual. Sugiro que baixar o compactado(zipado).
- Entre no Marketplace, localize e instale o JBoss Tools 4.x.x.final.
- No eclipse, crie um projeto (localizer **“static web projet”**). Dentro deste projeto, vamos organizar nossas aulas usando o recurso de pastas ou pacotes. Ex: aula01.
- Para criar um projeto vá em NEW >>> OUTROS >>> WEB>>> STATIC WEB PROJetc.
- Dentro de cada pacote/pasta, crie suas páginas em HTML. Clique com o botão direito sobre o pacote. Clique em NEW >>> OUTROS >>> filtre ou localize HTML FILE.
- É opcional, mas recomendo para organizar nossos trabalhos, criar pastas para cada aula.

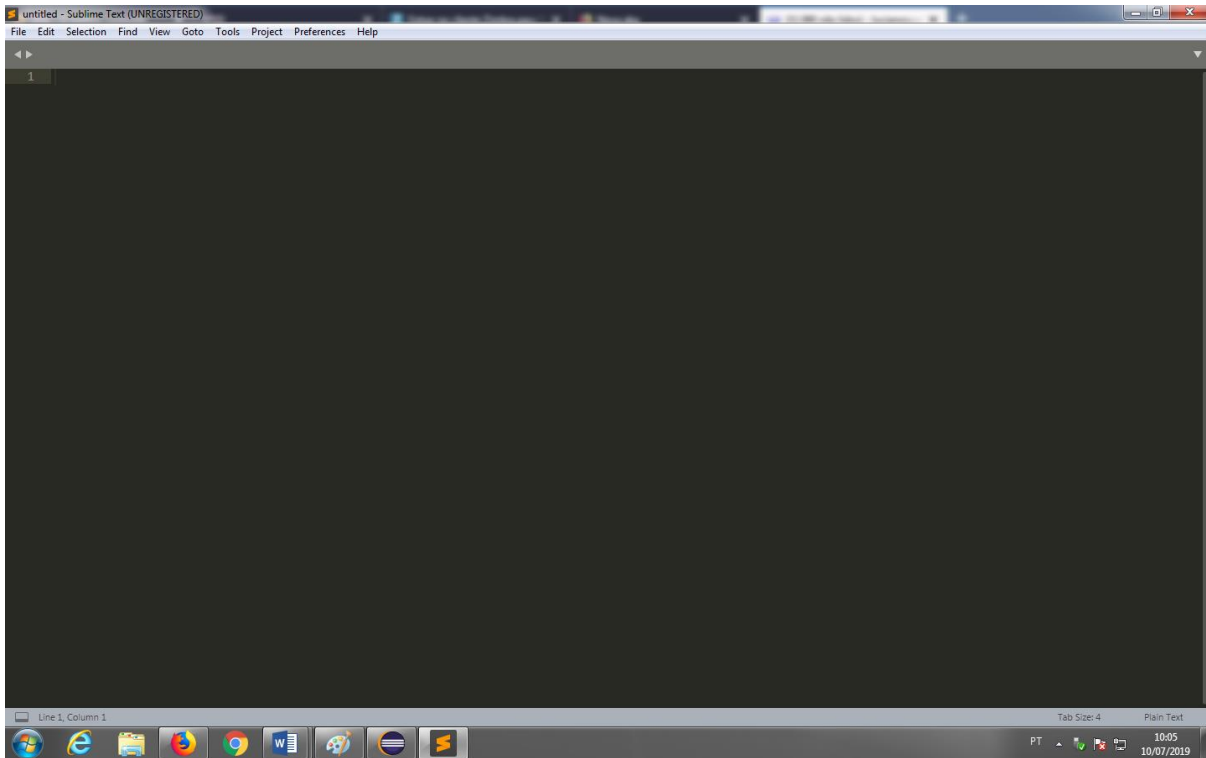


Ao trabalhar nossos exemplos com outro tipo de editor web, localize onde está o projeto Java com a nossas aulas. A pasta raiz será webcontent, que estará entro do projeto Java Criado.



Instalando o programa SUBLIME TEXT3

<https://www.sublimetext.com/3>



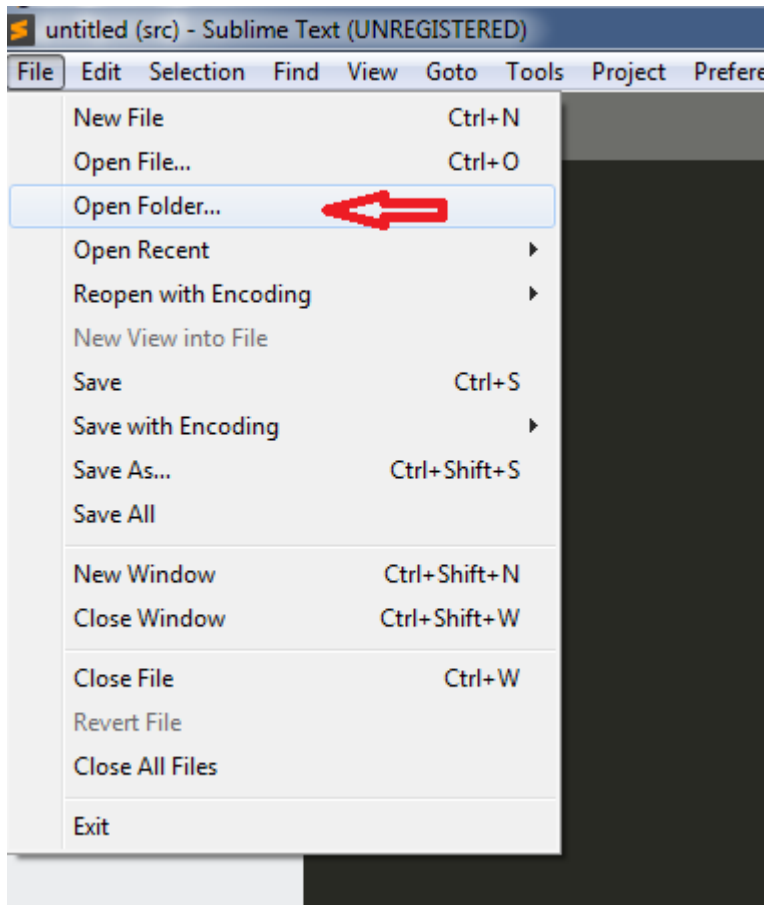
Defina o Local de trabalho.

Crie uma pasta para armazenar as nossas HTMLs.

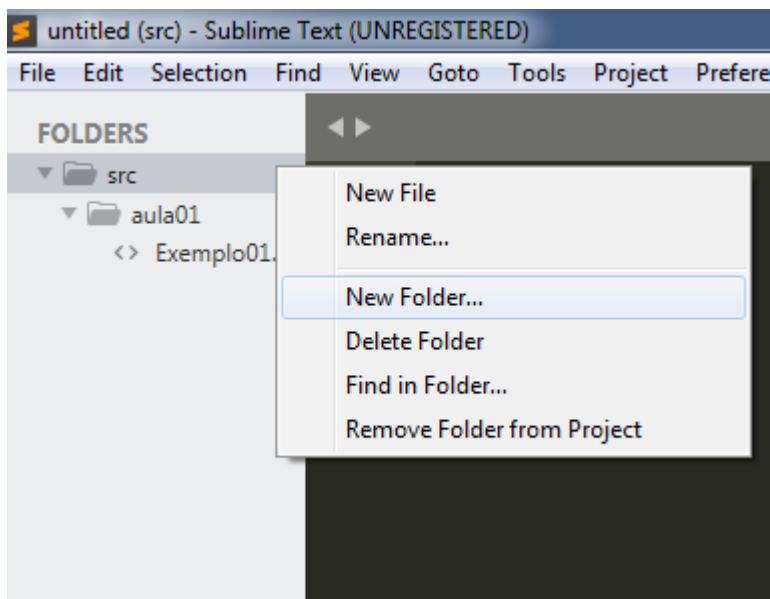
Dica: Se você deseja armazenar junto com o projeto Java existente. Basta abrir o diretório do “src” de seu projeto.

Clique em File >>> OPEN FOLDER e selecione o diretório onde serão armazenados suas HTMLs.

Outra opção, caso você tenha um servidor de WEB é salvar diretamente nele.



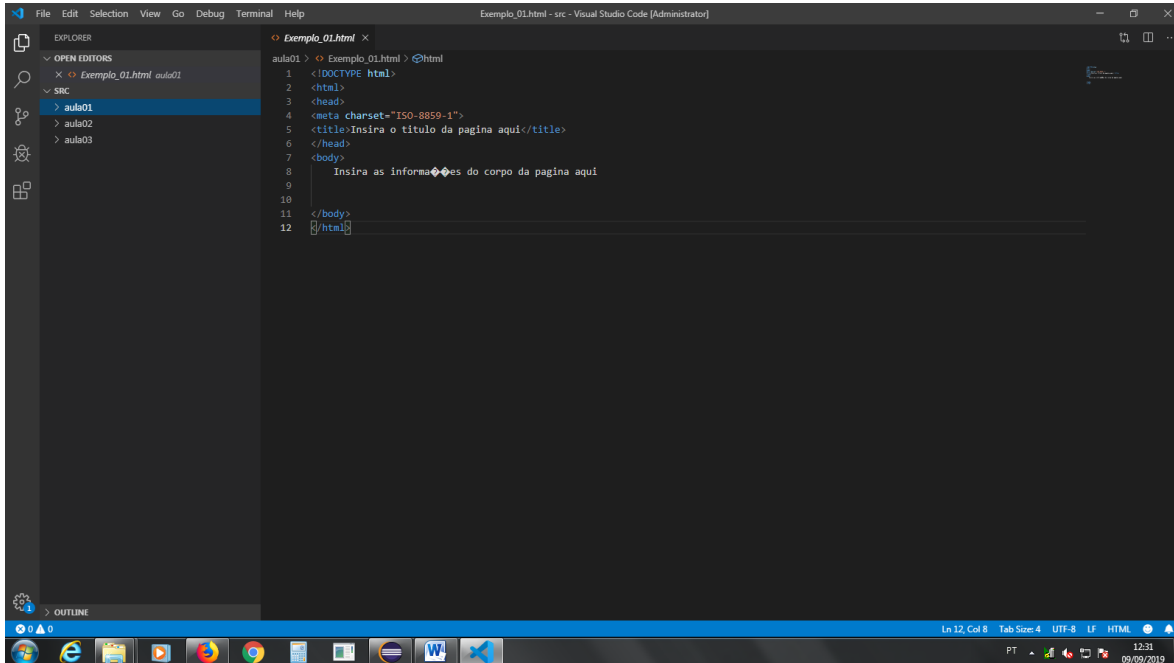
Dentro do SUBLIME, crie novos FOLDER, para organizar as aulas e os exemplos em HTML.



Dica: Usar a codificação *ISO-8859-1* ou *UTF-8*. (de preferencia utf-8)



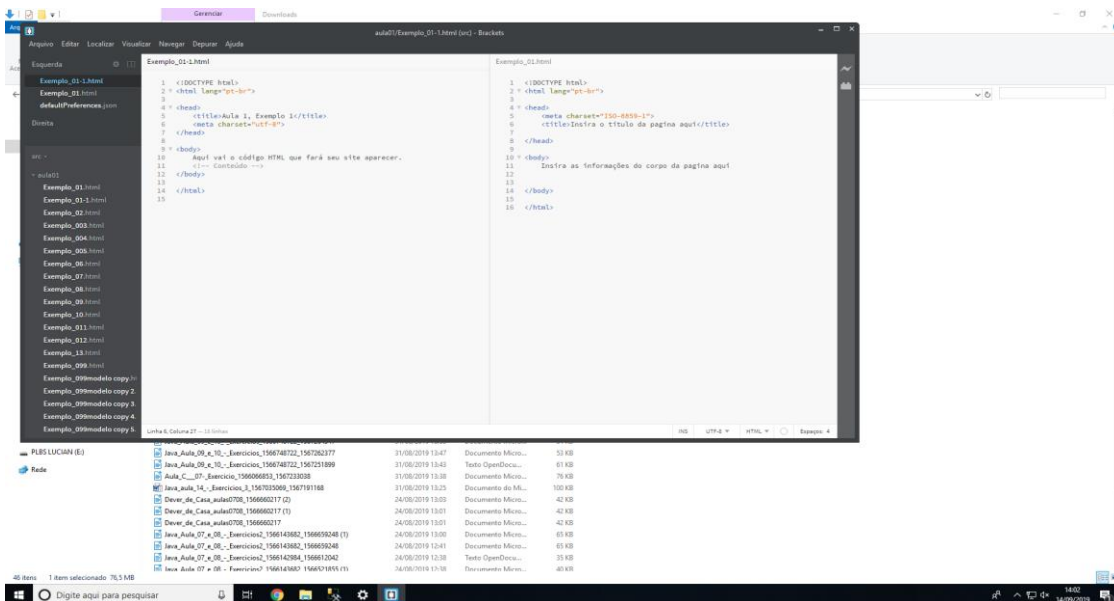
Instalando o programa VISUAL STUDIO CODE



Baixe o programa “Visual Studio Code”. Instale. Após a instalação abra-o e mande abrir a pasta (folder) onde concentra-se seus trabalhos.

Obs: O atalho para indentar no Visual Studio Code pelo Windows é Shift + Alt + F.

Outra opção é o programa **BRACKET**





O HTML5 e suas mudanças

Quando o HTML4 foi lançado, o W3C (World Wide Web Consortium) alertou os desenvolvedores sobre algumas boas práticas que deveriam ser seguidas ao produzir códigos client-side. Assuntos como a separação da estrutura do código com a formatação e princípios de acessibilidade foram trazidos para discussões e à atenção dos fabricantes e desenvolvedores.

O World Wide Web Consortium (W3C) é uma comunidade internacional que desenvolve padrões abertos para garantir o crescimento a longo prazo da Web.

