

HTML BÁSICO



HTML

SUMÁRIO

INTRODUÇÃO	1
• SURGIMENTO DA LINGUAGEM	1
• PARA QUE SERVE O HTML?	1
• TIPOS DE PÁGINAS WEB	1
• BROWSER/NAVEGADOR	2
• TAGS	2
ESTRUTURA INICIAL DO DOCUMENTO HTML	3
ATRIBUTOS DO HTML	3
CABEÇALHO DO DOCUMENTO HTML	4
CODIFICAÇÃO DOS CARACTERES DO DOCUMENTO HTML	5
TÍTULO DO DOCUMENTO HTML	5
CORPO DO DOCUMENTO HTML	6
INSERINDO ELEMENTOS NA PÁGINA	7
CABEÇALHO DA PÁGINA	8
TAG HR	12
QUEBRA DE LINHA	15
PARÁGRAFO	18
FORMATAÇÃO DE TEXTOS - PARTE I	21
ENTIDADE DE CARACTERE	24
FORMATAÇÃO DA PÁGINA	26
TABELA DE CORES	30
FORMATAÇÃO DO TEXTO - PARTE II	36

LINKS	39
CRIANDO PÁGINAS	42
LISTAS	44
IMAGENS NA PÁGINA	48
TABELAS	56
LAYOUT	62
EVOLUÇÃO DO HTML	65
CSS - FOLHA DE ESTILO EM CASCATA	67
SINTAXE DA PROPRIEDADE CSS	71
PRINCIPAIS ATRIBUTOS CSS	75
DIV	93
SPAN	97
FORMULÁRIOS	98
LAYOUT COM DIV	107
AGRADECIMENTO	115

Introdução

Surgimento da linguagem

O HTML foi criado em 1991, por **Tim Berners-Lee**, no **CERN (European Council for Nuclear Research)** na Suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade.

O HTML é baseado da linguagem **www (world wide web – ampla rede mundial)** que é interpretada por todos os Browsers (navegadores) disponíveis.

Para que serve o HTML?

HTML significa **Hiper Text Markup language**, ou seja, **Linguagem de Marcação de Hiper Texto**. A linguagem é usada para criar com simplicidade páginas na web, estabelecendo como um determinado elemento deverá ser visualizado, não sendo portanto, uma linguagem de programação, e sim, uma linguagem de formatação de exibição de textos, através de comandos especiais chamados de **TAGS**.

O documento HTML pode ser escrito em qualquer editor de texto (**Wordpad, Microsoft Word, Bloco de Notas etc.**), basta salvar o documento definindo a extensão **.html**, que é a extensão utilizada pelos documentos HTML.

Porém, a maneira mais rápida e fácil de criar um documento HTML é usar um editor de código, como o Dreamweaver, Sublime Text, Notepad++, etc. Estes encurtam os caminhos, pois possuem atalhos que ajudam o trabalho.

Uma página web nada mais é que um texto normal, daqueles que você digita no Bloco de Notas, com terminação **.html** para ser lido por um Browser (Navegador).

Por fim, o HTML representa o desenvolvimento da Internet, não é exagero dizer, que boa parte do que temos hoje, deve-se ao surgimento e desenvolvimento da linguagem ao longo de sua história.

Tipos de Página Web:



Browser/Navegador

São os softwares (**programas**) responsáveis pelo acesso do usuário aos sites disponíveis na internet. O **Browser** ou **Navegador** lê e interpreta todos os arquivos HTML que são enviados pela www, transformando-os em páginas web. Existem alguns tipos de Browsers como: Mozilla Firefox, Opera, Safari, Google Cromos, Internet Explorer entre outros.



Tags

Para que você possa entender o que são as tags, vou começar explicando o significado da palavra. Tag em inglês significa **etiqueta, rótulo**, a maioria das palavras na linguagem do computador são originadas do inglês. E etiqueta é algo que colocamos em objetos, mercadorias, arquivos, etc. para nos ajudar na vida diária, para nos orientar.



Podemos então dizer que tags são etiquetas que não aparecem na tela do computador, e que servem para orientar alguém. Esse alguém construiu um robot que vai a uma página da web e recolhe as informações contidas nessas etiquetas. Por exemplo, uma das informações que ele recolhe é o título da página, a qual precisa estar devidamente etiquetada, para que o robot "saiba" o que ela contém. O robot é um programa, e nestas circunstâncias, é instruído a recolher as informações que estão contidas dentro das etiquetas. Para ser mais específico, no exemplo do título, ao entrar na página, ele se depara com várias instruções, e uma delas é a seguinte: title. Isto significa que o nome da etiqueta (**tag**) é **title (= título)**. Logo, o início do título da página é precedido da palavra title, assim representada. Dizemos que a tag título está aberta, e que vai começar a ser colocado o título. Vamos supor que o título da página seja "**Início**". A tag **título** ficará assim: `<title>Início</title>`.

As tags são formadas por sinais de maior < e menor >, entre os sinais é onde ficam os atributos, como vimos no exemplo do **title**. As tags possuem **abertura, atributo, elemento e fechamento**. O seletor de atributos combina elementos baseado no valor de um atributo dado. Existem algumas tags que não possuem fechamento, como a **tag LINK**, por exemplo, assunto no qual comentaremos mais adiante.

Atributo	
<code><title></code>	Abertura
Início	Elemento
<code></title></code>	Fechamento

Tudo que estiver entre as tags de abertura e fechamento do documento HTML compõem o documento inicial ou seja, a página web. Os componentes do documento inicial são:

Abertura	Tag <!doctype html>	Exemplo:
cabeçalho	Tag <head></head>	<doctype html> Abertura inicial <head> </head>
Corpo	Tag <body></body>	<body> </body>
Fechamento	Tag </html>	</html> Fechamento do Documento

Estrutura Inicial do Documento HTML:

```
<!doctype html>  
<html lang =”pt-br”>  
<head>  
<meta charset =”UTF-8”>  
<title>Início</title>  
</head>  
<body>  
</body>  
</html>
```

Para que entenda melhor, explicarei cada atributo do Documento Inicial.

Atributos do HTML

Abertura do Documento HTML <!doctype html>

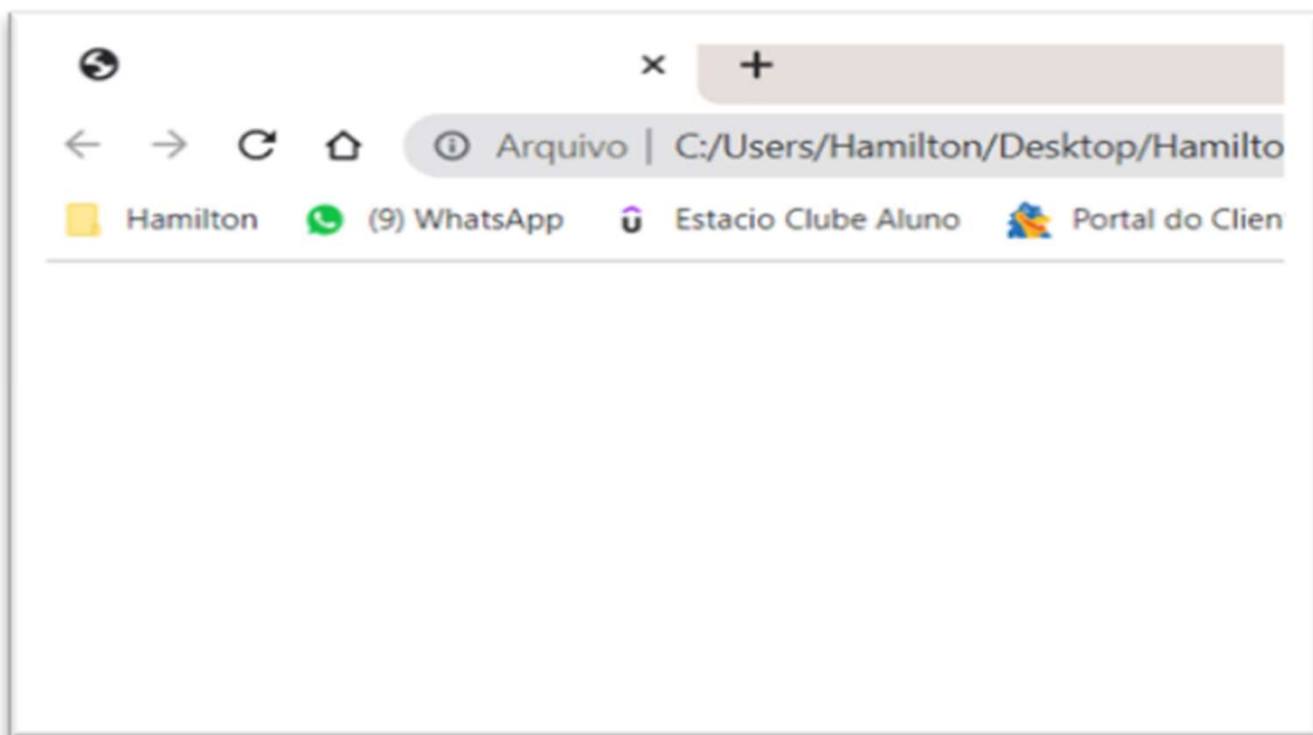
Este é o atributo principal do documento inicial, através deste, estamos informando ao Browser/Navegador que queremos abrir uma página na web.

Linguagem do Documento HTML <html lang>

A tag html lang informa ao Navegador /Browser o tipo de linguagem que queremos, no caso, aqui no Brasil, é o pt-br : <html lang =”pt-br”>

Fechamento do Documento HTML </html>

Este é o atributo de fechamento do documento. Sendo assim, tudo que estiver entre as tags de abertura e fechamento: <!doctype html> </html> formam uma página na web. Note que neste exemplo a página está vazia, isto porque ainda não inserimos nenhum conteúdo ainda. Não se preocupe, faremos isso mas adiante.



Cabeçalho do Documento HTML

O cabeçalho de um documento HTML é definido pela tag <head> que fornece a identificação lógica do documento com o navegador.

No cabeçalho estão as identificações da origem do documento, trata-se da primeira seção do documento. A tag `<head>` é uma tag dupla, pois ela deve ser aberta e fechada para apontar o início e o fim das informações do cabeçalho.

As informações do cabeçalho são apresentadas através de outros elementos: TITLE, META, STYLE, SCRIPT e LINK. Uma outra grande utilidade da tag `<head></head>` na criação de um site, é que lá ficam localizados scripts, como de JavaScript. Assunto à ser discutido no próximo módulo do curso.

Codificação dos Caracteres do HTML

`<meta charset>`

O atributo charset serve para indicar o formato de codificação de caracteres utilizado no documento. Cada 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. Ao usar o charset dentro da tag meta de nosso arquivo estamos dizendo ao navegador: quando for ler esse arquivo, use as regras do **CHARSET UTF-8**.

Título do Documento HTML

`<title>...</title>`

É o título que identifica primeiramente o documento para o usuário, é a primeira informação a ser mostrada enquanto a página é carregada. O título de uma homepage é criado pela tag `<title>`. O que é definido entre `<title>Início</title>` aparece na barra de títulos da página como referência em listas de links úteis criados pelo usuário, como os favoritos.



Corpo do Documento HTML

`<body> </body>`

A tag `<body>`, que junto com a tag `<html>` e a `<head>` formam a tríade de tags mais importantes e essenciais de qualquer site feito em HTML.

Quando pensamos em conteúdo, nos surge a mente textos, informações escritas. Sim, com certeza os textos são o tipo mais abundante de conteúdo existente na internet, mas conteúdo não se limita somente a isso, a letras. Imagens, vídeos, animações e qualquer coisa que o usuário possa ter contato, é conteúdo. Se muita coisa pode ser considerada conteúdo, muitas são as utilidades da tag **<body>** então, é por esse motivo que ela é uma tag tão importante e considerada básica/essencial para qualquer site da internet.

Portanto, tudo que estiver entre as tags **<body>**, formam o corpo da página de um site. É o corpo do documento, ou seja, toda a parte “física”. A tag **<body>** demarca tudo o que irá aparecer quando a página for exibida pelo navegador que identifica a tag **<body>**, interpreta e exibe todo o conteúdo do documento até encontrar a tag **</body>**.

Salvando o Documento HTML

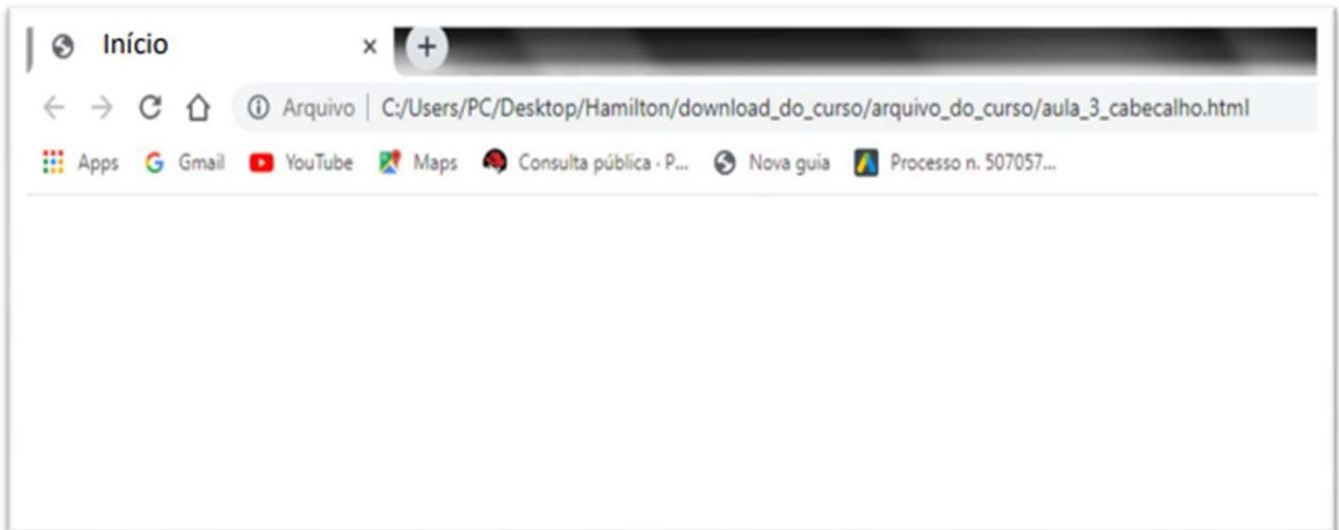
Digite a estrutura padrão no editor de códigos bloco de notas:

```
<!doctype html>
<html lang = "pt-br">
  <head>
    <meta charset = "UTF-8">
    <title> Início</title>
  </head>
  <body>
  </body>
</html>
```

Para visualizarmos a página na web basta salvar o arquivo com a extensão **.html**. Vamos ver como isso funciona na prática. Com o documento aberto no bloco de notas salve o arquivo como: **index.html** na área de trabalho. Por padrão a primeira página do documento a nomeamos como, **index.html** e sempre com letras minúsculas, as demais podem ser nomeadas pelo nome que achar conveniente na construção de seus projetos.

Exemplo: galeria.html, contato.html, etc.

Muito bem, feito isso abra o arquivo **index.html** que você acabou de salvar para ver o resultado.



Inserindo Elementos na Página

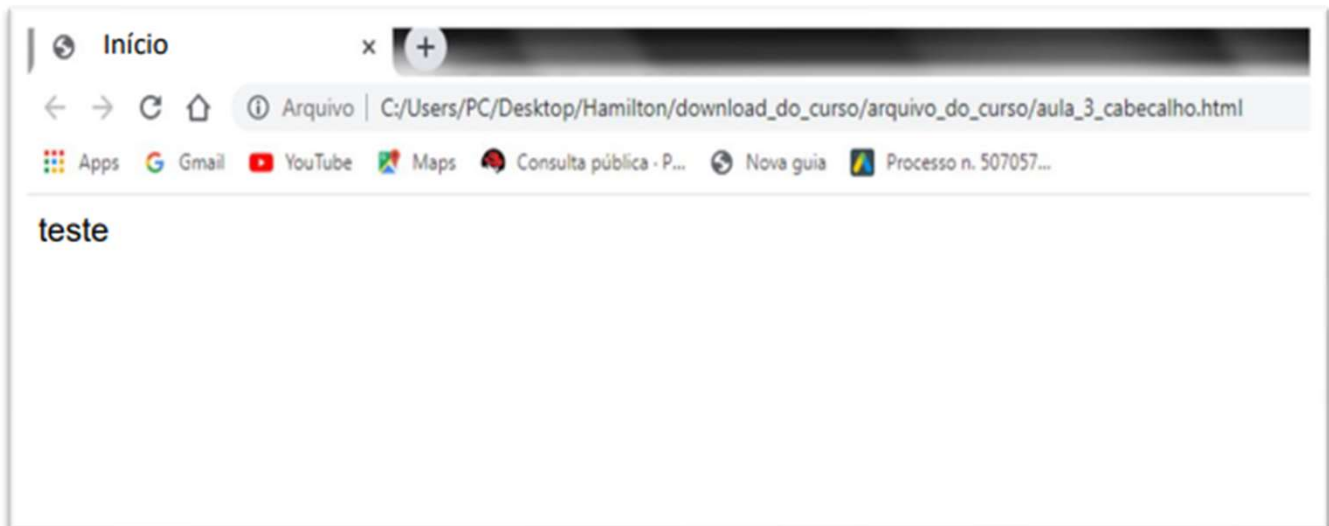
Ao digitar os códigos da estrutura no bloco de notas e visualizar no Navegador/Browser perceba que a página está sem conteúdo algum. Isso porque a tag `body` que é responsável pelo corpo da página está vazia. Para que o conteúdo seja exibido é necessário a inserção de textos, imagens, links etc. É exatamente isso que faremos agora, a começar pelo cabeçalho da página.

Nota: Não confunda **cabeçalho da página** com **cabeçalho do documento HTML**. Para que entenda melhor, faremos um teste.

Com a estrutura padrão digitalizada no bloco de notas, digite a palavra **teste** abaixo da tag `body`.

