



Curso Técnico em Desenvolvimento de Sistemas Online

DESENVOLVIMENTO WEB

GEEaD - Grupo de Estudo de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Expediente

GEEaD – CETEC
GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
PROGRAMAÇÃO WEB I

Autor:
Paulo Eduardo Cardoso Andrade

Revisão Técnica:
Eliana Cristina Nogueira Barion
Lilian Aparecida Bertini

Revisão Gramatical:
Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:
Flávio Biazim

APRESENTAÇÃO

Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.

O material foi elaborado para servir de apoio aos estudos dos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.

Esperamos que este livro possa contribuir para uma melhor formação e aperfeiçoamento dos futuros Técnicos.

AGENDA 12

HTML - UMA LINGUAGEM DE MARCAÇÃO?

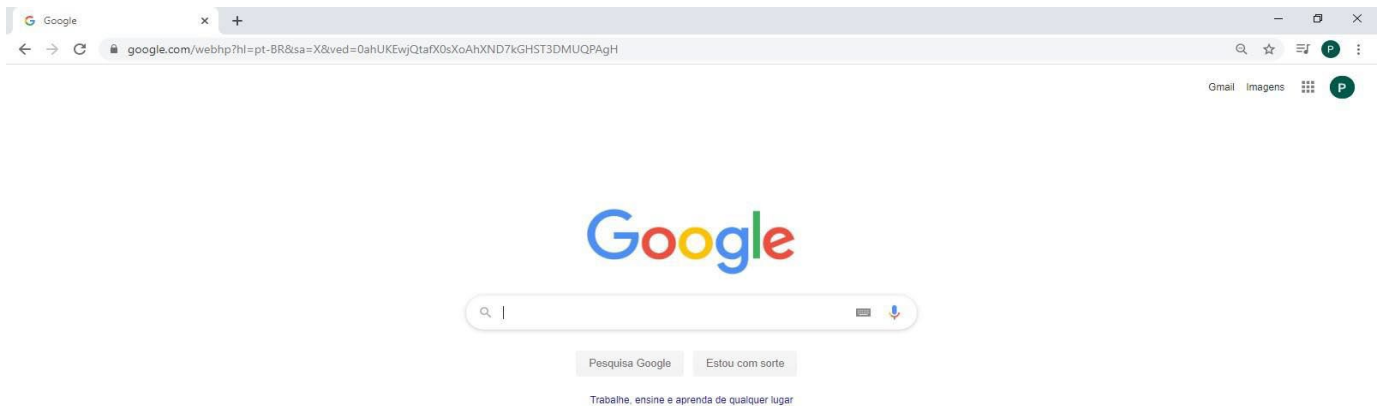




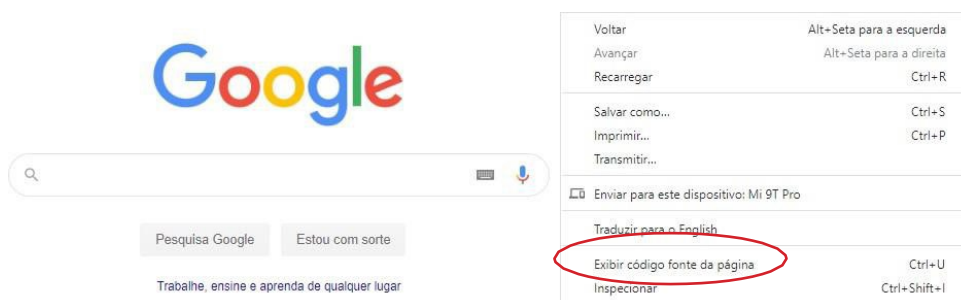
MERGULHANDO NO TEMA...

Como são feitas as páginas de internet?

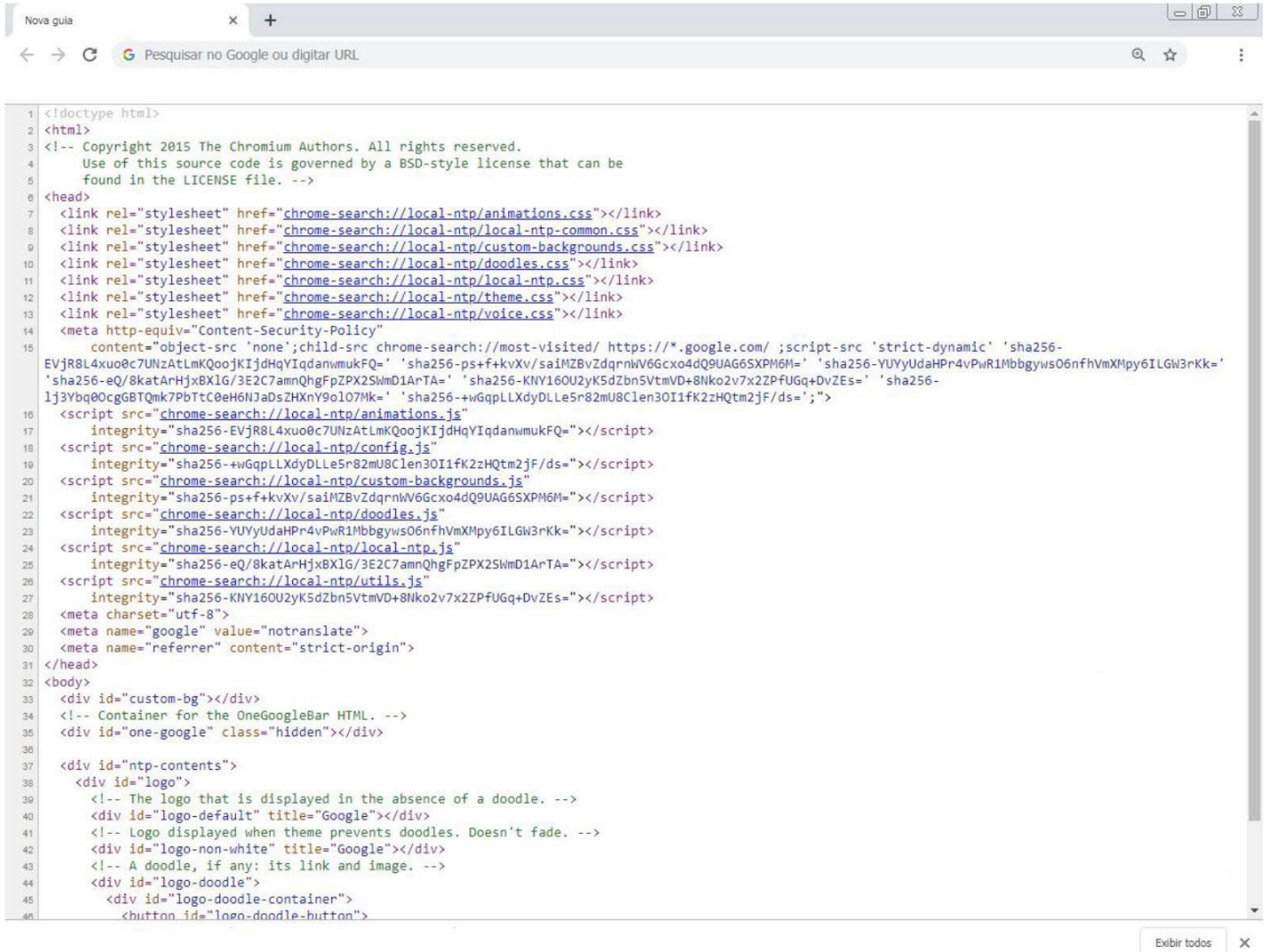
Para responder essa pergunta inicie seu navegador (Chrome, IE, Mozilla e etc.) e entre no site do Google.



Pressione o botão direito do mouse em uma área em branco da página, irão aparecer algumas opções, clique em **“exibir código fonte da página”**.



