

**HTML 5 E CSS3**

**PROFESSOR ANDRÉ – INFORMÁTICA**

**Email:** [**andre.santos@etec.sp.gov.br**](mailto:andre.santos@etec.sp.gov.br)

**Facebook: André Gonzaga**

**TÉCNICAS DE PROGRAMAÇÃO**

**PARA INTERNET**

SUMÁRIO

[1 – INTRODUÇÃO 3](#_Toc414226412)

[1.1 - Estrutura da Internet 3](#_Toc414226413)

[1.2 –Estrutura da Web 4](#_Toc414226414)

[1.3 - Formato de um domínio 4](#_Toc414226415)

[1.4– Competências e Habilidades do desenvolvedor WEB 5](#_Toc414226416)

[2 - PLANEJANDO UM SITE 5](#_Toc414226417)

[2.1 - Recursos mínimos necessários para o desenvolvimento de um site 6](#_Toc414226418)

[3 – O HTML 5 6](#_Toc414226419)

[3.1 - O que é tag? 6](#_Toc414226420)

[3.2 – Características do HTML 5 com CSS 3 7](#_Toc414226421)

[3.3 – Regras para programar em HTML 7](#_Toc414226422)

[4- ESTRUTURA DE UMA PÁGINA HTML5 7](#_Toc414226423)

[4.1 - Estrutura obrigatória de um documento HTML5 7](#_Toc414226424)

[4.2 – A Tag html 8](#_Toc414226425)

[4.3 – A Tag Head 8](#_Toc414226426)

[4.4 – A Tag LINK 9](#_Toc414226427)

[5- DESENVOLVENDO UMA PÁGINA SIMPLES COM HTML 5 10](#_Toc414226428)

[5.1 - Exemplo de um arquivo HTML 5 10](#_Toc414226429)

[6 –VALIDANDO UM DOCUMENTO 11](#_Toc414226430)

[6.1 - Testando páginas com erro 11](#_Toc414226431)

[7 – FORMATANDO NOSSO HTML 5 COM CSS3 12](#_Toc414226432)

[7.1 – Criando o CSS 13](#_Toc414226433)

[7.2–Entendendo os Codigos de CSS 15](#_Toc414226434)

[7.3 - Utilizando figuras como fundo 16](#_Toc414226435)

[8 – CRIANDO TITULOS E LISTAS NO HTML5 17](#_Toc414226436)

[8.1 - Criando Titulos no HTML 17](#_Toc414226437)

[8.2 - Listas Ordenadas Não Ordenadas e Listas de Definição 17](#_Toc414226438)

[9.0 - TAGS PERSONALIZADAS COM USO DE IDENTIFICADORES E CLASSES 19](#_Toc414226439)

[9.1 - Utilizando a tag <div> 21](#_Toc414226440)

[**9.2 - Utilizando a tag <span>** 21](#_Toc414226441)

[**9.4 – Inserindo comentários** 21](#_Toc414226442)

**TÉCNICAS DE PROGRAMAÇÃO PARA INTERNET**

# 1 – INTRODUÇÃO

Nesta primeira parte, faremos um breve estudo dos conceitos e estrutura básica de uma página web.

Para esta disciplina além do conhecimento teórico, serão necessários alguns exercícios práticos que poderão ser executados no computador em offline(sem conexão com a Internet), com o uso de qualquer editor de textos e um navegador web, porém, por motivos de padronização estaremos utilizando o editor **NotePad ++**Portátil e o navegador Google **Crome** bem como um **pen drive** para guardar todos os exercícios relacionados a Internet.

Usaremos também a linguagem HTML5 com CSS3 e para correção das páginas iremos validá-las no site da **W3C**.

Além das apostilas, contaremos também com a ajuda de vários sites, listados abaixo, com vídeos, fóruns elivros, para aperfeiçoarmos o nosso aprendizado.

[**www.w3.org**](http://www.w3.org)ou **http://www.w3c.br/Home/WebHome**

**http://www.maujor.com/**

[**http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf**](http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf)

[**http://www.htmlprogressivo.net/2014/01/Apostila-HTML5-completa-download-W3C.html**](http://www.htmlprogressivo.net/2014/01/Apostila-HTML5-completa-download-W3C.html)

## 1.1 - Estrutura da Internet

Criada pelo departamento de defesa dos Estados Unidos no final da década de 60, em plena guerra fria, a Internet é uma grande rede de computadores onde as informações são fracionadas, divididas em pequenos pedaços, antes de seguir destino. A principal vantagem da Internet é que em caso de bloqueio em um caminho a informação pode tomar caminhos alternativos até o destino, isso teoricamente permite que a Internet não pare nunca.

Para que exista essa rede são necessários cabos, geralmente de fibra ótica, onde equipamentos se comunicam recebendo e enviando informações em alta velocidade. É necessário também, o uso de protocolos de comunicação para que todos esses equipamentos falem a mesma língua.

Um dos serviços mais utilizado na Internet é a Web criada no final da década de 1980. Há uma con­fusão entre internet e Web, mas, como foi comentado, na verdade a internet já existia 15 anos antes do surgimento da Web.

A Web consiste em uma estrutura chamada de cliente/servidor. Os clientes são os navegadores, também chamados de **Browser,** como por exemplo, o Internet Explorer, Mozilla Firefox, Google Crome e outros. Os servidores são equipamentos que armazenam e gerenciam as informações na Internet são chamados de **Servidores Web**.

## 1.2 –Estrutura da Web

A comunicação entre os clientes e servidores se dá através do pro­tocolo HTTP, através de uma requisição de HTTP:

Usuário digita um endereço desejado na caixa de endere­ço ou clica em um link;

2. Navegador envia a requisição até o servidor de web;

3. Servidor de web envia o conteúdo requisitado para o navega­dor do usuário;

4. O Navegador apresenta a informação na tela do usuário.

A resposta de uma requisição HTTP vem em um formato chamado HTML. Trata-se de um arquivo em que as informações de texto são apre­sentadas por uma série de marcadores (tags) para indicar a apresentação daquele texto. Os marcadores indicam o que são: listas, tabelas, títulos, formulários, entre outros. Exemplo de uma Tag:**</p>**( informa ao navegador para pular de linha).

O navegador utiliza estes marcadores para construir a aparência da página requisitada.

As páginas de Internet também chamadas home pages, ficam armazenadas no servidor Web.

Site é um conjunto de páginas web (home pages), interligadas através de hipertextos. A web ou WWW (World Wide Web) é o conjunto de todos os sites existentes no mundo.

## 1.3 - Formato de um domínio

Na Internet, cada página tem um endereço próprio, o que chama­mos de URL (Uniform Resource Locator) – Localizador Uniforme de Recur­sos. Esses endereços são chamados de domínios. Como exemplo, utiliza­mos o endereço abaixo, descrevendo-o detalhadamente.

http://www.google.com.br

• http:// (Hyper Text Transfer Protocol) – é o protocolo de transferência de ar­quivos de hipertexto e o mais comumente usado na internet. Quando necessitamos enviar os dados criptografados utilizamos o https:// que envia as informações de forma mais segura.

• www indica que estamos na Web

• google nome do site

• com tipo do site (com – comercial, org – organização não gover­namental, adv – advogados, gov – governamental, etc)

• br país de origem do site (br – Brasil)

## 1.4– Competências e Habilidades do desenvolvedor WEB

O profissional de Web designer tem sob sua responsabilidade as se­guintes atividades: construção de sites, sistemas, projetos multimídia, solu­ções para educação à distância e comércio eletrônico na internet. Ele deve criar ações que trabalhem os conceitos de usabilidade e planejamento da interface, facilitando e assegurando a interação aos usuários finais.

O web designer precisa adquirir conhecimentos em diversas áreas da informática e se manter atualizado, além de:

• possuir uma base de design tradicional, como teoria das cores e tipografia;

• aprimorar seus conhecimentos com base na sua área de interes­se;

• empenhar-se em ter um bom entendimento de CSS, HTML, PHP, Java Script, Flash, Photoshop entre outros. Há a necessidade de conhecer um pouco de tudo para decidir qual tecnologia utilizar;

• buscar conhecimento sobre usabilidade e procurar sempre por novas tecnologias.

# 2 - PLANEJANDO UM SITE

Antes de sair fazendo um site, precisamos conhecer o “problema” que tentaremos resolver. Detectar o real objetivo do site é primordial para um bom planejamento, de modo que as ações sejam tomadas de forma correta, minimizando assim futuras correções para atender o cliente.

É necessário definir o público-alvo do site, o objetivo, os serviços oferecidos, qual será o diferencial em relação aos outros sites.

Após o estudo feito, será definida a estrutura do site, a tecnologia empregada e o layout.