Lado cliente (do inglês **client-side**), também conhecido como front-end refere-se às operações que são realizadas pelo cliente em um relacionamento cliente-servidor em uma rede de computadores.

Voltando ao assunto, o HTML4 ainda não trazia diferencial real para a semântica do código. O HTML4 também não facilitava a manipulação dos elementos via Javascript ou CSS. Se você quisesse criar um sistema com a possibilidade de Drag'n Drop de elementos, era necessário criar um grande script, com bugs e que muitas vezes não funcionavam de acordo em todos os browsers.

Arrastar e soltar (em Inglês: ***Drag-and-drop***), nomenclatura utilizada nas interfaces gráficas de computadores, é a ação de clicar em um objeto virtual e "arrastá-lo" a uma posição diferente ou sobre um outro objeto virtual. De maneira geral, ele pode ser usado para invocar diversos tipos de ações, ou criar vários tipos de associações entre dois objetos abstratos

O que é o HTML5?

O HTML5 é a nova versão do HTML4. Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final.

Ao contrário das versões anteriores, o HTML5 fornece ferramentas para a CSS e o Javascript fazerem seu trabalho da melhor maneira possível. O HTML5 permite por meio de suas APIs a manipulação das características destes elementos, de forma que o website ou a aplicação continue leve e funcional.



O HTML5 também criou novas tags e modificou as funções de outras. As versões antigas do HTML não continham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, menu e etc. Não havia um padrão de nomenclatura de IDs, Classes, tags e outros.

Motores de Renderização

Há uma grande diversidade de dispositivos que acessam a internet. Entre eles, há uma série de tablets, smartphones, pc's e etc. Cada um destes meios de acesso utilizam um determinado browser para navegar na web. Não há como os desenvolvedores manterem um bom nível de compatibilidade com todos estes browsers levando em consideração a particularidade de cada um. Uma maneira mais segura de manter o código compatível, é nivelar o desenvolvimento pelos motores de renderização. Cada browser utiliza um motor de renderização que é responsável pelo processamento do código da página.

Abaixo, segue uma lista dos principais browsers e seus motores:

Motor	Browser
Webkit	Safari, Google Chrome
Gecko	Firefox, Mozilla, Camino
Trident	Internet Explorer 4 ao 9
Presto	Opera 7 ao 10

É interessante que façamos códigos compatíveis com estes motores. Focando a compatibilidade nos motores de renderização você atingirá uma amplitude maior de browsers.

Por exemplo, se seu código funcionar no Webkit, você alcançará o Safari e o Chrome, dois dos principais browsers do mercado para desktops. Além disso, você também alcança aparelhos como Blackberry, iPhone, iPod Touch, iPad e dispositivos que rodam Android.



Professor: Luciano

Módulo HTML/CSS

Aula 01 e 02

Sintaxe para declaração de tag

<nome_tag> conteúdo **</nome_tag>**

Leia-se: Tag de abertura, conteúdo e tag de fechamento

<nome_tag atributos > conteúdo **</nome_tag>**

Leia-se: Tag de abertura com declaração de atributos, conteúdo e tag de fechamento.

Obs: Algumas tags não necessitam de tag de fechamento



Estrutura básica do HTML

A estrutura básica do HTML5 continua sendo a mesma das versões anteriores da linguagem, há apenas uma exceção na escrita do Doctype. Ao criamos uma pagina HTML será criado um arquivo HTML com a seguinte estrutura básica:

```
<!DOCTYPE html>
<html>

<head> </head>

<body> </body>

</html>
```

Usando fundo preto

```
<!DOCTYPE html>

<html>

    <head> </head>

    <body> </body>

</html>
```

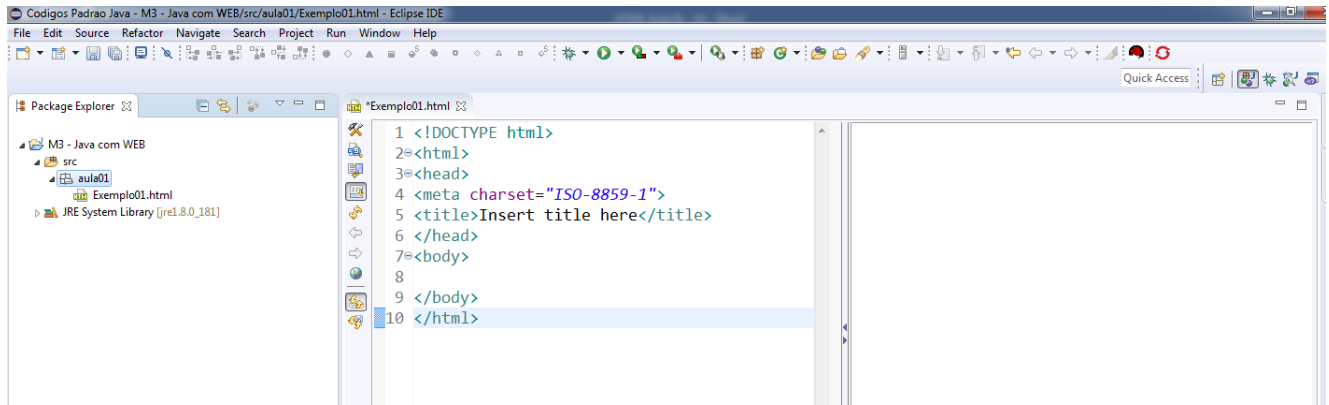
Usando fundo branco

Obs: O recurso de abrir o HTML no eclipse, será usado nos módulos/aulas mais à frente. Por enquanto nos códigos serão criados usando o **Visual Studio Code**, **Bracket** ou **SUBLIME TEXT3**.

Dica: Nos editores de HTML, use a mesma filosofia de organização, criando pastas para cada aula. Caso tenha algum projeto criado, aponte para a pasta ContentWeb do projeto ou SRC se houver.



Estrutura padrão



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Insira o título da página aqui</title>
</head>
<body>

</body>
</html>
```

Obs:

Metatag Charset

No nosso exemplo há uma metatag responsável por chavear qual tabela de caracteres a página está utilizando.

```
<meta charset="utf-8">
```

Nas versões anteriores ao HTML5, essa tag era escrita da forma abaixo:

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

Essa forma antiga será também suportada no HTML5. Contudo, é melhor que você utilize a nova forma.



Legenda:

Definição do documento (doctype)

Cabeça (head)

Corpo (body)

Doctype - Definindo o documento

SEMPRE deve existir o doctype, que é este código **<!DOCTYPE html>**.

O Doctype não é uma tag do HTML, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita. Desta forma eles sabem o que fazer para mostrar seu site da melhor forma possível. Lembre-se: o doctype é **OBRIGATÓRIO** e deve ser sempre a PRIMEIRA LINHA do seu documento.

HEAD

Contém informações que não são transpostas visivelmente para o usuário/leitor do documento. São dados implícitos de uso e controle do documento: vinculação com outros arquivos, aplicação de lógica de programação de scripts e metadados. Na prática, todo o conteúdo do cabeçalho fica delimitado entre a abertura e fechamento tag head.

A Tag HEAD é onde fica toda a parte informativa da página. No HEAD ficam os metadados. Metadados são informações sobre a página e o conteúdo ali publicado. O cabeçalho do documento é delimitado pelo par de tags <head> e </head>.

BODY

Trata-se do documento em si, ou seja, a informação legível para o usuário/leitor do documento. É todo e qualquer texto que se deseja apresentar, assim como toda e qualquer forma de mídia de saída (imagens, sons, mini aplicativos embutidos, conteúdo multimídia, etc). Além disso, toda a apresentação de entrada de dados (formulários) também se aplica nesta seção do documento. Na prática, o corpo do documento é delimitado pelo par de tags <body> e </body>.

Este é o preceito básico que deve estar muito bem claro: onde as marcações se aplicam, e quais são os resultados deste modelo. Por exemplo: se você deseja informar conteúdo textual para saída



legível ao usuário do seu sistema web, esta marcação deverá obrigatoriamente estar no bloco do corpo (body) da página. Ainda, para definir qual o tipo de codificação da página (uma meta informação do documento), esta deve obrigatoriamente estar marcada no cabeçalho do mesmo documento.

Dentro do elemento BODY sua estrutura de página terá os elementos semânticos da construção da sua página, onde serão declarados e identificados cabeçalhos, rodapé, conteúdo principal, etc.



Descrição dos componentes básicos do HTML

- **Tag** – simbolizam os comandos de marcação no HTML. Cada Tag representa uma funcionalidade. Tags são representadas desta forma: <tag> (sinal de menor, nome_tag, sinal de maior). A grande maioria das tags necessitam de tag de fechamento.

Uma tag de fechamento é simbolizado por </ tag>. (sinal de menor, barra, nome_tag, sinal maior).

- **Element** - é uma tag completa, tendo uma abertura <tag> conteúdo </ tag>.
- **Attribut** - é usada para mudar o valor de um elemento em HTML. Geralmente um elemento tem vários atributos.

Obs: Todas as páginas de web tem pelo menos os elementos básicos: **html, head e body**. Toda Tags tem três partes, como foi dito anteriormente, a abertura, o fechamento e o conteúdo.

Obs: Sintaxe de declaração de elemento:

<tag_abertura> conteúdo </tag_fechamento>

O nome da tag de abertura e de fechamento devem ser o mesmo. A diferença entre a tag de abertura e a tag de fechamento é que a tag de abertura consiste do nome da tag entre < > e a tag de fechamento consiste no mesmo nome da tag de abertura entre o sinal de </ >.

HTML

Todo documento HTML sempre começará com a tag <html> e finalizará com </html>. Essa é a estrutura padrão de um HTML. *

Sintaxe:

<html>

</html>

No HTML5

<!DOCTYPE html>

<html>

</html>



HEAD

O elemento `<head>` é o que vem em seguida. Ele é colocado entre a tag *html* e *body*. *Head* ou "**Cabeçalho**" não tem nenhuma função visível, mas oferece ao navegador informações úteis como outros códigos do tipo JavaScript , CSS e etc.

Sintaxe:

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

```
    <head>
```

```
    </head>
```

```
</html>
```

TITLE

Title (título) fica dentro da tag de cabeçalho (head). Este coloca o título na pagina do navegador.

Sintaxe:

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

```
    <head>
```

```
        <title>Minha primeira página</title>
```

```
    </head>
```

```
</html>
```


**BODY**

O elemento body (corpo) é o que define o início do conteúdo da página em si (títulos, parágrafos, fotos, músicas e qualquer outro conteúdo)

Sintaxe:

```
<html>

    <head>
        <title>Minha primeira página</title>

    </head>

    <body>

        Olá Alunos da Step!

    </body>

</html>
```

Vamos usar o padrão base de html5, conforme modelo abaixo

```
<!DOCTYPE html>
<html lang="pt-br">

    <head>
        <meta charset="utf-8">
        <title>Minha primeira página</title>

    </head>

    <body>
        Olá Alunos da Step!

    </body>

</html>
```



ATRIBUTOS

O que são Atributos?

Atributos são informações que passamos na Tag para que ela se comporte da maneira esperada. Existem atributos globais, que funcionam em todas as Tags e específicos, que são direcionados para cada Tag, através de especificação.

Os Atributos possuem nome e um valor. Existem Atributos que você irá usar praticamente sempre e existem outros que serão mais raros.

- **Atributos Globais** são atributos comuns a todos elementos HTML; eles podem ser usados em todos os elementos, embora os atributos não tenham efeito em alguns elementos.

Seguem alguns atributos globais importantes e suas descrições simplificadas:

- **accesskey**

Determina uma (ou mais) tecla(s) de atalho para o elemento. Para definir mais de uma tecla, coloque-as separadas por espaço.

O atributo *accesskey* nos permite especificar um atalho de teclado para acessar ou ativar um elemento em nossa página **HTML5**. A ideia aqui é fornecer ao usuário uma forma de navegar pelos elementos de nossa página usando atalhos. É importante mostrar de alguma forma aos visitantes do site que esses atalhos estão disponíveis.

- **class**

Determina uma (ou mais) classe(s) para o elemento. Para definir mais de uma classe, coloque-as separadas por espaço.



- **draggable**

Define se um elemento é arrastável ou não.

- **hidden**

Oculto o elemento onde for declarado.

- **id**

É o identificador único do elemento. Somente deve ser declarado um id por elemento. E este id não deve ser repetido na mesma página.

- **lang**

Determina o idioma em que está escrito o conteúdo do elemento.

- **style**

Determina propriedades CSS diretamente no elemento.

- **tabindex**

Organiza os elementos por ordem de tabulação. Deve-se usar valor numérico.

- **title**

Representa um auxílio ao elemento. Semelhante a dica do elemento.



Tipos de Meta Tags

Author - O nome do autor da página.

Copyright - Direito autoral da página.

Description - Descrição da página.

Expires - Data em que o documento deve ser considerado expirado.

Keywords - Aqui é um dos locais onde os motores de busca procuram informações a respeito da página.



Exemplo 1

Criando nossa primeira página HTML

```
<!DOCTYPE html>
<html lang="pt-br">

  <head>
    <meta charset="utf-8">
    <title>Insira o titulo da pagina aqui</title>