Uma nova aba será aberta com o código fonte desta página do Google. Apesar de ser muito código, não precisa se assustar.



```

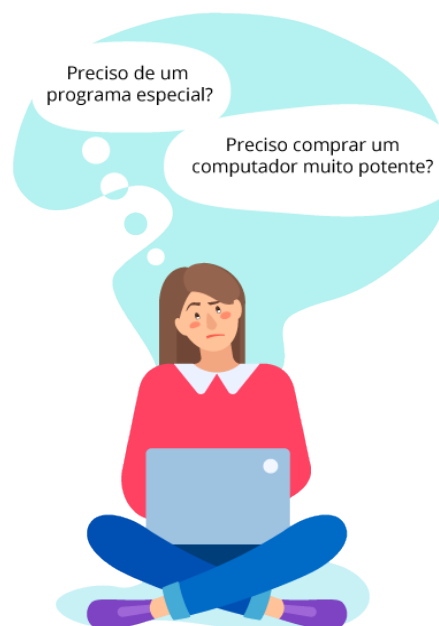
1 <!doctype html>
2 <html>
3 <!-- Copyright 2015 The Chromium Authors. All rights reserved.
4 Use of this source code is governed by a BSD-style license that can be
5 found in the LICENSE file. -->
6 <head>
7 <link rel="stylesheet" href="chrome-search://local-ntp/animations.css"></link>
8 <link rel="stylesheet" href="chrome-search://local-ntp/local-ntp-common.css"></link>
9 <link rel="stylesheet" href="chrome-search://local-ntp/custom-backgrounds.css"></link>
10 <link rel="stylesheet" href="chrome-search://local-ntp/doodles.css"></link>
11 <link rel="stylesheet" href="chrome-search://local-ntp/local-ntp.css"></link>
12 <link rel="stylesheet" href="chrome-search://local-ntp/theme.css"></link>
13 <link rel="stylesheet" href="chrome-search://local-ntp/voice.css"></link>
14 <meta http-equiv="Content-Security-Policy"
15 content="object-src 'none'; child-src chrome-search://most-visited/ https://*.google.com/ ; script-src 'strict-dynamic' 'sha256-
EVJR8L4xuo0c7UNzAtLmKQoojKIjdHqYIqdanwmukFQ=' 'sha256-ps+f+kvXv/saiMZBvZdqrnWV6Gcxo4dQ9UAG6SXPm6M=' 'sha256-YUYyUdaHP4vPwR1MbbgywsO6nfhVmXmPy6ILGw3rKk='
'sha256-eQ/8katArHjxBX1G/3E2C7amnQhgFpZPX2SWMd1ArTA=' 'sha256-KNY160U2yK5dZbn5VtmVD+8Nko2v7x2ZPFUGq+DvZEs=' 'sha256-
lj3Ybq0cgGBTQmk7PbTtC0eH6NJaDsZHXnY9o107Mk=' 'sha256-+wGqPLXdyDLLe5r82mU8Clen3OI1fK2zHQtm2jF/ds='";>
16 <script src="chrome-search://local-ntp/animations.js"
17 integrity="sha256-EVJR8L4xuo0c7UNzAtLmKQoojKIjdHqYIqdanwmukFQ="></script>
18 <script src="chrome-search://local-ntp/config.js"
19 integrity="sha256-+wGqPLXdyDLLe5r82mU8Clen3OI1fK2zHQtm2jF/ds="></script>
20 <script src="chrome-search://local-ntp/custom-backgrounds.js"
21 integrity="sha256-ps+f+kvXv/saiMZBvZdqrnWV6Gcxo4dQ9UAG6SXPm6M="></script>
22 <script src="chrome-search://local-ntp/doodles.js"
23 integrity="sha256-YUYyUdaHP4vPwR1MbbgywsO6nfhVmXmPy6ILGw3rKk="></script>
24 <script src="chrome-search://local-ntp/local-ntp.js"
25 integrity="sha256-eQ/8katArHjxBX1G/3E2C7amnQhgFpZPX2SWMd1ArTA="></script>
26 <script src="chrome-search://local-ntp/utls.js"
27 integrity="sha256-KNY160U2yK5dZbn5VtmVD+8Nko2v7x2ZPFUGq+DvZEs="></script>
28 <meta charset="utf-8">
29 <meta name="google" value="notranslate">
30 <meta name="referrer" content="strict-origin">
31 </head>
32 <body>
33 <div id="custom-bg"></div>
34 <!-- Container for the OneGoogleBar HTML. -->
35 <div id="one-google" class="hidden"></div>
36
37 <div id="ntp-contents">
38 <div id="logo">
39 <!-- The logo that is displayed in the absence of a doodle. -->
40 <div id="logo-default" title="Google"></div>
41 <!-- Logo displayed when theme prevents doodles. Doesn't fade. -->
42 <div id="logo-non-white" title="Google"></div>
43 <!-- A doodle, if any: its link and image. -->
44 <div id="logo-doodle">
45 <div id="logo-doodle-container">
46 <button id="logo-doodle-button">

```

O importante agora é refletir e perceber que muito do que faremos não será visto pelo usuário; porém, sem isso, não seria possível fazer uma página web. O objetivo desta agenda é entender sobre **HTML** e poder criar nossas páginas **WEB**.


Assim que começamos a pensar em como desenvolver uma página, logo nos perguntamos:

Como desenvolver?



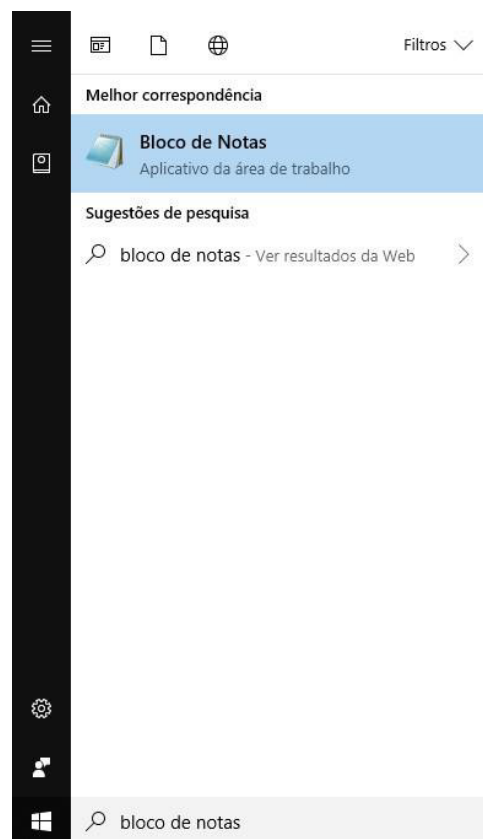
Para ambas as perguntas, a resposta é **NÃO!** Para provar isso, vamos criar nossa primeira página agora mesmo, vamos utilizar o programa **Bloco de Notas**.

Criando uma página utilizando o Bloco de Notas

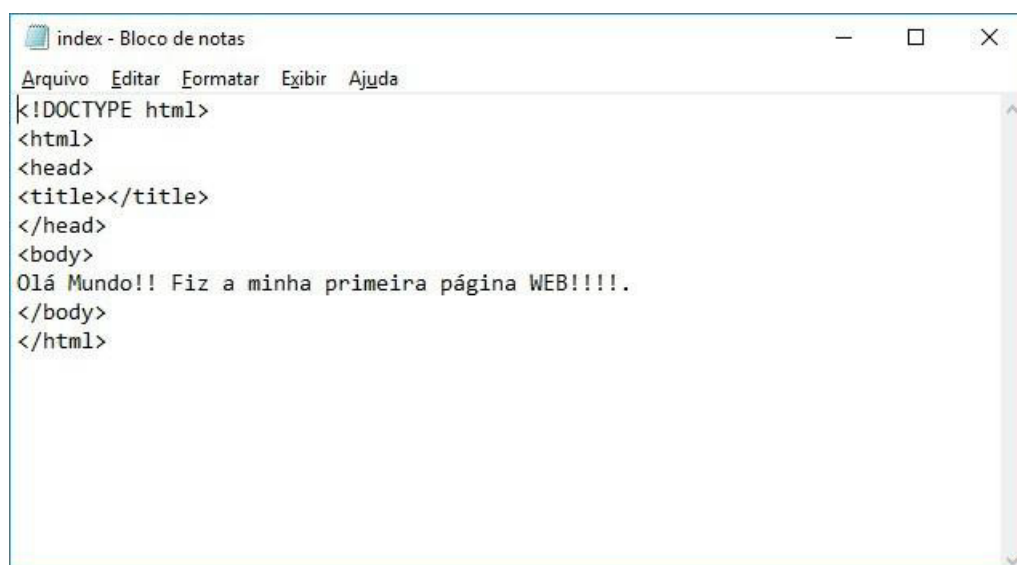
Para utilizá-lo, basta pressionar a tecla do Windows ou clicar em iniciar  depois digitar **Bloco de Notas** e pressionar a tecla “Enter” ou clicar em cima do ícone do Bloco de Notas.

Com o programa aberto, vamos digitar alguns códigos em **HTML** para a confecção da nossa página **Web**.