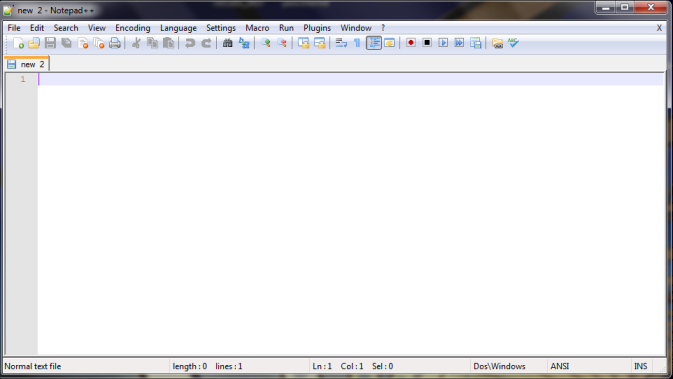
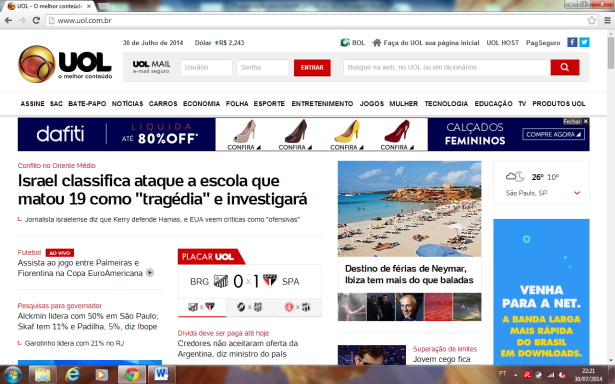
Quando alguém visita um site, o que faz com que ele tenha vontade de continuar navegando é a aparência, a facilidade de achar o que se procura e a sua utilidade. Por isso, devemos definir o design visual do site, que não precisa pular girar e pis­car, precisa sim ter um aspecto profissional seguindo os conceitos básicos do design, como: aproximação, contraste e alinhamento. O layout do site deve ser utilizado para transmitir o desejado ao visitante.

Devemos também desenvolver o site nos padrões web, ditados pela **W3C,** além de testá-lo em vários navegadores, pois há diferenças na visualização entre eles.

**Sites recomendados:** [**www.maujor.com**](http://www.maujor.com) **e** [**www.w3.org**](http://www.w3.org) **.**

## 2.1 - Recursos mínimos necessários para o desenvol­vimento de um site

Há vários softwares que facilitam a criação de páginas web, como por exemplo: Dream Weaver, Front Page, Fireworks, entre outros. Porém, nesta disciplina, o faremos através de comandos, utilizando para isto apenas um editor de texto e um navegador.

**Figura 1: Tela do Editor de Texto Notepad++ Figura 2: Tela do Google Crome - Navegador Web**

Conforme foi comentado estaremos utilizando o padrão Web, ou Web Standard ditado pela W3C, isso garante que os códigos da nossa página sejam abertos até em navegadores mais simples.

# 3 – O HTML 5

O HTML5 é a nova versão do HTML4 e um dos seus principais objetivos 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 cria novas tags e modifica a função de outras. As versões antigas do HTML não continham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, sidebar, menus e etc. Não havia um padrão de nomenclatura de IDs, Classes ou tags. Não havia um método de capturar de maneira automática as informações localizadas nos rodapés dos websites.

## 3.1 - O que é tag?

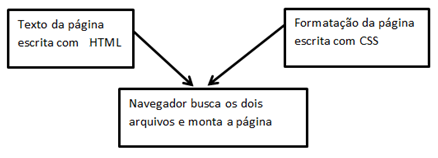
Tags são palavras de códigos contidas entre sinais de menor (<) e maior (>) na linguagem HTML de descrição do documento, por exemplo: **<h1>**AQUI VAI O TITULO DO DOCUMENTO**</h1>**.

A maioria das tags tem que ser abertas e fechadas, porém, algumas aparecem sozinhas, como por exemplo, a tag**<br />** que no HTML 5 não precisam mais ser fechadas.

## 3.2 –Características do HTML5 com CSS 3

Quando estamos escrevendo os códigos de uma página com o HTML5somente faremos o bloco de texto de forma semântica sem se preocupar com sua aparência. Para formatar a aparência criaremos outro grupo de códigos com o uso do CSS (**Cascading Style Sheets ou Folha de estilo em cascata**).

No momento que o navegador interpretar a página HTML5 ele automaticamente buscará os itens que formatarão sua aparência no CSS.



## 3.3 –Regras para programar em HTML

• Os textos devem ser bem formados.

• Todas as tags devem ser escritas com letras minúsculas.

• Tags devem estar convenientemente aninhadas.

• Algumas tags devem ser fechadas.

• Diferenças na sintaxe dos atributos.

# 4- ESTRUTURA DE UMA PÁGINA HTML5

## 4.1 - Estrutura obrigatória de um documen­to HTML5

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<title>TíTULO DO DOCUMENTO</title>

</head>

<body>

conteúdo do site

</body>

</html>

O Doctype deve ser a primeira linha de código do documento antes da tag HTML e indica para o navegador e para outros meios qual a especificação de código deverá ser utilizada. Em versões anteriores, era necessário referenciar o DTD diretamente no código do Doctype. Com o HTML5, a referência por qual DTD utilizar é responsabilidade do Browser (navegador).

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.

## 4.2–A Tag html

O código HTML indica que está começando uma página de Html.

<html lang=”pt-br”>

O atributo LANG é necessário para que os user-agents saibam qual a linguagem principal do documento.

Lembre-se que o atributo LANG não é restrito ao elemento HTML, ele pode ser utilizado em qual­quer outro elemento para indicar o idioma do texto representado.

Para encontrar a listagem de códigos das linguagens, acesse:

<http://www.w3.org/International/questions/qa-choosing-language-tags>.

## 4.3–A Tag Head

A Tag HEAD é onde fica toda a parte inteligente da página. No HEAD ficam os Metadados. Metadados são informações sobre a página e o conteúdo ali publicado.

Metatag Charset

No nosso exemplo há uma metatag responsável por chavear qual tabela de caractéres 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 também é suportada no HTML5. Contudo, é melhor que se utilize a nova forma.

**OBSERVAÇÃO:** A Web é acessada por pessoas do mundo inteiro. Ter um sistema ou um site que limite o acesso e pessoas de outros países é algo que vai contra a tradição e os ideais da internet. Por isso, foi criada uma tabela que suprisse essas necessidades, essa tabela se chama **Unicode**. A **tabela Unicode suporta algo em torno de um milhão de caracteres**. Ao invés de cada região ter sua tabela de ca­racteres, é muito mais sensato haver uma tabela padrão com o maior número de caracteres possí­vel. Atualmente a maioria dos sistemas e browsers utilizados por usuários suportam plenamente Unicode. Por isso, fazendo seu sistema Unicode você garante que ele será bem visualizado aqui, na China ou em qualquer outro lugar do mundo.

O que o Unicode faz é fornecer um único número para cada caractere, não importa a plataforma, nem o programa, nem a língua.

## 4.4 –A Tag LINK

Há dois tipos de links no HTML:

A tag **A**, que são links que levam o usuário para outros documen­tos e

A tag **LINK**, que são links para fontes externas que serão usadas no documento.

No nosso exemplo há uma tag **LINK** que importa o **CSS** para nossa página:

<link rel=”stylesheet” type=”text/css” href=”estilo.css”>

O atributo rel=”stylesheet” indica que aquele link é relativo a importação de um arquivo referente a folhas de estilo.

Há outros valores para o atributo REL, como por exemplo, o ALTERNATE:

<link rel=”alternate” type=”application/atom+xml” title=”feed” href=”/feed/”>

Neste caso, indicamos aos user-agents que o conteúdo do site poder ser encontrado em um cami­nho alternativo via Atom FEED.

No HTML5 há outros links relativos que você pode inserir como o rel=”archives” que indica uma referência a uma coleção de material histórico da página. Por exemplo, a página de histórico de um blog pode ser referenciada nesta tag.

# 5- DESENVOLVENDO UMA PÁGINA SIMPLES COM HTML5

## 5.1 - Exemplo de um arquivo HTML5

Crie uma Nova Pasta com o nome de **HOME PAGE**, Abra o editor **NOTEPAD ++**clique na guia **LANGUAGE** e escolha a opção **HTML**.

Copie o código abaixo e salve na pasta **HOME PAGE** com o nome de **index.html.**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<script src="arquivo.js"></script>

<title> ANIMAIS VENENOSOS</title>

</head>

<body>

<h1>BEM VINDOS A MINHA HOME PAGE!!!</h1>

<p>Esta Home Page se destina </p>

<p> a informar sobre os procedimentos </p>

<p> em caso de acidentes com animais venenosos.</p>

<p> Para maiores informações clique no link abaixo:</p>

<p><a href="http://www.butantan.gov.br/">INSTITUTO BUTANTA</a></p>

</body>

</html>

Abra o arquivo no navegador **Google Crome** para visualizá-lo.

**OBSERVAÇÃO**: Se o navegador não exibir acentuação gráfica, use a tag**<meta charset="iso-8859-2">** no lugar de**<meta charset="UTF-8">.**

As tags<h1> e </h1> criam um título com tamanho pré configurado e as tags<p> e </p> montam linhas de parágrafos.