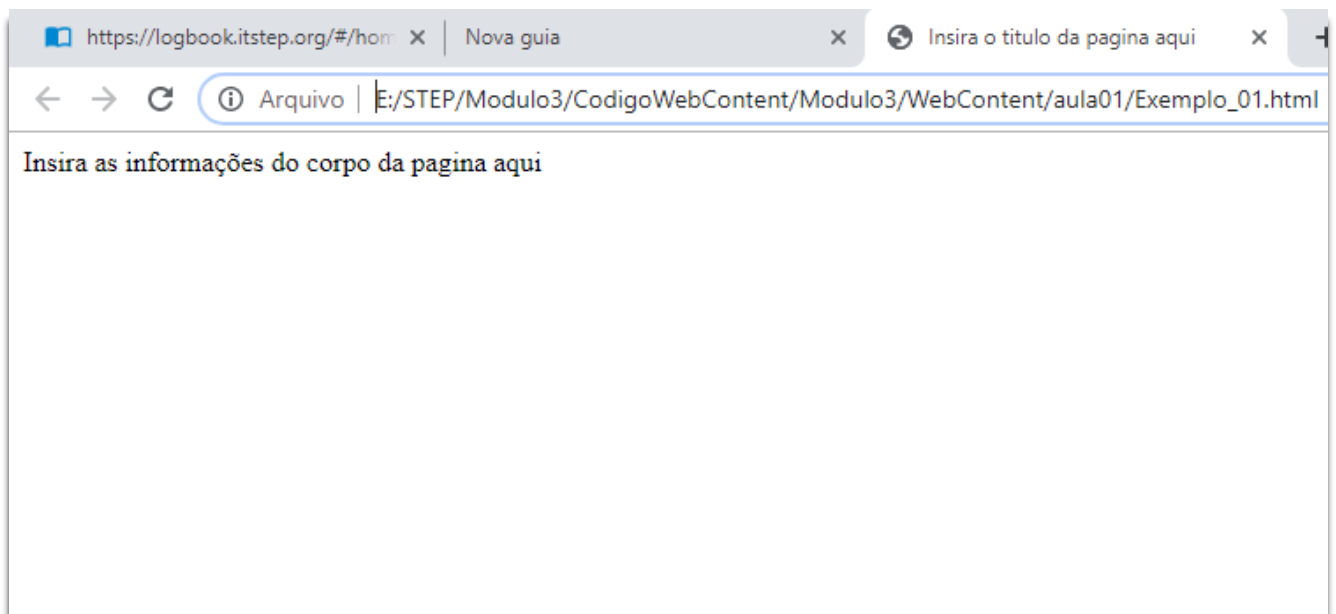
  </head>

  <body>
    Insira as informações do corpo da pagina aqui

  </body>

</html>
```

Resultado:





Exemplo 2

No Exemplo 2, temos o atributo “lang” (linguagem na tag do HTML).

Podemos observar o uso do recurso de comentário `<!-- conteúdo -->`. Este pode ser usado para comentar um ou várias linhas

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 2</title>
</head>

<body>

  No HTML <br> <br>
  A tag "< head >" informações gerais (metadados) sobre a
  página, incluindo seu título e links para scripts e folhas de estilos (CSS).
  <br>
  <br>

  A tag < meta > define qualquer informação de metadados que não podem
  ser definidos por outros elementos HTML (< base >, < link >, < script >, < style
  > ou < title >). <br>
  Neste código estamos definindo o padrão de formatação deste documento
  na hora de ser salvo. Por padrão usaremos o padrão "utf-8".

  <br>
  <br>

  a tag "< title >" (Título) define o título do documento,
  mostrado na barra de título de um navegador ou na aba da página.
  Deve conter somente texto.

  <br>
  <br>
  a tag < body >, define o corpo da pagina em HTML. tudo que será exibido no navega
dor
  estará declarado dentro do body.

  <br>
  <br>
```



```
Usamos o comando "<!--" e "-->" para fazermos comentários na página. <br>
Comentários

<br>
<br>
o comando <br> serve para pular de linha.

</body>

</html>

<!-- Exemplo de declaração de comentário
Podemos declarar comentários em qq parte do arquivo HTML.
Ele serve para o programador colocar alguma informações, dica,
sugestão ou comentário do código.
Os comentários não são exibidos no navegador ao abrir a página de HTML.

Sintaxe:
para declarar um comentário, basta iniciar o comando com "<!--"
e finalizar com "-->"
<!-- coloque seu comentário aqui -->

Obs:
*não use comentário dentro de comentário.
* o comando "<!--" e o comando "-->", escreve-se sem espaço
* alguns caracteres possuem funções específicas no HTML < > e etc.
para exibir o < use o comando &lt;
para exibir o > use o comando &gt;
-->
```

Resultado

No HTML

A tag "< head >" informações gerais (metadados) sobre a página, incluindo seu título e links para scripts e folhas de estilos (CSS).

A tag "< meta >" define qualquer informação de metadados que não podem ser definidos por outros elementos HTML (< base >, < link >, < script >, < style > ou < title >). Neste código estamos definindo o padrão de formatação deste documento na hora de ser salvo. Por padrão usaremos o padrão "utf-8".

a tag "< title >" (Título) define o título do documento, mostrado na barra de título de um navegador ou na aba da página. Deve conter somente texto.

a tag "< body >," define o corpo da página em HTML. tudo que será exibido no navegador estará declarado dentro do body.

Usamos o comando "<!--" e "-->" para fazermos comentários na página.

Comentários

o comando
 serve para pular de linha. "



Exemplo 3

Cabeçalho

Elementos de cabeçalho são implementados em seis níveis, `<h1>` é o mais importante e `<h6>` é o de menor importância. Um elemento de cabeçalho descreve brevemente o tópico da seção em que ele está.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 3</title>
</head>

<body>
  <h1>Criando Títulos/Cabeçalhos</h1>
  <p>Elementos de cabeçalho são implementados em seis níveis, < h1> é o mais import
ante e < h6> é o de menor importância. <br>
    Um elemento de cabeçalho descreve brevemente o tópico da seção em que ele est
á. <br>
    As informações de cabeçalho podem ser usadas para definir, dividir ou organiz
ar o conteúdo da página de html.

    <br>
    Sintaxe:
    < tag_abertura > conteúdo < / tag_fechamento>
  </p>
  <br>

  <!-- tag de títulos e seus níveis de 1 a 6-->
  <h1>Título nível 1</h1>
  <h2>Título nível 2</h2>
  <h3>Título nível 3</h3>
  <h4>Título nível 4</h4>
  <h5>Título nível 5</h5>
  <h6>Título nível 6</h6>

  <br>
  <!-- definindo um paragrafo-->
  <p>Usando a tag parágrafo </p>

  <br> <!-- formatando o estilo do paragrafo, deixando centralizado-->
  <center>
```




```
<h1>Título nível 1</h1>
<h2>Título nível 2</h2>
<h3>Título nível 3</h3>
<h4>Título nível 4</h4>
<h5>Título nível 5</h5>
<h6>Título nível 6</h6>

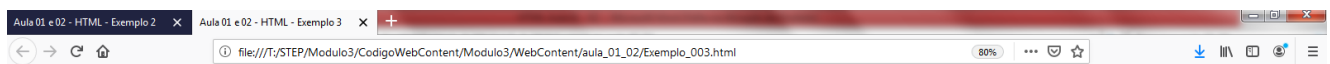
</center>
</body>

</html>

<!--

-->
```

Resultado:



Criando Títulos/Cabeçalhos

Elementos de cabeçalho são implementados em seis níveis, <h1> é o mais importante e <h6> é o de menor importância. Um elemento de cabeçalho descreve brevemente o tópico da seção em que ele está. As informações de cabeçalho podem ser usadas para definir, dividir ou organizar o conteúdo da página de html. Sintaxe: <tag_abertura> conteúdo </tag_fechamento>

Título nível 1

Título nível 2

Título nível 3

Título nível 4

Título nível 5

Título nível 6

Usando a tag parágrafo

Título nível 1

Título nível 2

Título nível 3

Título nível 4

Título nível 5

Título nível 6



Exemplo 4

Uso de algumas tags de formatação de texto.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 4</title>
</head>

<body>

  <!--      &lt;   representa o sinal de <
           e o &gt;   representa o sinal de > -->

  <p>
    <b>tag "b" Texto em negrito</b><br>
    <i>tag "i" Texto em itálico </i><br>
    <u>tag "u" Texto sublinhado </u><br>
    <sub>tag "sub" Texto subscrito </sub><br>
    <sup>tag "sup" Texto sobrescrito</sup><br>
  </p>

  <br> <br>

  <p>

    <b>tag &lt;b> - Texto em negrito </b><br>
    <i>tag &lt;i> - Texto em itálico </i><br>
    <u>tag &lt;u> - Texto sublinhado </u><br>
    A <sub>tag &lt;sub> - Texto subscrito</sub><br>
    A <sup>tag &lt;sup> - Texto sobrescrito</sup><br>
    A <big>tag &lt;big> - Texto com fonte maior do que o padrão </big><br>
    <small>tag &lt;small> - Texto com fonte menor do que o padrão</small><br>
    <em>tag &lt;em> - Texto em itálico</em><br>
    <strong>tag &lt;strong> - Texto em negrito</strong> <br>
    <mark>tag &lt;mark> - Texto destacado</mark>

  </p>

  Outros exemplos:
  <p>Exemplo de <b>Texto em negrito - tag &lt;b> </b></p>
  <p>Exemplo de <em>Texto enfatizado - tag &lt;em> </em></p>
  <p>Exemplo de <strong>Texto forte - tag &lt;strong> </strong></p>
  <p>Exemplo de <i>Texto em italico - tag &lt;i> </i></p>
```



```

<p>Exemplo de <sup>Texto superior - tag &lt;sup&gt;</sup></p>
<p>Exemplo de <sub>Texto inferior - tag &lt;sub&gt;</sub></p>
<p>Exemplo de <del>Texto taxado - tag &lt;del&gt;</del></p>
<p>Exemplo de <code>Texto de código de computador - tag &lt;code&gt;</code></p>

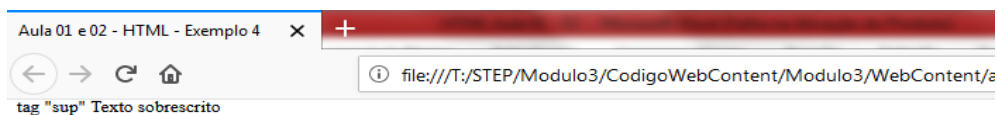
<p>Texto em negrito com <b>bold</b> e <strong>strong</strong>.</p>
<p>Texto em itálico com <i>italics</i> e <em>emphasis</em>.</p>
<p>Texto <sup>sobrescrito</sup> e <sub>subscrito</sub>.</p>
<p>Texto <ins>inserido</ins> e <del>excluído</del>.</p>
<p>Texto <small>pequeno</small> e <mark>destacado</mark>.</p>

</body>

</html>

```

Resultado:



tag - Texto em negrito

tag <i> - Texto em itálico

tag <u> - Texto sublinhado

A tag <sub> - Texto subscrito

A tag <sup> - Texto sobrescrito

A tag <big> - Texto com fonte maior do que o padrão

tag <small> - Texto com fonte menor do que o padrão

tag - Texto em itálico

tag - Texto em negrito

tag <mark> - Texto destacado

Outros exemplos:

Exemplo de **Texto em negrito** - tag

Exemplo de *Texto enfatizado* - tag

Exemplo de **Texto forte** - tag

Exemplo de *Texto em itálico* - tag <i>

Exemplo de Texto superior - tag <sup>

Exemplo de Texto inferior - tag <sub>

Exemplo de ~~Texto taxado~~ - tag

Exemplo de Texto de código de computador - tag <code>

Texto em negrito com **bold** e **strong**.

Texto em itálico com *italics* e *emphasis*.

Texto ^{sobrescrito} e _{subscrito}.

Texto inserido e ~~excluído~~.

Texto pequeno e **destacado**.



Exemplo 5

Uso do atributo “align” da tag “p”, responsável pelo alinhamento de parágrafos.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 5</title>
</head>

<body>
  <h1>Atributo alinhamento em paragrafo</h1>

  <p>O parágrafo é a base do texto editado. </p>

  <!-- Uso da atributo align , responsável pelo alinhamento do parágrafo-->
  <!-- shift + alt + f => ativa a indentação do código no Visual Studio Code-->

  <p align="justify">O parágrafo é o elemento base do texto editado. </p>

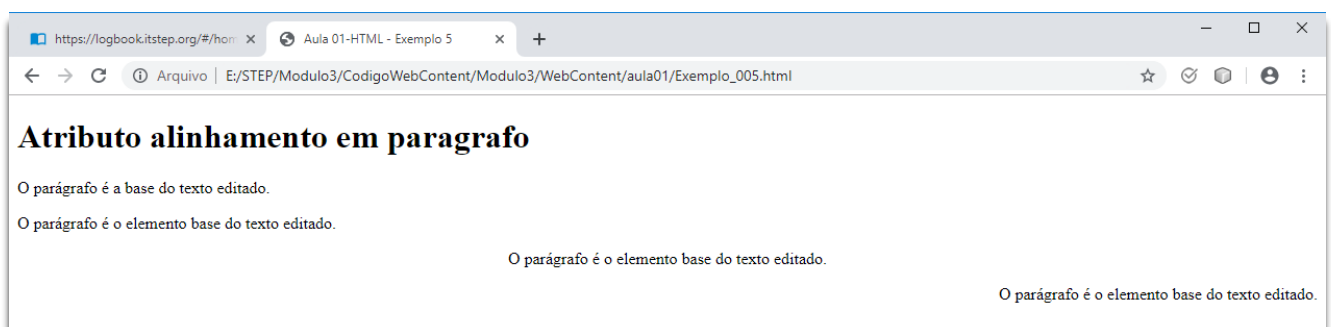
  <p align="center">O parágrafo é o elemento base do texto editado. </p>

  <p align="right">O parágrafo é o elemento base do texto editado. </p>

</body>
</html>

<!--
  O elemento HTML <p> (ou Elemento HTML Parágrafo) representa um parágrafo do texto.
  Parágrafos são, geralmente, representados em mídia visual, como bloco de texto
  que são separados dos blocos adjacente por espaços brancos verticais e/ou recuo
  de primeira-linha. Parágrafos são elementos em bloco .
-->
```

Resultado:





Exemplo 6

Uso de marcadores, tanto numéricos quanto pontos.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 6</title>
</head>

<body>

  <h4>Lista numérica</h4>

  <ol>
    <li>Café</li>
    <li>Almoço</li>
    <li>Jantar</li>
  </ol>

  <br>

  <!-- a tag <ol>, cria um marcador numérico-->
  <!-- a tag <ul>, cria um marcador de pontos-->
  <!-- A tag <li></li> serve para separar os itens da lista no meio das tag acima
citadas. -->

  <hr />
  <!-- Use <hr /> <hr> para criar uma linha horizontal. Essa tag é parecida com a
de Quebra de Linha. -->

  <br>

  <h4>Lista com marcadores</h4>

  <ul>
    <li>Café</li>
    <li>Almoço</li>
    <li>Jantar</li>
  </ul>
```



```
<hr />
<!-- Use <hr /> <hr> para criar uma linha horizontal. Essa tag é parecida com a
de Quebra de Linha. -->
</br>
<h4>Lista com marcadores</h4>

<ul Type="disc">
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
</ul>

<hr />
<!-- Use <hr /> <hr> para criar uma linha horizontal. Essa tag é parecida com a
de Quebra de Linha. -->
</br>

<h4>Lista com marcadores</h4>

<ul type="circle" >
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
</ul>

<hr />
<!-- Use <hr /> <hr> para criar uma linha horizontal. Essa tag é parecida com a
de Quebra de Linha. -->
</br>

<h4>Lista com marcadores</h4>

<ul type="square">
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
</ul>

</body>

</html>

<!--
```



O elemento HTML `` (ou elemento HTML de Lista desordenada) representa uma lista de

itens sem ordem rígida, isto é, uma coleção de itens que não trazem uma ordenação numérica e as suas posições, nessa lista, são irrelevantes. Caracteristicamente, os itens em uma lista desordenada são exibidos com um marcador que pode ter várias

formas, como um ponto, um círculo, ou um quadrado.