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
Olá Mundo!! Fiz a minha primeira página WEB!!!!.
</body>
</html>
```



O objetivo é ficar como a figura a seguir:

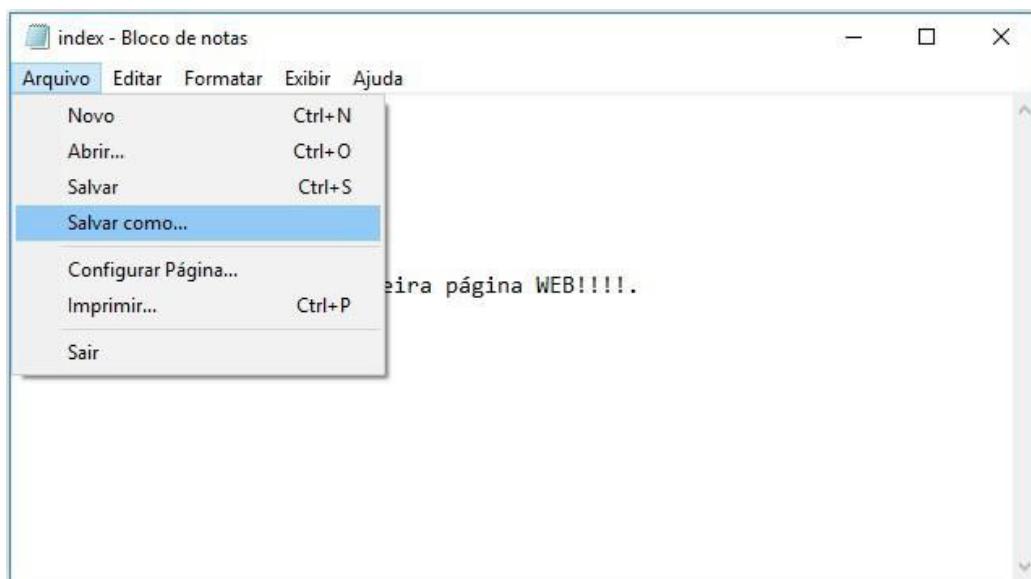


"Olá, mundo! Fiz a minha primeira página WEB!!!"

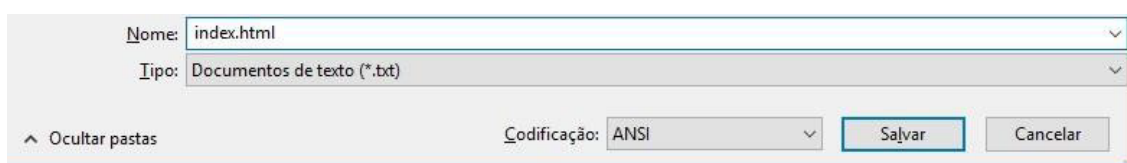
Assista as explicações do Prof. Sandro Valérius na videoaula a seguir:



O que acabamos de fazer foi codificar em **HTML** a nossa primeira página **Web**. Porém, agora precisamos salvar o código no formato correto para que qualquer navegador **Web** seja capaz de exibir essa página. Para salvar, basta clicar em “**Arquivo**” depois em “**Salvar Como**”, como mostra a imagem a seguir.



Para o nome do arquivo, coloque sem aspas: “**index.html**”. O nome “**index**” é utilizado como padrão para o arquivo **HTML** referenciá-lo como site principal. Tanto é que vários servidores vêm configurados para sempre exibirem esse arquivo como página inicial.



O “**.HTML**” é a extensão do arquivo que define que o texto é um código em **HTML**, ou seja, os navegadores serão capazes de interpretar e exibir essa página. Para saber mais sobre extensões, você pode acessar [este site](#).

Pronto, agora vá até a pasta em que você salvou o arquivo. Caso tenha um navegador instalado no seu computador, o arquivo aparecerá com o ícone desse navegador. Caso você tenha mais de um navegador instalado, o ícone será do seu navegador principal ou programa padrão escolhido para a execução de arquivos do tipo “.html”. Neste caso, o navegador padrão é o **Chrome da Google**, como mostra a imagem ao lado.



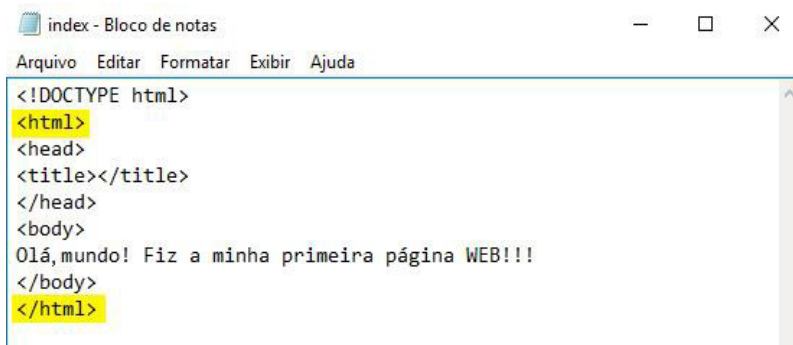
Dê dois cliques sobre o arquivo para que o seu navegador seja inicializado e, caso todas as etapas tenham sido realizadas corretamente, a primeira página será exibida no navegador, como na imagem a seguir.



Pronto, conseguimos codificar uma página HTML usando o bloco de notas!

Você percebeu que é uma página bem simples que só diz: “**Olá Mundo!! Fiz a minha primeira página WEB!!!!**”, mas é o primeiro passo para que você possa desenvolver qualquer site.

Agora que já desenvolvemos nossa primeira página, vamos entender melhor cada código que foi digitado. Perceba que foi criada a estrutura básica para uma página, começando com o código: **<!DOCTYPE html>**.



Todo o **HTML** é baseado em utilização de tag, porém, esse código não é uma tag e sim uma instrução especial que indica qual a versão do HTML os navegadores utilizarão para exibir uma página HTML, então solicitamos ao navegador que use o **HTML 5** para exibir a página criada.

Existem outros comandos nessa parte do **DOCTYPE** para outras versões e utilizações:

Doctype	Versão
<!DOCTYPE html>	HTML5
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"	HTML 4.01
"http://www.w3.org/TR/html4/strict.dtd">	Strict
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"	HTML 4.01
"http://www.w3.org/TR/html4/loose.dtd">	Transitional
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"	HTML 4.01
"http://www.w3.org/TR/html4/frameset.dtd">	Frameset

Atualmente, o recomendado é sempre usar a última versão do HTML

Tag <html> </html>

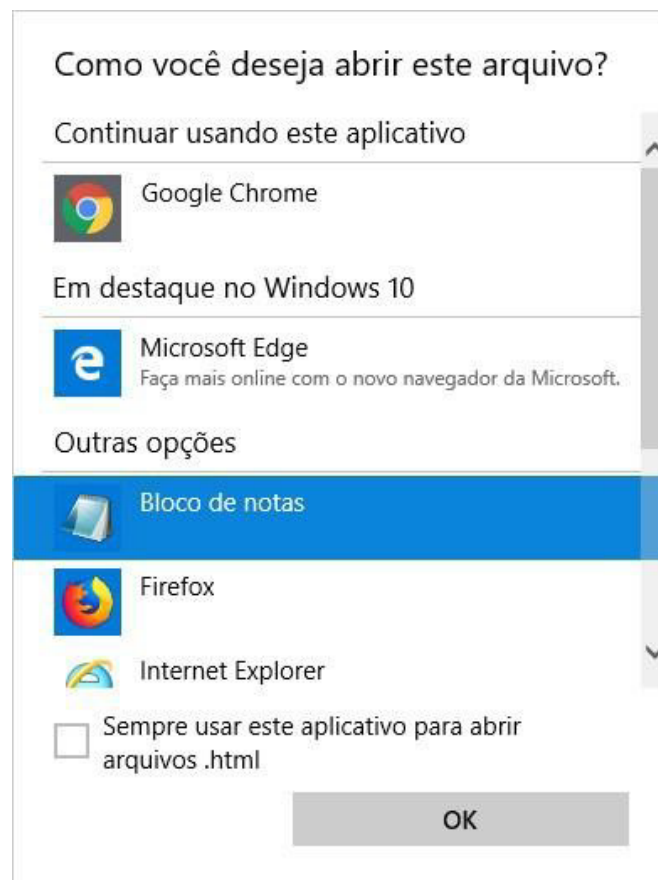
Na estrutura de uma página HTML, todas as tags ficarão entre o **<html>** - “aberto” e o **</html>** - “fechado”. Observe o código a seguir e repare que, exceto a instrução Doctype, todo o código que digitamos está entre as tag’s **<html>** e **</html>**. É importante que seja assim, porque essas tag’s demarcam todo o código **HTML**, ou seja, o navegador sabe que tudo que estiver entre essas duas marcações são da página **HTML**.

Tag <head> </head>

As tags **<head>** e **</head>** são usadas como **cabeçalho da página** e basicamente possuem o mesmo papel de um cabeçalho de um trabalho escolar, contendo um conjunto de informações de uma página. A especificação desta tag obriga que codifiquemos outra tag: **<title><title>**, “dentro” ou entre as tag’s **<head>**, onde é possível colocar um título que será exibido na aba da página.

Nas próximas etapas, retomaremos a tag **<head>** com mais informações. Agora, vamos utilizar essa tag para colocar um título na página que desenvolvemos. Vamos lá? Então, siga os seguintes passos:

- 1- Com o botão direito do mouse, clique em cima do arquivo **index** que criamos, escolha a opção “**abrir com**” e escolha o **Bloco de Notas**.



Observação: Não selecione o campo “Sempre usar este aplicativo para abrir arquivos.html”. Caso você selecione, sempre que quiser abrir a sua página no navegador, terá que fazer esse processo, mas, ao invés de escolher Bloco de Notas, escolherá o navegador.

- 2- Digite entre as tags **<title>Primeira Página</title>**.
- 3- Salve pressionando **CTRL + S** ou clicando em arquivo e depois em salvar. Após o salvamento, se sua página ainda estiver sendo executada no navegador, basta pressionar **F5** para atualizar a página, ou apenas abra novamente o arquivo, como feito anteriormente.

O resultado deve ficar como o demonstrado na imagem a seguir:

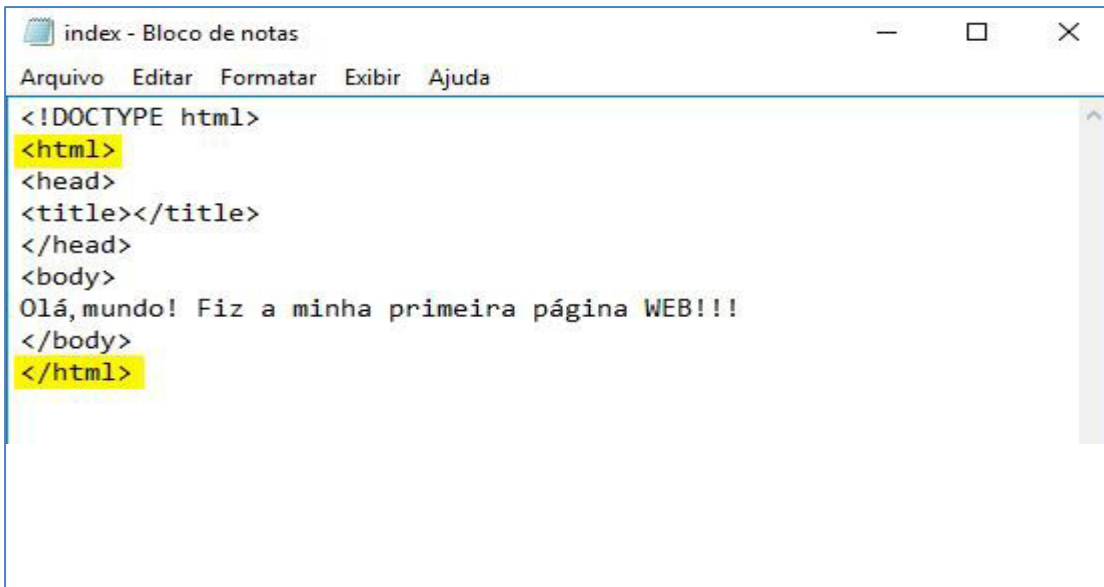


Note que, antes da alteração, quando não tinha nenhuma informação digitada entre as tags `<title>`, o navegador precisava colocar alguma informação na guia para representar o título da página ([index.html](#)).

Quando digitamos um título entre as tags, o navegador exibe o texto (Primeira Página) que foi “marcado” com as tags. Dessa forma, podemos entender melhor porque o HTML é definido como **linguagem de marcação**.

Tag `<body>` `</body>`

A tag `<body></body>` contém o corpo da página, ou seja, é o que será exibido pelo navegador em sua janela. Observando a página que criamos e o texto que digitamos: (“**Olá mundo! Fiz a minha primeira página WEB!!!**”), perceba que, no código, essa frase está entre as tags `<body>` e, na visualização pelo navegador, é apresentada no corpo da página, como pode ser observado na imagem a seguir:



```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
Olá, mundo! Fiz a minha primeira página WEB!!!
</body>
</html>
```

Pronto, agora a página faz mais sentido, não é? Então é com você!

VOCÊ NO COMANDO

Utilizando o que foi visto até agora

1. Crie uma página com os seguintes requisitos:

- A) O título será “Quem sou eu?”.**
- B) No corpo da página você deve colocar seu nome completo, idade, cidade/estado, site preferido.**

Agora, confira se você conseguiu resolver os desafios propostos!

Código:

```

ex1 - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<!DOCTYPE html>
<html>
<head>
<title>Quem sou eu?</title>
</head>
<body>
Nome: Zuleicka Zulmira.
Idade: 25.
Cidade: São Paulo / SP.
Site Preferido: www.google.com.br
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
<title> Quem sou eu?</title>
</head>
<body>
Nome: Zuleicka Zulmira.
Idade: 25.
Cidade: São Paulo / SP.
Site Preferido: www.google.com.br
</body>
</html>

```

Resultado no Navegador:

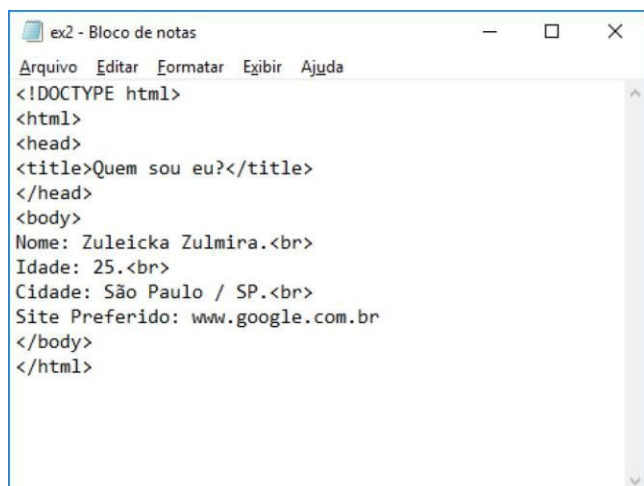


Observação: Você notou que mesmo escrevendo cada informação (nome, idade etc.) em cada linha separada no código, o navegador a exibe em uma mesma linha. Isso acontece porque o navegador não sabe que você pulou de linha. Por isso, é preciso informar onde existe quebra de linha.

A tag utilizada para resolver esse problema é a tag **
**. Esta tag não necessita de fechamento porque a sua função é apenas marcar onde será realizada uma quebra de linha. No exemplo do desafio acima, coloque a tag **
** nos locais onde você queira que o navegador quebre a linha.



O código ficaria assim:



```

ex2 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<!DOCTYPE html>
<html>
<head>
<title>Quem sou eu?</title>
</head>
<body>
Nome: Zuleicka Zulmira.<br>
Idade: 25.<br>
Cidade: São Paulo / SP.<br>
Site Preferido: www.google.com.br
</body>
</html>

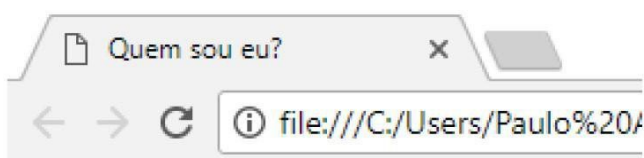
```

```

<!DOCTYPE html>
<html>
<head>
<title> Quem sou eu?</title>
</head>
<body>
Nome: Zuleicka Zulmira. <br>
Idade: 25. <br>
Cidade: São Paulo / SP. <br>
Site Preferido: www.google.com.br
</body>
</html>

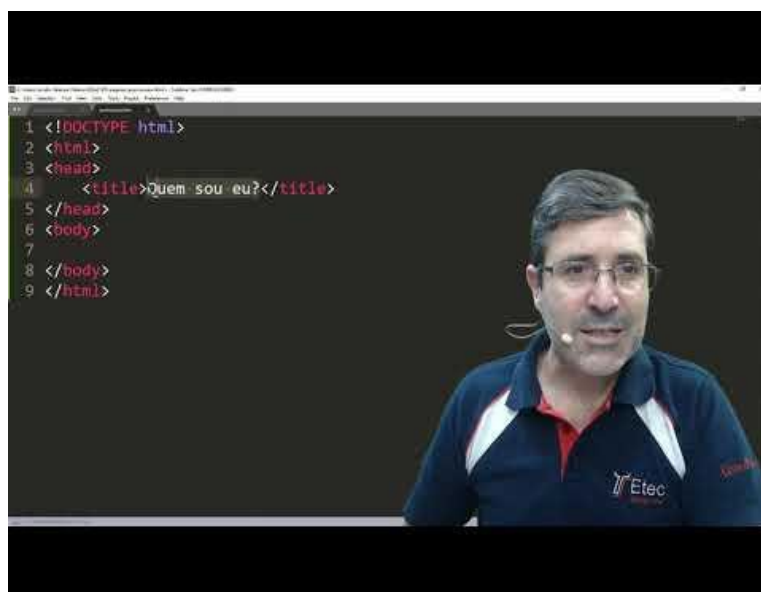
```

No navegador ficaria assim:



Nome: Zuleicka Zulmira. Idade: 25. Cidade:
São Paulo / SP.
Site Preferido: www.google.com.br

Assista as explicações do Prof. Sandro Valérius sobre a página “Quem sou Eu?”



É possível desenvolver um site inteiro, apenas com o Bloco de Notas?

A resposta para essa pergunta é SIM! Porém existem ferramentas mais eficientes que podem ser utilizadas pelos desenvolvedores para auxiliá-los em seu trabalho.

No Blog Umblar você encontra **9 tipos de ferramentas essenciais para todo desenvolvedor web**. Confira [nesse site!](#)



Disponível em: <https://blog.umblar.com/br/ferramentas-para-desenvolvedores-da-web>.

Como você pode perceber, existem várias opções de ferramentas para desenvolvimento Web e, além desses, existe também o **NotePad++**, que é bem simples e oferece alguns recursos. Veja como ficaria o código da 1ª página Web que desenvolvemos, utilizando esse software:

A screenshot of the Notepad++ text editor. The title bar shows the file path 'C:\Users\roni.dantas\Desktop\hello_world.html - Notepad++'. The menu bar includes 'Arquivo', 'Editar', 'Localizar', 'Visualizar', 'Formatar', 'Linguagem', 'Configurações', 'Tools', 'Macro', 'Executar', and 'Plug'. The toolbar contains various icons for file operations and editing. The main text area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7   Olá, mundo! Fiz a minha primeira página WEB!!!
8 </body>
9 </html>
```



Link para download do NotePad++

<https://notepad-plus-plus.org/downloads/>

Neste link você encontra um [tutorial](#) de download e instalação do programa Notepad++.

Repare que as tags foram colocadas na cor azul e que esse software possui um recurso de **IntelliSense** simples, porém de grande ajuda!

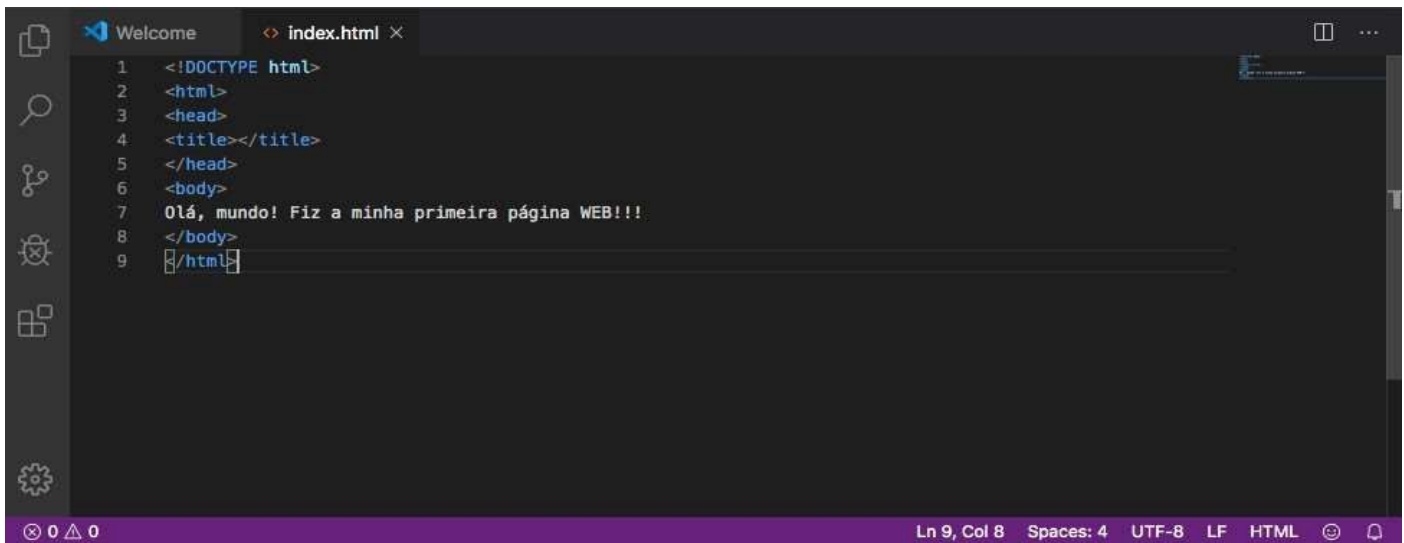
Você sabe o que é esse termo IntelliSense?