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
teste
</body>
</html>
```

Salve o documento como teste.html na área de trabalho e abra o documento no navegador/browser para ver como fica.



Cabeçalho da Página

"Cabeçalhos" normalmente são usados para títulos e subtítulos de uma página. O HTML possui seis níveis de cabeçalhos, numerados de 1 a 6, sendo o número 1 o de maior destaque. Cabeçalhos são exibidos em letras maiores e em negrito. O primeiro cabeçalho em cada documento deve estar marcado como **<h1>**.

ATENÇÃO: ao definir o tamanho de um cabeçalho, você não está definindo o tamanho da letra (fonte 10, fonte 14). Você apenas define que ele aparecerá com maior tamanho e destaque que o resto do texto. O tamanho exato com que ele será visualizado é definido pelo programa visualizador de **html (browser)** de cada pessoa que acessar a informação.

As notações relativas a cabeçalhos são: **<hy>** texto do cabeçalho **</hy>** onde y é um número entre 1 e 6 especificando o nível do cabeçalho.

<h1>Headings**</h1>**

<h2>Headings**</h2>**

<h3>Headings**</h3>**

<h4>Headings**</h4>**

<h5>Headings**</h5>**

<h6>Headings**</h6>**

HTML BÁSICO

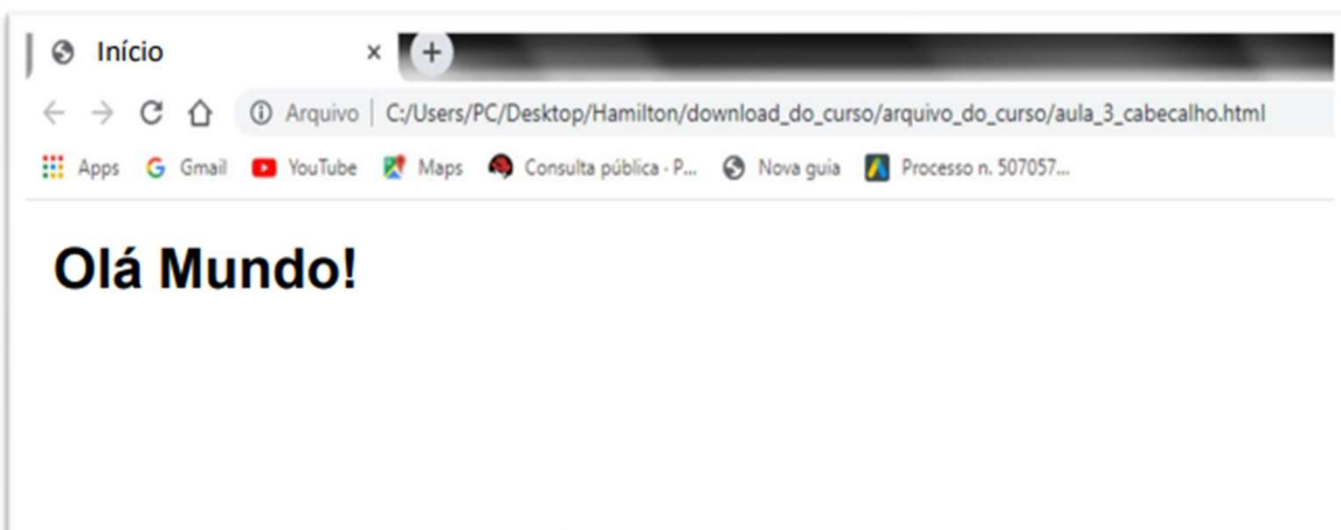


Para que entenda melhor faremos um outro teste. Com a mesma estrutura inicial aberta no editor bloco de notas, apague a palavra teste e digite:

```
<h1>Olá Mundo!</h1>
```

```
<!doctype html>  
<html lang = "pt-br">  
<head>  
<meta charset = "UTF-8">  
<title> Início</title>  
</head>  
<body>  
<h1>Olá Mundo!</h1>  
</body>  
</html>
```

Se você digitou tudo corretamente, ao salvar o documento com a extensão **.html**, o resultado ficará como está abaixo:



As tags **h1**, **h2**, **h3**, **h4**, **h5** e **h6** servem para organizar o seu conteúdo aos olhos do tio Google, por padrão elas já vem configuradas como sendo os Cabeçalhos que usamos como títulos em um artigo. Ou seja, quando você usa um título com cabeçalho.

HTML BÁSICO

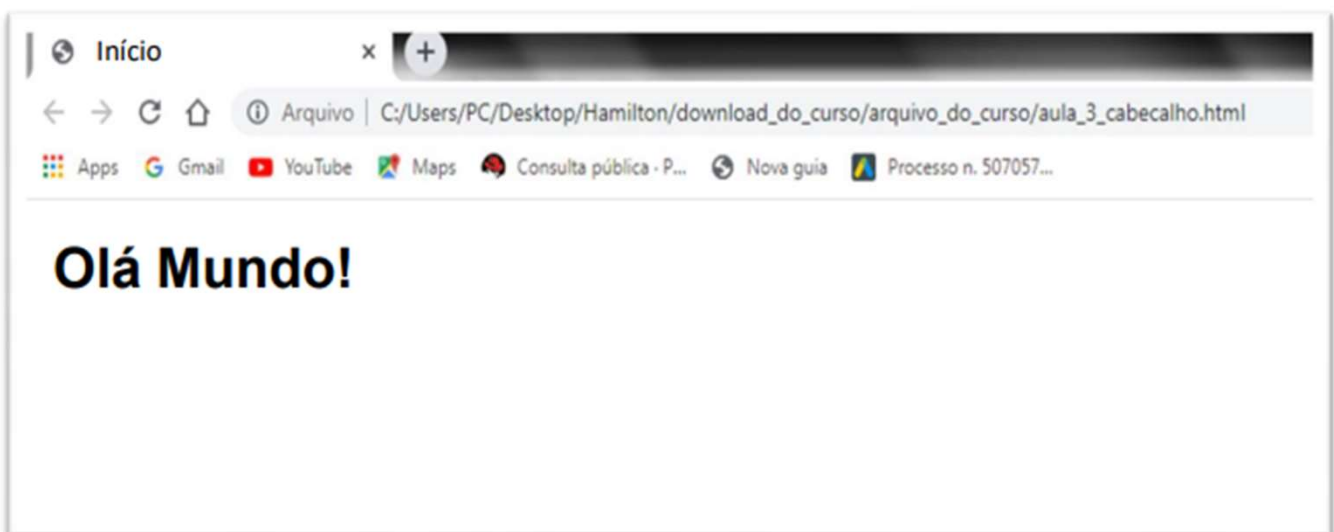


Para que entenda melhor faremos um outro teste. Com a mesma estrutura inicial aberta no editor bloco de notas, apague a palavra teste e digite:

```
<h1>Olá Mundo!</h1>
```

```
<!doctype html>  
<html lang = "pt-br">  
<head>  
<meta charset = "UTF-8">  
<title> Início</title>  
</head>  
<body>  
<h1>Olá Mundo!</h1>  
</body>  
</html>
```

Se você digitou tudo corretamente, ao salvar o documento com a extensão **.html**, o resultado ficará como está abaixo:



As tags **h1**, **h2**, **h3**, **h4**, **h5** e **h6** servem para organizar o seu conteúdo aos olhos do tio Google, por padrão elas já vem configuradas como sendo os Cabeçalhos que usamos como títulos em um artigo. Ou seja, quando você usa um título com cabeçalho.

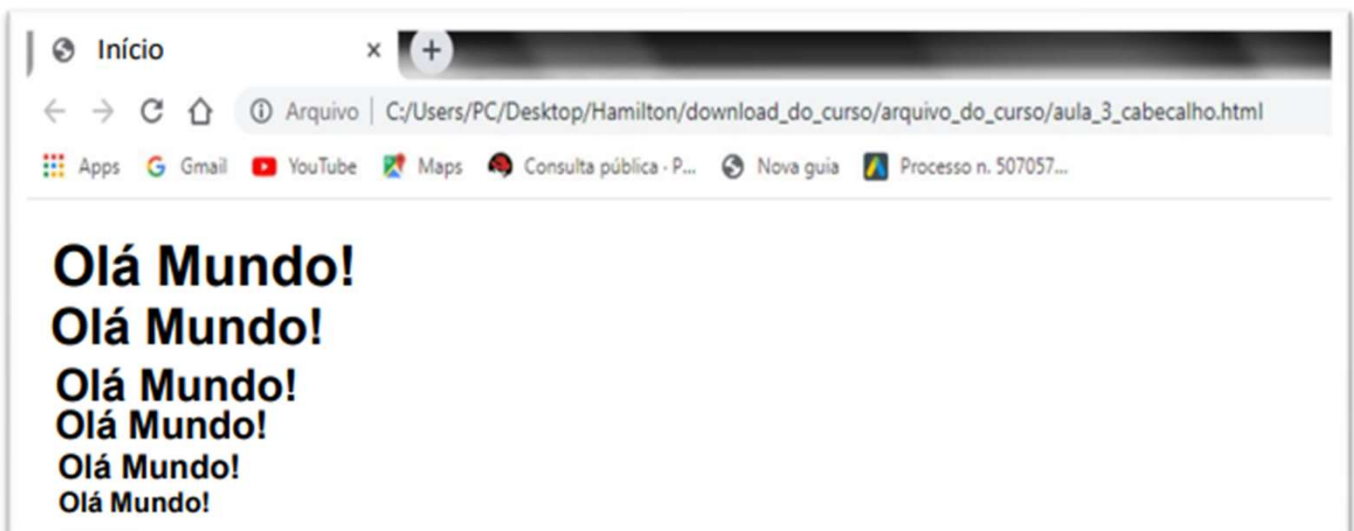
HTML BÁSICO



Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
<h1>Olá Mundo!</h1>
<h2>Olá Mundo!</h2>
<h3>Olá Mundo!</h3>
<h4>Olá Mundo!</h4>
<h5>Olá Mundo!</h5>
<h6>Olá Mundo!</h6>
</body>
</html>
```

Se você digitou corretamente, ao salvar o documento com a extensão .html, o resultado ficará como está abaixo:



Tag HR

O elemento HR está previsto nas recomendações do W3C para o HTML 4.01 e destina-se a inserir uma Régua Horizontal (HR é a abreviatura de Horizontal Rule – em inglês) em um documento Web, quando renderizado por um agente de usuário em modo gráfico ou visual. Trata-se de um elemento vazio, isto é, não existe a tag de fechamento, basta escrever simplesmente `<hr>`. Se você estiver codificando XHTML lembre-se: elementos vazios devem ter uma tag de fechamento ou a tag de abertura deve terminar com `/>`. Como exemplo; `<hr/>`.

Atributos do elemento HR

- **align** - Define o alinhamento da régua à esquerda, no centro ou à direita.
- **Size** - Controla a altura da régua.
- **Width** - Controla o comprimento da régua. O valor default é de 100% ou seja, a régua estende-se por toda a largura da tela.

Nota: Todos os atributos para régua horizontal estão em desuso. Use **Estilo CSS** para obter os efeitos do atributo. Mas não se preocupe, mas adiante abordaremos este assunto.

Marcação do elemento HR

A seguir alguns exemplos de marcação em desuso, mostrados como ilustração e comparações com emprego de folha de estilos.

Exemplos:

```
<hr/>  
<hr width=50%/>  
<hr width=50px/>  
<hr size=10 />  
<hr align=right />  
<hr align=center/>
```

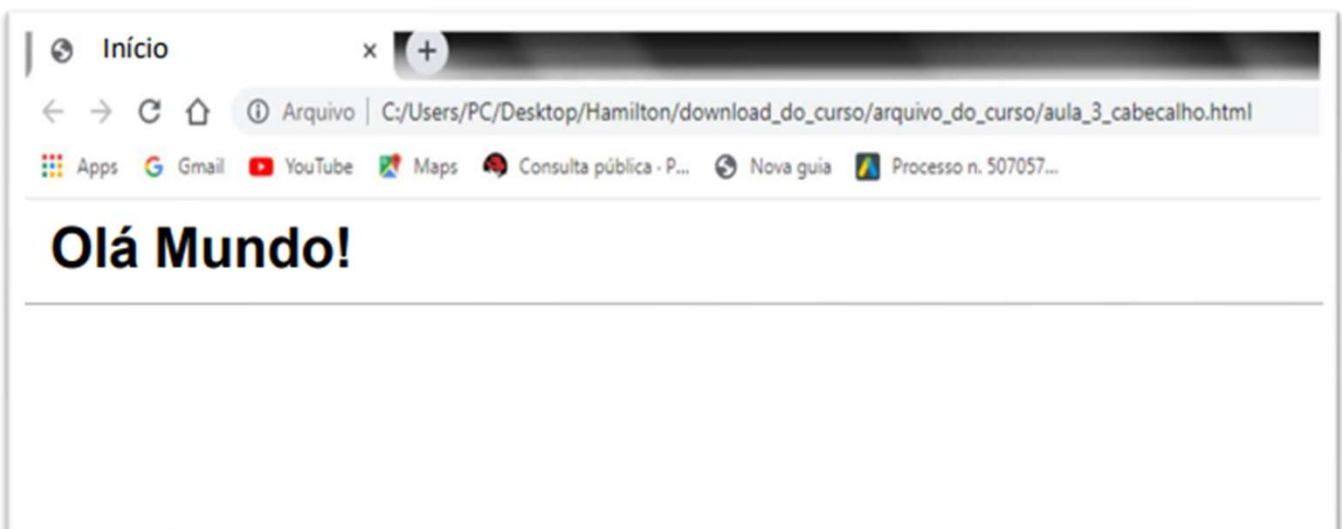
HTML BÁSICO



Com os conceitos abordados faremos um pequeno teste. Com a estrutura inicial aberta, digite no body o primeiro exemplo citado na página anterior dessa forma:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
<h1>Olá Mundo!</h1>
<hr/>
</body>
</html>
```

Salve o documento e abra no Navegador/Browser para ver como fica:



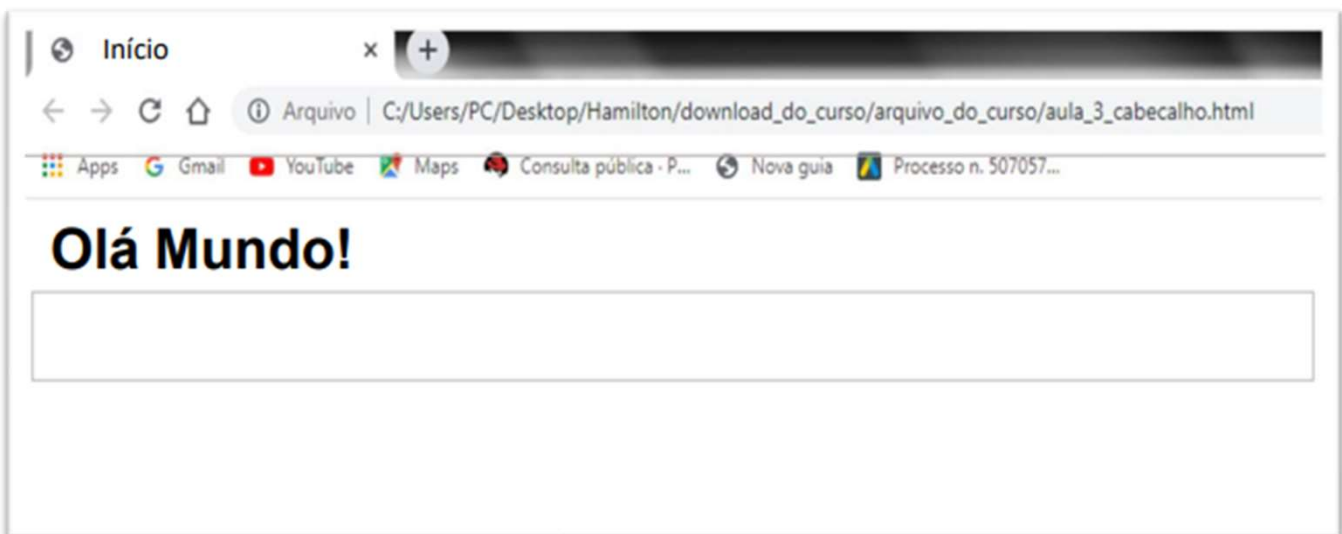
HTML BÁSICO



Veja esse outro exemplo.

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
<h1>Olá Mundo!</h1>
<hr size=50/>
</body>
</html>
```

Salve o documento e abra no Navegador/Browser para ver como fica:



Agora e com você, faça a mesma coisa com os outros exemplos e pratique bastante.

✓ **Lembre-se: Quanto mais você prática, melhor você fica!**

Quebra de Linha

Para quebrar a linha, ou seja, para o HTML dar um salto de uma linha para outra (o que acontece quando damos enter em um editor de textos), basta usarmos a tag: `
`. O br vem do inglês break. Você pode simplesmente usar a tag `
` para fazer uma quebra de linha, a qualquer momento. Porém, deixar uma tag aberta pode causar alguns problemas. O XML, por exemplo, não aceita só uma tag aberta assim, e também há algumas questões de compatibilidade com o XHTML. Para mais informações sobre essa discussão, acesse:

<http://stackoverflow.com/questions/1946426/html-5-is-it-br-br-or-br>

Vamos pegar uma letra de uma música, da abertura do antigo desenho "Cavalo de Fogo", ela é assim:

No meu sonho eu já vivi;
Um lindo conto infantil;
Tudo era magia;
Era um mundo fora do meu;
E ao chegar desse sono acordei;
Foi quando correndo eu vi;
Um cavalo de fogo ali;
Que tocou meu coração;
Quando me disse, então;
Que um dia pudesse acabar;
No mundo dos sonhos pudesse um dia rainha eu seria;
Se com a maldade chegar.

Você conseguiria escrever ela em um papel? Fácil, né. E em um editor de textos, como o bloco de notas ou Word? Mais fácil e rápido ainda. E no chat do Facebook? Mesma coisa. Pois vamos tentar escrever essa letra em um site, em HTML.

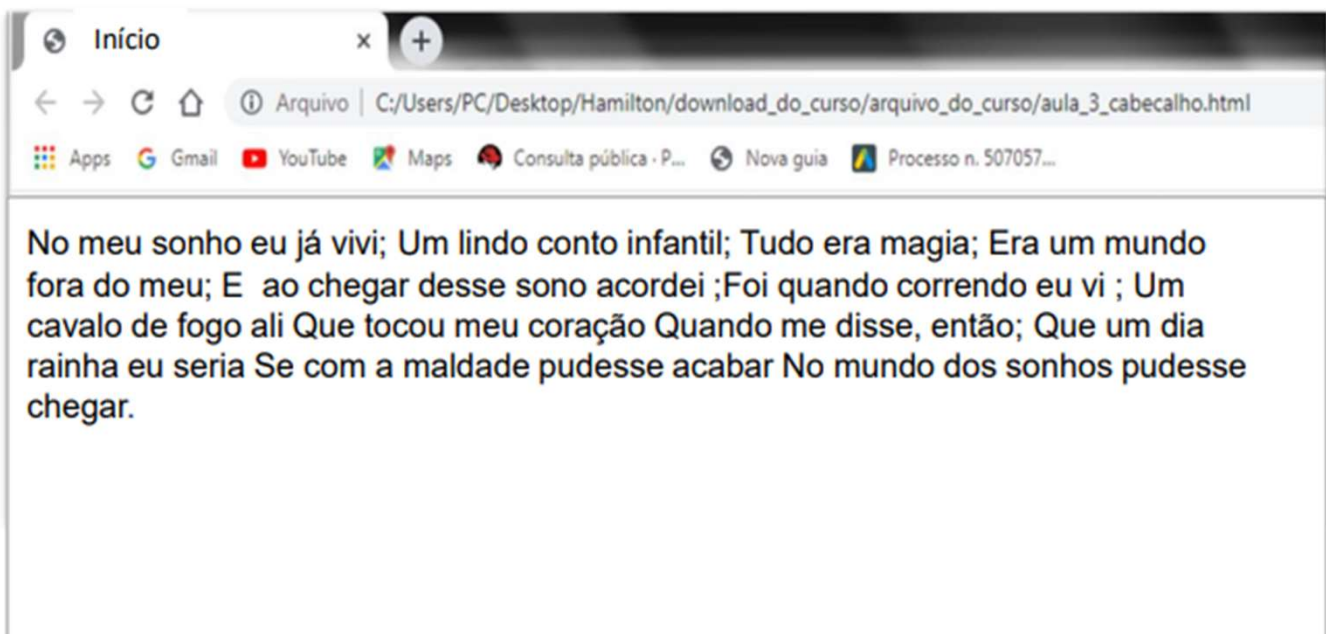
HTML BÁSICO



Com a estrutura inicial do documento HTML aberta, digite a letra da musica.