O tipo de marcador não é definido na descrição HTML da página, mas na CSS associada,

utilizando a propriedade `list-style-type`.

O elemento HTML `` (ou a Lista dos Itens de um elemento HTML) é usado para representar um item que faz parte de uma lista. Este item deve estar contido em um elemento pai:

uma lista ordenada (``), uma lista desordenada (``), ou um menu (`<menu>`) e representa uma única entidade dessa lista. Em menus e listas desordenadas a relação de itens é exibida, normalmente, usando pontos de marcação (as bolinhas). Em listas ordenadas eles são, comumente, mostrados com algum contador ascendente - como um número, ou letra - à sua esquerda.

O Elemento HTML `` (ou Elemento HTML de lista ordenada) representa uma lista de itens ordenados. De forma característica esses itens ordenados em uma lista são mostrados com uma contagem que os precede, que pode ser de qualquer tipo, como numerais, letras, algarismos romanos, ou simples símbolos. Esse modelo numerado não é definido na descrição html da página, mas na folha de estilos CSS associada, pela propriedade `list-style-type`.

O atributo (`type`) é utilizado para definir qual símbolo gráfico (bullet) entre os três disponíveis deverá ser inserido juntamente com os itens da lista. Os valores possíveis são:

Type="disc" - Insere um círculo preenchido.

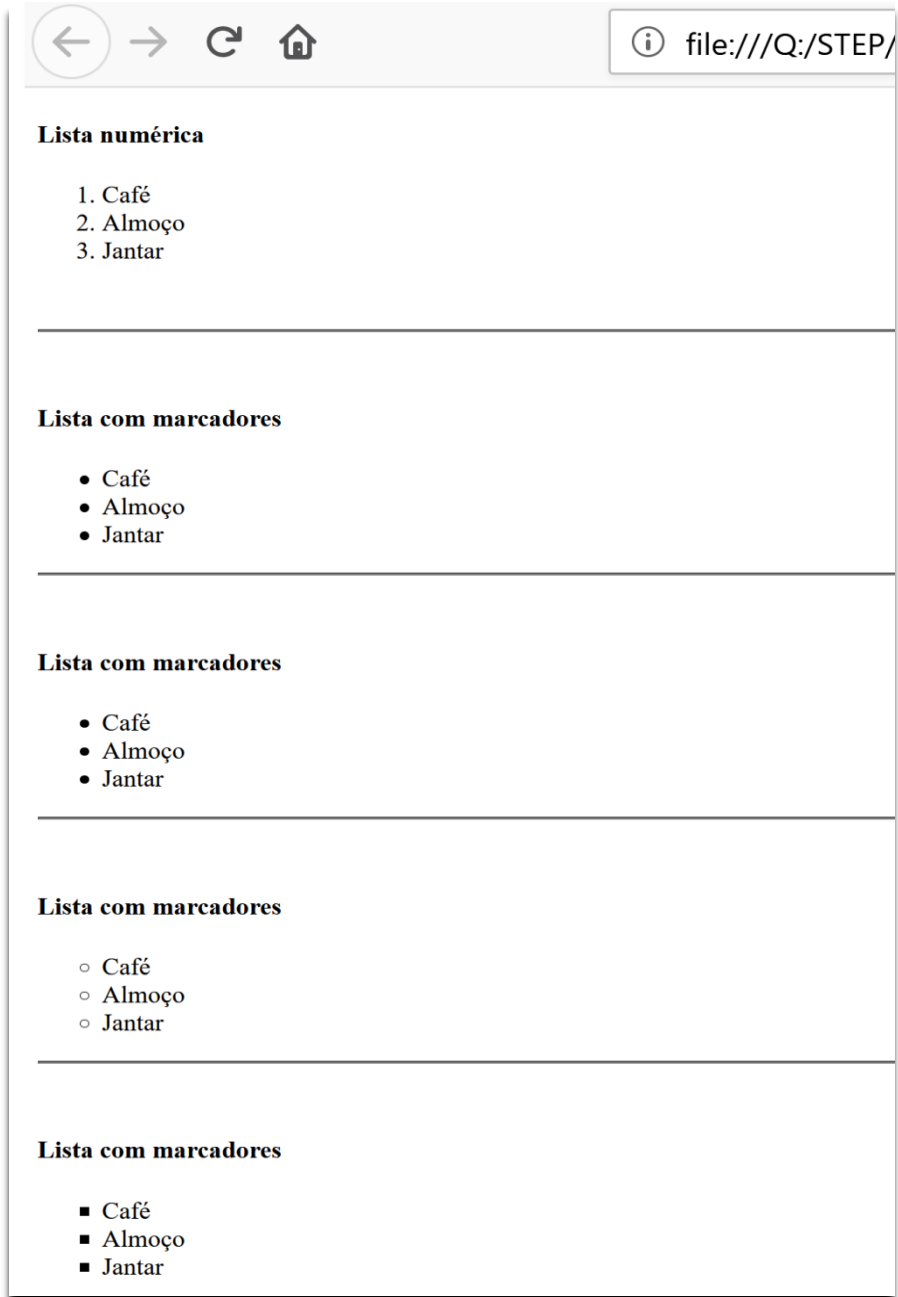
Type="circle" - Insere um círculo vazio.

Type="square" - Insere um quadrado preenchido.

-->



Resultado:





Exemplo 7

Uso de marcadores ordenados, passando como atributo o tipo ou forma da ordenação.

Letras minúsculas	Letras maiúsculas	Numerais romanos em minúsculas	Numerais romanos em maiúsculas
a. Trabalho b. Dinheiro c. Outra cidade	A. Trabalho B. Dinheiro C. Outra cidade	i. Trabalho ii. Dinheiro iii. Outra cidade	I. Trabalho II. Dinheiro III. Outra cidade

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 7</title>
</head>

<body>
  <!--
Outros tipos de listas:
<ol type="a">
<ol type="A">
<ol type="i">
<ol type="I">
-->

  <h4>Lista com indicadores alfabéticos</h4>

  <ol type="a">
    <li>Café</li>
    <li>Almoço</li>
    <li>Jantar</li>
    <li>Ceia</li>
  </ol>

  <hr>

  <h4>Lista com indicadores alfabéticos</h4>

  <ol type="A">
    <li>Café</li>
    <li>Almoço</li>
```



```
<li>Jantar</li>
<li>Ceia</li>
</ol>

<hr>

<h4>Lista com indicadores com indice</h4>

<ol type="i">
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
  <li>Ceia</li>
</ol>

<hr>

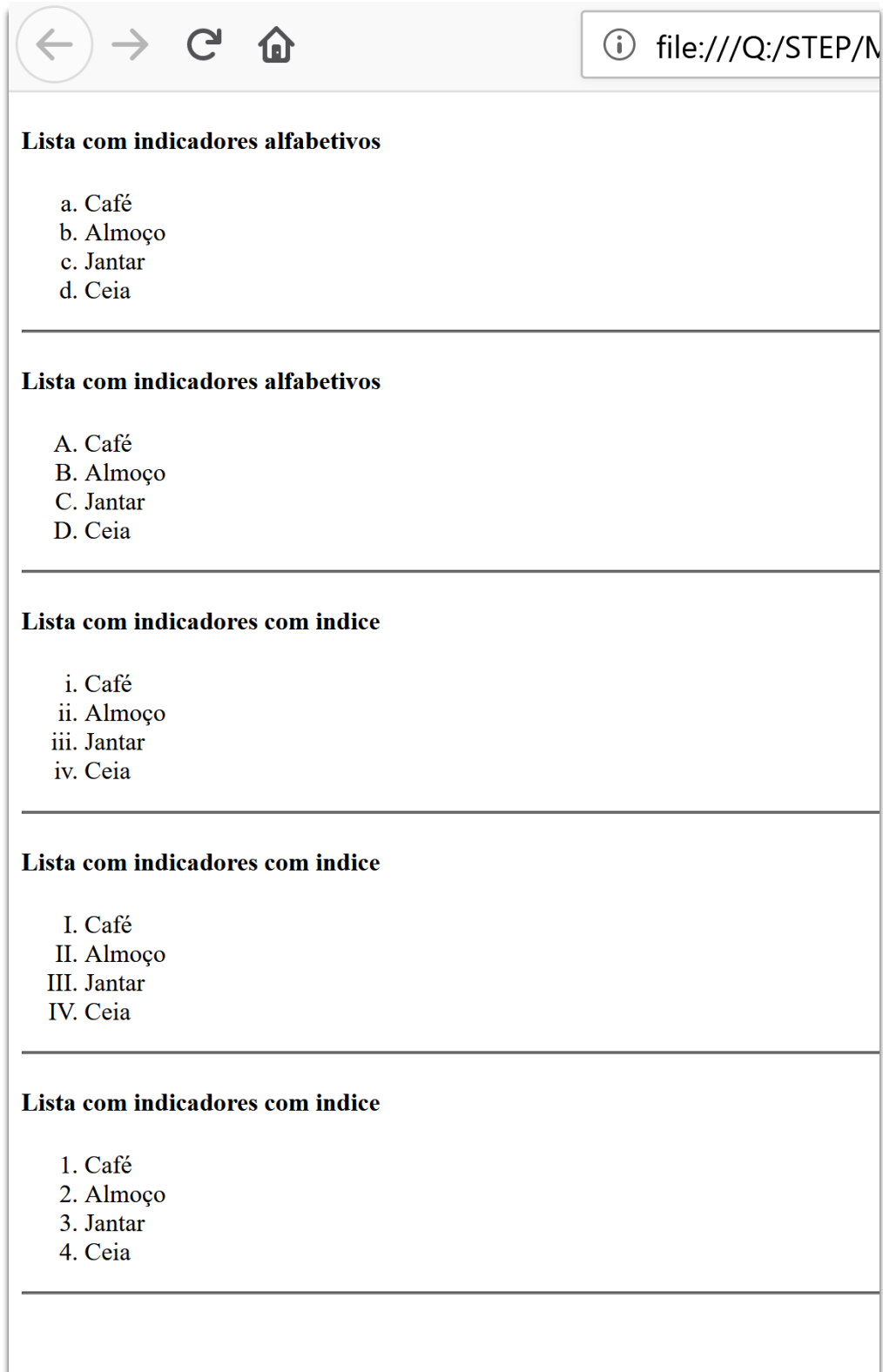
<h4>Lista com indicadores com indice</h4>

<ol type="I">
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
  <li>Ceia</li>
</ol>
<hr>

<h4>Lista com indicadores com indice</h4>

<ol type="1">
  <li>Café</li>
  <li>Almoço</li>
  <li>Jantar</li>
  <li>Ceia</li>
</ol>
<hr>

</body>
</html>
```

**Resultado:**

**Exemplo 8**

Uso de marcadores de definição.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 8</title>
</head>

<body>

  <h1>HTML</h1>
  <h2>Uso do recurso de detalhamento de descrição ou definição.</h2>

  <dl>
    <dt>Queso</dt>
    <dd>Palavra espanhola para queijo.</dd>

    <dt>Ordenador</dt>
    <dd>Palavra espanhola para computador.</dd>

    <br>

    <dt>Firefox</dt>
    <dt>Mozilla</dt>
    <dt>Internet Explorer</dt>
    <dd>São exemplos de navegadores </dd>
    <dd>São exemplos de programas de acesso a internet</dd>
    <dd>São programas de navegação</dd>

  </dl>

</body>

</html>
```



```
<!--  
    dl engloba uma lista de pares de termos e descrições. Um uso comum  
para este elemento  
    é para implementar um glossário ou exibir metadados(uma lista de pares  
chave e valor).  
  
    dt define um termo  
    dd descreve ou define o termo acima  
  
    em um dl, poderá haver vários dt.  
    em cada dt pode haver nenhum ou varios dd ( nenhum ou so um)
```

O elemento HTML Definition List (<dl>) engloba uma lista de pares de termos e descrições.