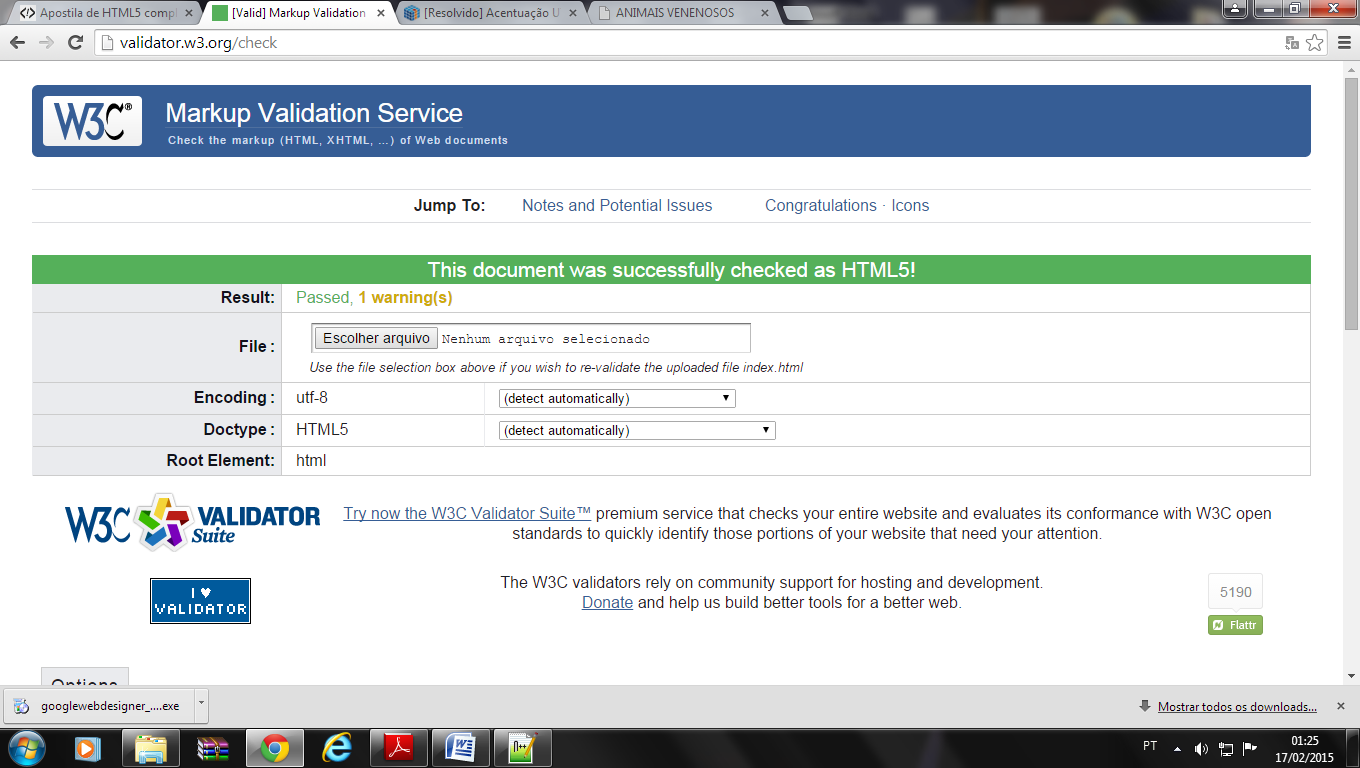
As Tags<a href="http://www.butantan.gov.br/home/">e </a>criam um link para a página do instituto butantan.

Nossos exercícios em HTML5 serão todos testados com o navegador aberto para a visualização dos resultados e o notepad++ aberto para digitação dos códigos.

# 6 –VALIDANDO UM DOCUMENTO

Para verificar se os códigos estão no padrão HTML 5, existe uma ferramenta online no W3C que faz o teste de conformidade.

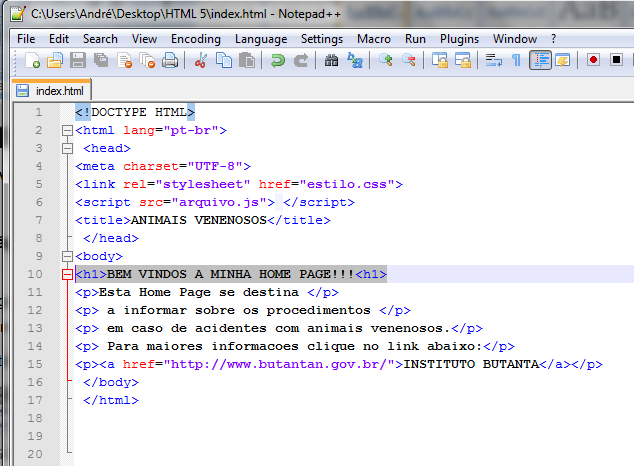
Entre na internet no endereço **http://validator.w3.org/**clique na aba **Vali­date by File Upload**, selecione o seu arquivo e clique em **“Check”.** Se tudo estiver certo verá a mensagem **“success fully checked”.** No caso de erros, o sis­tema apresenta as tags que não estão em conformidade com o html 5. Corrija os códigos e tente novamente.

****

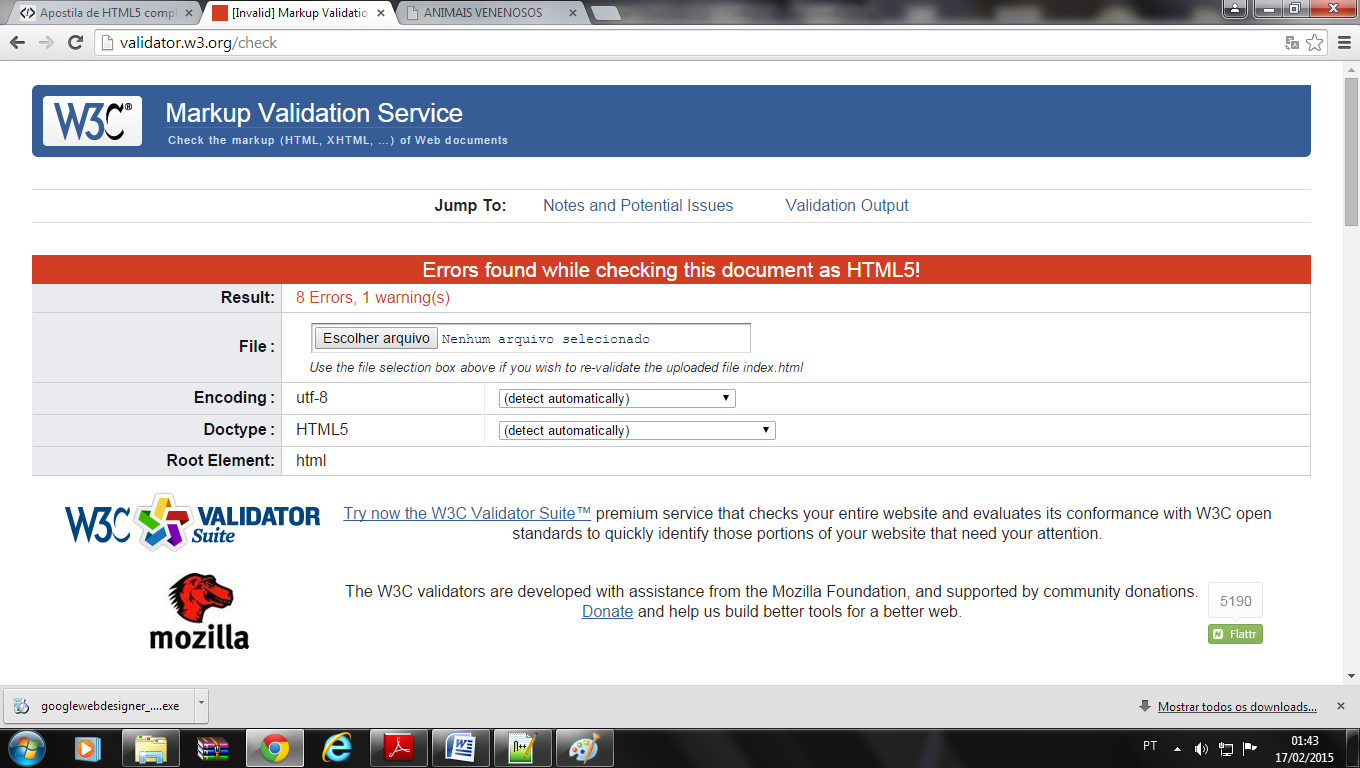
**Figura 4: Resultado da Validação bem-sucedida da Primeira Página**