O termo “IntelliSense” é de uso geral para vários recursos: Listar Membros, Informações do Parâmetro, Informação Rápida e Completar Palavra. Esses recursos ajudam você a aprender mais sobre o código que está usando, a manter o acompanhamento dos parâmetros que está digitando e a adicionar chamadas a métodos e propriedades pressionando apenas algumas teclas.

Visual Studio Code

Neste curso, utilizaremos o **Visual Studio Code** que, como você viu no Blog, é um software simplificado de editor de código com suporte para operações de desenvolvimento, como: depuração, execução de tarefas e controle de versão, ou seja, ele tem muito mais recursos a oferecer que o **NotePad++**.

O objetivo do **Visual Studio Code** é fornecer as ferramentas que um desenvolvedor precisa para um ciclo rápido de desenvolvimento e verificação de código.



Por que a escolha do Visual Studio Code?



Esse software também é gratuito e oferece muito mais recursos.

Além disso, o seu **IntelliSense** é muito melhor, o que nos ajudará a **economizar mais tempo**.



Link para download do Visual Studio Code

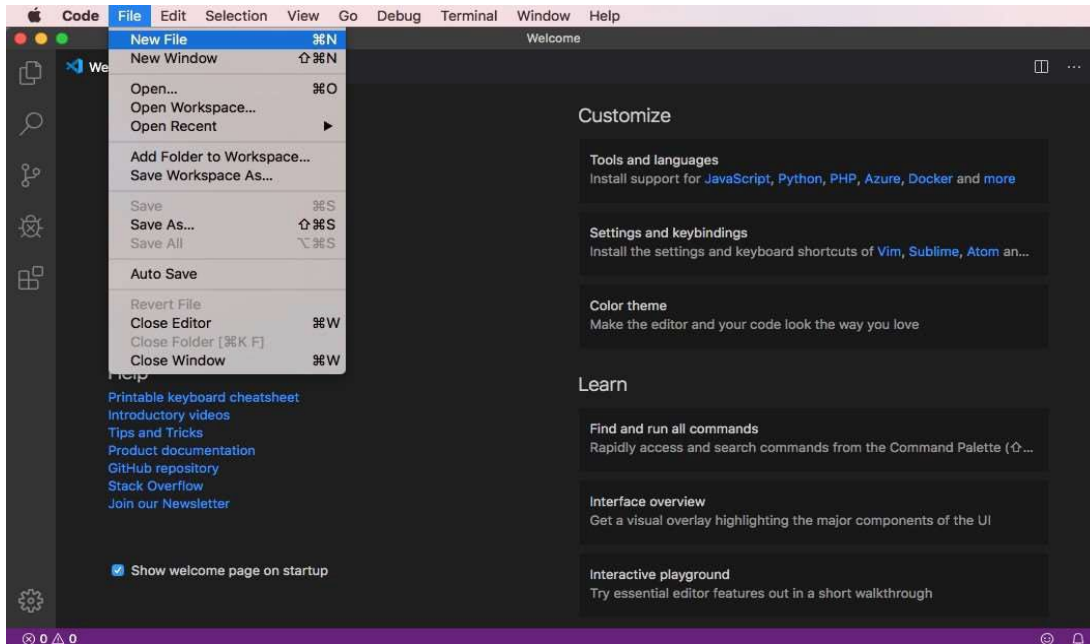
<https://code.visualstudio.com/Download/>

Observação:

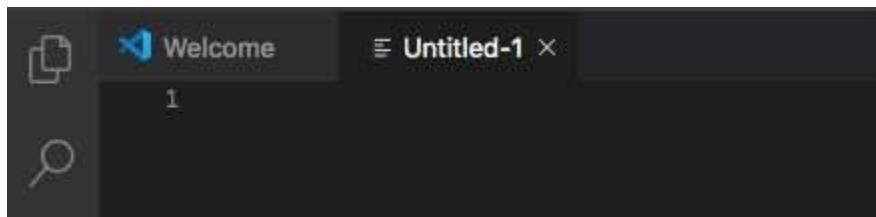
Lembre-se de que você é livre para escolher qualquer ferramenta, até mesmo o Bloco de Notas, para o desenvolvimento de todos os conteúdos.

Agora vamos ver um exemplo do porquê utilizar o **Visual Studio Code**:

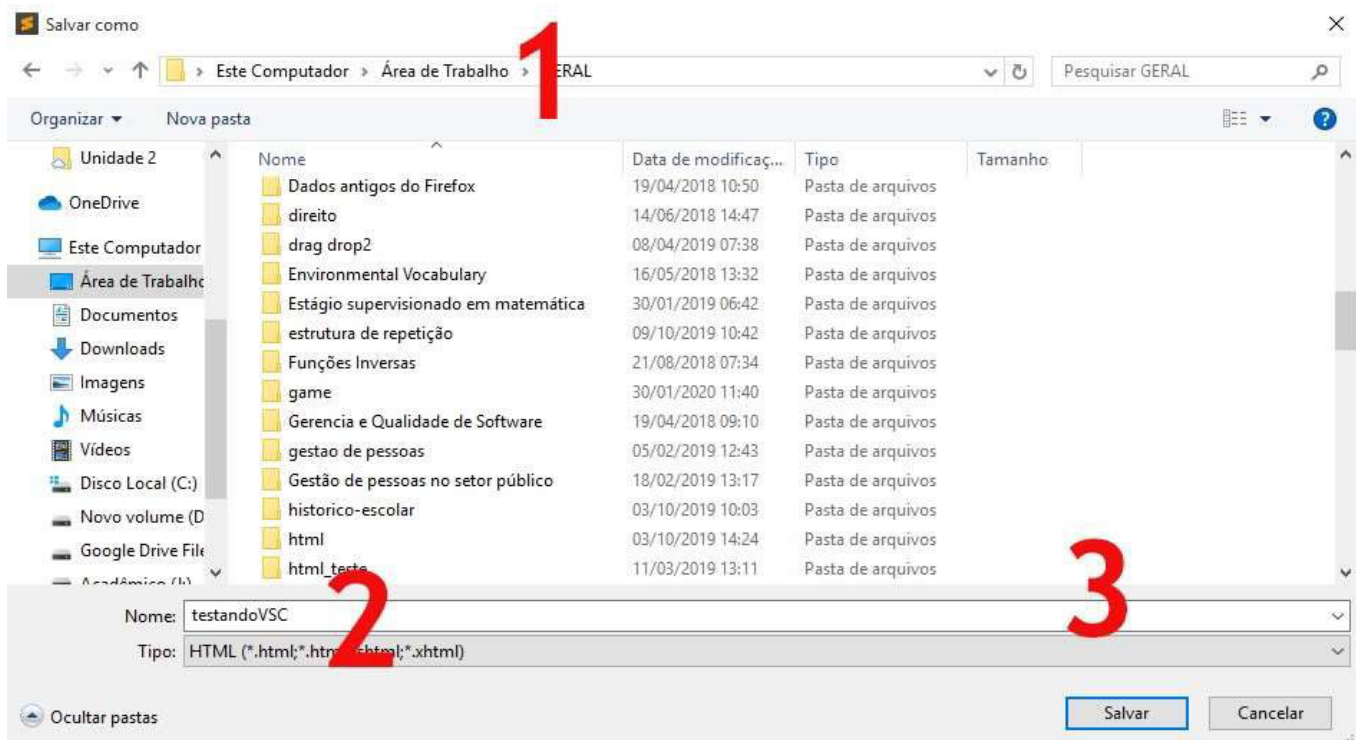
Abra o programa, clique em **“File”**, depois em **“New File”**, ou apenas pressione **CTRL + N**.



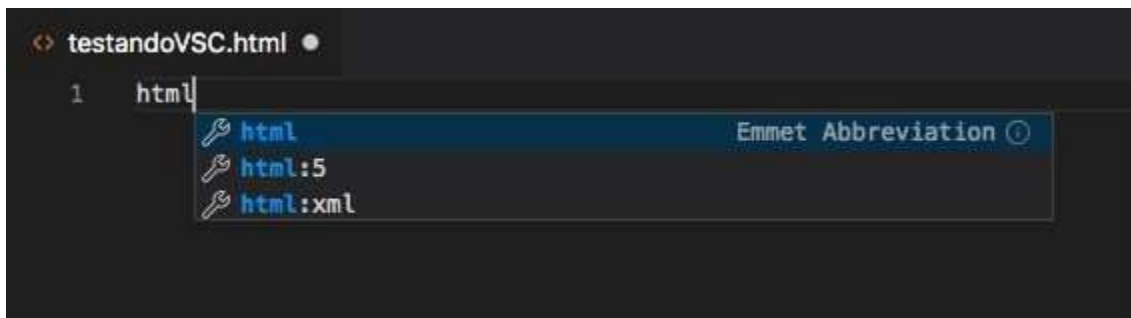
Será aberto um novo arquivo com um nome provisório **“Untitled-1”** que em tradução livre significa **sem título**, conforme mostra a imagem a seguir:



Agora vamos salvar o arquivo e nomeá-lo como **“testandoVSC”**. Clique no menu **“File”**, em **“Save”**, escolha o **“local”** (1) onde será salvo o arquivo e digite o **“nome de arquivo”** (2), sem se esquecer de escolher o **“tipo de arquivo”** (3) que queira salvar: **“HTML”**, conforme mostra a imagem a seguir.