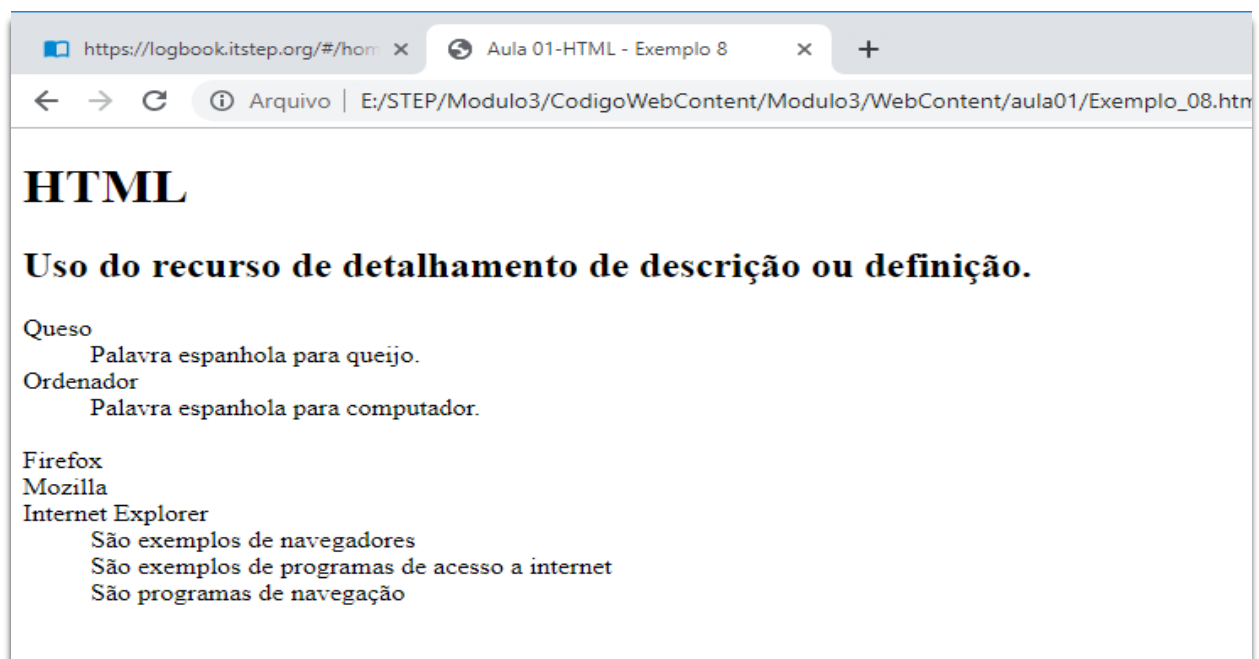
Um uso comum para este elemento é para implementar um glossário ou exibir metadados(uma lista de pares chave e valor).

O elemento HTML <dt> (ou Elemento HTML de Definição de Termo) identifica um termo na lista de definição. Este elemento pode ocorrer somente em um elemento filho de <dl>. Geralmente seguido por um elemento <dd>; ou multiplos <dt> na mesma linha indicam vários termos sendo definidos pelo próximo element <dd>.

O elemento HTML <dd> é usada para descrever um termo / nome em uma lista de descrição.

-->

Resultado:





Exemplo 9

Links - tag <a>

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 9</title>
</head>

<body>

  <h1>HTML</h1>
  Link para o Google: <a href="http://www.google.com.br" target="_blank" title="Google">Site de Pesquisa</a></br>
  </br>
  </br>

  Link para o Google: <a href="http://www.google.com.br" target="_blank" title="Google">Site de Pesquisa</a></br>
  Link para o UOL: <a href="http://www.uol.com.br" target="_blank" title="UOL">Site de Notícias</a></br>
  Link para o IG: <a href="http://www.ig.com.br" target="_blank" title="IG">Site de Notícias</a></br>
  Link para o Cade: <a href="http://www.cade.com.br" target="_blank" title="Cade">Site de Pesquisa</a></br>
  Link para o Terra: <a href="http://www.terra.com.br" target="_blank" title="Terra">Site de Notícias</a></br>

  <a href="https://www.imagem.na.internet.com.br/"> </a>

</body>

</html>

<!--
tag a, serve para criarmos link
o atributo "href" recebe o endereço do link a ser acessado
o atributo "target" diz ao navegador se ele precisa abrir outra página na mesma janela
o atributo "title" exibe uma dica ao usuário, na barra inferior do navegador
```



```
"blank" - Abrir uma nova janela  
"self"  - Abrir a página na mesma janela  
"parent" - Abrir nova página na mesma janela do link  
"top"   - Abre a nova página no mesmo navegador cancelando todos os demais frames
```

Obs: quando não sabemos o caminho, ou ainda teremos que definir, é recomendado declarar a href da seguinte maneira:

```
<a href="#">Nome do link</a>
```

também pode ser declarado como vazio

```
<a href="">Nome do link</a>
```

```
-->
```

Resultado:



**Exemplo 10**

Inserindo Imagens

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 10</title>
</head>

<body>
<h1>Inserindo imagens</h1>


  <!-- a imagem está no mesmo local que a pagina foi carregada-->


  <!--a imagem está no diretório "imagens", no mesmo nível
do local que a pagina foi carregada-->


<!-- a imagem está localizado na pasta imagens, na raiz da web atual -->


<!-- a imagem está localizado na pasta, um nível acima da pasta atual -->


<!-- a imagem está localizado na pasta "imagens" um nível acima da pasta atual -->

      <!--
a imagem está localizado na pasta "imagens" um nível acima da pasta atual -->

                <!--
a imagem está localizado na pasta, a 2 níveis acima da pasta atual -->

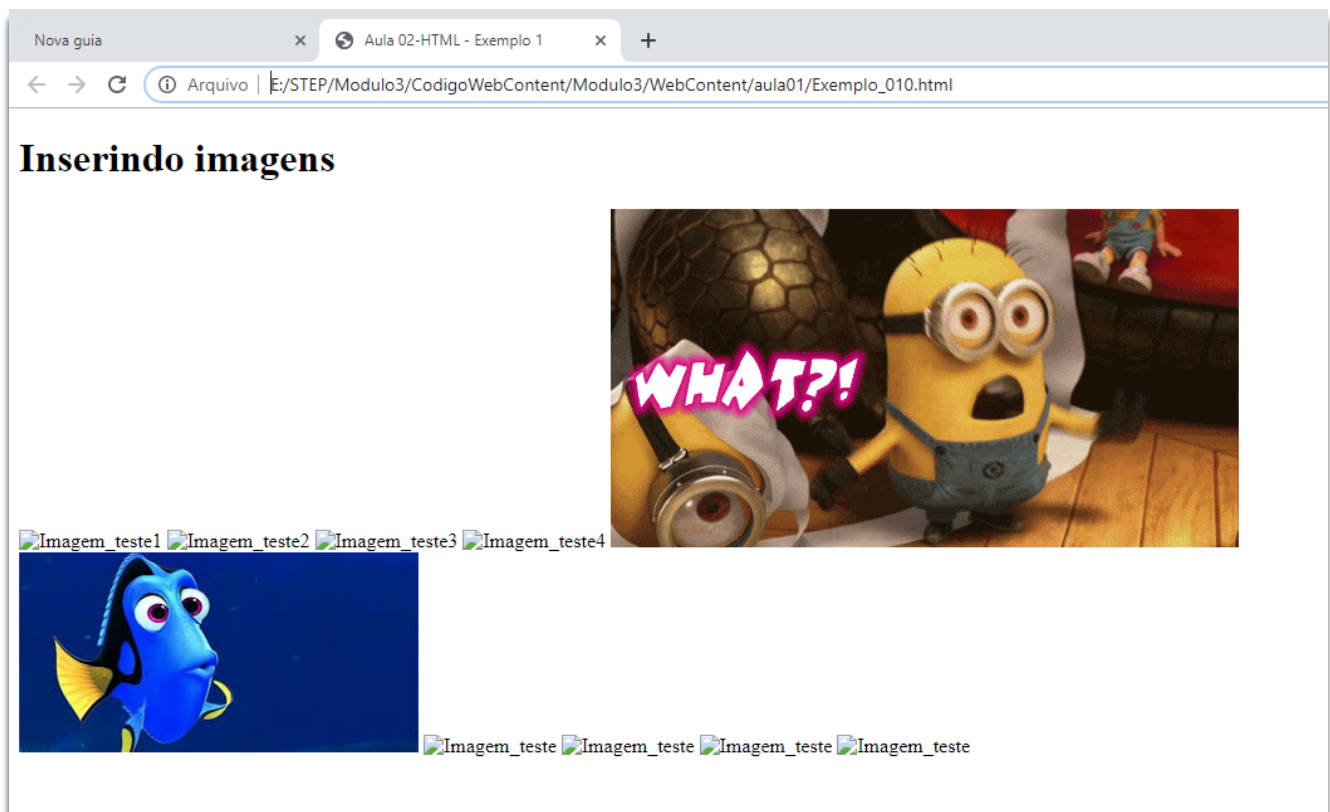
            <!--
a imagem está localizado na pasta a 3 níveis acima da pasta atual -->

              <!--
a imagem está localizado na pasta "diretório" no mesmo nível do local nde encontra-
se a pagina HTML -->
```




```
      <!--  
  a imagem está localizado no endereço URL na pagina Web -->  
  
</body>  
  
</html>  
  
  <!--  
No atributo "alt", de nomes as imagens, caso elas não sejam carregadas  
ele também ajuda na busca pelos motores ou robos de procura e  
quando ativado o recurso para deficientes visuais, o sistema irá ler o que foi descri  
to neste atributo  
Ex: alt="imagem com 3 filhotes de cachorros"  
  
Atributo alt  
Indica um texto alternativo, descrevendo brevemente a imagem, que é apresentado  
no lugar da imagem nos browsers texto, ou quando se desabilita o carregamento  
de imagens em browsers gráficos. É recomendável que esteja sempre presente.  
-->
```

Resultado:





Exemplo 11

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 11</title>
</head>
```

```
<body>
```

```
<h1>Atributos de Imagens</h1>
<p>
```

Ao inserir uma imagem é necessário saber o caminho da imagem. `
` Esta caminho deve ser atribuído ao atributo "src". `
`

O atributo "alt" define que informação irá aparecer caso a imagem não seja carregada pelo navegador. `
`

A tag img não necessita de tag de fechamento. `
`

O atributo "width" e "height" servem para especificar a largura e a altura de uma imagem. `
`

Sintaxe: `
`

```
< img src="caminho_da_imagem" alt="texto_caso_imagem_não_carregue" width="largura_em_pixel" height="altura_em_pixel" > <br>
```

`
`Também podemos usar o atributo "style" para definir o estilo da imagem.

```
</p>
```

```

```

```
</br>
```

```

```

```
<!--
```

o atributo width e height para especificar a largura e a altura de uma imagem .-->

```
</br>
```

```

```

```
<!--
```

o atributo style para especificar a largura e a altura de uma imagem (width e height).



de preferencia use o tamanho com style. isso evita que outros styles altere a imagem.
-->

```
</br>
```

```

```

```

```

```
<!--
```

Exemplo de como o navegador se comporta, quando não consegue carregar a imagem-->

```
</br>
```

```
<a href="www.pagina.com.br">
```

```
    
```

```
</a>
```

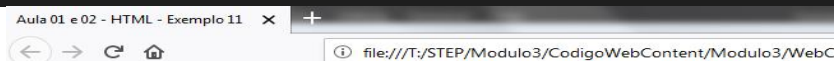
```
<!--
```

Criando um link, e associando a imagem ao link criado. Como resultado, ao clicar com o mouse na imagem, uma página será aberta-->

```
</br>
```

```
</body>
```

```
</html>
```



Atributos de Imagens

Ao inserir uma imagem é necessário saber o caminho da imagem.

Esta caminho deve ser atribuído ao atributo "src".

O atributo "alt" define que informação irá aparecer caso a imagem não seja carregada pelo navegador.

A tag img não necessita de tag de fechamento.

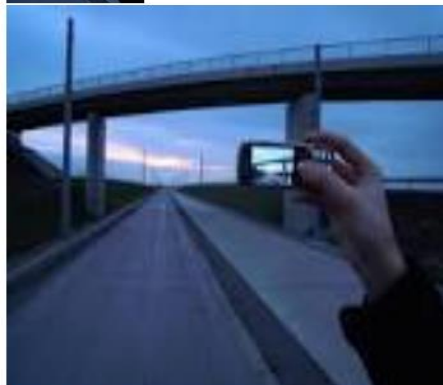
O atributo "width" e "height" servem para especificar a largura e a altura de uma imagem.

Sintaxe:

```

```

Também podemos usar o atributo "style" para definir o estilo da imagem.



Minha imagem de teste
HTML5 texto alternativo sem imagem
[HTML5 tutorial sem imagem](#)



Exemplo 12

Tag “p” - parágrafo

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8" name="Descrição" content="atributos de parágrafos">
  <title>Aula 01-HTML - Exemplo 12</title>
</head>

<body>
  <h1>Atributo alinhamento em paragrafo</h1>

  <p>0 parágrafo é a base do texto editado. </p>

  <!-- shift + alt + f => ativa a indentação do código no Visual Studio Code-->

  <!-- Uso da atributo align , responsável pelo alinhamento do parágrafo-->
  <p align="right">0 parágrafo é o elemento base do texto editado. </p>

  <p align="left">0 parágrafo é o elemento base do texto editado. </p>

  <p align="justify">0 parágrafo é o elemento base do texto editado. </p>

  <p align="center">0 parágrafo é o elemento base do texto editado. </p>

  <!-- colocando uma imagem no paragrafo-->
  <p>
    Este parágrafo contém uma imagem  de exemplo.
  </p>

  <!-- tag fonte-->
  <p>
    <font size="1">Esse é uma fonte de tamanho 1</font><br />
    <font size="2">Esse é uma fonte de tamanho 2</font><br />
    <font size="3">Esse é uma fonte de tamanho 3</font><br />
    <font size="4">Esse é uma fonte de tamanho 4</font><br />
    <font size="5">Esse é uma fonte de tamanho 5</font><br />
    <font size="6">Esse é uma fonte de tamanho 6</font><br />
    <font size="7">Esse é uma fonte de tamanho 7</font><br />
  </p>
```



```
</p>

<p>
  <font color="#990000">Esse texto está na cor hexadecimal #990000</font><br />
  <font color="red">Esse texto é vermelho</font>
</p>

<!-- atributo estilo da fonte-->
<p>
  <font face="Bookman Old Style">Esse parágrafo tem sua fonte formata com atrib
utos de
    face da fonte</font>
</p>

<p style="color:red;">This is a paragraph.</p>
<p style="font-family:courier;">This is a paragraph.</p>
<p style="font-size:160%;">This is a paragraph.</p>
<p style="text-align:center;">Centered paragraph.</p>

</body>

</html>
<!--
```

Tags são o conjunto de caracteres que formam um elemento, ou seja, quando nos referenciamos à Tag "p" estamos falando disso: <p>

Existem dois tipos de Tags, as que necessitam de fechamento e as que não necessitam de fechamento.