## 6.1 - Testando páginas com erro

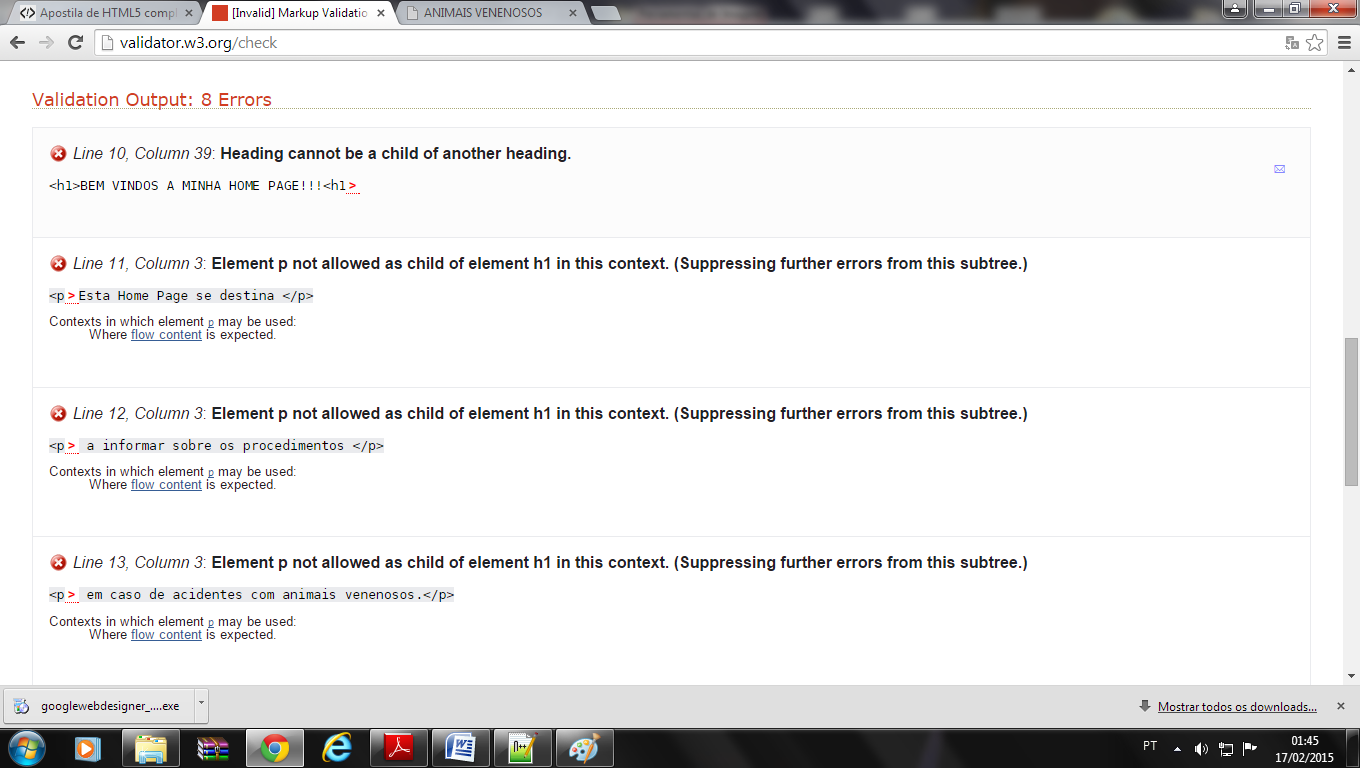
Vamos agora gerar um arquivo com erro para aprendermos o que ocorre quando este é validado. Para isso, iremos inserir um erro na linha 10 onde não será fechado a tag de cabeçalho <h1>.

****

Resultado da Validação:

****

Erros apontados pelo validator:

****

Podemos observar que apesar do erro se encontrar no final da linha 10, o validador mostrou vários outros erros. Corrija a linha 10, salve o arquivo novamente e revalide-o.

# 7 – FORMATANDO NOSSO HTML 5 COM CSS3

Podemos perceber até agora que os códigos **HTML5** que escrevemos não possuem nenhum tipo de formatação (cores, tamanho de letras, espaçamento), essa formatação é feita em um arquivo externo, com extensão **.css**.

**CSS** significa **CascadingStyleSheet**, traduzindo: **folhas de estilo em cascata** e toda a tarefa de estilização da pagina HTML ficará a cargo do CSS, que nada mais é do que um arquivo independente do arquivo HTML no qual são declarados propriedades e valores de estilização para os elementos do HTML.

## 7.1 – Criando o CSS

Com o NOTEPAD ++ aberto, configure na guia **LANGUAGE** para a linguagem **CSS**. Copie o código de CSS abaixo e salve na pasta HOME PAGE com o nome estilo.css.

BODY {background-color: #00ff00;}

H1 {

font-family: 'Comic Sans MS';

font-size: 36pt;

color: blue;

}

P {font-family: 'Courier';

margin-left: 0.5in;

color: red;}

A {color: black;}

Podemos perceber que deixamos em nos códigos de HTML um link chamando um arquivo de css com o nome de estilo.css, então é só atualizar a página para que os novos estilos sejam aplicados. Observe na seta abaixo o formato da tag e sua posição na pagina **index.html.**

<!DOCTYPE HTML>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="estilo.css">

<script src="arquivo.js"></script>

<title>ANIMAIS VENENOSOS</title>

</head>

<body>

<h1>BEM VINDOS A MINHA HOME PAGE!!!</h1>

<p>Esta Home Page se destina <p>

<p> a informar sobre os procedimentos <p>

<p> em caso de acidentes com animais venenosos.<p>

<p> Para maiores informacoes clique no link abaixo:<p>

<p><a href="http://www.butantan.gov.br/">INSTITUTO BUTANTA</a><p>

</body>

</html>

Vamos agora entender como utilizar o CSS.

Podemos definir regras de CSS de três formas diferentes podendo combiná-las livremente nas suas páginas. A maneira como as regras interagem entre si está relacionada à parte “em cascata”. As três formas são:

**1) Em um documento separado fora de todos os documentos HTML, chamado método EXTERNO:**

O Método Externo significa que você coloca as regras de CSS em um arquivo separado, e então sua página HTML pode fazer um link para esse arquivo. Essa abordagem lhe permite definir regras em um ou mais arquivos que podem ser aplicadas em alguma página do seu web site.

Para definir um conjunto de regras de estilo, que você pode facil­mente aplicar em alguma página do seu site, é preciso colocar as regras em um arquivo de texto. Você pode criar este arquivo com um editor de textos simples e dar ao nome desse arquivo a extensão .css.

Sempre que quiser utilizar esses estilos em uma nova página, basta colocar uma tag<LINK> no cabeçalho que referencie esse arquivo .css, como o exemplo abaixo:

Não é mais utilizado em HTML 5

<link rel="stylesheet" href="estilo1.css" type="text/css" />

Caso o arquivo css esteja na internet, a tag ficaria assim:

<link rel="stylesheet" href="http://www.industrial.com.br/estilo1.css">

**2) No cabeçalho de um documento HTML. Chamamos esse método de INCORPORADO.**

Incorporado significa que você especifica as regras de CSS no ca­beçalho do documento. As regras incorporadas afetam somente a página atual.

Se quisermos criar um conjunto de estilos que se aplicam somente a uma úni­ca página, podemos configurar os estilos exatamente como fizemos no método externo, porém em vez de criar um arquivo separado colocamos os comandos entre as tags<style>...</style>dentro da própria página HTML. A estrutura básica de uma página da web que utiliza estilos incorporados é semelhante ao seguinte código:

<head>

<title>Exemplo Estilos Incorporados</title>

<style>

P {background-color: #FFFFFF;

Como exercício, insira esses códigos dentro da página inicial.html para visualizar os resultados.

font-family:'Comic Sans MS';

font-size: 14pt;}

</style>

</head>

<body></body></html>

**3) Dentro de uma tag de HTML, pelo método denominado INLINE:**

Inline significa que você especifica as regras de CSS dentro da tag de HTML. Os estilos inline são os que têm menos efeitos. Eles afetam somente a tag atual, não outras tags na página e tampouco outros documentos. A sinta­xe para definir um estilo inline é a seguinte:

<style="regrascss">

Exemplo:

<p><a style="color: green; text-decoration: none" href="http://www.uol.com.br">UOL</a></p>

Note que em vez das tags<style>...</style>, apenas utilizamos um atributo style dentro da tag para definir o estilo, e, em vez de colocar as regras de CSS entre colchetes, colocamos entre aspas, separando-as com ponto-e-vírgula como de costume.

## 7.2–Entendendo os Códigos de CSS

Cada estilo que você cria é definido como uma regra CSS. Cada regra deve utilizar a seguinte sintaxe:

elemento {atributo1: valor; atributo2: valor ...;}

**Elemento** - descreve o elemento de design ao qual o estilo será aplicado. A mesma tag HTML, mas sem os sinais de maior e menor. Essa parte da regra é, às vezes, chamadas de **seletor**.

**Atributo** - o aspecto específico do elemento que você quer usar como estilo. Deve ser um nome de atributo CSS válido, como o atributo font-size.

**Valor** - a configuração aplicada ao atributo. Deve ser uma confi­guração válida para o atributo em questão, como 20pt (20 pontos) para font-size.

Podemos atribuir múltiplas declarações no “**Atributo:Valor**” , bastando para isso separá-los com ponto-e-vírgula (;).