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Cavalo de Fogo</title>
</head>
<body>
No meu sonho eu já vivi;
Um lindo conto infantil;
Tudo era magia;
Era um mundo fora do meu;
E ao chegar desse sono acordei ;
Foi quando correndo eu vi ;
Um cavalo de fogo ali Que tocou meu coração Quando me disse, então;
Que um dia rainha eu seria Se com a maldade pudesse acabar No mundo dos sonhos
pudesse chegar.
</body>
</html>
```

Agora salve o documento e abra no navegador para ver como fica:



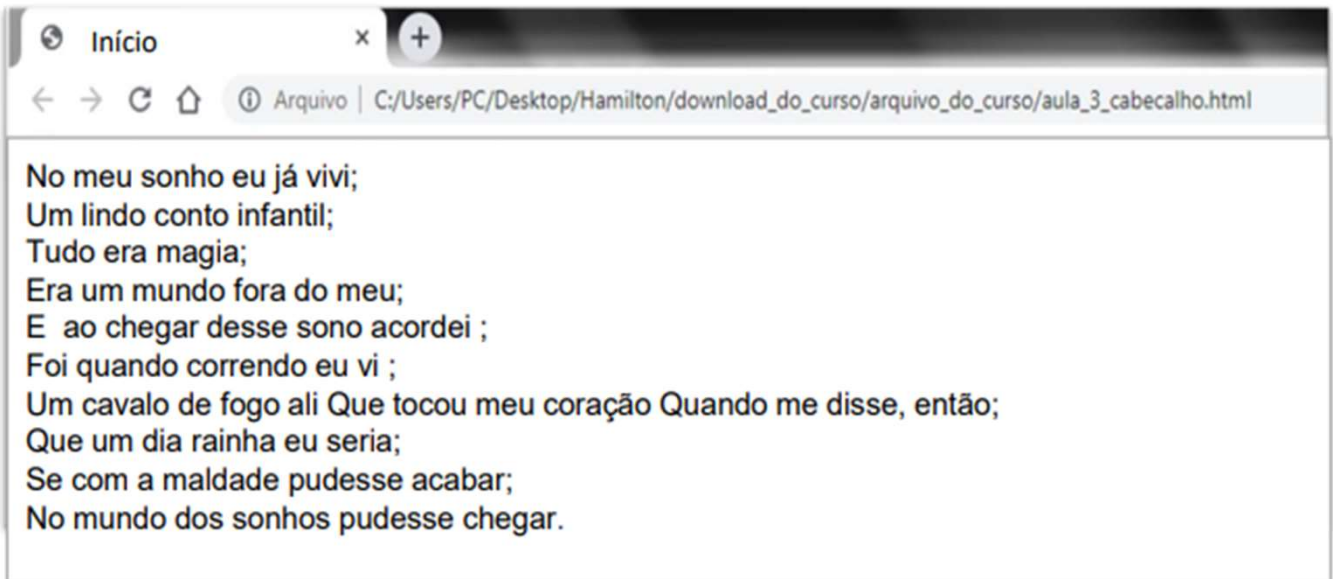
Você percebeu que ao carregar a página no navegador, a letra da musica digitada no editor ficou sem formatação alguma? Já deve ter notado também que esses códigos são uma maneira de se comunicar com os navegadores /browsers.

Não é à toa que o L de HTML é de "language", pois é uma linguagem, é uma maneira diferente de se 'conversar'. Neste artigo vamos ter mais uma prova que HTML é uma linguagem diferente. Não é como escrever em um chat ou em um editor de textos.

Pode estar sendo confuso, mas você vai entender agora o que estou falando, exatamente. Para adicionar quebra de linha no documento, basta digitar a tag
 no final de cada frase da musica, para que sendo assim, ao visualizar no navegador a letra da musica apareça formatada.

Vamos ver como isso se comporta na prática: Digite dessa forma no documento HTML. Salve e depois abra no navegador para ver como fica.

```
<!doctype html>
<html lang ="pt-br">
<head>
<meta charset ="UTF-8">
<title>Cavalo de Fogo</title>
</head>
<body>
No meu sonho eu já vivi;<br/>
Um lindo conto infantil;<br/> Tudo era magia;<br/>
Era um mundo fora do meu; <br/>E ao chegar desse sono acordei ;<br/>
Foi quando correndo eu vi;<br/>
Um cavalo de fogo ali;<br/>Que tocou meu coração;<br/>Quando me disse,
então;<br/>Que
um dia rainha eu seria;<br/>Se com a maldade pudesse acabar;<br/>No mundo dos
sonhos pudesse chegar.
</body>
</html>
```



Parágrafo

Todos os textos são divididos em parágrafos e em uma homepage não é diferente. Com isso em uma programação HTML é utilizada a tag **<P>** que, através dela, é especificado onde começa e termina um parágrafo. Todo o texto que for dividido entre as tag's **<p>** e **</p>** será reconhecido pelo browser como um parágrafo. Essa tag tem o seu próprio atributo e suas propriedades de alinhamento, que são:

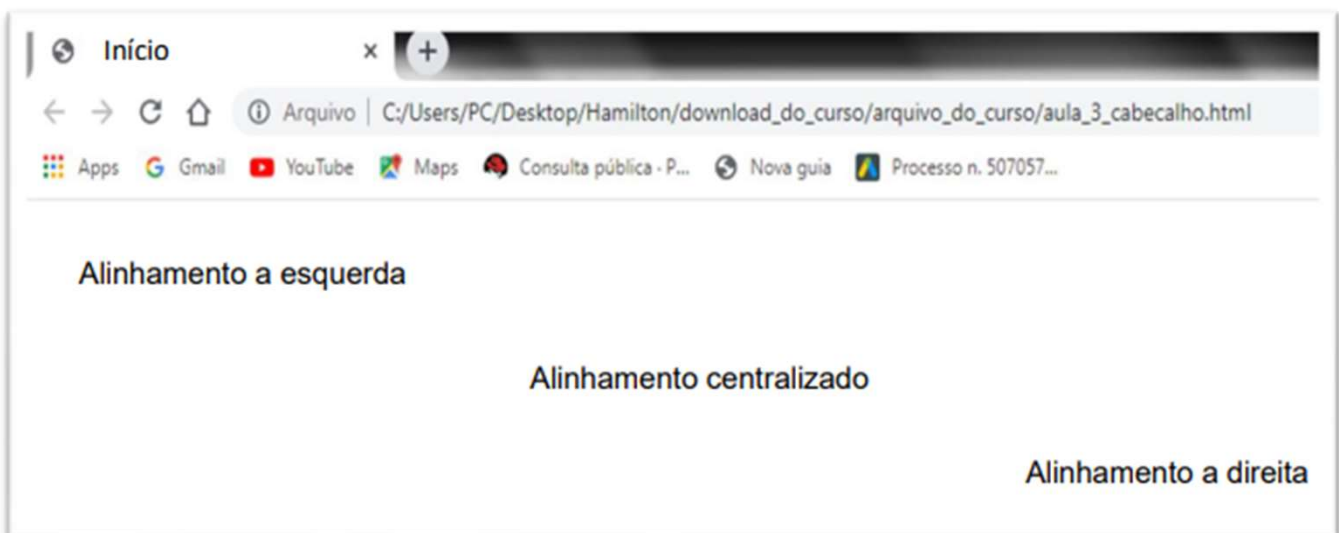
- **Atributo - Align** - Usado para especificar como será alinhado o texto: esquerda (padrão), direita, centralizado e justificado.

Vamos ver dois exemplos:

Exemplo 1:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body>
<p align="left">
Alinhamento a esquerda
</p>
<p align="center">
Alinhamento centralizado
</p>
<p align="right">
Alinhamento a direita
</p>
</body>
</html>
```

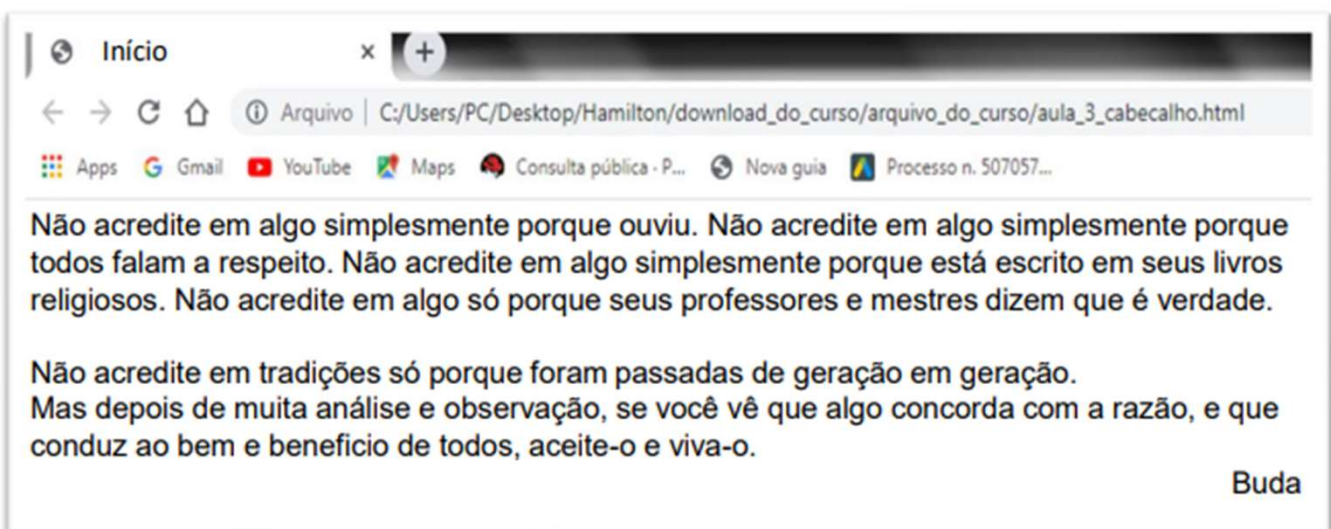
Salve e veja no navegador para ver como fica:



Exemplo 2:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body>
<p>Não acredite em algo simplesmente porque ouviu.
Não acredite em algo simplesmente porque todos falam a respeito.
Não acredite em algo simplesmente porque está escrito em seus
livros religiosos. Não acredite em algo só porque seus professores e
mestres dizem que é verdade.
Não acredite em tradições só porque
foram passadas de geração em geração.</p>
<p>Mas depois de muita análise e observação, se você vê que algo
concorda com a razão, e que conduz ao bem e benefício de todos,
aceite-o e viva-o.</p>
<p align=right> Buda</p>
</body>
</html>
```

Salve e veja no navegador para ver como fica:



Agora é com você, digite outros textos, no editor de códigos bloco de notas acrescentando as tag's <p> </p> conforme os exemplos acima.

✓ **Lembre-se: O segredo para aprender e ficar bom é treinar bastante!**

Formatação de Textos Parte |

Estes exemplos demonstram como você pode formatar texto em um documento HTML.

TAG	DESCRIÇÃO
<code></code>	texto em negrito
<code><big></code>	texto grande
<code></code>	texto enfatizado
<code><i></code>	texto em itálico
<code><small></code>	Texto pequeno
<code></code>	texto forte

Vamos ver como isso se comporta no documento HTML:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title> Início</title>
</head>
<body>

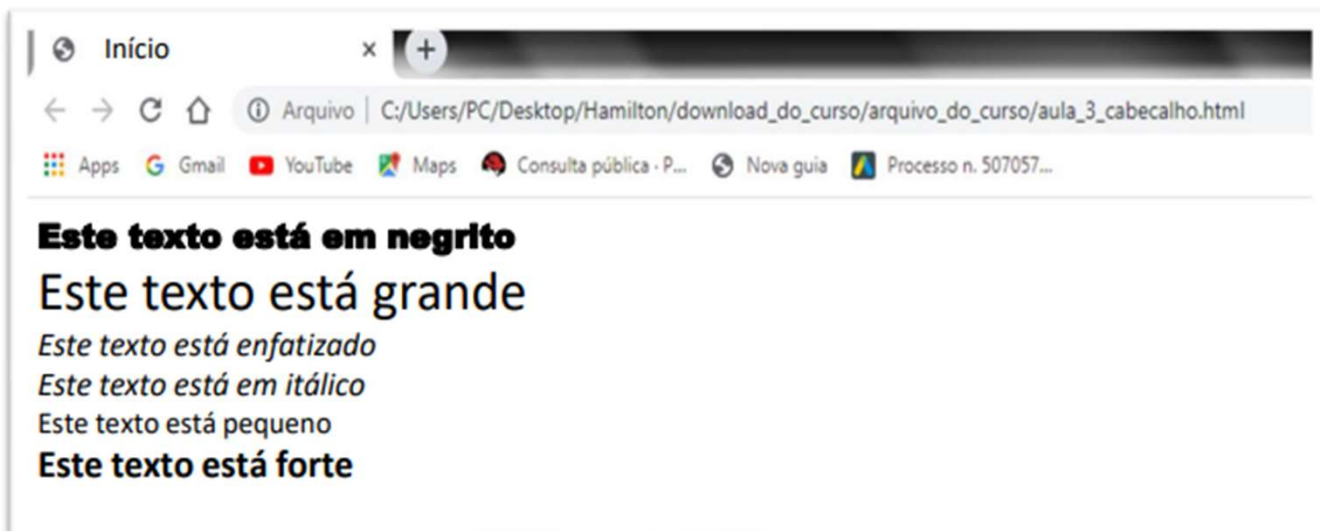
<b>Este texto está em negrito</b><br/>
<big>Este texto está grande</big><br/>
<em>Este texto está enfatizado</em><br/>
<i>Este texto está em itálico</i><br/>
<small>Este texto está pequeno</small><br/>
<strong>Este texto está forte</strong>

</body>
</html>
```


HTML BÁSICO



Salve o documento e veja como fica:



Vamos ver mais exemplos:

TAG	DESCRIÇÃO
<sub>	Texto sobescrito
<sup>	Texto subscrito
<ins>	Texto inserido
	Texto deletado
<s>	usa-se
<strike>	Usa-se
<u>	Usa-se estilos/css

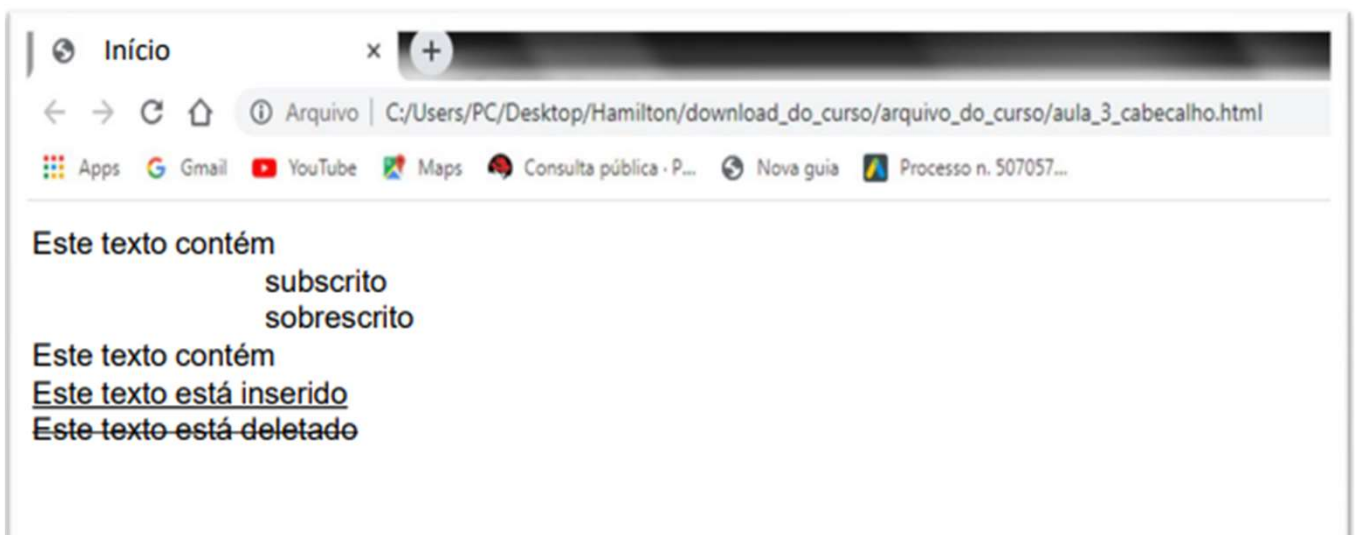
HTML BÁSICO



Digite assim no documento HTML:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
Este texto contém<sub>subscrito</sub><br/>
Este texto contém<sup>sobrescrito</sup><br/>
<ins>
Este texto está inserido
</ins><br/>
<del>
Este texto está deletado
</del>
</body>
</html>
```

Salve o documento e veja como fica:



Entidade de Caractere

Alguns caracteres têm significado especial no HTML, como o sinal de menor do que (<) que define o início de uma tag de HTML. Se quisermos que o navegador exiba de fato estes caracteres devemos inserir entidades de caractere na fonte da HTML.

Uma entidade de caractere tem três partes: um e comercial (&), um nome de entidade ou um # e um número de entidade, e finalmente um ponto e vírgula (;). Para exibir um sinal de menor do que em um documento HTML devemos escrever: < ou < A vantagem de usar um nome em vez de um número é que o nome é mais fácil de lembrar.

A desvantagem é que nem todos os navegadores suportam os nomes de entidades mais novos, enquanto que o suporte para números de entidade é muito bom em quase todos os navegadores. Observe que as entidades são sensíveis à caixa. Alguns caracteres como o caractere <, têm um significado especial no HTML e por isso não podem ser usados no texto.

Para exibir um sinal de menor do que (<) em HTML, temos que usar uma entidade de caractere.

Este exemplo permite que você experimente entidades de caracteres:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title> Início</title>
</head>
<body>
<p> Exemplos de entidade de caractere:</p>
<p>&#60; &#62; &#169; &nbsp; &#174; &nbsp;
&quot; &nbsp; &cent;</p>
</body>
</html>
```

Salve o documento e veja como fica:



Entidade de Caractere mais Comuns:

Resultado	Descrição	Nome	Número
	Espaço	 	
<	Menor que	<	<
>	Maior que	>	>
&	E comercial	&	&
“	Aspas	"	"
'	Apóstrofe		'
¢	Cent	¢	¢
£	Libra	£	£
¥	Yen	¥	¥
§	Parágrafo	§.;	§
©	Copyright	&cop;	©
®	M. registrada	®	®
×	Multiplicação	×	×

Formatação da Página

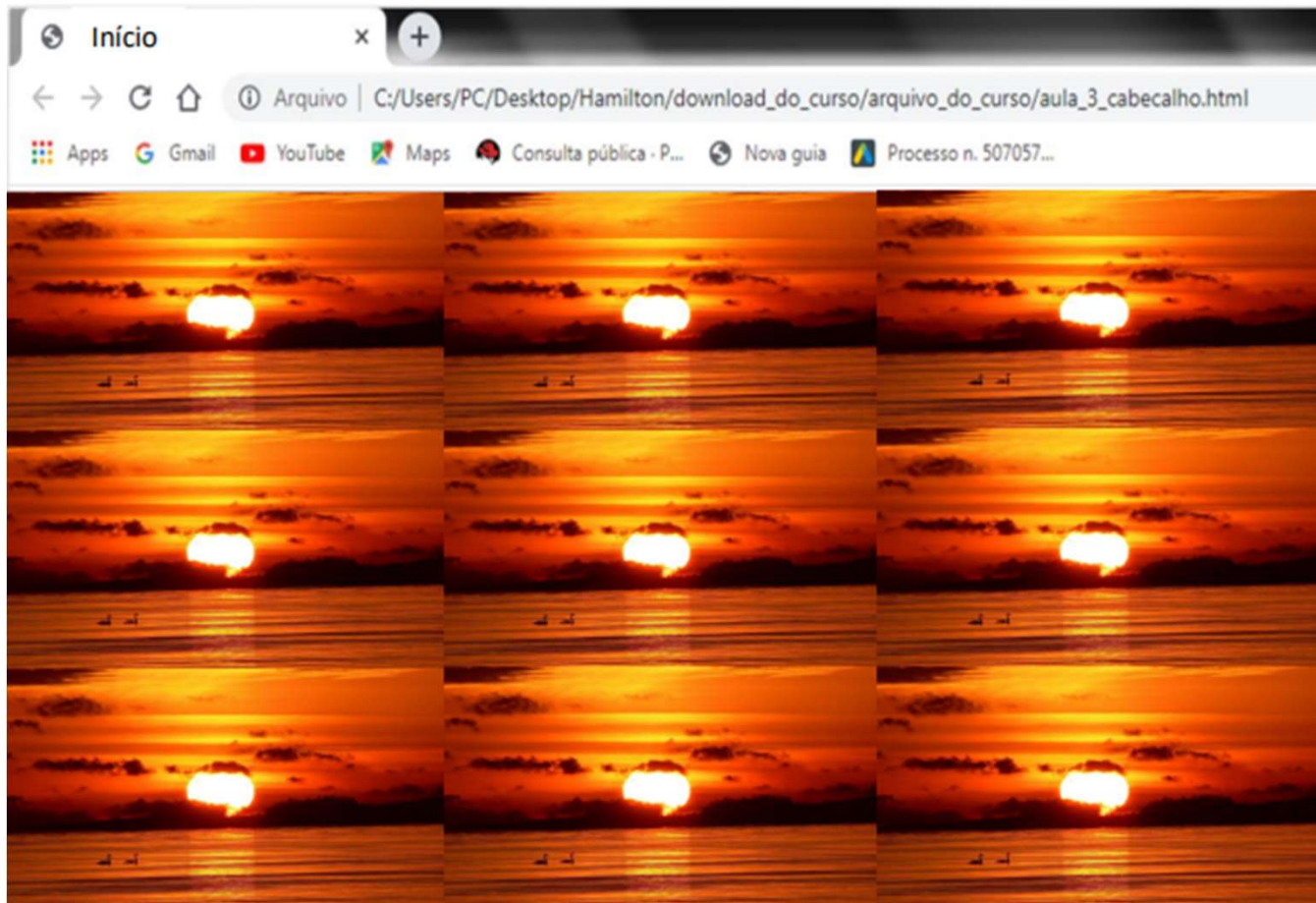
Além de demarcar e marcar o corpo do documento HTML, a tag <body> pode, opcionalmente, determinar as cores do fundo do texto, de link ou, ainda definir a imagem que será utilizada como um “plano de fundo” da página. Esses efeitos são atingidos através do uso dos seguintes atributos:

- **BACKGROUND** - Define a figura que será usada como plano de fundo da página.