Pronto! Agora que está tudo configurado, digite **html** e repare que irão aparecer algumas opções. Esse é o **IntelliSense**! Ele já prevê que o arquivo **html** está vazio e, quando você começa a digitar, já aparecem possíveis complementos. Selecione com a seta do teclado a opção "**html: 5**" e pressione a tecla **Enter**.



Automaticamente, o **Visual Studio Code** criará toda a **estrutura básica**, facilitando o trabalho de digitação, evitando erros e ganhando tempo no desenvolvimento. Veja:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>

</body>
</html>
```

O que você achou? É válida ou não a utilização de um software para auxiliar no desenvolvimento?

Você já imaginou quantas tags tem a Linguagem HTML?

A resposta é “muitas!”. Confira acessando esse [link](#):



São muitas tags e vamos aprender a utilizar diversas!

Para começar é melhor entender as tags e os novos códigos que apareceram assim que utilizamos o **IntelliSense** do Visual Studio Code. Observe a codificação:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>

</body>
</html>
```

Entenda a codificação desse exemplo:

Linha 2: <html lang="en">

Repare que existe um novo código dentro da tag html. Trata-se de informações que passamos dentro da tag, ou seja, entre os símbolos “<>” para que essa tag se comporte da maneira esperada ou notifique algum tipo de informação para navegadores ou motores de busca (Google, Bing etc), essas informações são chamadas de atributos. Existem basicamente dois tipos de atributos:

Globais

Que funcionam em todas as Tags

Específicos

Que são direcionados para cada Tag, por meio de especificação

Por definição, os atributos possuem um nome e um valor e o uso de atributos dependerá das necessidades da sua página.

O atributo HTML **lang** pode ser usado para declarar o idioma de uma página da Web ou parte de uma página da Web, servindo de referência para motores de busca e navegadores. Nesse exemplo, o atributo está definido no idioma inglês. Para alterá-lo para português do Brasil, coloque entre as aspas “**pt-br**”, como representado a seguir:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
```

Neste link você pode encontrar várias opções de valores para o atributo lang:

http://www.w3schools.com/tags/ref_language_codes.asp

Linha 4: <metacharset="UTF-8">

O código trata-se de uma Tag com o atributo charset. Perceba que:

Meta é uma Tag implementada no código de páginas web entre as tags head do site (<head> e </head>). Algumas têm como função passar instruções a programas externos. Por exemplo, passar aos sites de busca, como o Bing e o Google, as instruções sobre a página web desenvolvida. Então, podemos definir que são **metadados**, ou seja, dados sobre os dados da página.

O atributo colocado nessa tag **meta** é o **charset** com o valor "**UTF-8**", que serve para indicar o formato de codificação de caracteres utilizados no documento. Algumas codificações existentes como o **ASCII**, o **UTF8**, o **ANSI** e **ISO-8859-1** são exemplos de **charsets**.

Para cada caractere que digitamos ou lemos em uma página ou um site, existe uma representação dele, em byte, na memória. Cada tipo de codificação **charset** representa o caractere em memória de uma forma diferente. O charset **UTF-8** é o que usamos na web atual e faz parte de um padrão chamado **Unicode**. A ideia é que com ele possamos representar qualquer caractere, de qualquer idioma.

Linha 4: <meta name="viewport" content="width=device-width, initial-scale=1.0">

Na linha 4, encontramos outra tag meta, porém com atributos diferentes da primeira:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Trata-se de uma configuração de visualização da área visível de uma página da web. Essa área varia de acordo com o dispositivo, por exemplo: a visualização em um celular será menor do que em uma tela de computador. Antes dos tablets e dos telefones celulares, as páginas da web eram desenvolvidas apenas para telas de computador, ou seja, eles adotavam um design estático e tamanho fixo. Logo que foi possível navegar na Internet, usando tablets e telefones, as páginas web ficavam desproporcionais, gerando dificuldades para visualizar o seu conteúdo em telas pequenas.

O valor **width=device-width**, definido ao atributo **content**, define largura da página, de acordo com a tela do dispositivo que está exibindo a página. Com o uso da vírgula, é possível colocar dois ou mais valores em um atributo. O valor **initial-scale=1.0** define o zoom inicial quando a página é carregada pela primeira vez pelo navegador. É possível observar, na imagem a seguir, a diferença de exibição com o uso da **meta tag** e sem o uso dela.

Linha 6: <meta>

Por fim, na **linha 6**, encontra-se outra **meta tag**. Observe que, apesar de existirem padrões a cada evolução, os navegadores tratam os códigos de maneira diferente. Esse código é utilizado para forçar o Internet Explorer a renderizar a página web com a versão mais recente. A partir da versão IE11, não é mais necessário o uso dessa meta, mas, se o navegador for anterior a essa versão, é preciso fazer seu uso, conforme apresentado:

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```



Sem meta tag

Com meta tag

Criação de Páginas

Agora que você já conhece a estrutura HTML, vamos começar a criar páginas com várias tags diferentes. Vamos lá?

Crie um arquivo **HTML** novo, com o nome de “**título**”.

Vale ressaltar: nunca utilize espaços em branco, caracteres especiais (ç, á, ~ etc.) para os nomes dos arquivos.

Tag <h>

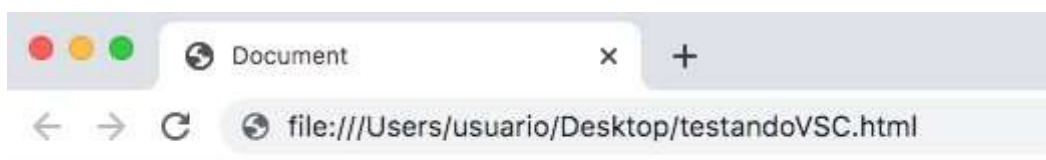
A **tag header** (<h1>, <h2>, <h3>, <h4>, <h5> e <h6>) é utilizada para destacar títulos e subtítulos de uma página. H1 é a abreviação do inglês para Header 1, sendo ele o de maior importância. Puramente, o **H1** possui um destaque maior, uma fonte maior e é geralmente o elemento de texto mais visível da página.

A **tag H1** é um importante elemento que os motores de busca utilizam para determinar o principal assunto abordado, visto que o título de uma página, conceitualmente, define seu conteúdo.

Com o arquivo aberto, digite **html: 5** e pressione a tecla “Enter” para que toda a estrutura básica seja criada pelo **VSCode**. Entre as tag <body> e </body>, insira as tags <h1> e </h1> e digite a seguinte frase: “**HTML 5 – Vamos Programar**”.


```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title> Document </title>
</head>
<body>
<h1>HTML 5 – Vamos Programar!</h1>
</body>
</html>
```

Salve a codificação pressionando **CTRL + S**, localize a pasta onde tenha salvo o arquivo e execute-o. O resultado deve ser parecido com a imagem a seguir:



HTML 5 – Vamos Programar!

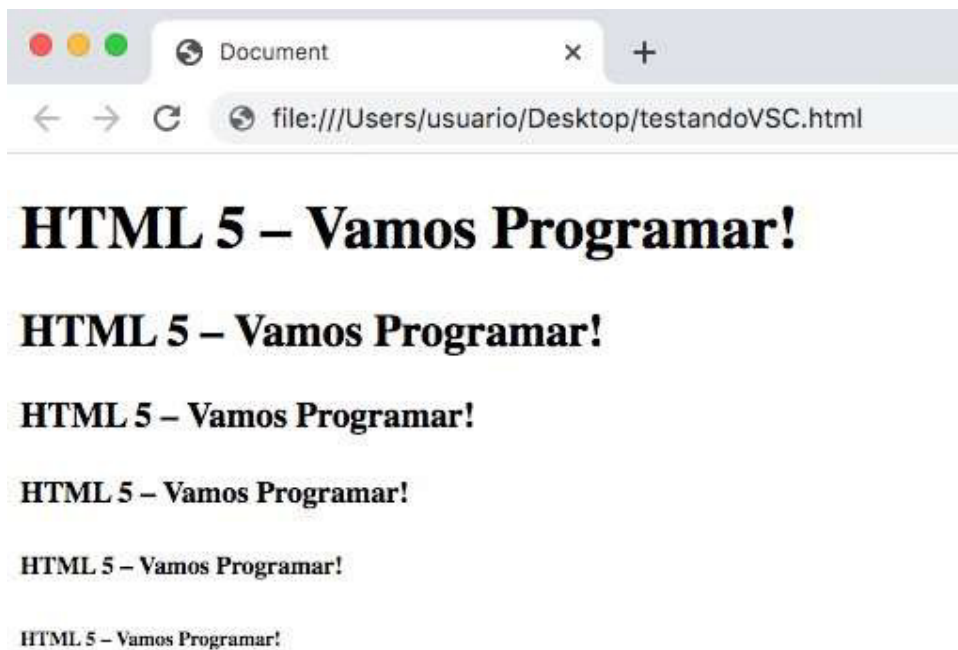
É notável que tudo o que fica entre as tags **<h1>** e **</h1>** é marcado como título e recebe uma configuração padrão: o aumento do tamanho da letra.

Faça uma alteração no código do mesmo arquivo, deixando-o dessa forma:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>HTML 5 – Vamos Programar!</h1>
<h2>HTML 5 – Vamos Programar!</h2>
<h3>HTML 5 – Vamos Programar!</h3>
<h4>HTML 5 – Vamos Programar!</h4>
<h5>HTML 5 – Vamos Programar!</h5>
<h6>HTML 5 – Vamos Programar!</h6>
</body>
</html>
```

Salve (**CTRL + S**), volte ao navegador em que a página está sendo exibida e pressione a tecla F5. Você deve

obter um resultado como o mostrado na imagem a seguir.