Para as que necessitam de fechamento, utilizamos o sinal de menor (<), seguido do nome do elemento e o sinal de maior (>) para abertura. Para fechamento, utilizamos o sinal de menor (<), seguido de barra (/), nome do elemento e o sinal de maior (>).

As Tags servem para marcar onde começam e terminam os Elementos. Já os Elementos são uma parte conceitual/semântica que têm um começo e fim determinados

Atributos são informações que passamos na Tag para que ela se comporte da maneira esperada.



Existem atributos globais (que funcionam em todas as Tags) e específicos (que são direcionados para cada Tag, através de especificação).

Os Atributos possuem nome e um valor, existem Atributos que você vai usar praticamente sempre e existem outros que serão mais raros.

-->

<!--

Características de TAG parágrafo <p>

A tag de abertura é obrigatória. A tag de fechamento pode ser omitida caso o elemento <p> seja imediatamente sucedido pelos elementos

<address>, <article>, <aside>, <blockquote>, <div>, <dl>, <fieldset>, <footer>, <form>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>, <hr>, <menu>, <nav>, , <pre>, <section>, <table>, , ou por outro elemento <p>, ou, caso não haja mais conteúdo no elemento pai e o elemento pai não for um elemento <a>.

-->

<!--

O elemento HTML <style> contém informações de estilo para um documento ou uma parte do documento. As informações de estilo específico estão contidas dentro deste elemento, geralmente no CSS.

Dentre os atributos existe um específico denominado style. Este atributo permite definir estilos CSS para seu elemento.

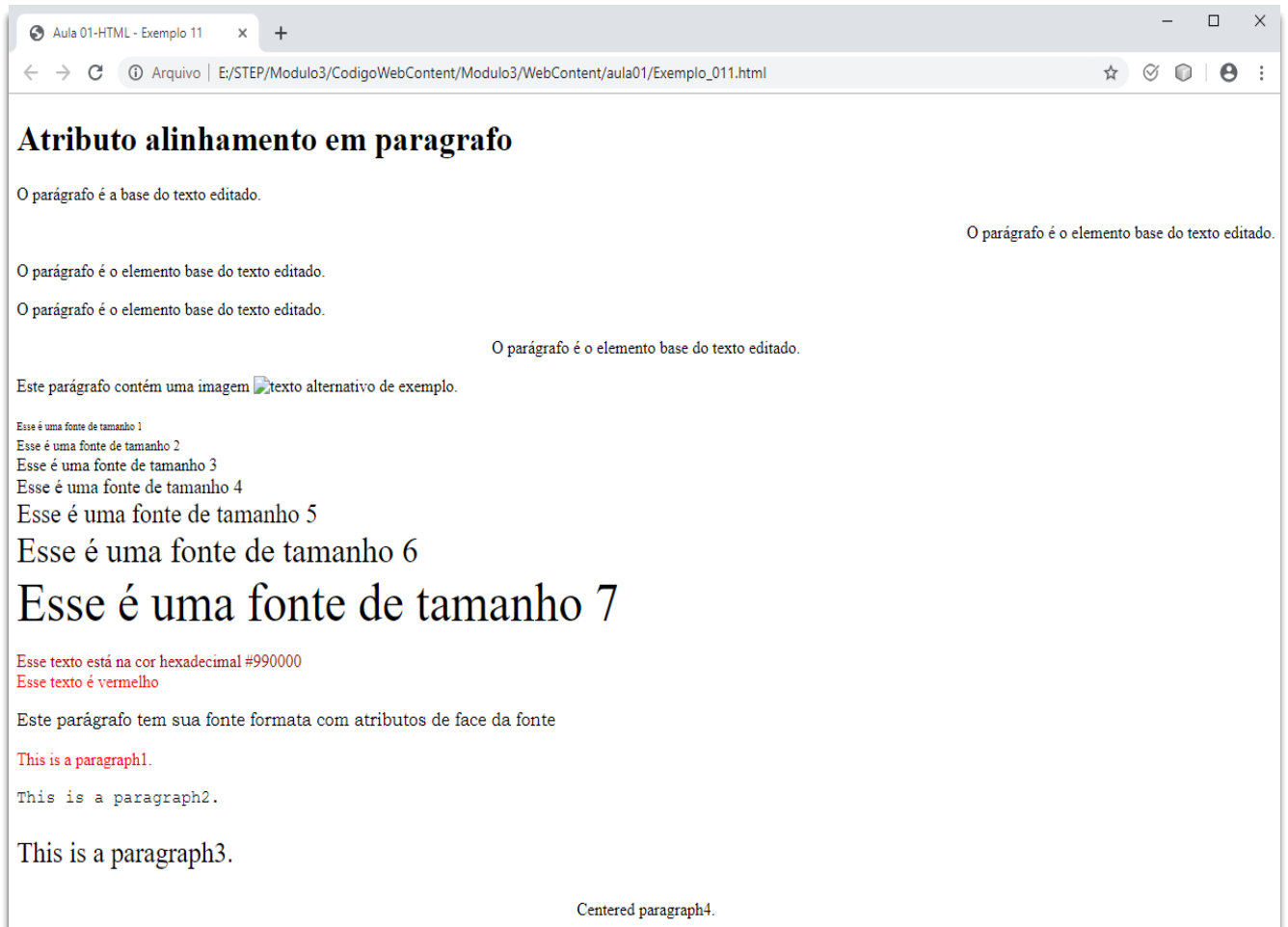
O CSS (Cascade Style Sheets, significa Folha de Estilo em Cascatas) é uma linguagem de estilização que serve para alterarmos

a estilização de nosso html, obviamente. é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação. Ele segue a seguinte sintaxe:

sintaxe:

<nomedatag style="propriedade: valor;">conteúdo aqui...</nomedatag>

-->

**Resultado:**



Exemplo 13

Tag “p” – parágrafo com style

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 12</title>
</head>
<body>
  <h1>Usando o atributo "style" para formatar o parágrafo, alterando sua cor</h1>

  <p style="color:black;">Este é um parágrafo</p>
  <p style="color:blue;">Este é um parágrafo</p>
  <p style="color:rgb(158, 47, 47);">Este é um parágrafo</p>
  <p style="color:chartreuse;">Este é um parágrafo</p>
  <p style="color:darkcyan;">Este é um parágrafo</p>
  <p style="color:orange;">Este é um parágrafo</p>
  <p style="color:fuchsia;">Este é um parágrafo</p>
  <p style="color:red;">Este é um parágrafo</p>
</body>

</html>
```

<!--

O CSS trabalha com quatro tipos de cores: RGB, RGBA, Hexadecimal e o nome da cor em inglês.

RGB

rgb(0, 0, 0);

RGBA

rgba(0, 0, 0, 0);

Hexadecimal

#000000

Formato dos Códigos HTML:



Cada código HTML contém o símbolo "#" e 6 letras ou números. Estes números estão no sistema numérico hexadecimal. Por exemplo, "FF" em hexadecimal representa o número 255 em Decimal.

Significado dos símbolos em hexadecimal:

Os primeiros dois símbolos no código de cor HTML representam a intensidade da cor encarnada. 00 é o mínimo e FF o mais intenso. O terceiro e o quarto representam a intensidade de verde e o quinto e o sexto representam a intensidade de azul.

Portanto, combinando as intensidades de encarnado, verde e azul podemos misturar qualquer cor que o nosso coração deseje:)

Nome da cor em inglês

red, blue, green, yellow, purple, etc

Nesse exemplo de mudar a cor de fundo com background-color ficou um pouco difícil de entender as letras não é? Então teremos que alterar a cor das letras também.

Color: cor do texto

A propriedade color altera a cor do texto de um elemento

```
<h1 style="color:blue;">Este é um cabeçalho</h1>  
<p style="color:red;">Este é um parágrafo</p>
```

Font-family: Fonte do texto

A propriedade font-family define a fonte a ser usada no elemento html.

```
<h1 style="font-family:verdana;">Este é um cabeçalho</h1>  
<p style="font-family:courier;">Este é um parágrafo</p>
```

Font-size: tamanho do texto

A propriedade font-size define o tamanho da fonte utilizada no texto

```
<h1 style="font-size:18px;">Este é um cabeçalho</h1>  
<p style="font-size:24px;">Este é um parágrafo</p>
```

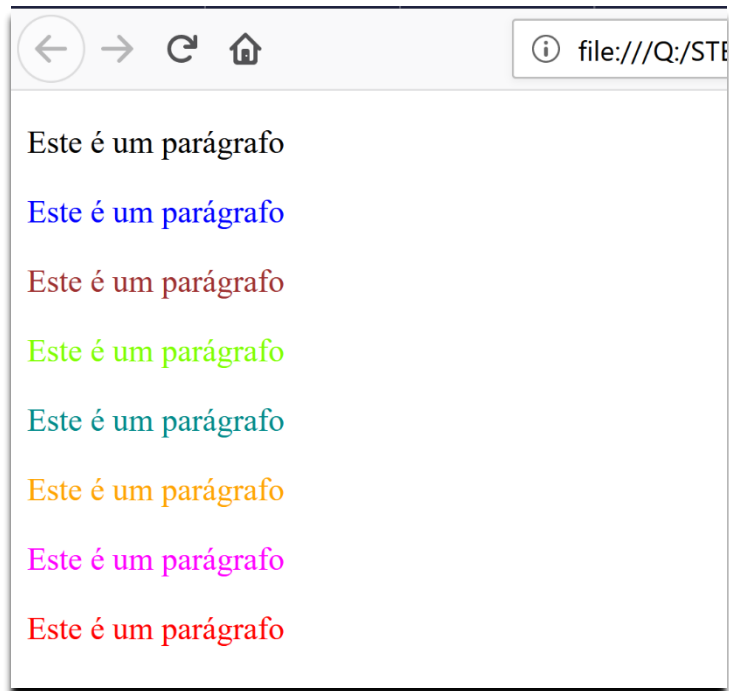
Text-align: alinhamento do texto

A propriedade text-align define o alinhamento do texto, que pode ser left, right, center e justify.

```
<p style="text-align:center;">Parágrafo centralizado</p>  
<p style="text-align:left;">Parágrafo alinhado a esquerda</p>
```



```
<p style="text-align:right;">Parágrafo alinhado a direita</p>
<p style="text-align:justify;">Parágrafo com alinhando justificado</p>
-->
```





Exemplo 14

Tag “body”

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Aula 02-HTML - Exemplo 14</title>
</head>
<body bgcolor="aqua" >
  <h1>Body e seus atributos</h1>

</body>
</html>
```

<!--

O atributo bgcolor do elemento <body>

Supondo que a cor do fundo seja branca, o atributo bgcolor permite escolher a cor de fundo da página. Há três modos para indicar a cor de fundo da página usando código de cor hexadecimal, em base 10 ou o nome de cor.

Estas são as codificações possíveis: (cor no formato hexadecimal, RGB ou nome da cor em ingles)

```
<body bgcolor="#ffffff">
<body bgcolor="rgb(255,255,255)"> , cada tom de cor varia de 0 a 255 (vermelho, verde e azul)
<body bgcolor="white">
```

| Color | Hex | Color | Hex | Color | Hex | Color | Hex |
|---------|---------|--------|---------|--------|---------|--------|---------|
| aqua | #00FFFF | green | #008000 | navy | #000080 | silver | #C0C0C0 |
| black | #000000 | gray | #808080 | olive | #808000 | teal | #008080 |
| blue | #0000FF | lime | #00FF00 | purple | #800080 | white | #FFFFFF |
| fuchsia | #FF00FF | maroon | #800000 | red | #FF0000 | yellow | #FFFF00 |

O atributo background do elemento <body>

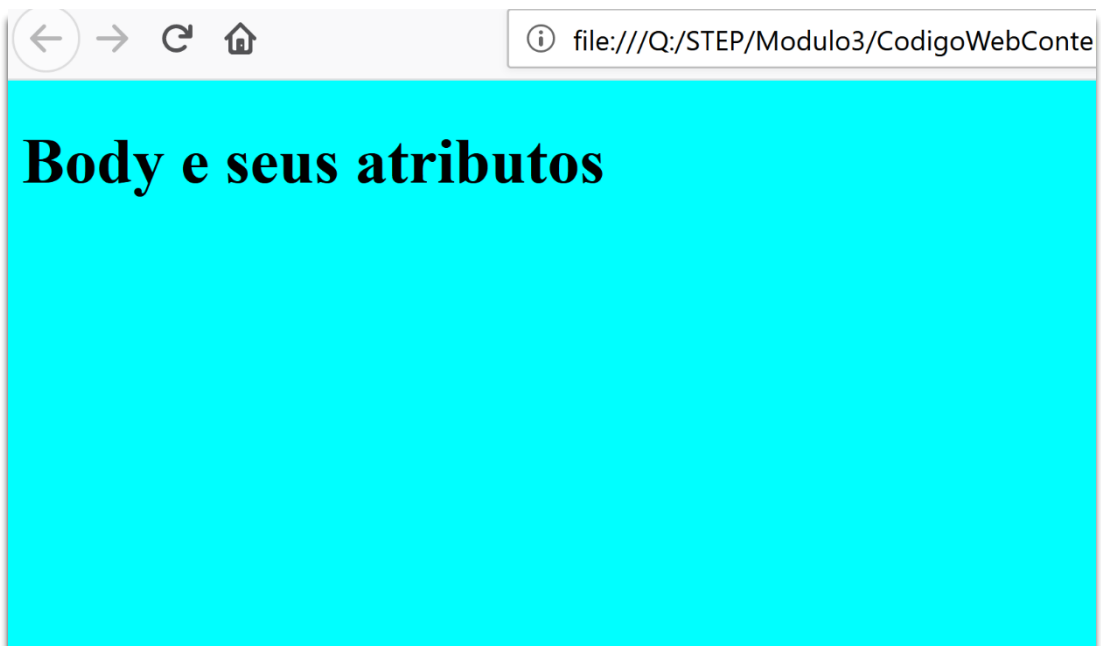
O atributo "background" estabelece que o padrão de fundo da página é dado por uma imagem e o valor deste atributo indica o local onde se encontra a imagem.