Vamos ver como funciona isso na prática: Abra o Google Chrome imagens e salve uma imagem qualquer de tamanho 1920 X 1080. Renomeie essa imagem com nome fundo e salve a mesma na área de trabalho. Esse é o link da imagem que vamos usar no exemplo abaixo:

<https://wall.alphacoders.com/big.php?i=94708&lang=Portuguese>

Nota: Se a imagem for menor que a página ela será repetida até que ocupe toda a página.



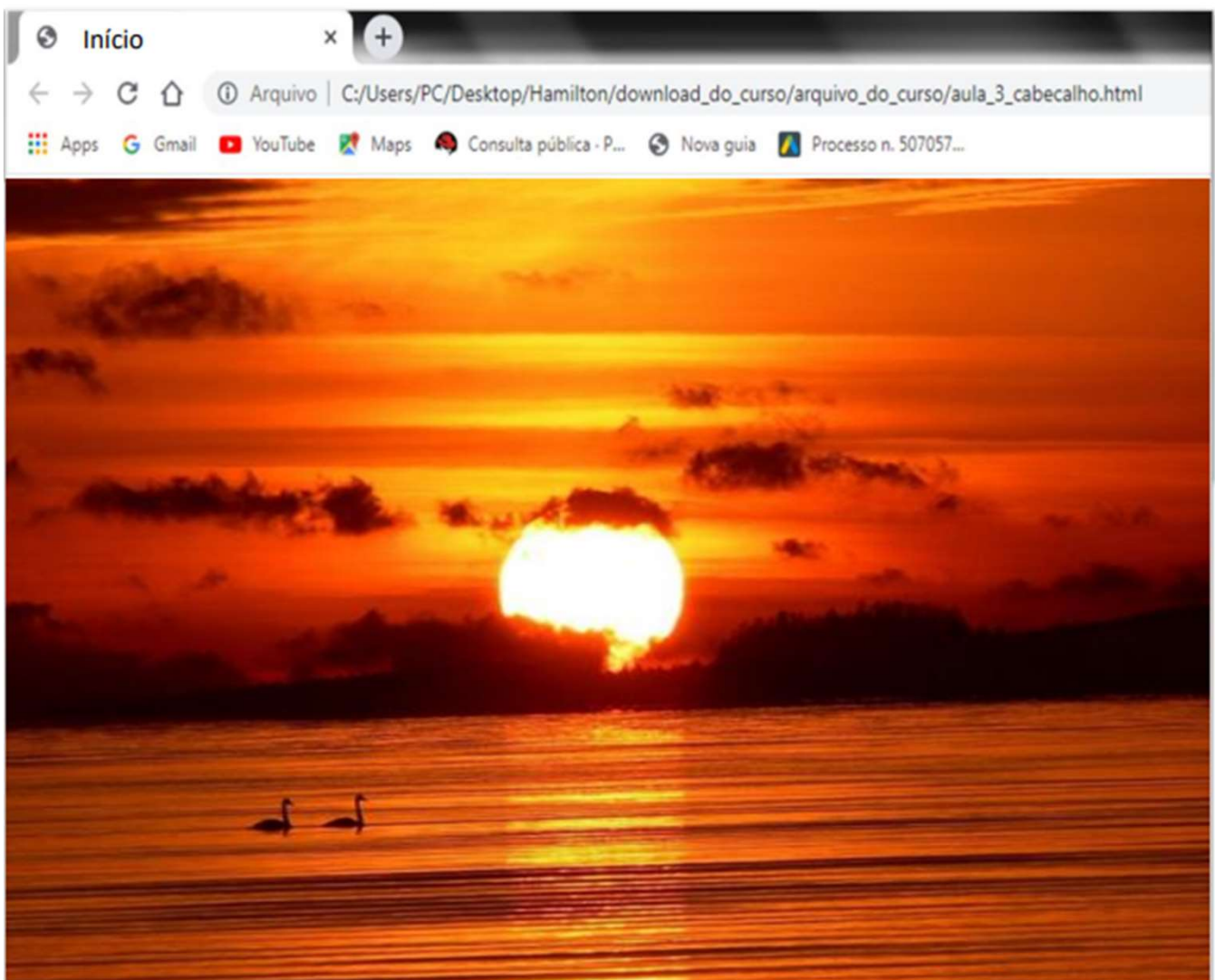
HTML BÁSICO



Exemplo:

```
<!doctype html>  
<html lang = "pt-br">  
<head>  
<meta charset = "UTF-8">  
<title>Início</title>  
</head>  
<body background = img/fundo.jpg>  
</body>  
</html>
```

Salve a pagina como fundo-1.html e veja como fica:



HTML BÁSICO



- **BGCOLOR** - Define a cor que será utilizada no fundo da página. Você pode usar as cores pelo: Nome, Código Hexadecimal ou Código RGB:

Exemplo:

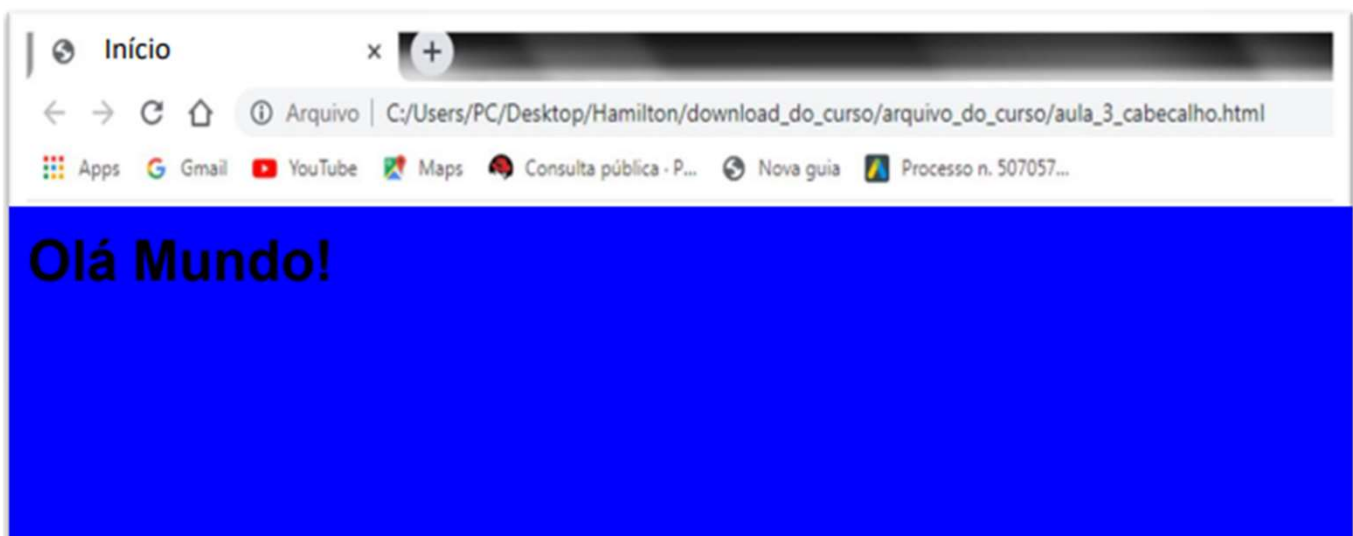
- Nome: Blue
- Código Hexadecimal: #000080
- Código RGB: rgb(0,0,255)

Vamos testar na prática.

Exemplo 1:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body bgcolor=blue>
<h1>Olá Mundo</h1>
</body>
</html>
```

Salve a página como: fundo-2.html e veja como fica:



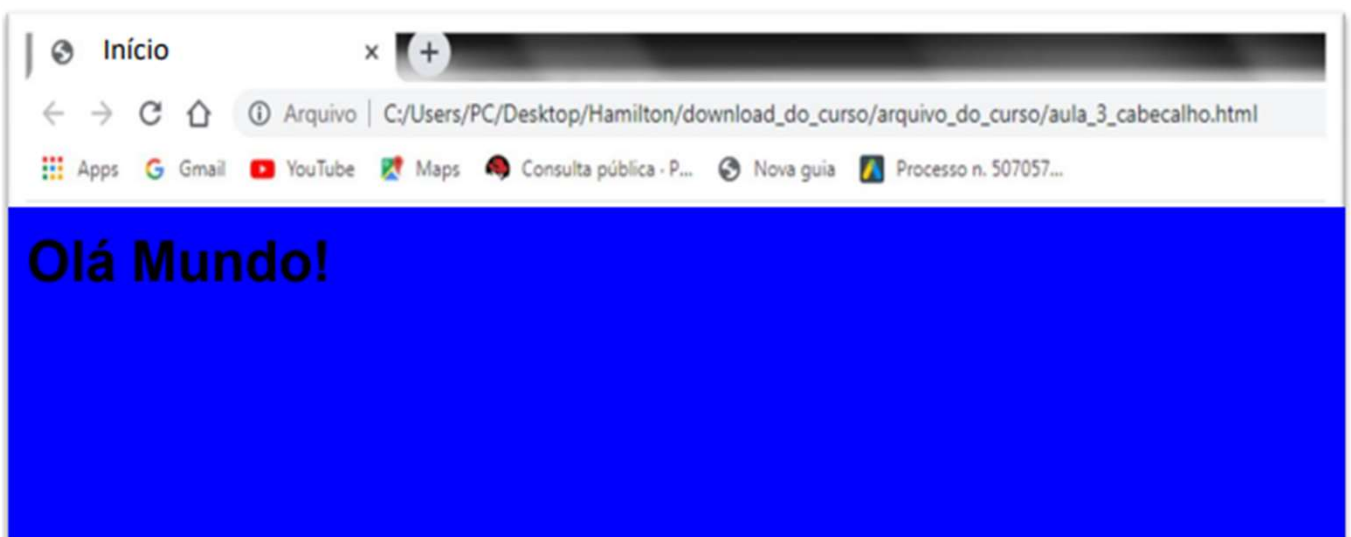
HTML BÁSICO



Exemplo 2:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body bgcolor=#000080>
<h1>Olá Mundo</h1>
</body>
</html>
```

Salve a página como: fundo-3.html e veja como fica:



Agora é com você! Faça este ultimo exemplo utilizando o código RGB e depois salve o documento como fundo-4.html para ver como fica:

Lembre-se: Quanto mais você prática, melhor você fica!

Tabela das cores

Veja os outros exemplos segundo a tabela das cores abaixo:

DimGray	#696969	(105,105,105)
Gray	#808080	(128,128,128)
DarkGray	#A9A9A9	(169,169,169)
Silver	#C0C0C0	(192,192,192)
LightGrey	#D3D3D3	(211,211,211)
Gainsboro	#DCDCDC	(220,220,220)
White	#FFFFFF	(255,255,255)

Tons de Laranja

OrangeRed	#FF4500	(255,69,0)
DarkOrange	#FF8C00	(255,140,0)
Orange	#FFA500	(255,165,0)

Tons de azul

Navy	#000080	(0,0,128)
DarkBlue	#00008B	(0,0,139)
MediumBlue	#0000CD	(0,0,205)
Blue	#0000FF	(0,0,255)
CornflowerBlue	#6495ED	(100,149,237)
RoyalBlue	#4169E1	(65,105,225)
DodgerBlue	#1E90FF	(30,144,255)
SlateBlue	#6A5ACD	(106,90,205)
DarkSlateBlue	#483D8B	(72,61,139)
MidnightBlue	#191970	(25,25,112)

Tons de Ciano

Aqua / Cyan	#00FFFF	(0,255,255)
DarkTurquoise	#00CED1	(0,206,209)
Turquoise	#40E0D0	(64,224,208)
MediumTurquoise	#48D1CC	(72,209,204)
LightSeaGreen	#20B2AA	(32,178,170)
DarkCyan	#008B8B	(0,139,139)
Teal	#008B8B	(0,128,128)
Aquamarine	#7FFFD4	(127,255,212)
MediumAquamarine	#66CDAA	(102,205,170)
CadetBlue	#5F9EA0	(95,158,160)

Tons de verde

DarkSlateGray	#2F4F4F	(47,79,79)
MediumSpringGreen	#00FA9A	(0,250,154)
SpringGreen	#00FF7F	(0,255,127)
PaleGreen	#98FB98	(152,251,152)
LightGreen	#90EE90	(144,238,144)
DarkSeaGreen	#8FBC8F	(143,188,143)
MediumSeaGreen	#3CB371	(60,179,113)
SeaGreen	#2E8B57	(46,139,87)
DarkGreen	#006400	(0,100,0)
Green	#008000	(0,128,0)
ForestGreen	#228B22	(34,139,34)

Tons de Amarelo

Gold	#FFD700	(255,215,0)
Yellow	#FFFF00	(255,255,0)
Khaki	#F0E68C	(240,230,140)

Tons de Roxo

MediumPurple	#9370DB	(147,112,219)
BlueViolet	#8A2BE2	(138,43,226)
Indigo	#4B0082	(75,0,130)
DarkViolet	#9400D3	(148,0,211)
DarkOrchid	#9932CC	(153,50,204)
MediumOrchid	#BA55D3	(186,85,211)
Purple	#800080	(128,0,128)
DarkMagenta	#8B008B	(139,0,139)
Fuchsia / Magenta	#FF00FF	(255,0,255)
Violet	#EE82EE	(238,130,238)
Orchid	#DA70D6	(218,112,214)
Plum	#DDA0DD	(221,160,221)

Tons de Rosa

MediumVioletRed	#C71585	(199,21,133)
DeepPink	#FF1493	(255,20,147)
HotPink	#FF69B4	(255,105,180)
PaleVioletRed	#DB7093	(219,112,147)
LightPink	#FFB6C1	(255,182,193)
Pink	#FFC0CB	(255,192,203)
LightCoral	#F08080	(240,128,128)
IndianRed	#CD5C5C	(205,92,92)
Crimson	#DC143C	(220,20,60)

Tons de Verme

Brown	#A52A2A	(165,42,42)
Salmon	#FA8072	(250,128,114)
Maroon	#800000	(128,0,0)
Tomato	#FF6347	(255,99,71)
Red	#FF0000	(255,0,0)

Formatação de Texto – Parte II

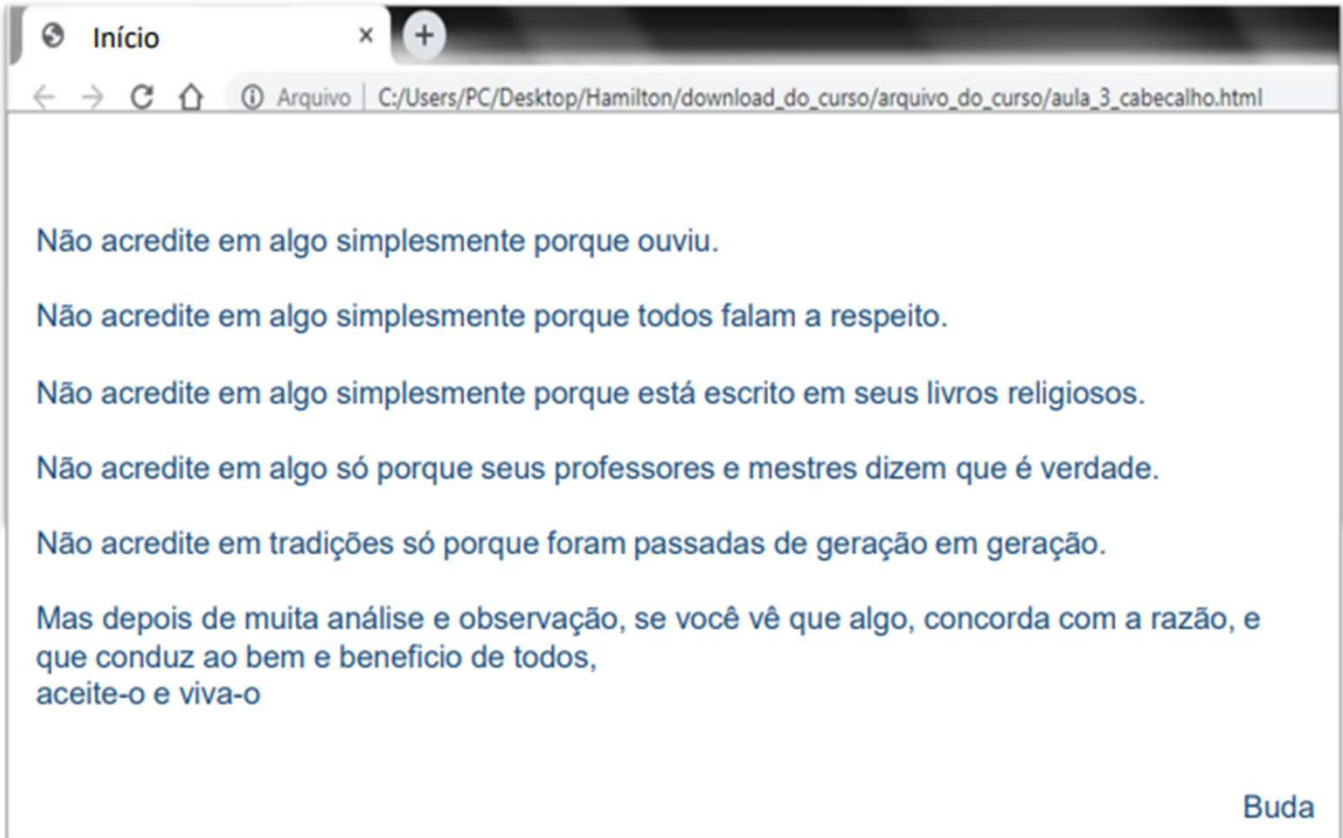
- Cor do texto

➤ Define a cor simples do texto da página.