Como exemplo criaremos uma regra CSS que especifica que todos os títulos de nível 1 (tags<h1>) sejam exibidos em uma fonte de 36 pontos, a regra ficaria como o exemplo abaixo:

H1 {font-size: 36pt;}

Agora outro exemplo de uma regra que diz que todos os títulos de nível 2 (tags<h2>) devem ter tamanho de 24 pontos e cor azul;

H2 {font-size: 24pt; color: #0000FF;}

Podemos pode inserir quebras de linha e espaços em branco dentro das regras como quisermos. Assim, é possível ver mais facilmente todas as declarações e certi­ficar-se de que foi colocado todos os sinais de ponto-e-vírgula e colchetes nos lugares corretos. Por exemplo: aqui está uma regra que diz que os parágra­fos aparecerão em fonte Times, 12 pontos, azul e recuados meia polegada a partir da margem esquerda da página:

P {

font-family: Times;

font-size: 12pt;

color: blue;

margin-left: 0.5in;

}

É possível mudar praticamente qualquer aparência. Por exemplo: para a cor de fundo usa-se background-color:

BODY{

color: red;

background-color: green;}

**Importante:** Podemos visualizar a tabela de cores utilizadas na web no site:

[**http://www.mxstudio.com.br/Conteudos/Dreamweaver/Cores.htm**](http://www.mxstudio.com.br/Conteudos/Dreamweaver/Cores.htm)ou no site

**http://www.flextool.com.br/tabela\_cores.html**

Muitas vezes não queremos todo o texto da página da mesma cor, quase sempre não. Então, podemos aplicar mudanças para cada tagHTML individualmente:

H1 { color: blue;}

H2, H3, H4, H5, H6 {color: gray;}

O segundo exemplo mostra que é possível aplicar a mesma modifi­cação a vários seletores simultaneamente. Aqui indicamos que a fonte dos títulos H1 são azuis e dos outros títulos cinza.

## 7.3 - Utilizando figuras como fundo

Você já deve ter alguma imagem gravada na sua pasta de trabalho, caso ainda não tenha, pesquise e salve alguma da internet, \*.gif ou \*.jpg. É possível aplicar um padrão repetido por todo o bloco usando:

BODY{background-image: url("rosa.jpg");}

OU

P{background-image: url("rosa.jpg");}

Repetindo a imagem, dentro da formatação do parágrafo acima, escolha uma das opções abaixo:

background-repeat: no-repeat; /\* a imagemsomente uma vez \*/

background-repeat: repeat-y; /\* a imagem repetena vertical \*/

background-repeat: repeat-x; /\* a imagem repetena horizontal \*/

background-repeat: repeat; /\* a imagem repete navertical e horizontal (padrão) \*/

Para manter o fundo fixo enquanto rola-se a página:

background-attachment: fixed;

background-attachment: scroll; /\* rolagem de forma padrão \*/

Posicionar a imagem em um local específico:

background-position: x-pos y-pos ; x-% y-% ; top

left ; top center ; top right ; center left ;

centercenter ; center right ; bottom left ;

bottom center ; bottom right ;

Exemplo:

p{background-image: url("nomearquivo.gif");

background-repeat: no-repeat;

background-position: center center;}

Como os outros parâmetros, podemos escrever um grupo de op­ções de uma vez só:

p{background: url("nomearquivo.gif") no-repeat center center fixed;}

## 

# 8 – CRIANDO TITULOS E LISTAS NO HTML5

## 8.1 - Criando Títulos no HTML

Como em todo texto, as páginas Web também possuem vários títu­los, há 6 níveis de títulos: h1, h2, h3, h4, h5 e h6. O nível 1 é para os títulos mais importantes da página e os outros devem ser usados para ca­racterizar subtítulos.

A hierarquia dos títulos é construída de acordo com a importância das informações e não em relação à aparência. Quanto a aparência, podemos alterá-la através do CSS.

Abra o arquivo índex.html no notepad++,inserir os códigos de títulos abaixo salve novamente e abra no navegador.

<p>NIVEIS DE TITULOS</p>

<h1> primeiro nível</h1>

<h2> segundo nível</h2>

<h3> terceiro nível</h3>

<h4> quarto nível</h4>

<h5> quinto nível</h5>

<h6> sexto nível</h6>

## 8.2 - Listas Ordenadas Não Ordenadas e Listas de Definição

Utilizamos listas para enumerar ou listar itens, é um recurso bastan­te utilizado. Existem dois tipos: numeradas e não numeradas ou ordenadas e não ordenadas. As listas numeradas (listas ordenadas) utilizam a tag<ol>e </ol> e para cada item da lista usamos o <li> e </li> como mostrado abaixo:

<p> LISTAS ORDENADAS </p>

<ol>

<li> PRIMEIRO ITEM DA LISTA </li>

<li> SEGUNDO ITEM DA LISTA </li>

<li> TERCEIRO ITEM DA LISTA </li>

<li> QUARTO ITEM DA LISTA </li>

</ol>

Da mesma forma as listas não numeradas (não ordenadas) utilizam a tag<ul></ul> e para cada item da lista usamos o <li> e </li> como mostrado abaixo:

<p> LISTAS NAO ORDENADAS </p>

<ul>

<li> PRIMEIRO ITEM DA LISTA </li>

<li> SEGUNDO ITEM DA LISTA </li>

<li> TERCEIRO ITEM DA LISTA </li>

<li> QUARTO ITEM DA LISTA </li>

</ul>

Temos também um outro tipo de lista chamada lista de definição que cria uma formatação especifica para textos. Nesta lista temos a tag<dl> e </dl> para criar a lista. As tags<dt> e </dt> para criar os títulos da lista e as tags<dd> e </dd> para detalhar os textos dos títulos. Veja o exemplo abaixo:

<p> LISTAS DE DEFINICAO </p>

<dl>

<dt>CARBOIDRATOS</dt>

<dd>Abrangem um dos grandes grupos de bio moleculas na natureza,

alem de serem a mais abundante fonte de energia.</dd>

<dt>LIPIDIOS</dt>

<dd>São constituídos por carbono(em maior numero), hidrogênio e oxigênio,

fornecendo 2,23 vezes mais energia/kg quando da oxidação</dd>

<dt>PROTEINAS</dt>

<dd>São compostos orgânicos de alto peso molecular,

sao formadas pelo encadeamento de aminoácidos</dd>

</dl>

**TABELAS**

# 9.0 -TAGS PERSONALIZADAS COM USO DE IDENTIFICADORES E CLASSES

Existem situações que desejamos modificar um único parágrafo, um título apenas ou um grupo de parágrafos e não todos os elementos da página, para isso podemos utilizar o identificador (**id**) ou a classe (**class**)**.**

• **id**: identifica **um único elemento na página**. É como um RG do elemento, e justamente por isso, **não pode ser repetido**. Por exemplo: Nos códigos abaixo iremos inserir um **id** para o parágrafo ALUNOS e um para os PROFESSORES.

<p id="Alunos">ALUNOS</p>

<p>Marcos Augusto</p>

<p>Marcela Mendes</p>

<p>Jose Carlos</p>

<p>Antonio Marcos</p>

<p id="Professores">PROFESSORES</p>

<p>Josimar Lacerda</p>

<p>Maria Medeiros</p>

<p>Elismar Goncalves</p>

Após definir o **id** iremos criar o CSS para formatá-los. Para chamar um **id** no CSS, usaremos o caractere “**#**”. Veja o exemplo do CSS abaixo:

BODY {background-color:#FFFF00;}

P {color:#FF0000;}

#Alunos {

color:#FFFFFF;

background-color:#000000;}

#Professores {

color:#000000;

background-color:#FFFFFF;}

• **class**: Utilizamos o class quando for necessário utilizar uma só formatação para um grupo de tag (elementos), o class **pode ser repetido na página**.

<p>PERGUNTAS E RESPOSTAS</p>

<p class="pergunta">O QUE E HARDWARE?</p>

<p class="resposta">E A PARTE FISICA DA MAQUINA</p>

<p class="pergunta">O QUE E SOFTWARE</p>

<p class="resposta">CONJUNTO DE INSTRUCOES QUE FAZ O HARDWARE FUNCIONAR</p>

Uma vez definido a **class**, é possível aplicar um CSS diretamente a eles usando o caracter “**.**” (ponto).

Veja o exemplo do CSS abaixo:

BODY {background-color:#ffff00;}

P {color:#FF0000;}

.pergunta {

color: #000000;

word-spacing: 1.2em;}

.resposta {

background-color:#00FF00;

color: #ffffff;}

Algumas vezes a mesma tag pode ser identificado por um **id** e fazer parte de um **class**:

Vamos fazer uma modificação no exemplo anterior:

**Código HTML**(Foi inserido a **id** tamanho da fonte)

<p>PERGUNTAS E RESPOSTAS</p>

<p class="pergunta" id="tamanho1">O QUE E HARDWARE?</p>

<p class="resposta" id="tamanho2">E A PARTE FISICA DA MAQUINA</p>

<p class="pergunta" id="tamanho3">O QUE E SOFTWARE</p>

<p class="resposta" id="tamanho4">CONJUNTO DE INSTRUCOES QUE FAZ O HARDWARE FUNCIONAR</p>

**Código CSS** (Foi inserido a **id** formatando a **Class** pergunta e resposta)

BODY {background-color:#FFFF00;}

P {color:#FF0000;}

.pergunta {

color: #000000;

word-spacing: 1.2em;}

.resposta {

background-color:#00FF00;

color: #ffffff;}

#tamanho1 {

font-size: 8pt;}

#tamanho2 {

font-size: 12pt;}

#tamanho3 {

font-size: 16pt;}

#tamanho4 {

font-size: 18pt;}

## 9.1 - Utilizando a tag<div>

As tags<div>...</div> podem ser usadas para formatar um grande bloco de texto abrangendo diversos parágrafos e outros elementos, com isso podemos criar grandes seções no documento e definir estilos para formata-los. Observe o exemplo abaixo:

No arquivo HTML:

<div>

<p>EX PRESIDENTES</p>

<p>Franklin D. Roosevelt</p>

<p>Harry S. Truman</p>

<p>John F. Kennedy</p>

<p>Lyndon B. Johnson</p>

<p>Jimmy Carter</p>

<p>Bill Clinton</p>

</div>

No arquivo CSS:

DIV {font-family: "Arial";

font-size: 12pt;

text-align: right;

background-color: #C0C0C0;

margin-left: 100px;

margin-right: 100px;}

**9.2 - Utilizando a tag <span>**

As tags<span>...</span> são como tags<div>...</div> no sentido de que podemos utilizá-las para definir estilos que formatam um bloco de texto, porém, ao contrário de <div>, que é utilizada para divisões de texto grandes, a tag<span> é especializada para blocos de textos menores que podem ser tão pequenos como um único caractere. Deve ser utilizado com o seletor class. Observe o exemplo abaixo:

No arquivo HTML:

<p>Dormir cedo e acordar cedo faz o homem <span class="caract">saudavel</span>,

<span class="caract">rico</span> e <span class="caract">sabio</span>.</p>

No arquivo CSS:

span.caract {color:green;

text-decoration: underline;}

**9.4 - Inserindo comentários**

Podemos inserir comentários no CSS para explicar seu código ou ajudar a relembrá-lo em caso de esquecimento. Os comentários são ignorados pelo navegador e começam com o "/\*", e termina com ”\*/" . Veja o exemplo abaixo:

/\* escreva seu comentário aqui\*/

BODY{

color: red; /\* ou aqui\*/

background-color: green;}

COMENTARIO NO HTML:

<!-- comentário no html -->

## 10 - Margens, espaçamento, larguras e bordas

Abra um novo arquivo HTML e inserir um parágrafo com um texto, como o exemplo abaixo:

Código HTML:

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<script src="arquivo.js"></script>

<title> CRIANDO BLOCOS DE PARAGRAFOS</title>

</head>

<body>

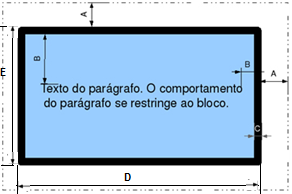
<p>Dormir cedo e acordar cedo faz bem ao homem.</p>

</body>

</html>

Cada bloco definido por um par de tags (<div>...</div>, <p>… </p>, ou <img… />) tem um conjunto de opções de aparência de bloco customizáveis com CSS: marging, padding, height, width, border.

Vejamos a figura abaixo:



Neste bloco as medidas referem-se a:

• **A** Margin: Margem transparente com a separação de um bloco e o bloco vizinho.

• **B** Padding: Espaçamento interno do bloco. Espaço entre o conteúdo do bloco e a borda.

• **C** Border: Largura da borda.

• **D** Width: Largura da Caixa de Texto.

• **E** Height: Altura da Caixa de texto.

Todas estas opções são configuráveis para cada um dos 4 lados do bloco individualmente: top, left, bottom, right.

O CSS ficaria assim:

P { margin-top: 12px;

margin-right: 10px;

margin-bottom: 20px;

margin-left: 12px;

padding-top: 10px;

padding-right: 12px;

padding-bottom: 12px;

padding-left: 12px;

padding-right: 12px;

color: #ffffff;

background: #000000;}

Pode-se também configurar em conjunto:

P{

margin: 15px 10px 20px 12px; /\* define na sequência: topo, direita, base, esquerda,

no sentido do relógio \*/

padding: 10px 12px; /\* define topo e base como 10px e laterais como 12px \*/

border-style: solid; /\* borda cheia para todos os lados \*/

border-width: 20px; /\* toda borda com 20pxde largura\*/

}

Para as bordas temos ainda estilo e cor:

p {

border-width: 10px 12px; /\* borda 10px de largura no topo e base e 12px nas laterais \*/

border-style: solid; /\* borda cheia para todos os lados \*/

border-color: #ff0000;}

As opções para estilo de borda são:

• dotted – série de pontos,

• dashed – tracejados,

• solid – linha contínua,

• double – duas linhas sólidas,

• groove – uma linha em baixo relevo

• ridge – uma linha em alto relevo

• inset – simula o bloco todo inserindo na tela

• outset – simula o bloco todo saltando para fora da tela

Já para a largura da borda (width) além das medidas em pixels (px) ou caracteres (em), temos as opções abaixo:

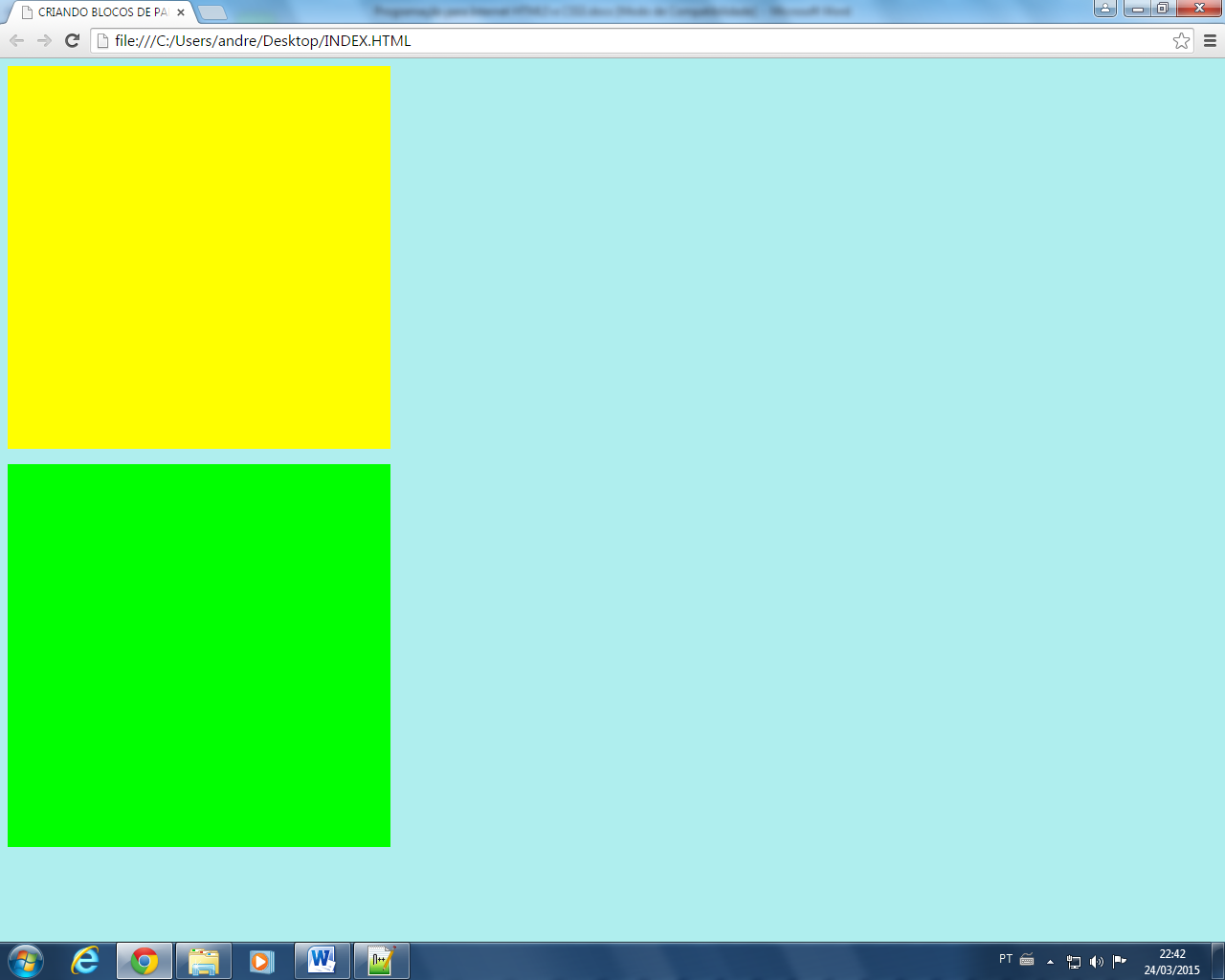
• thin: fina

• medium: média

• thick: grossa

# 11 - PARÂMETRO FLOAT

Os blocos <div> se comportam exatamente como os parágrafos <p>. Quando colocamos um parágrafo na página e definimos uma largura menor que a largura da página é evidente que sobrará espaço a direita do bloco. Digamos que o bloco em azul claro é o fundo do navegador, e os blocos amarelo e verde são dois parágrafos. Veja a figura abaixo:



Vamos inserir os códigos de HTML abaixo para gerar essa tela:

Códigos de **HTML:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<title> CRIANDO BLOCOS DE PARAGRAFOS</title>

</head>

<body>

<div id="amarelo"></div>

<p></p>

<div id="verde"></div>

</body>

</html>

Usamos CSS para definir o parágrafo amarelo e o bloco verde com largura 100px:

Codigos em **CSS:**

BODY {background-color: #afeeee;}

#amarelo {background-color: #ffff00;}

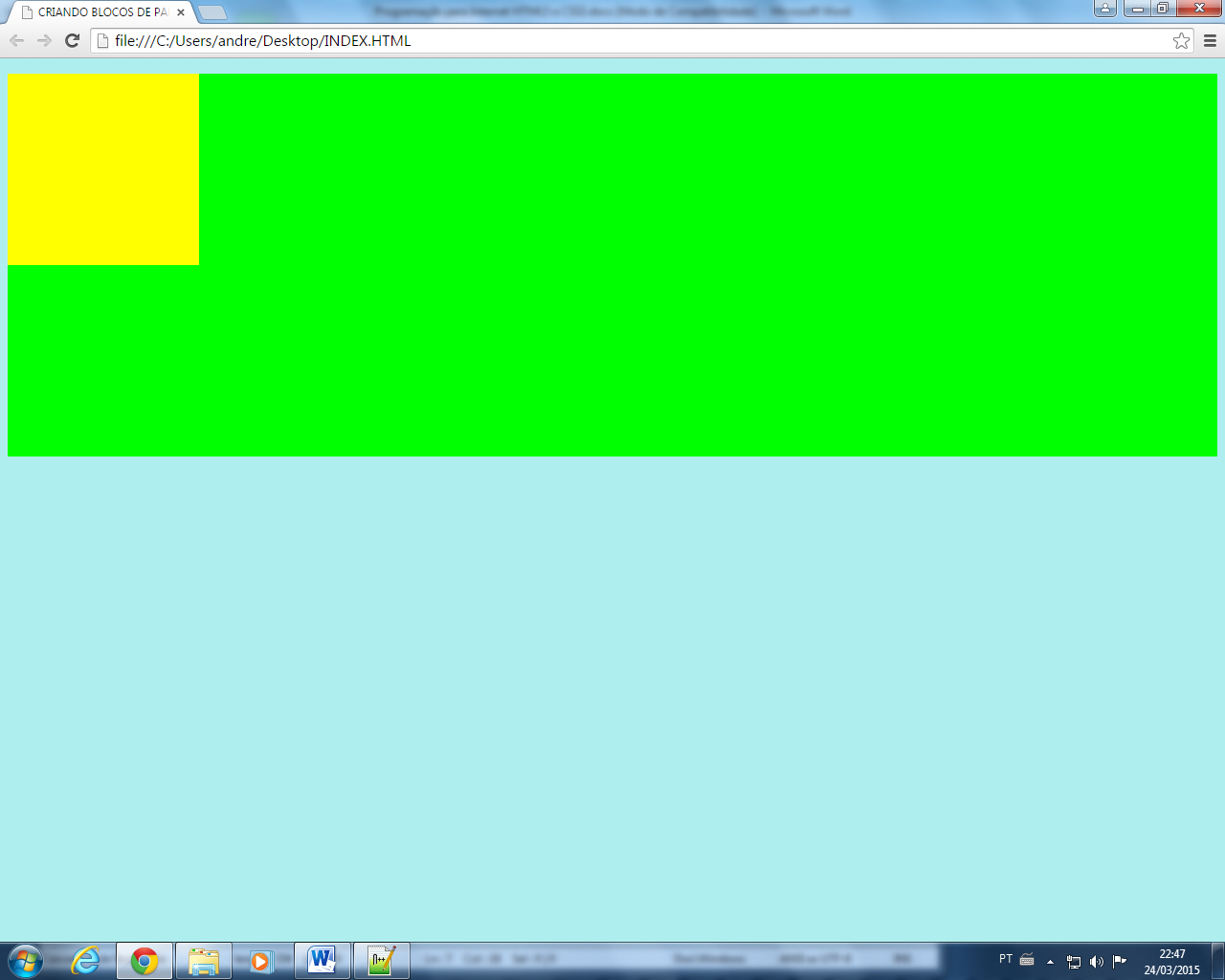
#verde {background-color: #00ff00;}

#amarelo, #verde {width:400px;

height: 400px;}

Mesmo existindo espaço para os dois parágrafos ficarem lado a lado, como são blocos, eles terminam com quebra de linha e o bloco verde continua abaixo.

O parâmetro float permite que um parágrafo simplesmente não quebre a linha no final. O comportamento é de flutuação. O bloco pode ser colocado a esquerda e o restante vai contornar conforme a figura:



Aqui os códigos de **CSS** formatou o bloco amarelo com os seguintes parâmetros:

BODY {background-color: #afeeee;}

#amarelo {background-color: #ffff00;}

#verde {background-color: #00ff00;}

#amarelo { height: 200px;}

#verde { height: 400px;}

#amarelo{width: 200px;float: left; /\* parágrafo amarelo flutuando a esquerda \*/}

#verde{width:100%;/\* parágrafo verde ocupando toda largura da tela horizontal \*/}

Para colocar os dois parágrafos um ao lado do outro acrescentamos **float: left;** no paragrafo verde:

#amarelo{width: 200px;float: left; /\* parágrafo amarelo flutuando a esquerda \*/}

#verde{width:500px; float: left;/\* parágrafo verde flutuando também a esquerda \*/}

# 12 - ESTRUTURA DO HTML 5

**12.1 - Marcações semânticas e layout de página**

Quando estruturamos uma página, geralmente utilizamos o **Div** com **Class** ou **Id** para fazer a divisão do documento. Apesar de funcionar muito bem esse marcador não diz nada sobre o conteúdo do bloco.

A marcação Semântica são Tags desenvolvidas para o HTML 5 que descrevem seu significado ou propósito claramente ao navegador e ao desenvolvedor.

O W3C investigou milhões de páginas da Web existentes para descobrir as IDs e nomes de classe que os desenvolvedores já estavam usando e com isso chegaram a uma lista de elementos descritivos ricos que já estavam sendo usados e os transformaram nos padrões.

Abaixo estão alguns dos novos elementos semânticos em HTML5:

* article
* aside
* figcaption
* figure
* footer
* header
* hgroup
* mark
* nav
* section
* time

Abaixo temos um exemplo de estrutura utilizada com estes novos marcadores:



**Headers** e **footers** criam cabeçalhos e rodapés.

**nav** cria uma barra de navegação ou de menu.

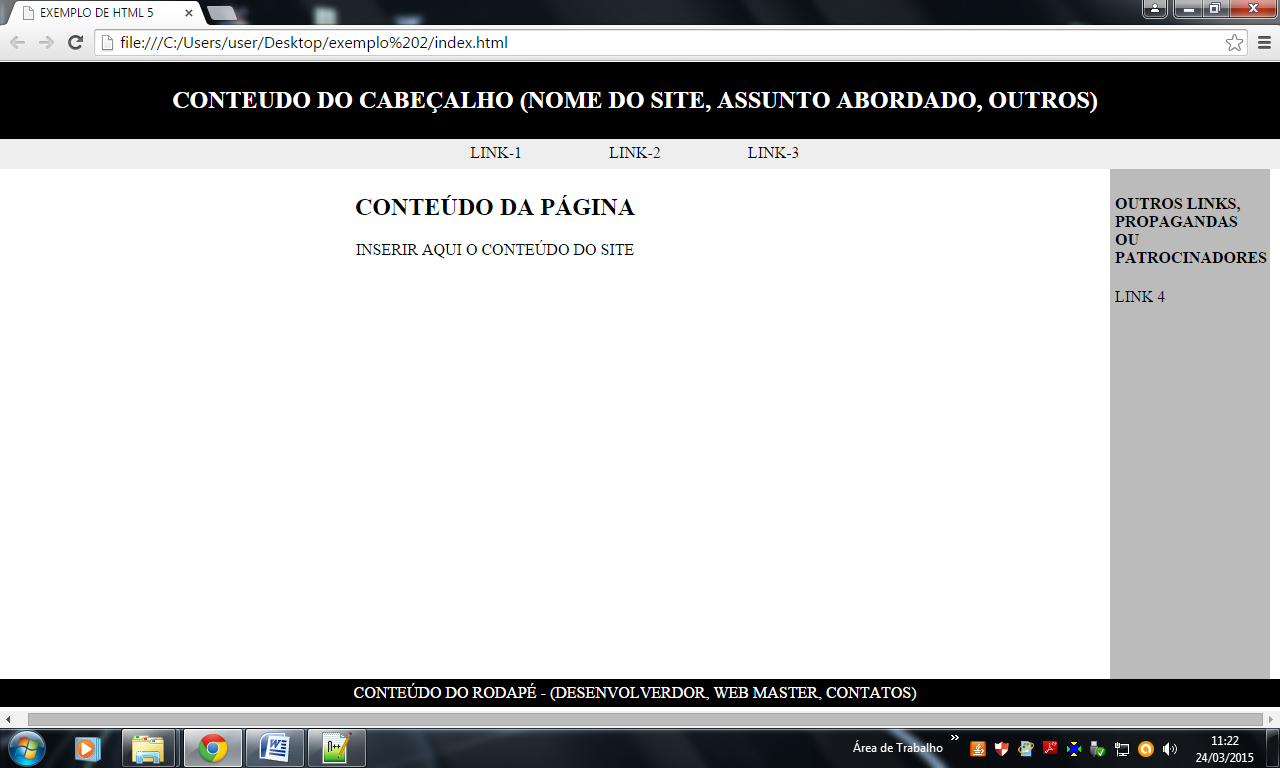
Podemos usar **sections** e **articles** para agrupar um conteúdo.