Tag <p>

Crie um arquivo **HTML** novo, com o nome de “parágrafo”. Não esquecendo de criar a estrutura básica.

Sempre que for necessário ou quiser que um texto seja visto como parágrafo, devemos colocá-lo entre as tags **<p> e </p>** e o navegador irá se encarregar de dispor o parágrafo da melhor maneira possível.

Para demonstrar isso, vamos escrever as seguintes frases, **uma em cada tag <p>**.

1. Basicamente você assiste à TV para desligar seu cérebro e usa o computador quando você o quer de volta!
2. Já dizia o velho sábio: O computador veio para resolver os problemas que nós ainda não tínhamos.
3. Quando a esmola é demais, o Santo desconfia que tem vírus anexado.

O código deve ficar como o demonstrado a seguir:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<p> Basicamente você assiste à TV para desligar seu cérebro e usa o computador quando você o quer de volta! </p>
<p> Já dizia o velho sábio: O computador veio para resolver os problemas que nós ainda não tínhamos. </p>
<p> Quando a esmola é demais, o Santo desconfia que tem vírus anexado. </p>
</body>
</html>
```

Salve a codificação pressionando **CTRL + S**, localize a pasta onde foi salvo o arquivo e execute-o.

O resultado dever ser parecido com a imagem a seguir:



Basicamente você assiste à TV para desligar seu cérebro e usa o computador quando você o quer de volta!

Já dizia o velho sábio: O computador veio para resolver os problemas que nós ainda não tínhamos.

Quando a esmola é demais, o Santo desconfia que tem vírus anexado.

O princípio da marcação é sempre o mesmo, tudo que estiver entre as tags será marcado com as configurações da respectiva tag.

Tag

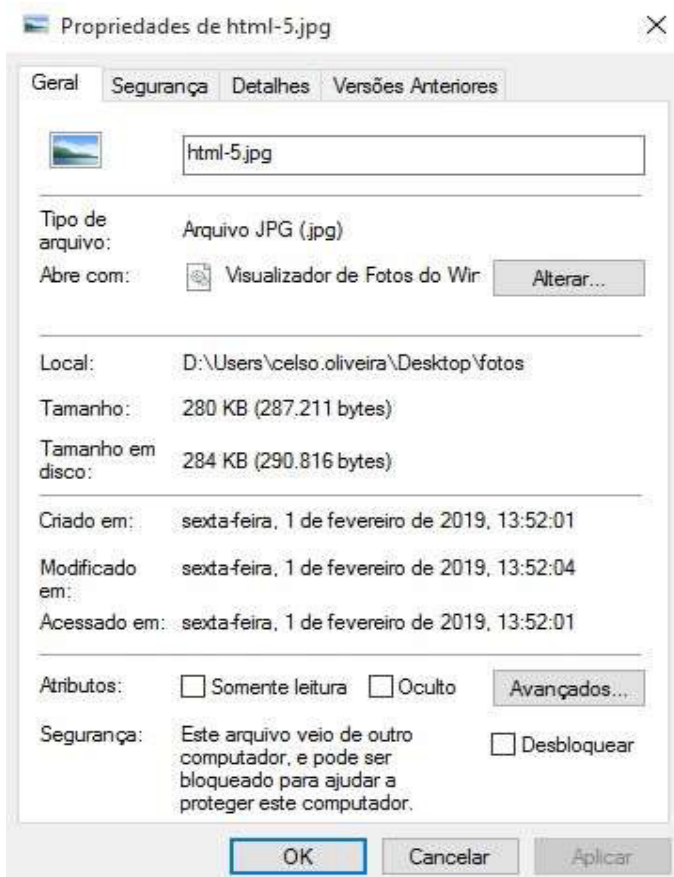
A tag **** é uma abreviação da palavra image (imagem). Com ela, é possível colocar imagens no site, viabilizando dessa forma, melhorar o design e a aparência de uma página da web.

Crie um arquivo **HTML** novo e salve-o com o nome de "imagem".

Não se esqueça de criar a estrutura básica.

Primeiro precisamos fazer o download de uma imagem e colocá-la na mesma pasta do arquivo.

Para descobrir qual o tipo da imagem que você salvou, basta clicar com o botão direito do mouse em cima da imagem e escolher a opção "propriedades". Logo no início da caixa de diálogo, é informado qual é o tipo da imagem. Nesse caso, é do tipo **JPG**.



Com o arquivo HTML e a imagem salvos na mesma pasta, é preciso agora, programar a tag ****. Para isso, utilizaremos um atributo chamado **src**, responsável por informar ao navegador onde se encontra a imagem para que seja possível exibi-la de forma adequada.

O código deve ser: ****

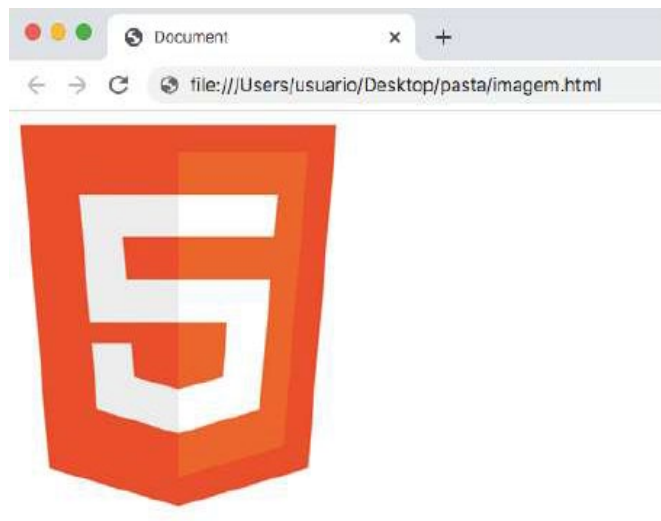
Lembrando que a imagem deve estar na mesma pasta do arquivo html. A imagem escolhida tem o nome **html-5** e é do tipo **jpg**.

O código deverá ficar conforme demonstrado a seguir:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>

</body>
</html>
```

O resultado será uma página com uma imagem, como pode ser visualizado a seguir:



O atributo **src** é referente ao local da imagem. Esse local pode ser de uma página na internet e, se você possuir o link da imagem, é possível exibi-la sem a necessidade de fazer o download do arquivo. Vamos a um exemplo: o link a seguir se refere a uma imagem, basta copiar e colar esse link dentro do atributo **src=""**.

O código deverá ficar conforme demonstrado a seguir:

```
<img src =
"https://upload.wikimedia.org/wikipedia/commons/thumb/1/10/CSS3_and_HTML5_logos_and_word
marks.svg/1280px-CSS3_and_HTML5_logos_and_wordmarks.svg.png">
```

O resultado será uma página com uma imagem que foi obtida por meio do link inserido no atributo `src`, conforme representado na imagem a seguir:



A tag `img` ainda possui um outro atributo: o `alt`.

- Quando você o utiliza e a imagem, por qualquer motivo, não é exibida, o conteúdo desse atributo é colocado no lugar da imagem.

Observe este exemplo:

- Insira o atributo `alt` na tag `img` com o valor "Html 5 e Css 3" e deixe o atributo `src` vazio.

```
<img src="" alt = "Html 5 e CSS 3">
```

O resultado será com o demonstrado na imagem a seguir:



Existem também outros atributos, tais como `width` e `height` (largura e altura, respectivamente), porém é melhor utilizá-los a partir do CSS, que é o assunto da próxima agenda!

A partir do HTML 5, foi incorporada uma nova tag que auxilia muito o desenvolvimento de páginas para definir imagens, gráficos etc., de forma mais adequada ao HTML. Trata-se de tag `<figure>`, com ela é possível definir uma marcação para a legenda da imagem. Apesar de parecer simples, é de extrema importância para os motores de busca, que as utilizam para obter melhores resultados para seus usuários.

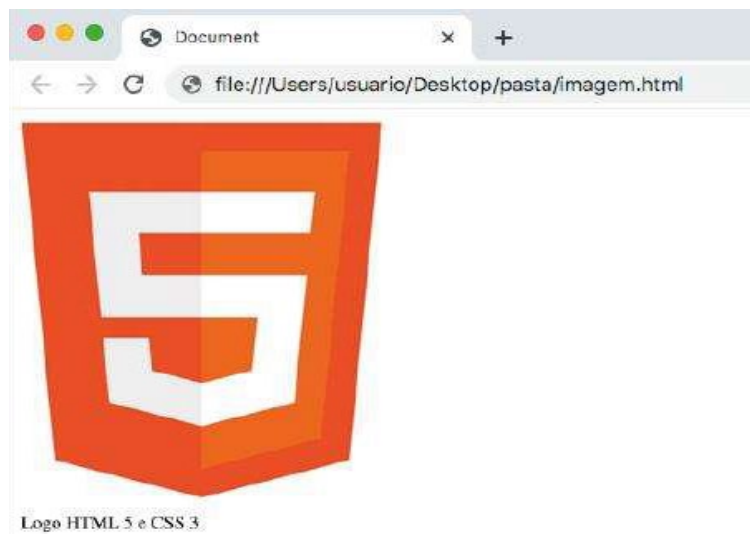
Observe o código do arquivo `imagem.html` com as tags `<figure>` e `<figcaption>`:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<figure>

<figcaption> Logo HTML 5 e CSS</figcaption>
</figure>

</body>
</html>
```

No navegador, teríamos o resultado conforme a imagem a seguir:



Visualizando o resultado na página, a diferença é mínima: aparece, no canto esquerdo da tela, uma legenda sem nenhuma formatação com o conteúdo escrito na tag `<figcaption>`: **Logo HTML 5 e CSS 3**. Apesar de simples, essa legenda é muito importante para o próprio navegador e motores de busca como o Google.

Tag `<a>`

Com certeza, você já entrou em algum site e clicou em algum texto ou imagem que fez com que o navegador abrisse uma nova página. O local em que você clicou é um **link**, que pode estar em forma de texto, geralmente sublinhado, ou de uma imagem.

O link é facilmente notado quando posicionamos o mouse sobre ele e o cursor se transforma em uma mão com o indicador esticado, sinalizando que aquilo é um link.

Você está usando um link quando:



- procura algo no Google e clica no resultado desejado.
- Abre seu e-mail e clica em “Caixa de Entrada”.
- clica em ‘curtir’ no Facebook.

Enfim, o uso e importância dos links é notório e você já deve ter certeza disso.

A tag `<a>` é responsável por fornecer este recurso para páginas web.



- Crie um novo arquivo **HTML** e salve com o nome de **"links"**.
- Não se esqueça de criar a estrutura básica.
- Vamos começar criando um link para o Google em nosso site. Para isso, será necessária a utilização do atributo **href**.