Exemplo:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body text=blue>
<p>Não acredite em algo simplesmente porque ouviu. <p>
<p>Não acredite em algo simplesmente porque todos falam a respeito. </P>
<p>Não acredite em algo simplesmente porque está escrito em seus livros
religiosos.</p>
<p>Não acredite em algo só porque seus professores e mestres dizem que é
verdade.</p>
<p>Não acredite em tradições só porque
foram passadas de geração em geração.</p>
<p>Mas depois de muita análise e observação, se você vê que algo
concorda com a razão, e que conduz ao bem e benefício de todos,
aceite-o e viva-o.</p>
<p align=right> Buda</p>
</body>
</html>
```


Salve o documento e veja o resultado:



● **Valor do espaço** - Define a quantidade de espaço na margem do topo que terá a página.

➤ **Leftmargin - Exemplo:**

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
```

```
<body leftmargin=50px text=red>
```

Não acredite em algo simplesmente porque ouviu. Não acredite em algo simplesmente porque todos falam a respeito.

```
</body>
</html>
```


HTML BÁSICO

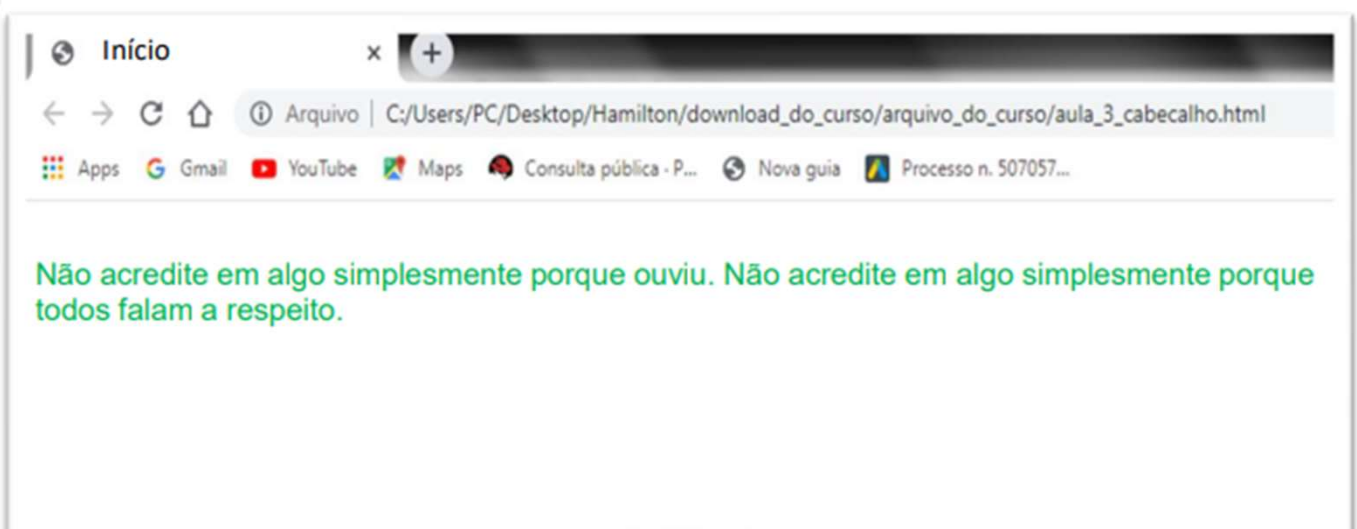


Salve o documento e veja o resultado:



➤ **topmargin - Exemplo:**

```
<!doctype html>  
<html lang = "pt-br">  
<head>  
<meta charset = "UTF-8">  
<title>Início</title>  
</head>  
<body topmargin=50px text=green>  
Não acredite em algo simplesmente porque ouviu. Não acredite em algo simplesmente  
porque todos falam a respeito.  
</body>  
</html>
```



Links

Link é uma palavra em inglês que significa elo, vínculo ou ligação.

No âmbito da informática, a palavra link pode significar hiperligação, ou seja, uma palavra, texto ou imagem que quando é clicada pelo usuário, o encaminha para outra página na internet, que pode conter outros textos ou imagens. Os Links servem para criar palavras quentes, que permitem a interligação entre documentos HTML e outros documentos ou até arquivos FTP.

Como criar um link em HTML

Para criar um link vamos usar mais uma tag, chamada anchor (âncora) tag: `<a>`. Porém, você não verá seu uso sozinho, sem atributos. Como dissemos previamente, o uso dos links são inúmeros, bem como a quantidade de atributos dessa tag. Por hora, nesse artigo de HTML, vamos aprender a maneira mais simples e usar de criar um link, com o atributo href.

A sintaxe é: ` Texto visível `

Como exemplo, vamos criar o código de um link para o site o Site de compras Mercado Livre. Como de costume, digite a estrutura básica novamente acrescente no body o seguinte atributo:

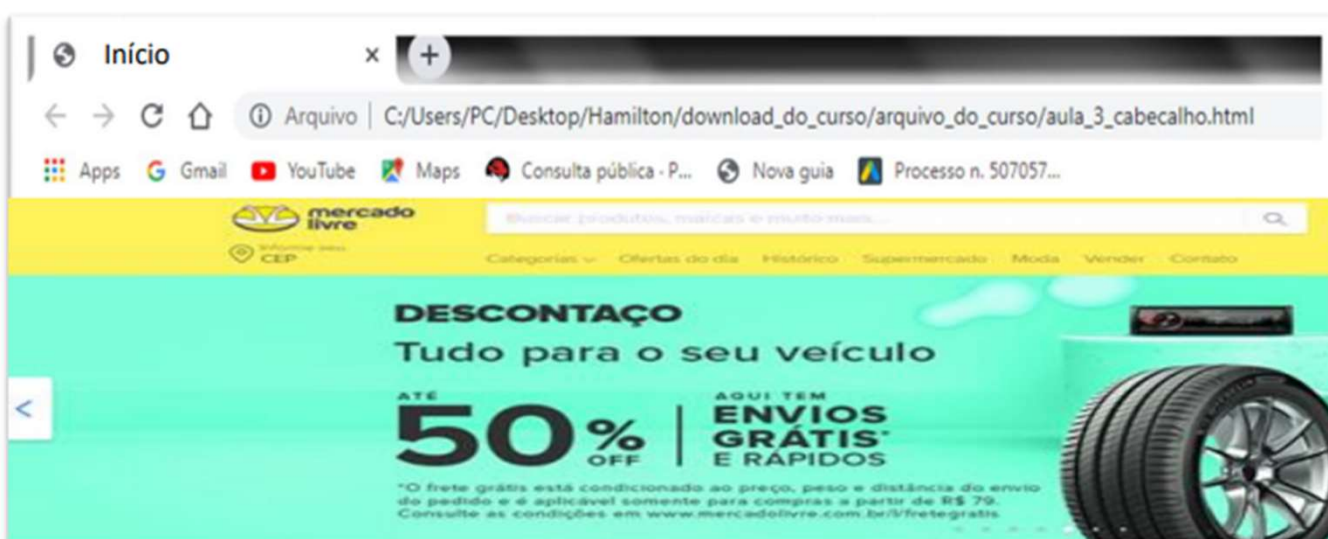
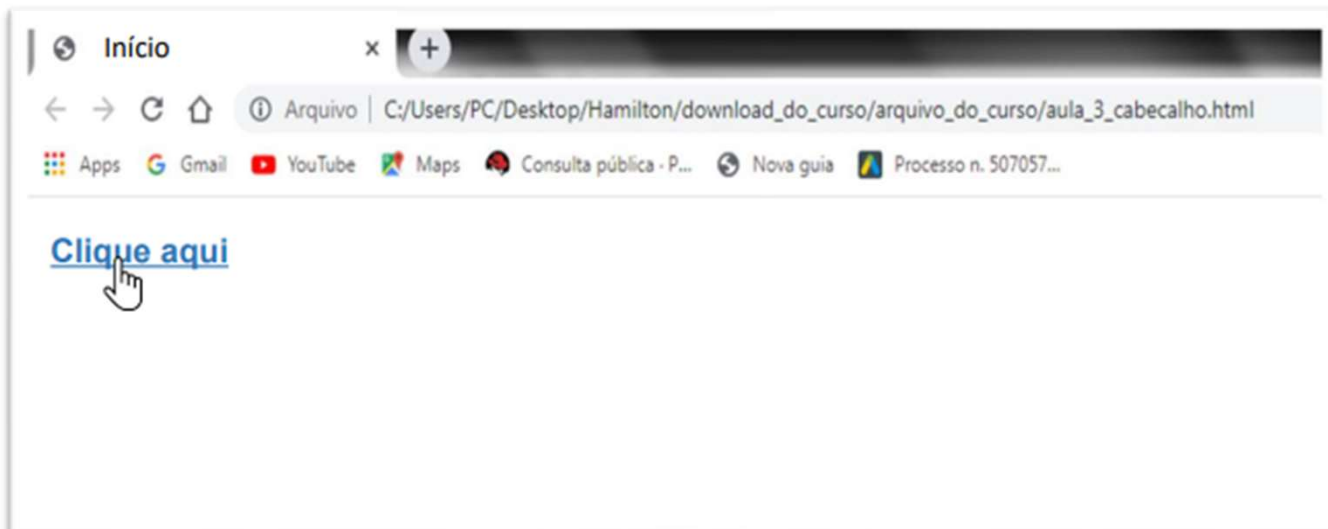
```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body>
<a href=http://www.mercadolivre.com.br/>clique aqui</a>
</body>
</html>
```

E o que veríamos, na prática, ao clicar no Mercado Livre, seria simplesmente a página do Mercado livre aberta no seu Navegador/Browser.

HTML BÁSICO



Salve o documento e veja o resultado:



E isso é bem comum em textos do tipo: "Se quiser acessar um site de compras online, [Clique aqui](#)". Ou seja, não importa o que você escreve entre as tags `<a>` ``, é apenas a parte visível ao usuário.

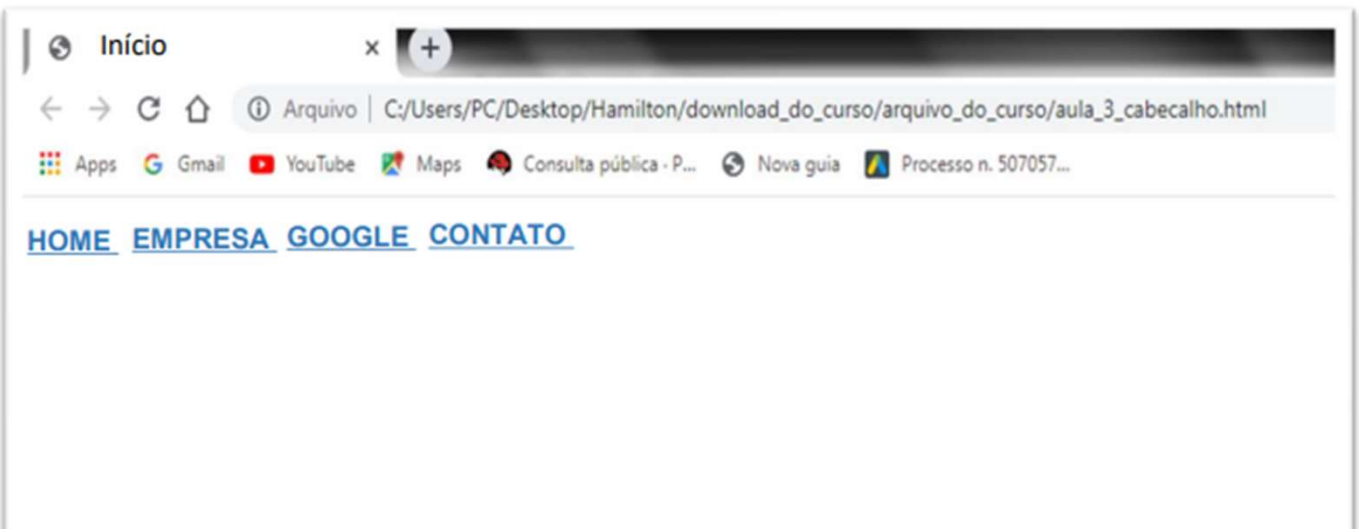
Os menus de navegação são links que possibilitam levar o usuário de uma página a outra dentro do site. Eles possuem papel importante na estrutura de um site, não só porque permitem essa mobilidade entre os conteúdos mas, também, funcionam como uma referência da estrutura para esse conjunto de páginas eletrônicas.

HTML BÁSICO



Exemplo de um Menu:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body>
<a href=index.html>HOME</a>
<a href=empresa.html>EMPRESA</a>
<a href=http://www.google.com.br>GOOGLE</a>
<a href=contato.html>CONTATO</a>
</body>
</html>
```



Ao clicar nos links que acabou de criar você deve ter percebido que somente o link do **GOOGLE** foi carregado corretamente. Isso ocorreu porque ainda não criamos as páginas para ser redirecionados aos seus retrospectivos links. É exatamente isso que faremos no próximo tópico.

Criando Páginas

O processo para criar páginas é muito simples, basta salvar o documento com a extensão **.html** como vimos na aula salvando um documento HTML (**pág. 5**). Lembra que comentamos também que a primeira página por padrão a nomeamos de **index.html** e as outras com qualquer nome? Se a resposta for sim, parabéns, você está tendo um ótimo desempenho no curso!

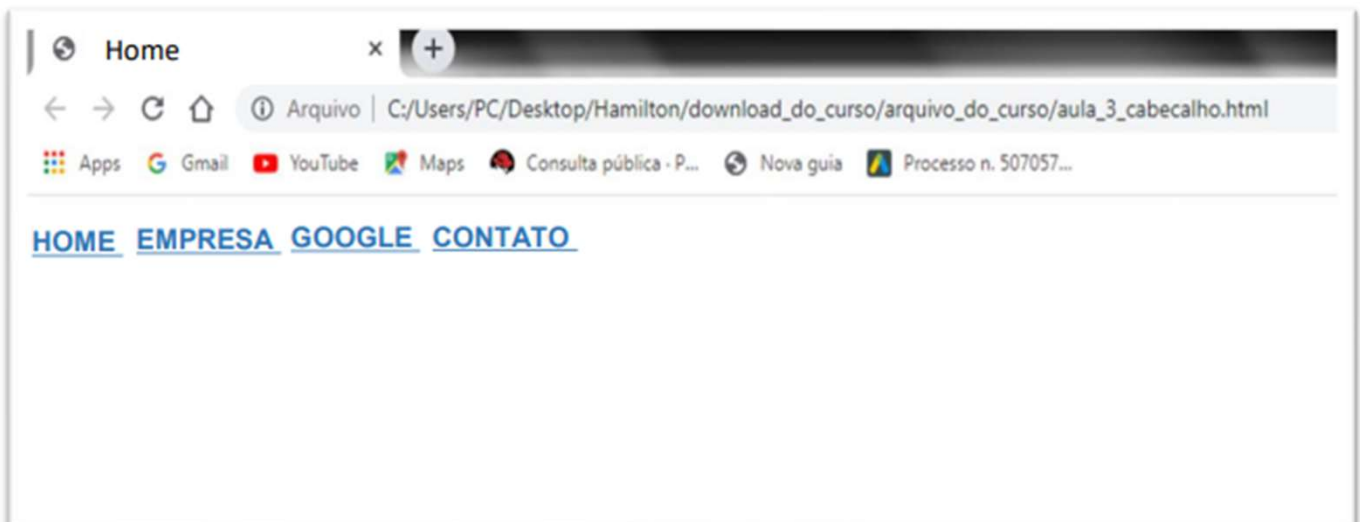
Para carregar os outros links, é muito simples: Com o documento **index.html** aberto basta salvar a página com o nome dos **links** que acabamos de criar.

Exemplo:

Na página **index.html** salvamos como: **empresa.html** e na página **empresa.html** salvamos como: **contato.html**

Agora é com você: Salve o documento **index.html** com o nome dos **links** que acabamos de criar no menu.

Nota: A cada página criada, lembre sempre de colocar o título referente a cada item do menu.



Atributo Target

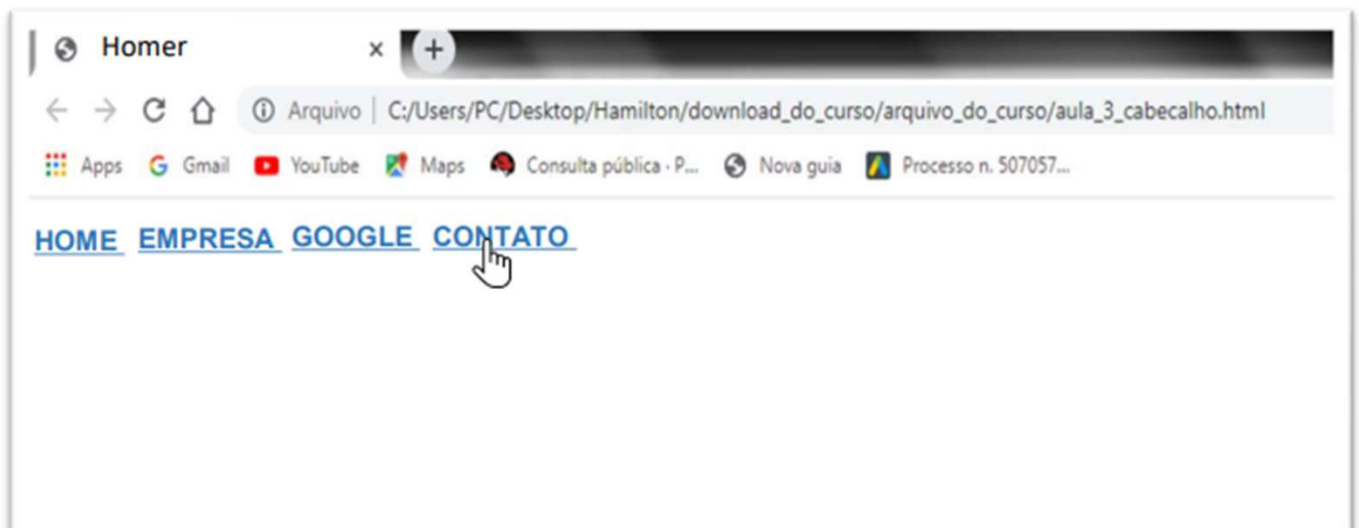
A função do atributo target basicamente é indicar o nome de um frame na página onde um documento deve ser aberto. Outro controle que você pode ter com ele é se o hyperlink deve ser aberto na própria janela ou no próprio frame.

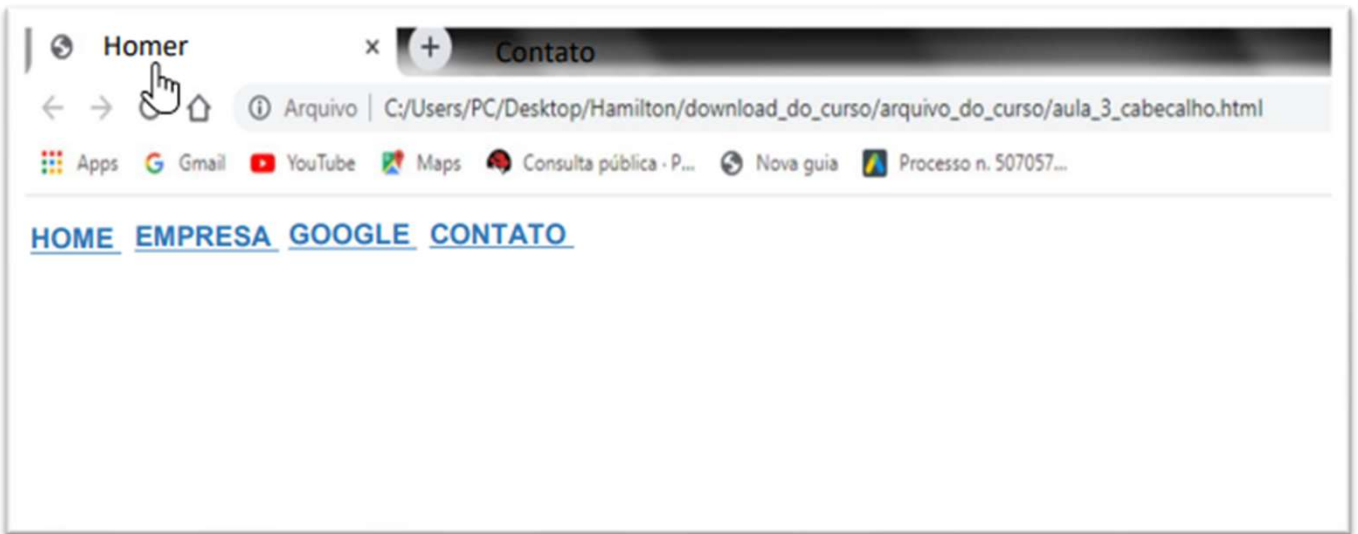
Os atributos target são:

- **_blank**: abre a página em uma nova janela/aba;
- **_self**: abre a página na mesma janela;
- **_parent**: abre a página na mesma janela do link;
- **_top**: cancela todos os demais frames e abre a nova página no mesmo navegador.

Veja o exemplo do menu com o atributo target:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<a href=index.html target=“_parent”> HOME</a>
<a href=empresa.html target=“_top”>EMPRESA</a>
<a href=http://ww.google.com.br target=“_self”>GOOGLE</a>
<a href=contato.html target=“_blank”>CONTATO</a>
</body>
</html>
```





Listas

A linguagem aceita diversos tipos de listas formatadas para apresentação dos dados. Todas as listas exigem um par de tags que indica o tipo de lista, além de uma tag no início de cada item da lista.