O elemento **aside** pode ser usado para conteúdo secundário, por exemplo, uma barra lateral de links relacionados.

Aqui está um exemplo simples de um código que usa esses elementos.

**12.2 – Exemplos de estrutura para Sites:**

**Exemplo 1:**



No **HTML 5:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<title>EXEMPLO DE HTML 5</title>

</head>

<body>

<header>

<h2>CONTEUDO DO CABEÇALHO (NOME DO SITE, ASSUNTO ABORDADO, OUTROS)</h2>

</header>

<nav>

LINK-1

LINK-2

LINK-3

</nav>

<section>

<h2>CONTEÚDO DA PÁGINA</h2>

<p>

INSERIR AQUI O CONTEÚDO DO SITE

</p>

</section>

<aside>

<h4>OUTROS LINKS, PROPAGANDAS<br>OU PATROCINADORES</h4>

<p>

LINK 4

</p>

</aside>

<footer>

CONTEÚDO DO RODAPÉ - (DESENVOLVERDOR, WEB MASTER, CONTATOS)

</footer>

</body>

</html>

No **CSS:**

body {margin: 1px;

width: 100%;

height: 100%;

}

header {

width: 100%;

height: 13%;

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

background-color:#eeeeee;

height:20px;

width:100%;

padding:5px;

text-align: center;

word-spacing: 5.2em;

}

section {

height:630px;

width:1000px;

float:left;

padding:5px;

text-align:center;

}

aside {

height:630px;

width:150px;

background-color:#bbbbbb;

float:right;

padding:5px;

}

footer {

width: 100%;

height: 8%;

background-color:black;

color:white;

clear:both;

text-align:center;

padding:5px;

}

**Exemplo 2:**



No  **HTML 5:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="estilo.css">

<title>EXEMPLO DE HTML 5</title>

</head>

<body>

<header>

<h1>CONTEUDO DO CABEÇALHO (NOME DO SITE, ASSUNTO ABORDADO, OUTROS)</h1>

</header>

<nav>

LINK 1<br>

LINK 2<br>

LINK 3<br>

</nav>

<section>

<h1>CONTEÚDO DA PÁGINA</h1>

<p>

INSERIR AQUI O CONTEÚDO DO SITE

</p>

</section>

<aside>

<h3>OUTROS LINKS, PROPAGANDAS<br>OU PATROCINADORES</h3>

<p>

LINK 4

</p>

</aside>

<footer>

CONTEÚDO DO RODAPÉ - (DESENVOLVERDOR, WEB MASTER, CONTATOS)

</footer>

</body>

</html>

No **CSS:**

body {margin: 1px;

width: 100%;

height: 100%;

}

header {

width: 100%;

height: 13%;

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

line-height:30px;

background-color:#eeeeee;

height:630px;

width:100px;

float:left;

padding:5px;

}

section {

height:630px;

width:350px;

float:left;

padding:5px;

}

aside {

height:630px;

width:150px;

background-color:#eeeeee;

float:right;

padding:5px;

}

footer {

width: 100%;

height: 8%;

background-color:black;

color:white;

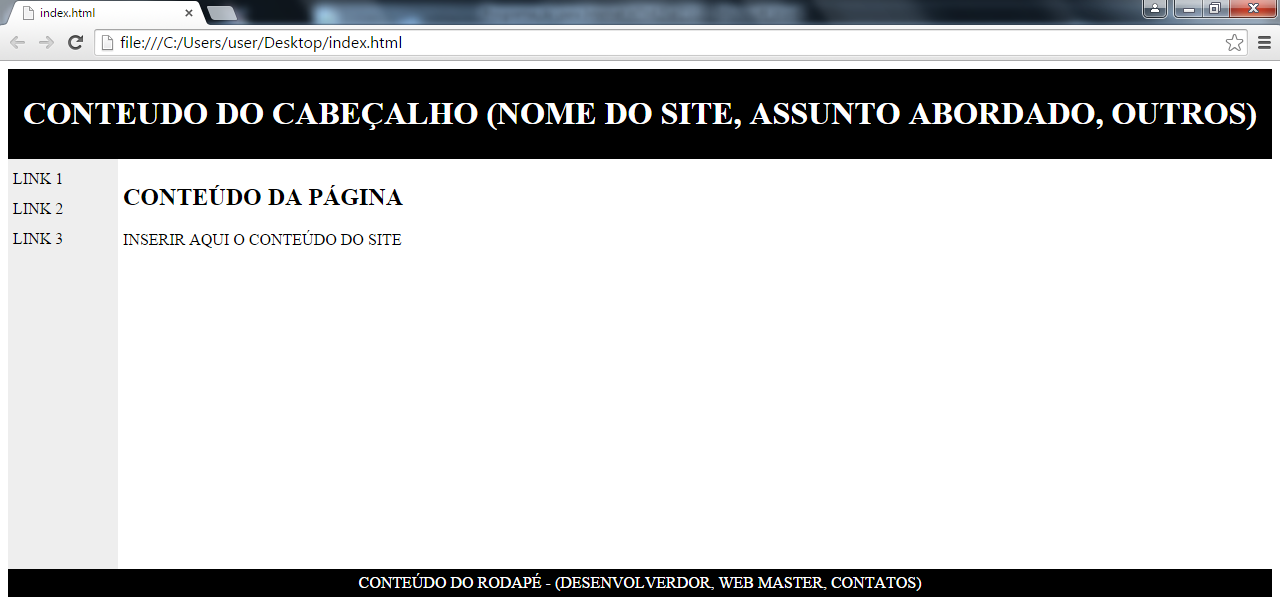
clear:both;

text-align:center;

padding:5px;

}

**Exemplo 3**:



Códigos de **HTML 5:**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="8751-2">

<link rel="stylesheet" href="estilo.css">

<title>EXEMPLO DE HTML 5</title>

</head>

<body>

<header>

<h1>CONTEUDO DO CABEÇALHO (NOME DO SITE, ASSUNTO ABORDADO, OUTROS)</h1>

</header>

<nav>

LINK 1<br>

LINK 2<br>

LINK 3<br>

</nav>

<section>

<h1>CONTEÚDO DA PÁGINA</h1>

<p>

INSERIR AQUI O CONTEÚDO DO SITE

</p>

</section>

<footer>

CONTEÚDO DO RODAPÉ - (DESENVOLVERDOR, WEB MASTER, CONTATOS)

</footer>

</body>

</html>

Códigos de **CSS:**

header {

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

line-height:30px;

background-color:#eeeeee;

height:630px;

width:100px;

float:left;

padding:5px;

}

section {

height:630px;

width:350px;

float:left;

padding:5px;

}

footer {

background-color:black;

color:white;

clear:both;

text-align:center;

padding:5px;

}

**X.1 – Ajustando o Layout para qualquer resolução de tela**

Usando o exemplo 3, vamos ajustando as tagspara adaptar a qualquer resolução de vídeo.

No **CSS** acrescentar width e height com porcentagens:

body {margin: 0px;

width: 100%;

height: 100%;}

header {

width: 100%;

height: 13%;

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

width: 20%;

height: 81%;

line-height:30px;

background-color:#eeeeee;

height:400px;

width:100px;

float:left;

padding:5px;

}

section {

width: 80%;

height: 81%;

height:400px;

width:350px;

float:left;

padding:5px;

}

footer {

width: 100%;

height: 6%;

background-color:black;

color:white;

clear:both;

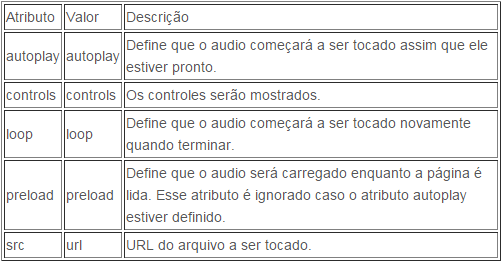
text-align:center;

padding:5px;

}

# 13 - INSERINDO MUSICA COM HTML 5

O HTML5 nos trouxe diversas novidades e facilidades, uma delas é permitir que possamos inserir players de áudio diretamente em nosso site através da [tag áudio](http://www.devmedia.com.br/html5-as-tags-audio-e-video/26018), sem precisar de nenhum tipo de plugin externo para isso, e o melhor de tudo, de uma forma super simples e intuitiva.  
  
Existem diversas formas de colocarmos um áudio em nossa página, como mostra a tabela a seguir:



## Autoplay

Define que o áudio começara automaticamente assim que for carregado.

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<title>HTML5 Audio Autoplay</title>

</head>

<body>

<audio autoplay="autoplay">

<source src="seu\_arquivo\_de\_audio.mp3" type="audio/mp3" />

</audio>

</body>

</html>

## Controls

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<title>HTML5 Audio Autoplay</title>

</head>

<body>

<audio controls="controls">

<source src="seu\_arquivo\_de\_audio.mp3" type="audio/mp3" />

</audio>

</body>

</html>

Referencias

<http://www.linhadecodigo.com.br/artigo/3621/html5-entendendo-a-tag-audio.aspx>