```
<a href="https://www.google.com"> Link para google </a>
```

Repare em dois detalhes:

Primeiro: o atributo href tem como valor definido "https://www.google.com", ou seja, ele será o local onde a página será direcionada quando clicar no link;

Segundo: o texto "Link para Google" foi marcado com as tag `<a>` e `` para que se torne o link para a página do Google.

O código completo da página ficaria dessa forma:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<a href="https://www.google.com">Link para google </a>

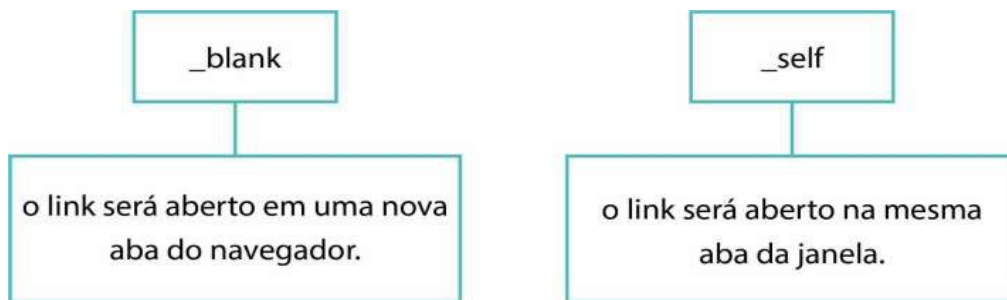
</body>
</html>
```

Veja o resultado no navegador e repare que o texto fica no formato de link (sublinhado e na cor azul), mudando de cor quando o link já foi visitado.



Você já deve ter notado que, algumas vezes, quando clica em um link, uma nova aba aparece e, outras vezes, a página é carregada na própria aba. Isso acontece porque a tag `<a>` tem um atributo nomeado **target**, que faz essas escolhas dependendo da codificação.

Observe: os principais valores são **_blank** e **_self** e suas funções são:



Observação: a opção **_self** é adotada como padrão quando a tag `<a>` tiver o atributo **target** definido. Confira os dois exemplos:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<a href="https://www.google.com" target="_blank">Link para google </a>
<a href="https://www.google.com" target="_self">Link para google </a>
</body>
</html>
```

Tag `<video>`

A tag `<video>` surgiu no **HTML 5**. Anteriormente, para usar vídeos em páginas web, era necessário utilizar um plug-in, com o flash, para conseguir reproduzi-los corretamente. A tag `<video>` em **HTML 5** especifica uma maneira padrão de incorporar um vídeo em uma página da web.

A imagem a seguir mostra os principais navegadores e quais versões suportam totalmente a utilização da tag `<video>`. Veja:



Novamente, vamos criar um arquivo HTML para inserir um vídeo em uma página. Salve com o nome “**video**” e crie a estrutura básica do HTML.

Observação: da mesma maneira que você insere imagens numa página, você deve escolher um vídeo e salvá-lo na mesma pasta do arquivo **html** que será desenvolvido.

O vídeo desse exemplo está nomeado como “**geeadIntro**”, do tipo **mp4**. Utilizaremos a tag **<video>**, com o atributo **controls**, e a tag **<source>**, com os atributos **src** e **type**. O código do arquivo video.html ficará assim:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<video controls>
  <source src="geeadIntro.mp4" type="video/mp4">
</video>
</body>
</html>
```

Note que no atributo **controls** não é necessária a definição de nenhum valor. Ele tem a função de colocar controles como play, pause e modo tela cheia para o usuário da página, conforme mostra a imagem a seguir:



Os atributos da tag **<source>**, **src** e **type**, se referem, respectivamente, à localização, nome e tipo do arquivo. No caso do exemplo, o vídeo está nomeado como “**geeadIntro**”, está localizado na mesma pasta do arquivo html e o tipo do vídeo é **mp4**.

O resultado exibido pelo navegador pode ser conferido na imagem a seguir.



Outros atributos da tag **<video>**, são:



Há outra tag chamada **<audio>** que tem a mesma formatação da tag **<video>**, mas que, ao invés de reproduzir o vídeo, reproduz áudio.

Comentário

Como o próprio nome define, são notas, informações ou observações que podem ser incluídas no código fonte. Esses comentários não são renderizados pela página e auxiliam o programador a organizar melhor seus códigos. Você pode adicionar comentários a seu arquivo HTML usando a seguinte sintaxe:

<!-- Seu comentário -->

Perceba que há um ponto de exclamação "!" no início da tag, mas não no final da tag. Com comentários, você pode inserir notificações e lembretes em seu **HTML**. Veja:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<!-- Vídeo Introdução Geead -->
<video controls>
<source src="geeadIntro.mp4" type="video/mp4">
</video>
<!-- Colocar mais vídeos sobre o GEEAD.-->
</body>
</html>
```


Observação: Os comentários não são exibidos para o usuário, mas podem ajudar, e muito, a documentar seu código-fonte HTML.

Outra função do comentário é fazer a verificação do código HTML, deixando as tags como comentário quando suspeitar que existe algum erro, já que o comentário não é utilizado pelo navegador.

Ao comentar uma tag, você a desativa, logo, poderá constatar se ela está ocasionando algum possível erro em sua página. Observe:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<!-- Não será exibido esse vídeo
<video controls>
<source src="geeadIntro.mp4" type="video/mp4">
</video>
-->
</body>
</html>
```

Repare que o Visual Studio Code deixa o comentário na cor verde para facilitar o reconhecimento dos códigos comentados. Você perceberá, ao longo de sua trajetória, como programador Web, que comentar códigos poderá auxiliá-lo bastante, principalmente em projetos longos.



VOCÊ NO COMANDO

Utilizando o que foi visto até agora....

1.Crie uma página com os seguintes requisitos:

1. Colocar título da página entre as tag <title> e </title>;
2. Usar a tag <h1> para título do filme;
3. Colocar a imagem do filme na mesma pasta do arquivo HTML
4. Ano Produção, Dirigido por, Duração, Classificação, Gênero, País de Origem – utilizar tag <h2>
5. Utilizar a tag <p>, para a descrição da sinopse e tag <h2> para o título “sinopse”;
6. Link interessante para encontrar informações do seu filme favorito: <https://filmow.com/>.
7. Opcional: Se você quiser testar seus conhecimentos, tente colocar o trailer do filme também na página.

No título (aba) da página:

Qual meu filme favorito?

No corpo da página:

Como título, o nome do
Filme. Abaixo, uma foto do
filme.

Algumas informações como subtítulo:

Ano/Produção,
Dirigido por,
Duração,
Classificação (se houver),
Gênero,
País de Origem.

No final a sinopse do filme.

Confira abaixo se você conseguiu resolver os desafios propostos!

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Qual Meu Filme Favorito?</title>
</head>
<body>
<h1>Uma mente brilhante</h1>
<figure>

<figcaption>Fonte: (FILMOW,2013) </figcaption>
</figure>
<h2>Ano de produção: 2001</h2>
<h2>Dirigido por: Ron Howard (I)</h2>
<h2>Duração:135 minutos </h2>
<h2>Classificação:12 - Não recomendado para menores de 12 anos</h2>
```



```

<h2>Gênero: Biografia; Drama</h2>
<h2>País de Origem:EUA</h2>
<h2>Sinopse: </h2>
<p>
    Uma Mente Brilhante é baseado no livro A Beautiful Mind: <br>
    A Biography of John Forbes Nash Jr., de Sylvia Nasar. O <br>
    filme conta a história real de John Nash que, aos 21 anos, <br>
    formulou um teorema que provou sua genialidade. Brilhante, <br>
    Nash chegou a ganhar o Prêmio Nobel. Diagnosticado como <br>
    esquizofrênico pelos médicos, Nash enfrentou batalhas em <br>
    sua vida pessoal, lutando até o fim de sua vida.<br>
</p>
</body>
</html>

```

Observação 1: Você notou que foi escolhido o filme “Uma mente brilhante” e esse código é a resposta para esse filme.

Observação 2: Lembre-se ainda que a imagem está na mesma pasta do meu arquivo HTML, por este motivo foi informado apenas o nome da imagem e sua extensão (tipo de arquivo: jpg), no código acima.

Resultado no Navegador:



Imagens: crédito Freepik.com