As mais importantes são as seguintes: Listas ordenadas, Listas não ordenadas e Listas de definição.

➤ Listas ordenadas:

As listas ordenadas são usadas para listar itens por sequência. Chamamos de lista numeradas apenas. As tags usadas para criar uma lista ordenada são `` e ``. A tag `` é usada para definir a lista e a tag `` é usada para cada item da lista.

HTML BÁSICO

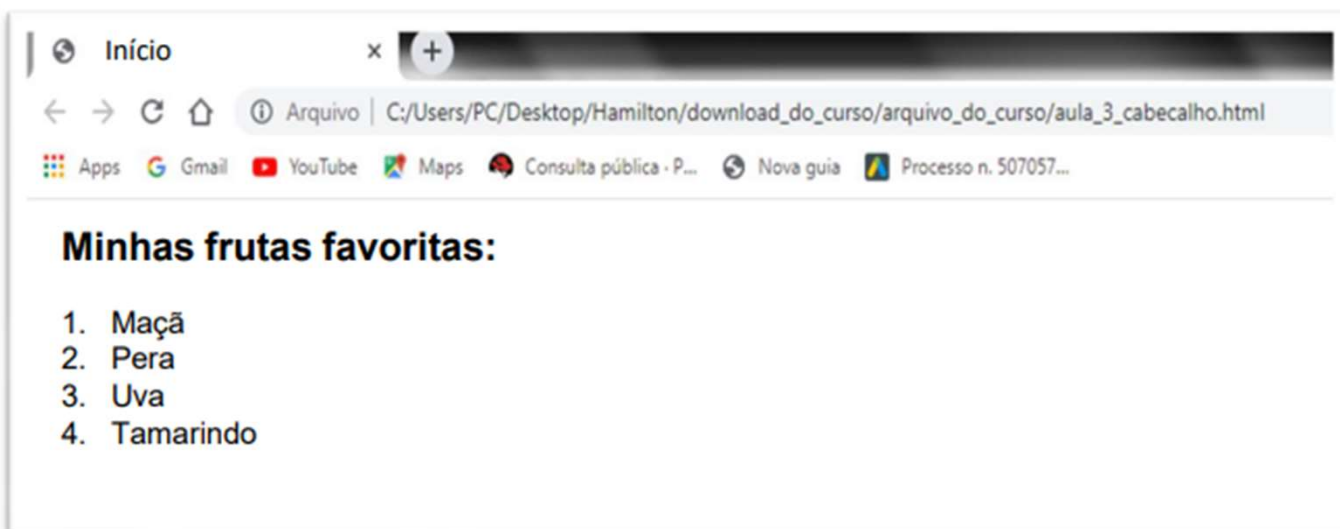


Vamos testar na pratica:

Insira no documento **HTML**, inicialmente uma lista ordenada com as suas frutas favoritas.

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<h3>Minhas frutas <u>favoritas</u></h3>
<ol>
<li>Maçã</li>
<li>Pêra</li>
<li>Uva</li>
<li>Tamarindo</li>
</ol>
</body>
</html>
```

Salve e veja e resultado:



➤ Listas não ordenadas:

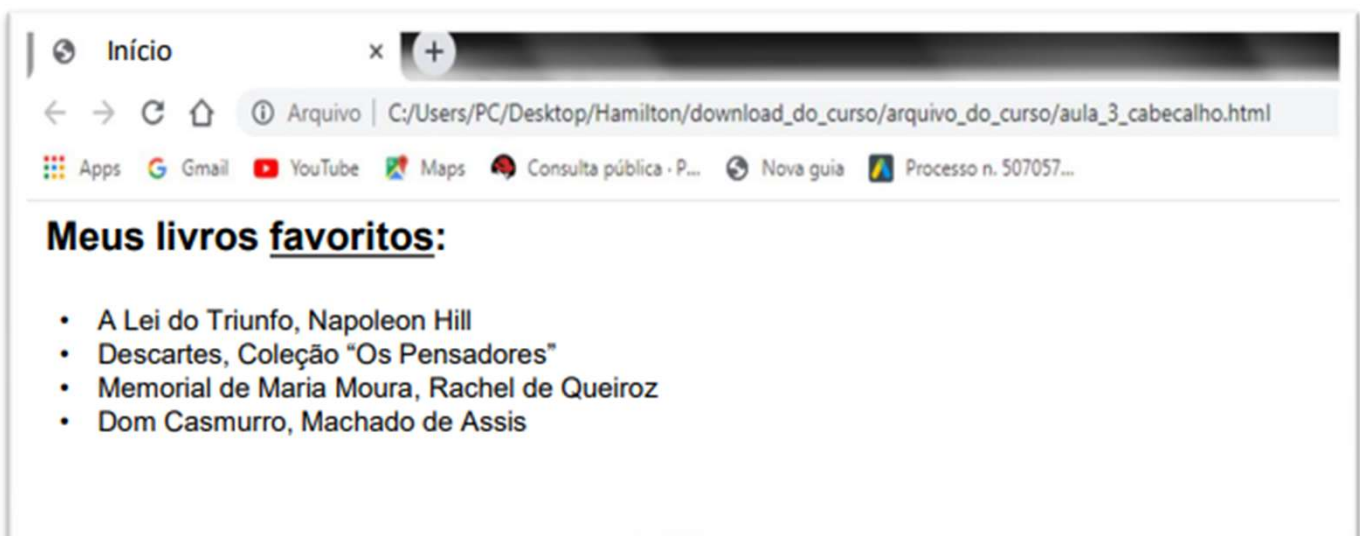
As listas não ordenadas são usadas para listar itens, sem se preocupar com sua sequência. Chamamos de lista de marcadores apenas.

As tags usadas para criar uma lista não ordenada são `` e ``. A tag `` é usada para definir a lista e a tag `` é usada para cada item da lista.

Agora, vamos formatar uma lista não ordenada e comparar as duas.

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<h3>Meus livros <u>favoritos</u></h3>
<ul>
<li>A Lei do Triunfo, Napoleon Hill</li>
<li>Descartes, coleção “Os Pensadores”</li>
<li>Memorial de Maria Moura, Rachel de Queiroz</li>
<li>Dom Casmurro, Machado de Assis</li>
</ul>
</body>
</html>
```

Salve e veja o resultado:



➤ Listas de Definição

Listas de definição são usadas em assuntos onde há um termo a em sequência a sua definição, estilo perguntas e respostas. As tags usadas são `<dl>`, `<dt>` e `<dd>`.

- `<dl>` **definition list** - cria a lista de definição
- `<dt>` **definition term** - onde o termo é proposto
- `<dd>` **definition definition** - onde o termo é definido

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<dl>
<dt>HTML</dt>
<dd>Linguagem de marcação de texto</dd>
<dt>CSS</dt>
<dd>Usado para formatação de documentos</dd>
</dl>
</body>
</html>
```

Salve e veja o resultado:

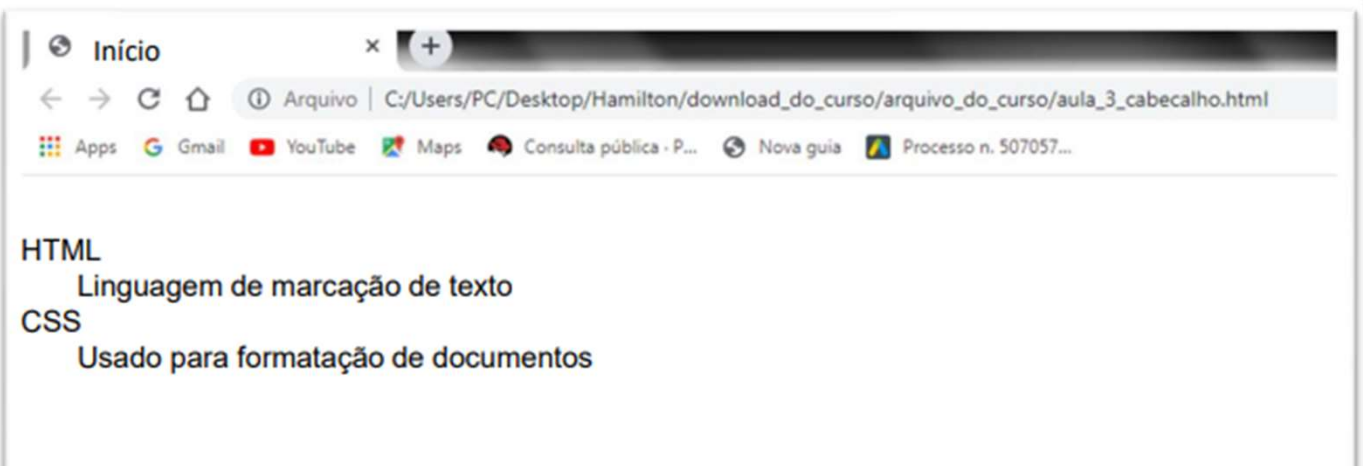


Imagem na Página

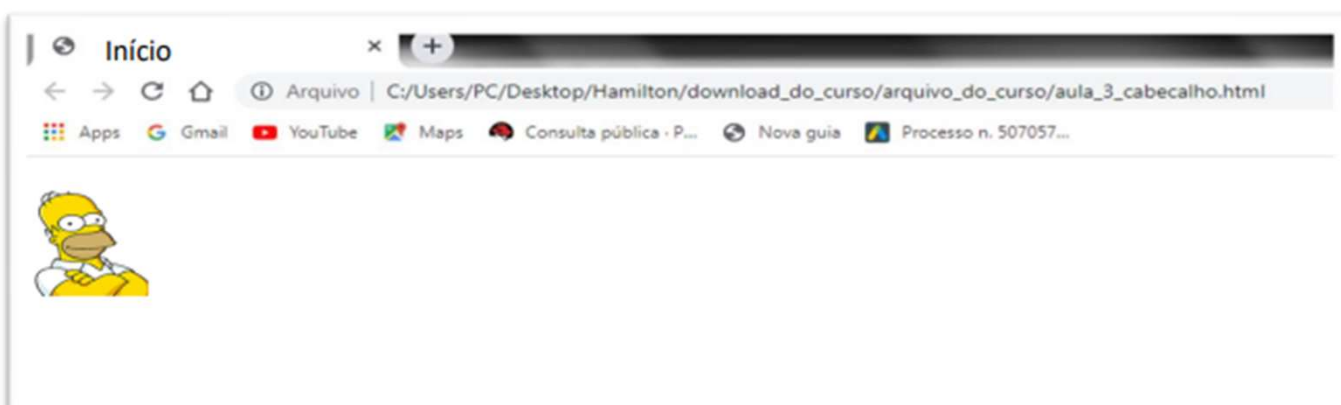
- Inserção da imagem

Para inserir uma imagem na página, utiliza-se a seguinte notação: `` "src" é a forma curta para "source" (fonte). Esse atributo é usado para indicar a localização da imagem. Como explicado no tutorial sobre links você pode usar qualquer **URL** para mostrar um arquivo. Para visualizar a imagem na pagina devemos salvar primeiro a imagem no diretório atribuído pelo source (**src**).

Abra o bloco de notas e digite essa estrutura abaixo:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body >
<img src=img/homer.png/>
</body>
</html>
```

Feito isso, renomeio o arquivo como index.html e salve.
Abra o arquivo index.html e veja como fica:



Nota: É importantante lembrar sempre que salvar uma **imagem** certifique-se que o arquivo esteja **dentro** da pasta **img** e o arquivo **index.html** **fora** da pasta **img**.

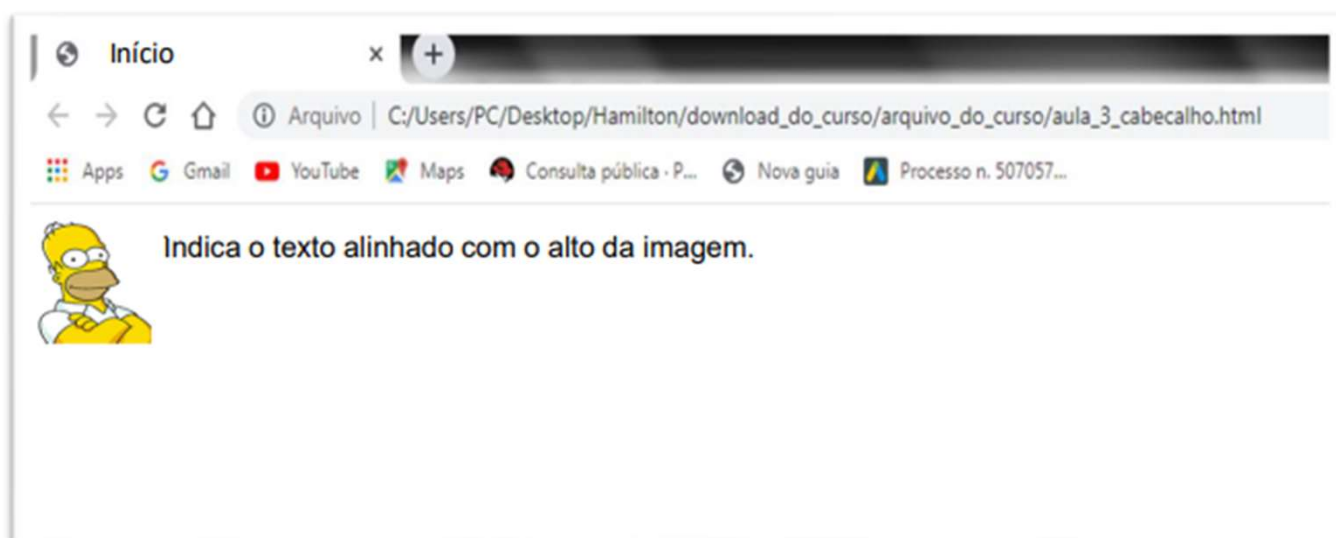
- **Alinhamento da imagem**

Toda imagem inserido pode ser alinhada através do atributo **ALIGN**:
Nestes três exemplos a posição da imagem está definida com relação ao texto:

Exemplo 1:

```
<!doctype html>  
<html lang =”pt-br”>  
<head>  
<meta charset =”UTF-8”>  
<title>Início</title>  
</head>  
<body >  
<img align=top src=img/homer.png />  
Indica o texto alinhado com o alto da imagem.  
</body>  
</html>  
</html>
```

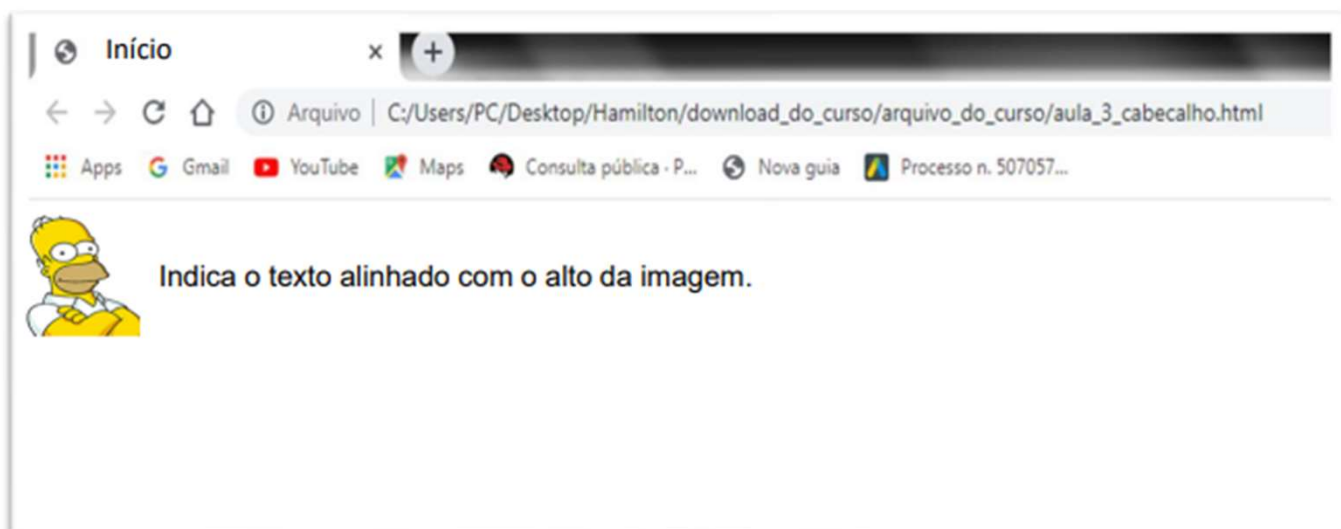
Para visualizar a pagina o processo é o mesmo, só que agora renomeio o arquivo como exemplo1.html e abra a página para ver como fica:



Exemplo 2:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<img align=middle src=img/homer.png />
Indica o texto alinhado com o meio da imagem.
</body>
</html>
```

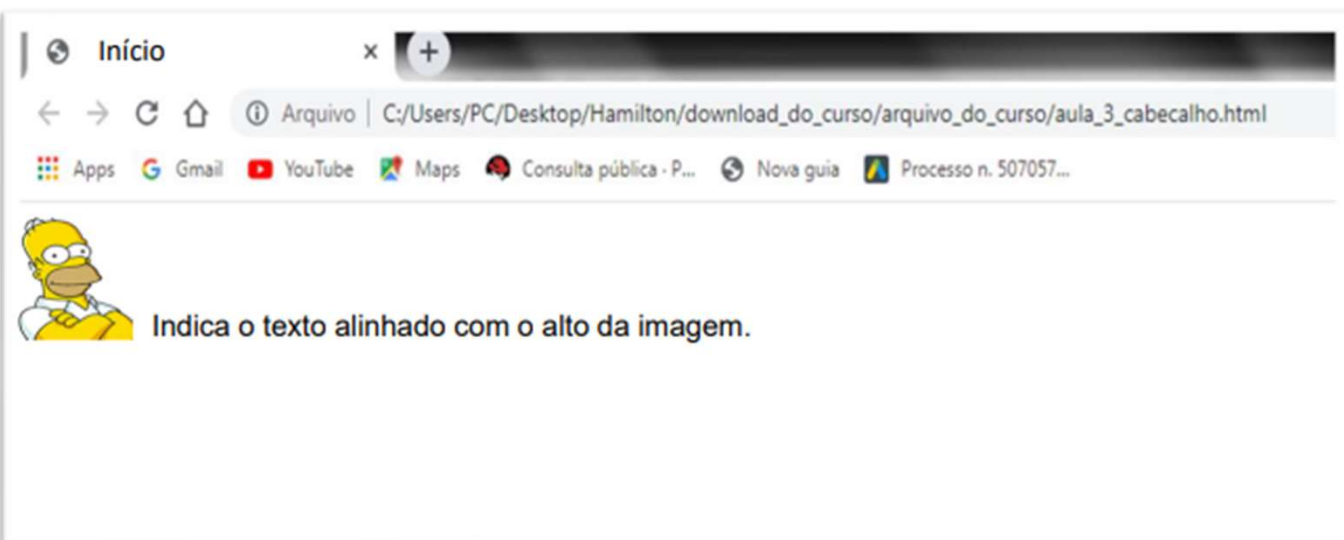
Para visualizar a pagina o processo aqui também é o mesmo, salve o arquivo, renomeie como exemplo-2.html e abra a página para ver como fica:



Exemplo 3:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<img align=botton src=img/homer.png />
Indica o texto alinhado com o rodapé da imagem.
</body>
</html>
```

Salve e veja o resultado: (lembre-se que o processo é mesmo, renomeio o arquivo como exemplo-3.html).