Se as dimensões da imagem forem menores que as dimensões da página o navegador



```
repetirá a imagem de modo a ocupar todo o fundo da página.  
  
Use a mesma sintaxe de carregar imagens na tag <img> para carrega-la;  
passando o caminho ou o endereço de URL.  
<body background="figura.gif">  
<body background="http://www.figura.com.br/dicas/fig/figura.gif">.
```

-->



**Exemplo 15**

Tag “**hr**” linha horizontal

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01-HTML - Exemplo 15</title>
</head>

<body>
  <p>Linha horizontal</p>
  <hr>
  <br>

  <p>Linha horizontal, centralizado, 75% da pagina, largura 15 na cor azul</p>
  <hr align=center width=75% size=15 color=blue>
  <br>

  <p>Linha horizontal, alinhado a direita, 50% da pagina na cor vermelha</p>
  <hr align=right width=50% size=10 color=red>
  <br>

  <p>Linha horizontal, alinhado a direita, 50% da pagina na cor vermelha</p>
  <hr align=left width=50% size=5 color=yellow>
  <br>

</body>

</html>

<!--
```

A marcação <hr> insere uma linha divisória na posição onde for colocado.

Não existe a marcação</HR>.

Principais atributos

size , recebe um valor numérico(pixel). Este define a largura da linha.

width, este valor pode receber um número ou um numero seguido de %.

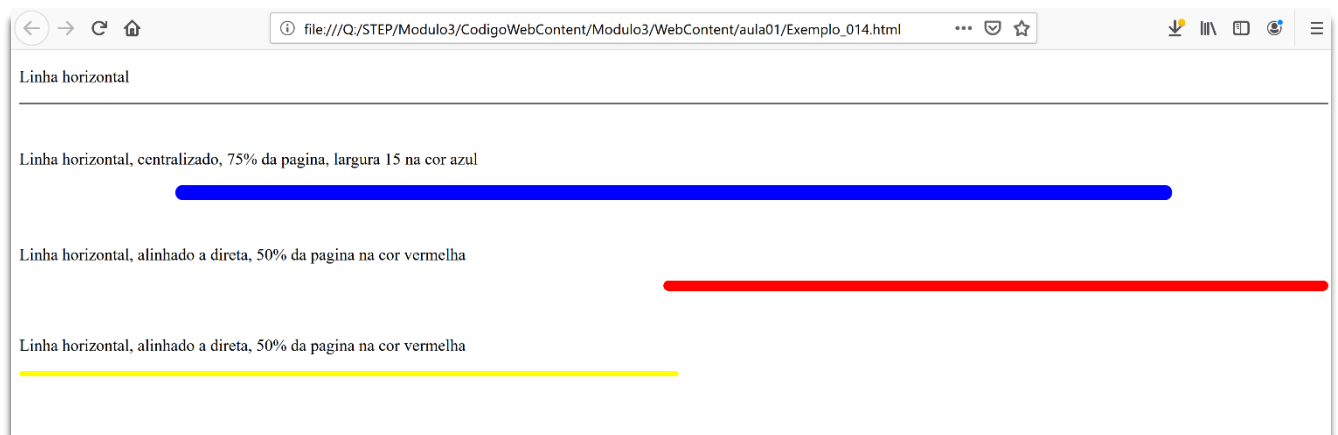
o primeiro define a larguma da linha. O segundo define a larguma baseado na relação e % da pagina.



```
align, ( recebe valores como "left ou center ou right"). especifica  
o alinhamento da linha.
```

```
color, especifica a cor da linha
```

```
-->
```





Unidades para valores

Existem inúmeros valores específicos para propriedades. Só que alguns são de uso comum para muitas propriedades e vira e mexe você estará usando. Veja uma pequena relação a seguir:

px – é a unidade de pixels. Muita usada em dimensões de blocos e fontes. Exemplo:

font-size: 14px;

width: 200px;

em – Indicada para tamanhos de fontes. Ao usarmos o tamanho padrão do dispositivo do usuário podemos expressar o tamanho da fonte de outros elemento através dessa unidade.

Assim um elemento com tamanho de fonte de 2em (font-size:2em) tem duas vezes o tamanho padrão. Caso o tamanho padrão for 16 pixels, uma fonte de 2em equivalerá a 32 pixels. Exemplo de sintaxe:

font-size: 1.25em;

font-size: 0.5em;

% – é, obviamente uma unidade de porcentagens. Ela é relativa ao bloco onde o elemento está contido. Se um bloco tem 150 pixels e for atribuído a outro bloco contido nele uma largura de 50%, equivalerá a 75 pixels.

font-size: 120%;

width: 50%;

Hexadecimal – é um sistema de numeração que representa os números em base 16, empregando assim 16 símbolos. Este sistema é composto por 10 números, de 0 a 9, e seis letras adicionais de A a F.

Usamos no CSS para atribuir valores para propriedades de cor para fontes (color), cor de fundo (background-color), cor de borda (border-color) entre outras. Exemplo:

background-color: #cccccc;

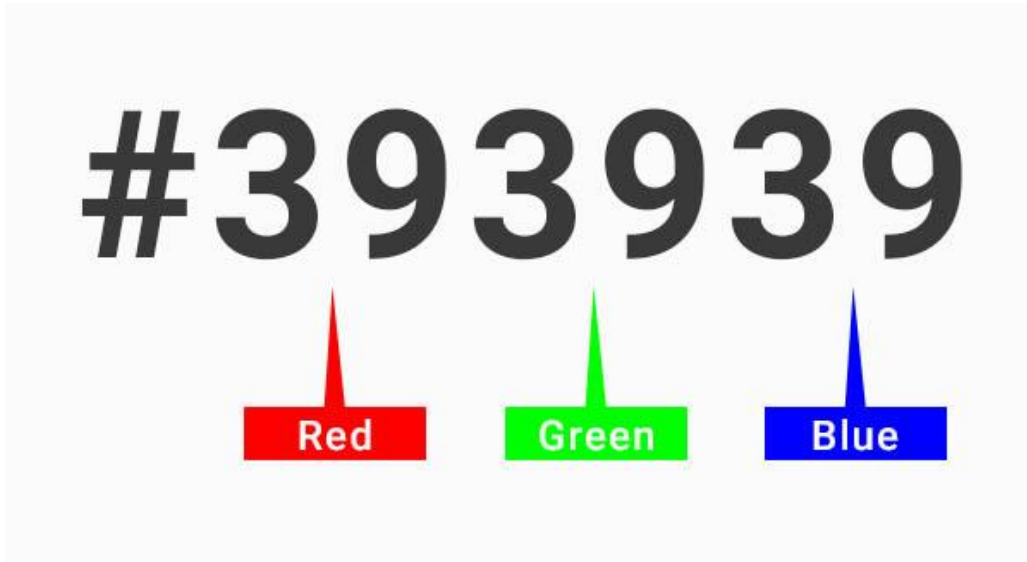
color: #ff0000;

Neste caso temos a cor vermelha para textos. O valor em hexadecimal em CSS é precedido do símbolo #.



O valor em hexadecimal é formado por três pares de caracteres. A ordem do pares, da esquerda para direita, representa o sistema de cores RGB (Red, Green, Blue) reproduzindo cores através da “mistura” em certa quantidade das cores aditivas Vermelha, Verde e Azul.

Na imagem abaixo vemos uma representação de um cinza escuro obtido pela mistura das três cores:





A Subclasse Style

Subclasse nada mais é que uma classe que define um objeto que deve ser incluído como atributo de um objeto pai, alterando as características desse objeto pai. Complicou? até parece, mas fora a definição, o conceito é simples: o objeto definido por uma subclasse não pode existir sozinho, deve ser sempre atributo de outro objeto, pois seus métodos e propriedades não afetam diretamente à ele, mas sim ao objeto que o contém. Assim acho que fica mais simples. Vamos ver então o que é a subclasse **Style**:

A Subclasse **Style** define um objeto com atributos próprios, mas que não pode existir sozinho no Html, deve sim ser um atributo de qualquer tag Html (salvo raras exceções). Os atributos desse objeto não alteram diretamente o próprio objeto, mas sim propriedades do objeto Html que o contém. Mais adiante isso ficará mais claro. Para que o objeto pai reconheça o objeto criado pela subclasse Style e para que este possa alterar propriedades diretamente no objeto pai, é convencionalizado que ele deve ser atribuído a um atributo do objeto pai chamado **style**. Adiante veremos a sintaxe correta de declaração e definição do objeto style e suas propriedades.

A Subclasse Style não acompanha o Html desde o advento da Internet, pois só se fez necessário o acesso à propriedades internas dos objetos Html após o advento do DHTML. Por isso a Subclasse Style só pode ser encontrada no HTML 4.0, e só é reconhecida nos navegadores IE e Netscape a partir das versões 4.0. Outros navegadores também podem reconhecer a subclasse style se estes interpretarem Html 4.0.

Como navegadores que interpretam o Html 3.0 estão fora de linha há mais de 8 anos, creio que vale a pena usar a subclasse style para programar os comportamentos DHTML. Vai ser muito raro um usuário nos dias de hoje, cujo navegador não interprete Html 4.0. Mesmo os usuários de outros Sistemas Operacionais, como o Linux e o Mac OS/2 já contam com suporte à DHTML e JavaScript há anos...

Definição e Sintaxe do Objeto Style

Por definição, qualquer atributo de objeto Html deve ser declarado seguindo a sintaxe a seguir:

```
nome_atributo="valor"
```

Assim também se faz para declarar o objeto style em um objeto Html:

```
style="definição_do_objeto"
```

A definição do objeto style nada mais é que uma sequência de caracteres que definem as propriedades do objeto. Como o objeto style possui mais de uma centena de atributos, todos eles já possuem um valor default definido, e só devemos informar na definição do objeto os atributos para os quais desejamos alterar o valor default.

A sintaxe da definição dos atributos da subclasse style se parece muito com a definição dos atributos dos objetos Html. Abaixo a sintaxe da declaração de cada atributo da subclasse style:



nome_atributo: valor;

Como você pode ver, as diferenças mais relevantes são: não se usa aspas (") no valor, o sinal de atribuição é o dois-pontos (:) e após o valor usa-se ponto-e-virgula (;).

A definição do objeto nada mais é que uma sequência de duplas **nome_atributo: valor;**. Todo valor deve ser sucedido de ponto-e-virgula, exceto o último, onde este sinal é facultativo (eu gosto de usar...).

Assim a declaração de um objeto style segue a sintaxe seguinte:

style="nome_atributo1: valor1; nome_atributo2: valor2; ... nome_atributoX: valorX"

Um exemplo de definição de um objeto style (este define o tamanho da fonte, a cor do texto e a cor de fundo):

style="font-size: 12px; color: green; background-color: #E9F7E1;"

Os Principais Atributos da Subclasse Style

Vamos conhecer os principais atributos da subclasse style. Alguns deles tem atributos correspondentes em grande parte dos objetos Html, mas mesmo assim é necessário conhecê-los, pois quando iniciarmos nosso estudo sobre CSS (próximo artigo) será necessário conhecer os nomes e valores permitidos de cada atributo.

Vamos primeiro aos mais usados, depois vou passar a definição dos mais simples e menos usados.

font-family

É um dos atributos mais complexos de se usar, por ter uma sintaxe diferenciada. Não definimos qual a fonte que será usada, mas sim sugerimos uma lista, que o navegador deve analisar e usar a primeira fonte da lista que estiver instalada no cliente.

O termo family se refere ao fato de normalmente informarmos famílias de fontes, e não seus nomes, nessa lista. Por exemplo: quero que um parágrafo meu seja exibido com a fonte Comic Sans MT2, mas pode ser que meu cliente não tenha essa fonte instalada, então informo apenas Conic como sendo a família da fonte, e se ele não possuir a fonte Comic Sans MT2, será exibida a fonte Comic Sans MS (padrão do windows, quase todo mundo tem) ou mesmo alguma outra da mesma família. Uma família define fontes com desenho gráfico parecidos, que tem sempre o primeiro nome em comum.

Mas ainda seguindo meu exemplo acima, pode ser que o usuário tenha as duas fontes (MT2 e MS) instaladas no seu micro. Então eu quero que o navegador exiba com a fonte Comic Sans MT2, mas a outra é padrão. Para que o navegador saiba qual é a minha primeira opção, normalmente eu inicio a lista com o nome da fonte que desejo, depois coloco o nome da sua família, para que seja usada outra fonte da família só se a fonte definida não estiver instalada.



Mas ainda pode ser que não tenha nenhuma fonte daquela família instalada, então devo informar outras famílias na lista, para que a primeira que for encontrada no cliente seja usada. Se nenhuma fonte, das famílias informadas for encontrada, então será usada a fonte padrão do navegador.

Sintaxe: A sintaxe do font-family segue algumas regras:

- Os itens da lista são separados por vírgulas, seguidas sempre por espaço (antes do próximo item);
- Os nomes de fonte devem estar entre aspas (") ou apóstrofes (');
- Os nomes de famílias de fonte não devem estar entre aspas ou apóstrofes.

Assim, um exemplo de definição da fonte seria:

font-family: "Times New Roman", Times, serif;

Assim, se for encontrada a fonte Times New Roman, ela será usada, senão será usada qualquer uma da família Times. Se não existir nenhuma instalada, será usada a fonte padrão da família serif (que é default do IE).

font-size

O atributo font-size é um dos que não podem ser acessados diretamente. As definições de tamanho de fonte nos objetos Html usam a notação de pontos que variam de 1 a 7, e os tamanhos reais (pixels ou pontos de impressão) variam de navegador para navegador. Por exemplo: no IE o tamanho 2 equivale a 14 pixels, já no Netscape equivale a 12 pixels.

Mas o atributo font-size permite definir o tamanho da fonte em pixels (px) ou pontos de impressão (pt), bastando informar o número seguido da notação **px** ou **pt**.

Exemplo: **font-size:** 18px;

font-weight

Atributo que define o peso da fonte. Assume valores como Negrito, Negrito Forte, Negrito Claro ou um valor em uma escala numérica. Não vou entrar na definição da escala numérica, vou me ater aos valores definidos. São eles:

- **bold:** Negrito
- **bolder:** Negrito Forte
- **lighter:** Negrito Claro
- **normal:** Normal

Como **normal** é o valor default, não é comum definir esta propriedade com esse valor. Os valores **bolder** e **lighter** dependem de como o navegador interpreta-os. Já o valor **bold** é equivalente as tags **** ou ****.

Exemplo: **font-weight:** bold;



font-style

O Estilo define se a fonte é inclinada. Os estilos disponíveis são Itálico e Oblíquo, sendo o estilo oblíquo idêntico ao itálico tanto no IE quanto no Netscape. O Estilo Itálico é equivalente às tags `<i>` ou ``. Vamos aos valores:

- **italic:** Itálico
- **oblique:** Oblíquo
- **normal:** Normal

Novamente, o valor normal é default, por isso não se usa definir a propriedade com esse valor.

Exemplo: `font-style: italic;`

text-decoration

A decoração do texto são os efeitos gráficos que podem ser alcançados para exibição do texto. Os valores permitidos são:

- **underline:** Sublinhado. Equivale a tag `<u>`
- **line-through:** Tachado. Equivale a tag `<s>`
- **overline:** Sobrelinha.
- **blink:** Piscar. Equivale a tag `<blink>`

Estes valores podem ser combinados, devendo ser separados por um espaço em branco, sem vírgulas ou qualquer outro sinal de pontuação. O valor default desse atributo é **none**, que define que o texto não deve ter nenhuma decoração.

Exemplo: `text-decoration: underline blink;`

text-align

Define o alinhamento do texto. Os valores são:

- **left:** alinha à esquerda.
- **center:** alinha ao centro.
- **right:** alinha à direita.
- **justify:** justifica o texto, ou seja, alinha às duas margens.

Exemplo: `text-align: center;`

vertical-align

Define o alinhamento vertical do texto na área do objeto. Os valores são:

- **top:** alinha junto à margem superior do objeto.



- **middle**: alinha ao centro vertical do objeto.
- **bottom**: alinha junto à base do objeto.

Exemplo: `vertical-align: middle;`

color

Define a cor da fonte. São permitidas todas as palavras-chaves do Html, como green, blue, black ... Também são permitidos valores Hexadecimais que definem a cor (precedidos sempre por #).

Exemplo: `color: #FF9966;`

background-color

Define a cor de fundo do objeto. São permitidos os mesmos valores do atributo **color**.

Exemplo: `background-color: yellow;`



Exemplo 16

Diferença entre a tag
 e <wbr>

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 16</title>

</head>

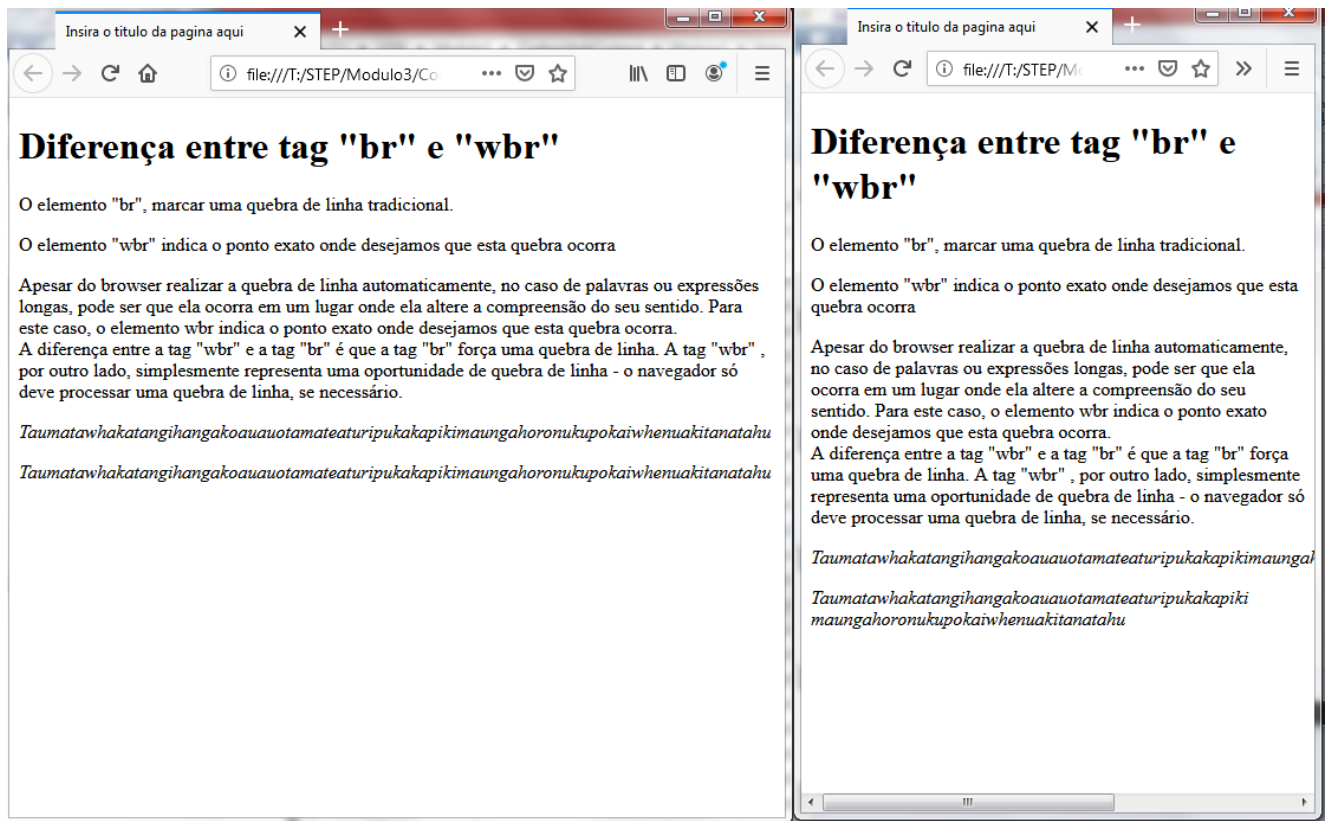
<body>
  <h1> Diferença entre tag "br" e "wbr"</h1>
  <p> O elemento "br", marcar uma quebra de linha tradicional.</p>
  <p> O elemento "wbr" indica o ponto exato onde desejamos que esta quebra ocorra</p>

  <p>Apesar do browser realizar a quebra de linha automaticamente, no caso de palavras ou expressões longas, pode ser que ela ocorra em um lugar onde ela altere a compreensão do seu sentido. Para este caso, o elemento wbr indica o ponto exato onde desejamos que esta quebra ocorra. <br>

  A diferença entre a tag "wbr" e a tag "br" é que a tag "br" força uma quebra de linha. A tag "wbr" , por outro lado, simplesmente representa uma oportunidade de quebra de linha - o navegador só deve processar uma quebra de linha, se necessário. </p>

  <p><i>Taumatawhakatangihangakoauauotamateaturipukakapikimaungahoronukupokaiwhenuakitanatahu</i></p>
  <p><i>Taumatawhakatangihangakoauauotamateaturipukakapiki<br>maungahoronukupokaiwhenuakitanatahu</i></p>
</body>

</html>
```





Exemplo 17

Tag “abbr” abreviação ou acrônimo. Ao passar o mouse sobre uma abreviação, esta por sua vez exibirá sua definição por extenso.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 17</title>
</head>

<body>
  <h1>A tag <abbr> define uma abreviação ou acrônimo, como "HTML", "Mr.", "Dec.", "
AAA", "BR".</h1>

  <p>Abreviação e acrônimo são versões reduzidas de outra coisa. Ambos são freq
uentemente representados como uma série de letras.</p>

  <p>Sintaxe:</p>
  <p> < abbr title="nome_por_extenso">abreviação < / abbr > </p>
  A <abbr title="Associação Brasileira de Normas Técnicas">ABNT</abbr> é uma fu
ndação nacional responsável por ....
</body>
</html>
```





Exemplo 18

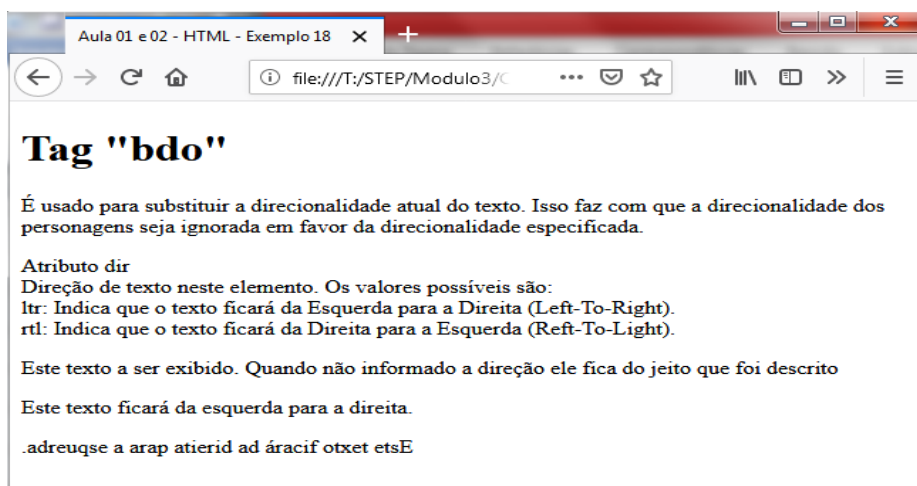
Tag <bdo>, define o direcionamento do texto.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 01 e 02 - HTML - Exemplo 18</title>
</head>
<body>
  <h1>Tag "bdo"</h1>
  <p>É usado para substituir a direcionalidade atual do texto. Isso faz com que a d
  irecionalidade dos personagens seja
    ignorada em favor da direcionalidade especificada.</p>
  <p> Atributo dir <br>
    Direção de texto neste elemento. Os valores possíveis são:

    <br> ltr: Indica que o texto ficará da Esquerda para a Direita (Left-To-Right).
    <br> rtl: Indica que o texto ficará da Direita para a Esquerda (Reft-To-Light).

  </p>
  <p>Este texto a ser exibido. Quando não informado a direção ele fica do jeito que
  foi descrito</p>
  <p><bdo dir="ltr">Este texto ficará da esquerda para a direita.</bdo></p>
  <p><bdo dir="rtl">Este texto ficará da direita para a esquerda.</bdo></p>
</body>
</html>
```





**Resumo**

Tag	Descrição
<html>	Define um documento HTML
<head>	Define informações de cabeçalho. Não é mostrada na janela do navegador.
<body>	Define o corpo do documento
<h1> a <h6>	Define título 1 a título 6
<p>	Define um parágrafo
 	Insere uma quebra de linha
<hr>	Define um filete horizontal
<!-->	Define um comentário
	Define texto em negrito
<big>	Define texto grande
	Define texto enfatizado;
<i>	Define texto em itálico
<small>	Define texto pequeno
	Define texto forte
<sub>	Define texto subscrito
<sup>	Define texto superescrito
<ins>	Define texto inserido
	Define texto cancelado
<s>	Desaprovada. Use em vez
<strike>	Desaprovada. Use em vez
<u>	Desaprovada. Use estilos em vez
<a>	Define uma âncora
	Define uma lista ordenada
	Define uma lista não ordenada
	Define um item de lista
<dl>	Define uma lista de definições
<dt>	Define um termo de definição
<dd>	Define uma descrição de definição
	Define uma lista ordenada
<wbr>	Indica um local para possível quebra de linha
<abbr>	Define uma abreviação ou acrônimo
<bdo>	Define o direcionamento do texto

**Entidades de Caractere Mais Comuns:**

Resultado	Descrição	Nome da Entidade	Número da Entidade
	espaço não separável	 	
<	menor do que	<	<
>	maior do que	>	>
&	e comercial	&	&
"	aspas	"	"
'	apóstrofe		'

Algumas Outras Entidades de Caractere Comumente Usadas:

Resultado	Descrição	Nome da Entidade	Número da Entidade
¢	cent	¢	¢
£	libra	£	£
¥	yen	¥	¥
§	parágrafo (em inglês section)	§	§
©	copyright	©	©
®	marca registrada	®	®
×	multiplicação	×	×
÷	divisão	&division;	÷