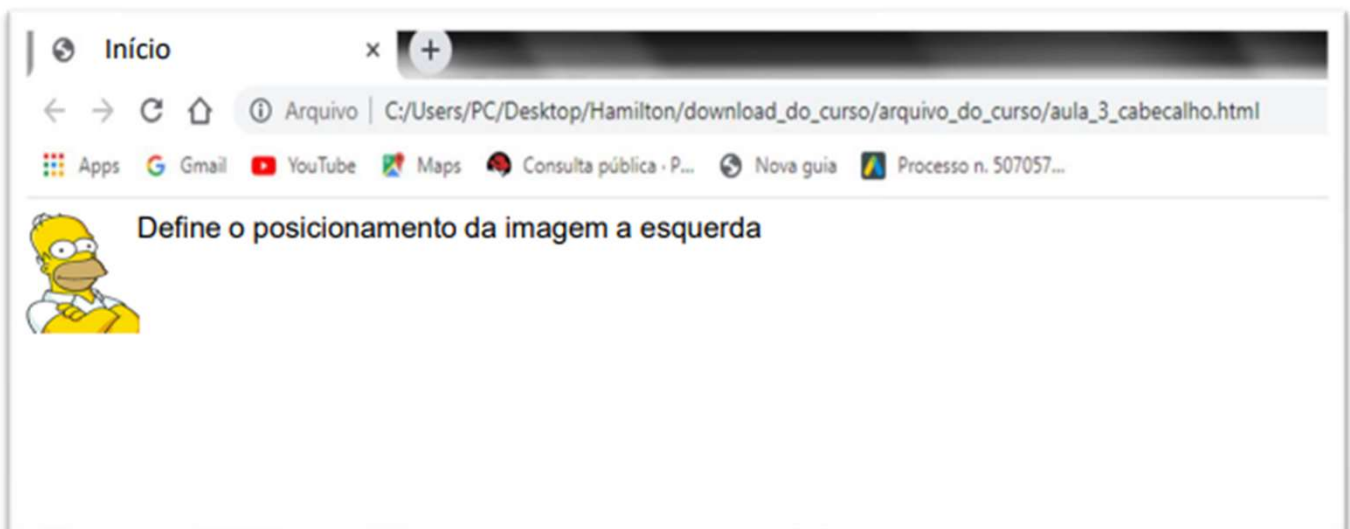
- **Posicionamento da imagem**

Neste dois exemplos define a posição da imagem na página:

Exemplo 1: a esquerda

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<img align=left src=img/homer.png />
Define o posicionamento da imagem a esquerda
</body>
</html>
```

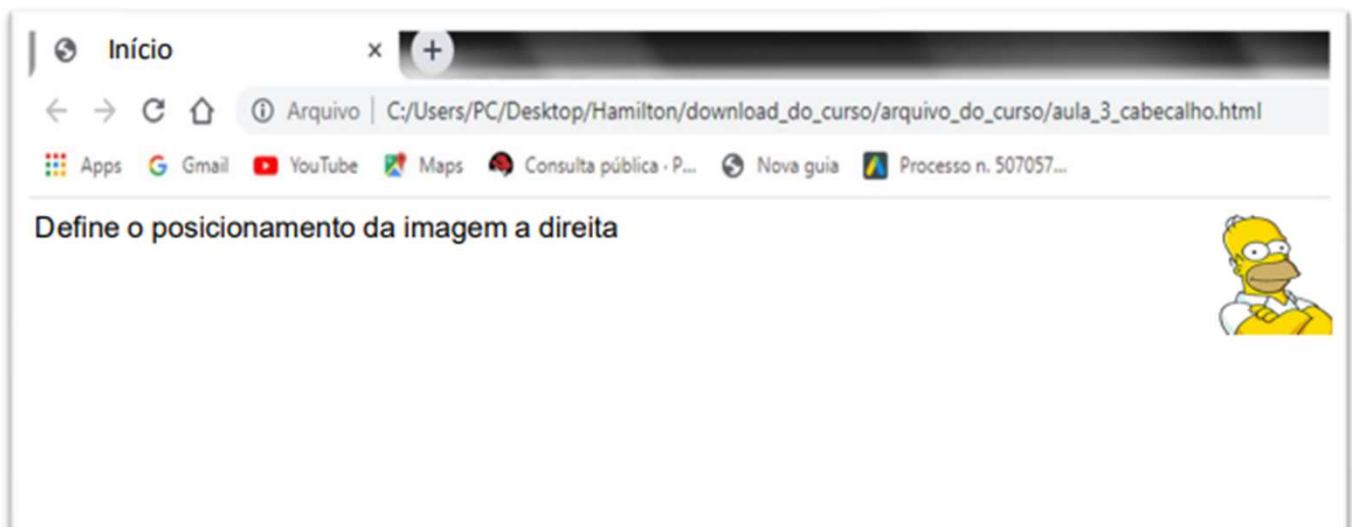
Salve e veja o resultado: (Salve o arquivo como: exemplo-4.html lembrando que o processo é mesmo, faça isso nas demais aulas.



Exemplo 2: a direita

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body >
<img align=right src=img/homer.png />
Define o posicionamento da imagem a direita
</body>
</html>
```

Salve e veja o resultado: (Salve o arquivo como: exemplo-5.html).



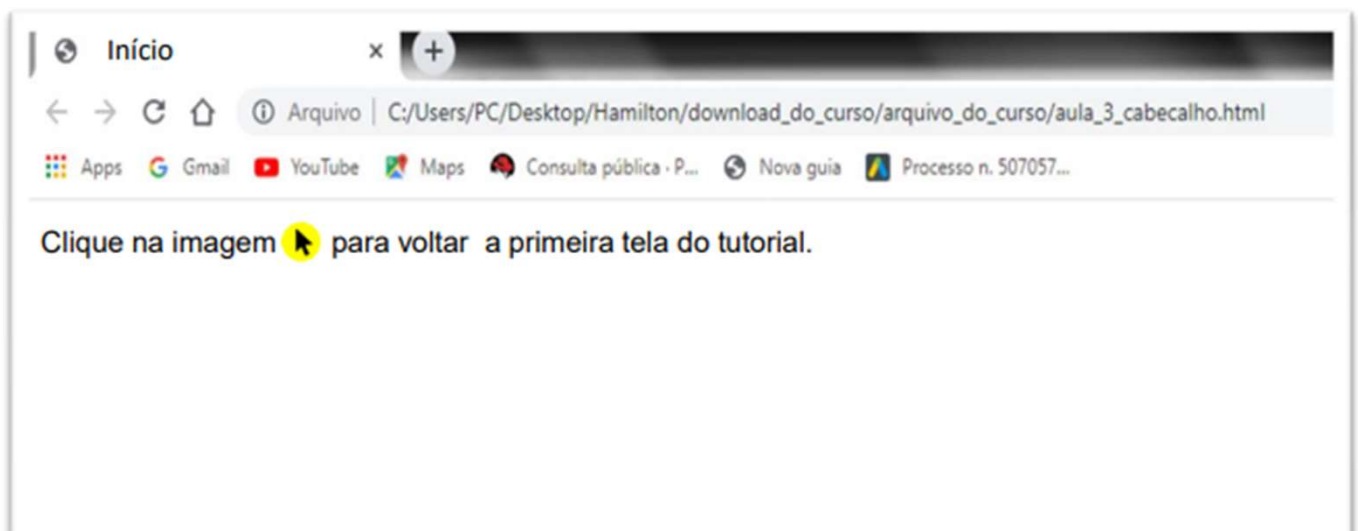
- Atribuindo imagem a botão

Uma imagem ainda pode ser atribuída como um botão. Para isso basta incluir uma imagem no interior de uma âncora.

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
Clique na imagem<a href=index.html target=“blank”>
<img src=img/pointer.png></a>para voltar a primeira tela do tutorial.
</body>
</html>
```

Salve e veja o resultado: (Salve o arquivo como: exemplo-6.html).



HTML BÁSICO



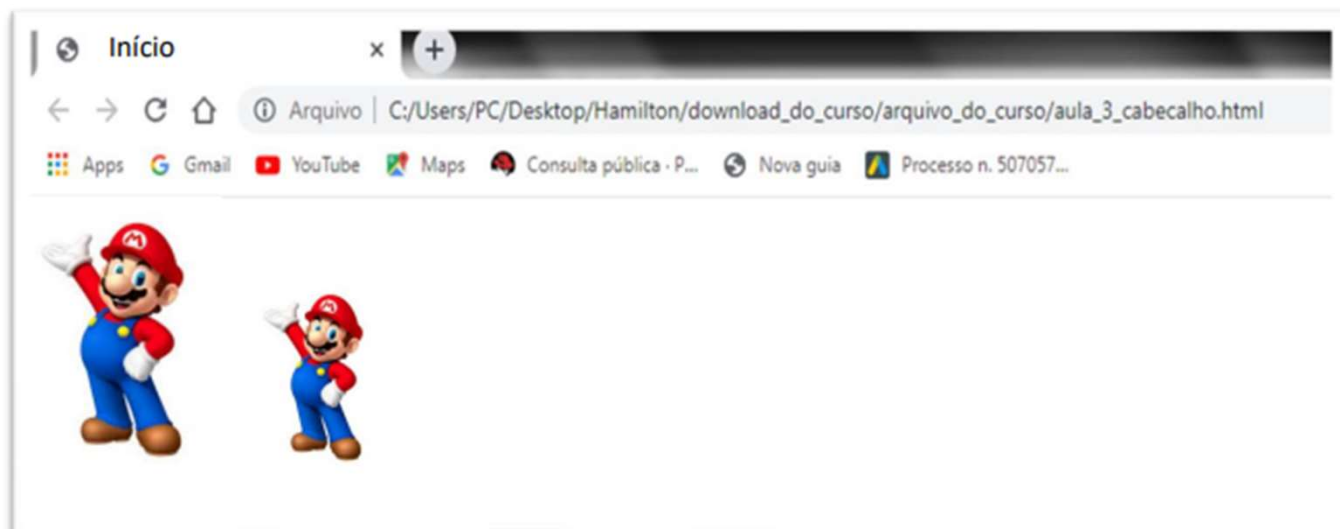
- Tamanho da imagem

Você pode também definir um largura e uma altura para imagem usando os parâmetros: **Width** para largura e **Height** para altura.

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<img src=img/img-atual.png />
<img src=img/img-definida.png width=200px height=200px/>
</body>
</html>
```

Salve e veja o resultado: (Salve o arquivo como: exemplo-7.html).



Tabelas

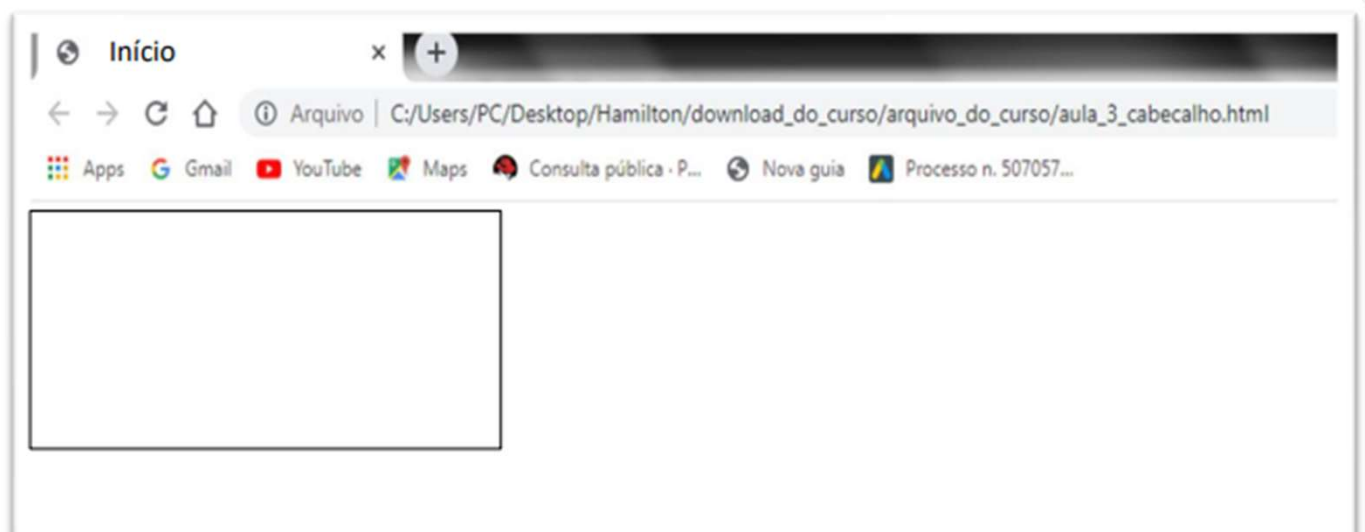
Segue abaixo o conjunto de marcação essenciais para desenhar tabelas em HTML. Toda tabela deve ser iniciada com a tag `<table>` e encerrada com a tag `</table>`. Dessa forma a tabela vai aparecer sem borda, mas dividindo linhas e células. para que tabela apareça com bordas basta definir dessa forma:

```
<table border=1 width=600px height=300px></table>
```

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<table border=1 width=600px height=300px>
</table>
</body>
</html>
```

Salve e veja o resultado:



• Componentes da Tabela

• `<tr></tr>` - Este demarca cada linha de uma tabela deve sempre aparecer entre as marcações `<tr>` e `</tr>`. (TR = Table Row).

• `<td></td>` - Esta é a marcação que define a célula de uma tabela. As células de uma tabela devem sempre aparecer entre as marcações de linhas `<tr>` e `</tr>`. Como padrão o texto nas células é alinhado a esquerda.

• `<th></th>` - Desta forma são definidos os títulos de uma tabela. Estes podem ser pensionados em qualquer célula. A diferença entre marcação de célula e título de célula é que o título aparece em negrito. Vamos ver então o exemplo de uma tabela utilizando essas marcações básicas:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<table border>
<tr>
<th>Itens/Mês</th> <th>Janeiro</th><th>Fevereiro</th><th>Março</th>
</tr>
<tr>
<td>Usuarios</td>
<th>10</th><th>20</th><th>50</th>
</tr>
<tr>
<td>Linhas</td>
<th>100</th><th>40</th><th>95</th>
</tr>
</table>
</body>
</html>
```

HTML BÁSICO



Salve e veja o resultado:



As marcas das tabelas podem apresentar resultados diferentes, se acompanhadas de alguns atributos.

Vamos ver os principais:

- **Border** - Esse atributo aparece junto com a marcação table. Caso esse atributo não apareça, a tabela não terá bordas.
- **Align** - Esse atributo pode ser aplicado à tr, th e td. Controla como será o alinhamento do texto dentro de uma célula, com relação as bordas laterais. Aceita os valores left, center, right, respectivamente para alinhar a esquerda, centralizar ou alinhar a direita.

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body>
<table border>
<tr>
<td>Primeira célula</td>
<td align=center>Centro</td>
<td align=left >Esquerda</td>
<td align=right >Direita</td>
</tr>
</table>
</body>
</html>
```

Salve e veja o resultado:



- Valign - Pode ser aplicado a TD, TR, TH, e define o alinhamento do texto nas células com relação a borda superior e inferior.

Aceita valores top, middle e bottom.

Veja o exemplo:

```
<!doctype html>  
<html lang = "pt-br" >  
<head>  
<meta charset = "UTF-8" >  
<title>Início</title>  
</head>  
<body>  
<table border >  
<tr >  
<td >  
Teste para alinhamentos <br >  
com relação a bordas <br >  
inferior e superior <br >  
</td >  
<td valign=top>TOP</td >  
<td valign=middle>MIDDLE</td >  
<td valign=bottom>BOTTOM</td >  
</tr >  
</table >  
</body >  
</html >
```

HTML BÁSICO



Salve e veja o resultado:



- Colspan Pode ser aplicado a TH ou TD. Define quantas colunas uma célula poderá abranger. Por padrão cada célula a uma coluna, ou seja colspan=1.

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body >
<table border>
<tr>
<td colspan=3>3 Colunas</td><td>Normal</td><td>Normal</td>
</tr>
<tr>
<td>Coluna 1</td>
<td>Coluna 2</td>
<td>Coluna 3 </td>
<td>Coluna 4</td>
<td>Coluna 5</td>
</tr>
</table>
</body>
</html>
```

Salve e veja o resultado:



- Rowspan - Pode ser aplicado a TH ou TD. Define quantas linhas uma mesma célula pode abranger. Assim como na marcação anterior, o padrão é uma célula corresponder a uma linha.

Exemplo:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Início</title>
</head>
<body>
<table border>
<tr>
<td rowspan=3> 3 Linhas</td>
<td>Col 2</td> <td>Col 3</td>
<td>Col 4</td> <td>Col 5</td>
</tr>
<tr>
<td>Col 2</td> <td>Col 3</td>
<td>Col 4</td> <td>Col 5</td>
</tr>
<tr>
</td><td>Col 2</td> <td>Col 3</td>
<td>Col 4</td> <td>Col 5</td>
</tr>
</table>
</body>
</html>
```


Salve e veja o resultado:



Layout

Layout é uma palavra inglesa, muitas vezes usada na forma portuguesa "leiaute", que significa plano, arranjo, esquema, design, projeto.

Na área da arte gráfica, o layout é um esboço ou rascunho que mostra a estrutura física de uma página de um jornal, revista ou página na internet (como um blog, por exemplo). layout engloba elementos como texto, gráficos, imagens e a forma como eles se encontram em um determinado espaço.

O layout gráfico pressupõe o trabalho de um designer gráfico, que vai trabalhar no formato e números de página e suas margens, números de colunas de texto e outros aspectos relevantes. Para montarmos um layout em HTML é necessário algumas atribuições para que se elabore um projeto.

Bom, apenas usando tabelas vamos criar o layout de um site bem interessante.

Obviamente é algo básico e usando tags e conceitos do HTML 4, onde várias de suas tags já estão em desuso.

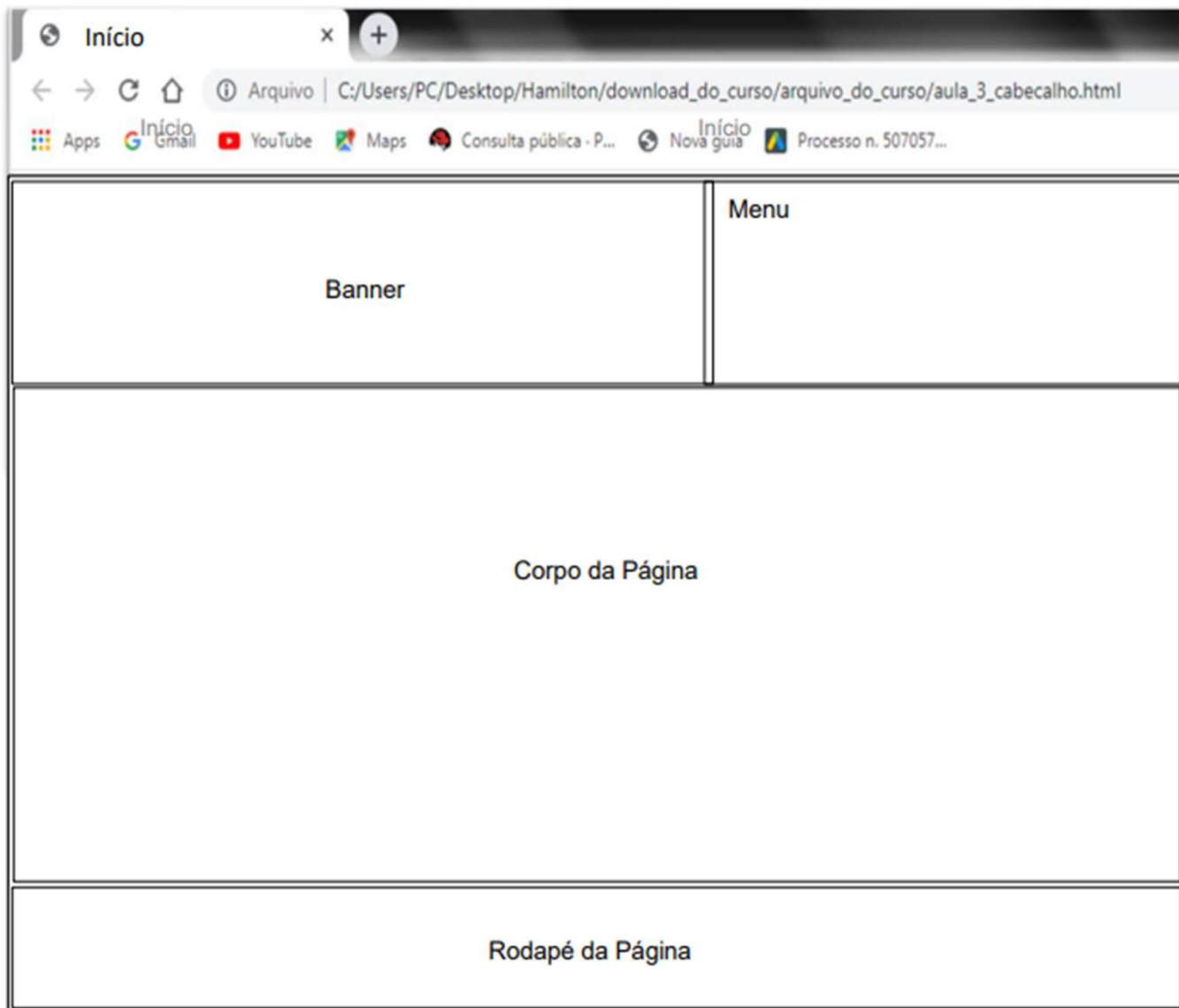
Porém, é de suma importância que se aprenda essas tags, e principalmente: é essencial fazer um site na mão, na unha, escrevendo todos os código. Só assim você se tornará um profissional, um webmaster bem competente.

Vamos então desenvolver o layout que terá: Uma área para **banner** e um menu de **navegação horizontal**, que na parte superior. Lembrando que no banner é onde vamos inserir a logomarca. Teremos também o **corpo** do layout, e **um rodapé**.

Agora vamos traduzir isso para o HTML:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>Início</title>
</head>
<body >
<table border=1 align=center>
<td width=400 height=100 valign=middle align=center>
Banner
</td>
<td width=400 height=100 valign=top>
Menu
</td>
<tr>
<td colspan=2 height=400 valign=middle align=center>
Corpo da Página
</td>
</tr>
<tr>
<td colspan=2 height=180 align=center>
Rodapé da Página
</td>
</tr>
</table>
</body>
</html>
```

Salve e veja o resultado:



Porém, elaborar um projeto, usando tabelas para construir um layout não é a maneira mais viável. Para elaborarmos um projeto na construção de layouts é necessário o uso da CSS. Para que entenda melhor, irei explicar as alterações sofridas nas versões da linguagem de marcação HTML, assim como viu o básico na introdução do curso.

Bom, sem mais delongas, preste bem atenção:

O HTML surgiu em 1990 e até o seu quinto ano de vida sofreu várias revisões e alterações na sua especificação. Nesta época, quem controlava o padrão era o CERN e a IETF. Após 1995, o padrão passou a ser regularizado pela W3C, entidade que regula os padrões Web e que será detalhada com maior profundidade posteriormente.

Evolução do HTML

Em 1993, Dave Raggett propôs uma evolução do padrão HTML, denominada HTML+. Entretanto tal proposta nunca foi implementada.

O verdadeiro sucessor do HTML foi o HTML 2.0, o qual foi apresentado na primeira conferência mundial sobre Web, a World Wide Web Conference.

Essa versão do HTML pode ser vista apenas como uma correção da versão anterior, ou seja, apenas formalizava as características do HTML que já estavam em uso.

Depois, Dave Raggett escreveu o HTML 3.0 baseado em seu mais recente rascunho do HTML+. Entretanto, mais uma vez, a versão do HTML não foi implementada.

Esta especificação foi superada pela versão 3.2, a qual novamente foi uma correção e providenciou total compatibilidade com a versão 2. O HTML 3.2 implementou características tais como tabelas, applets e texto flutuante ao redor de imagens.

Posteriormente, Raggett foi co-author do HTML 4 e ajudou com o desenvolvimento das

linguagens como XHTML, XForms, MathML e além de outras especificações mais modernas da W3C.

Em dezembro de 1999 o HTML 4.01 foi publicado trazendo como principal premissa a compatibilidade com as suas versões anteriores através de 3 implementações:

- Strict (estrita) – implementação na qual fica proibida a utilização de elementos obsoletos da linguagem;
- Transitional (transitória) – implementação na qual os elementos obsoletos são permitidos;
- Frameset – implementação direcionada para site que usam frames.

World Wide Web Consortium W3C

Em agosto de 1994 foi fundada a W3C, um consórcio formado por instituições comerciais

e educacionais, com o objetivo de definir padrões para as respectivas áreas relacionadas

à Web.

Em dezembro de 1997 a W3C publicou o HTML 4 e após dois meses publicou o XML1. A W3C reformulou o grupo responsável pelo HTML para criar uma “suíte de tags XML”. O primeiro passo foi dado em dezembro de 1998 quando o grupo reescreveu o HTML em XML sem adicionar elementos ou atributos novos. Essa especificação foi chamada de XHTML 1.

O objetivo seguinte da W3C foi a reestruturação dos formulários Web. Em agosto de 1999

o mesmo grupo responsável pelo HTML publicou o primeiro rascunho da extensão dos formulários para XHTML. Alguns meses depois, essa “extensão dos formulários para XHTML” foi rebatizada de “XForms”.

Foi criado um grupo específico, responsável pelo XForms. Esse grupo trabalhou em paralelo com o grupo de desenvolvimento do HTML com a finalidade de publicar a primeira versão do XForms em outubro de 2003. Enquanto isso, com a transição do XML completa, os membros do grupo do HTML criaram “a nova geração do HTML”. Em maio de 2001 eles publicaram a primeira versão do XHTML 1.1.

Em julho de 2004 a W3C organizou um workshop chamado “Workshop on Web Applications and Compound Documents”. Nesse workshop estavam presentes membros da W3C e companhias desenvolvedoras de navegadores como a fundação Mozilla e a Opera. Neste workshop foi apresentada uma visão do futuro da web com uma evolução do padrão HTML 4 incluindo novas características para aplicações modernas. No workshop a W3C deixou claro que não seriam desenvolvidas novas tecnologias de extensões do HTML e CSS que não fossem criadas pelo atual grupo

de trabalho da W3C. Diante desse impasse, o grupo que desenvolvia o HTML e os formulários HTML tinha duas escolhas: ou se uniam à W3C, ou se separavam. Dessa forma, em julho de 2004 surgiu o WHATWG.

Por muitos anos o W3C e o WHATWG ignoraram um ao outro. Enquanto o WHATWG focava nos formulários e nas novas características no HTML, o W3C estava ocupado com a versão 2.0 do XHTML. Mas em outubro de 2006, ficou claro que a WHATWG estava vivendo um momento mais sério, enquanto o XHTML 2.0 estava sendo lançado como rascunho e não estava sendo implementado pela maioria dos navegadores. Em outubro de 2006, Tim Berners-Lee, o fundador da W3C, anunciou que a W3C deveria trabalhar em conjunto com a WHATWG. O primeiro passo do trabalho em conjunto foi renomear o “Web Application 1.0” para “HTML5”.

Desenvolvedor Front End, entusiasta em HTML5, CSS3, Web Semântica, SEO e jQuery.

David William

CSS – Folha de Estilo em Cascata

O Cascading Style Sheets (CSS) 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 (como XML, HTML e XHTML). O CSS define como serão exibidos os elementos contidos no código de uma adotando cada vez mais estilos e variações para deixá-las mais elegantes a página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento.

Por que o CSS foi criado?

Com a evolução dos recursos de programação as páginas da internet estavam atrativas para os usuários. Com isto, linguagens de marcação simples como o HTML, que era destinada para apresentar os conteúdos também precisou ser aprimorada.

Foram criadas novas tags e atributos de estilo para o HTML e em resumo ele passou a exercer tanto a função de estruturar o conteúdo quanto de apresentá-lo para o usuário final. Entretanto, isto começou a trazer um problema para os desenvolvedores, pois não havia uma forma de definir, por exemplo, um padrão para todos os cabeçalhos ou conteúdos em diversas páginas. Ou seja, as alterações teriam que ser feitas manualmente, uma a uma.

A partir destas complicações, nasceu o CSS. Primariamente, foi desenvolvido para habilitar a separação do conteúdo e formato de um documento (na linguagem de formatação utilizada) de sua apresentação, incluindo elementos como cores, formatos de fontes e layout. Esta separação proporcionou uma maior flexibilidade e controle na especificação de como as características serão exibidas, permitiu um compartilhamento de formato e reduziu a repetição no conteúdo estrutural de uma página.

Com isto, as linguagens de marcação passaram novamente a exercer sua função de marcar e estruturar o conteúdo de uma página enquanto o CSS encarregou-se da aplicação dos estilos necessários para a aparência dela. Isto é feito por meio da criação de um arquivo externo que contém todas as regras aplicadas e, com isto, é possível fazer alterações de estilo em todas as páginas de um site de forma fácil e rápida.

O CSS também permite que as mesmas marcações de uma página sejam apresentadas em diferentes estilos, conforme os métodos de renderização (como em uma tela, impressão, via voz, baseadas em dispositivos táteis, etc.). A maioria dos menus em cascata, estilos de cabeçalho e rodapé de páginas da internet, por exemplo, atualmente são desenvolvidos em CSS.

Iniciando e criando arquivos CSS

Tag <style>

Agora que já conheceu o que é CSS e para que serve, você já deve estar convencido da importância da sua utilidade. É impossível ser um webmaster ou web designer profissional sem dominar completamente o uso dos estilos com CSS.

Como tudo em HTML e desenvolvimento web, há diversas maneiras de se fazer um site, sistema ou um simples efeito, o que torna o HTML uma linguagem muito flexível, principalmente se usada com o CSS. Essa flexibilidade se estende ao uso de estilos também, sendo possível estilizar um site inteiro, uma página ou mesmo apenas uma tag.

A maneira mais direta, e talvez simples, e com certeza mais trabalhosa, de se estilizar um elemento de nosso site, é usando o style como atributo de alguma tag. Por exemplo, se quisermos escrever uma frase usando a tag de cabeçalho h2 na cor vermelha, podemos fazer: `<h2 style="color: red;">` tag h2 estilizada diretamente. Veja como esta sintaxe difere das que tínhamos visto até o momento no curso. Como de costume, colocamos o valor do atributo dentro de aspas. Dentro das aspas podemos estilizar de várias maneiras.

No exemplo dado, estilizamos a cor da fonte, e para tal usamos o atributo color. Para este atributo receber um valor, usamos o dois pontos ":" e em seguida o valor do atributo color, que é red para vermelho. Podemos, inclusive, usar diversas regras de estilo com o atributo style. Para isso, basta separar como ponto e vírgula ";" as diferentes propriedades.

Por exemplo, para fazer nossa tag h2 ter tamanho da fonte 12, além de azul, fazemos assim: `<h2 style="color: red; font-size: 12pt;">`. Perceba que a tag `<h2>` estilizada diretamente com o atributo `style` `</h2>`

Inserindo arquivos CSS no Documento HTML

Há três maneiras de inserir o arquivo CSS em um documento HTML.

1. A primeira delas é inserindo a tag `Style` entre a própria tag `HTML`, ou seja `CSS INLINE`, que significa Regra de CSS. Para que entenda melhor, preste bem atenção no exemplo abaixo:

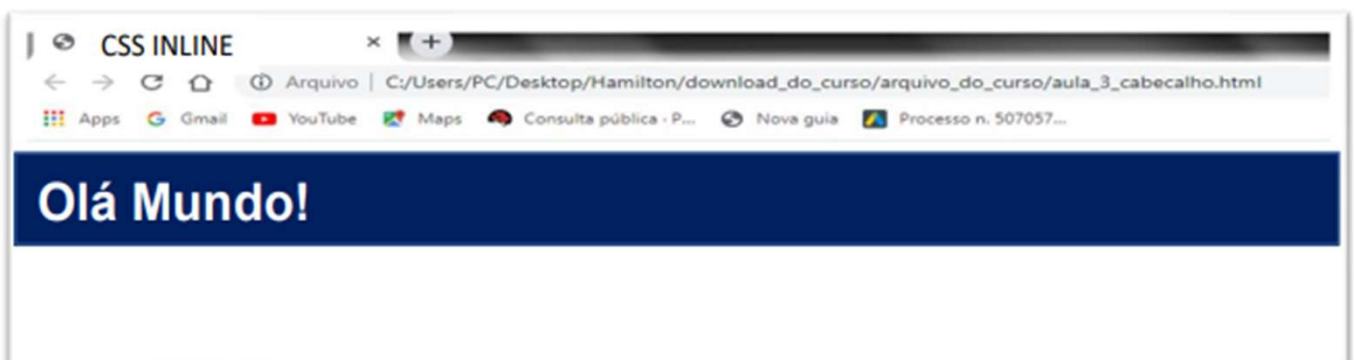
Vamos estilizar a página da seguinte forma:

No título, `<h1>Olá Mundo</h1>` vamos mudar a cor do texto para branco e a cor de fundo para azul.

Veja com fica na prática:

```
<!doctype html>
<html lang ="pt-br">
<head
<meta charset ="UTF-8">
<title>CSS INLINE</title>
</head>
<body >
<h1 style="color:white; background:blue;">Ola Mundo!</h1>
</body>
</html>
```

Salve e veja o resultado:



2. A segunda maneira de inserir o arquivo CSS em uma página web é através da tag `<style>` e o atributo `type` em CSS ou seja, criando um arquivo CSS interno dentro do documento HTML.

Obviamente, estilizar e colocar diversas propriedades dentro do atributo `style` de cada tag dentro de uma página daria muito trabalho. Seria mais interessante se pudéssemos declarar esse estilo apenas uma vez.

E é possível declarar o estilo somente uma vez em uma página, e estilizar todo o documento de uma vez. Isso é feito através da tag `<style>`, que tem toda uma sintaxe especial.

Antes de mais nada, devemos definir o tipo de estilo que vamos usar. Iremos usar o CSS. Fornecemos essa definição do estilo através do atributo `style` que recebe o valor `text/css`. Assim, a sintaxe da tag `style` é: `<style type="text/css"> </style>`

Nota: Essa tag é usada dentro da tag `head`. Outro importante conceito é a sintaxe em CSS de estilo.

Vamos ver como funciona isso na prática:

```
<!doctype html>
<html lang = "pt-br">
<head
<meta charset = "UTF-8">
<title>CSS INTERNO</title>
<style type="text/css">
h1{
color:white;
background: red;
}
</style>
</head>
<body >
<h1>Ola Mundo!</h1>
</body>
</html>
```

Salve e veja o resultado:



Sintaxe da Propriedade CSS

A sintaxe para usarmos CSS em sites recebe um formato padrão e diferente de tudo que já vimos até aqui em no curso. Dentro desta tag `<style>` que acabamos de apresentar, as propriedades do estilo de na página estarão declaradas no seguinte formato:

```
<style type="text/css">
Elemento 1 {
Propriedade 1: valor 1;
Propriedade 2: valor 2;
}
Elemento 2 {
Propriedade 1: valor 1;
Propriedade 2: valor 2;
}
</style>
```

Onde os 'elementos' são os mais diversos tipos de uma página, como a tag `h1`, tag `h2`, tag `body`, uma imagem, a tag `<p>` de parágrafo e praticamente qualquer coisa em um site. Pois é possível estilizar praticamente tudo com CSS.

Para criar um site você pode definir os estilos para as tags `h1` e `h2`.

- Para a tag `h1`: cor vermelha.
- Para a tag `h2`: cor azul e tamanho de fonte 12pt.
Basta definir esse estilo na tag `<head>`, dentro da tag `<style> </style>`.

Dentro do corpo do site (tag <body>), sempre que usar as tags h1 e h2 elas terão, automaticamente, o estilo definido via CSS.

Pode usar quantas vezes quiser as tags h1 e h2 , e sempre vão ter o estilo que você definiu só uma vez na tag <style>. Não precisa mais se preocupar em estilizar individualmente cada tag h1 e h2, basta fazer isso somente uma vez.

3. A terceira e ultima maneira de inserir o arquivo CSS em uma página web é criando um arquivo CSS externo ou seja uma folha de estilo fora do documento HTML. Para criarmos um arquivo separado do documento é necessário criar um link, ou seja, um elo de ligação chamando o arquivo CSS.

Como já vimos em aulas anteriores, é no cabeçalho do documento <head> </head> que inserimos o link para nos levar ao arquivo externo CSS. É exatamente isso que faremos agora.

Vamos ver como tudo isso se comporta na pratica:

```
<!doctype html>
<html lang ="pt-br">
<head>
<meta charset ="UTF-8">
<title>CSS EXTERNO</title>
<link rel="stylesheet" type="text/css" href="css/estilo.css"/>
</head>
<body >
<h1>Ola Mundo!</h1>
</body>
</html>
```

Salve e veja o resultado:



Ao salvar o documento perceba que nada se altera. Isso ocorre porque nós criamos apenas o link com o endereço onde o arquivo externo será redirecionado, mas não criamos a pasta, para que os arquivos sejam exibidos na folha de estilo CSS. É exatamente isso que faremos agora. Crie uma pasta e a renomeie com o nome: CSS, abra a pasta e veja que dentro da pasta contem o arquivo estilo.css

Foi exatamente isso que fizemos: informamos ao navegador/browser que queremos uma folha de estilo separada do documento HTML ao digitar:

`<link rel="stylesheet" type="text/css" href="css/estilo.css">` no cabeçalho do documento.

Nota: Para salvar o arquivo CSS Externo você deve criar a pasta CSS primeiro, depois abrir o bloco de notas e salvar o documento com o nome de estilo.css ou o nome que achar melhor, desde que a extensão do arquivo sempre seja .css. Com o arquivo CSS aberto vamos estilizar a tag de cabeçalho `<h1>` do documento HTML.

```
h1{
color:white;
background-color:blue;
}
h1:hover{
color:red;
background-color:black;
cursor:pointer;
}
```

HTML BÁSICO



Perceba que agora você tem dois arquivos separados: um documento **HTML** e uma folha de estilo **CSS**. E dessa forma que criamos um arquivo **CSS Externo**

```
<!doctype html>
<html lang = "pt-br">
  <head
    <meta charset = "UTF-8">
    <title>CSS EXTERNO</title>
    <link rel="Stylesheet" type="text/css" href="css/estilo.css">
  </head>
  <body >
  <h1>Ola Mundo!</h1>
  </body>
</html>
```

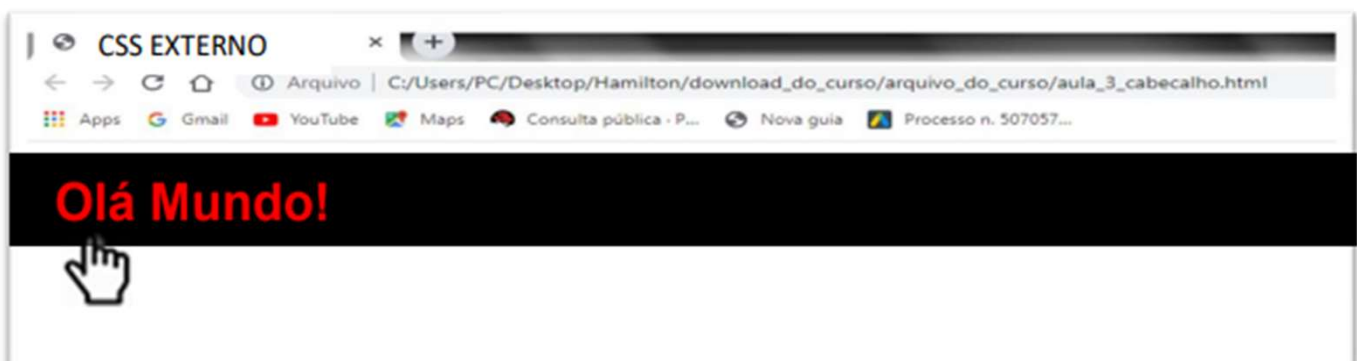
```
h1{
  color:white;
  background-color:blue;
}

h1:hover{
  color:red;
  background-color:black;
  cursor:pointer;
}
```

Salve o documento veja como fica:



Perceba também que ao passar o mouse sobre o titulo da página: **Olá Mundo!** O titulo fica com a cor vermelha e o fundo preto, no qual definimos no arquivo **CSS Externo**.



Já vimos na aula sobre tabelas, que é possível construir **layouts** com o uso das tabelas (**tables**), mas essa não é maneira mais prática de desenvolver projetos web na atualidade.

Vamos desenvolver um projeto final para web da forma correta, com todos os conceitos abordados no curso, onde eu o auxiliarei passo a passo.

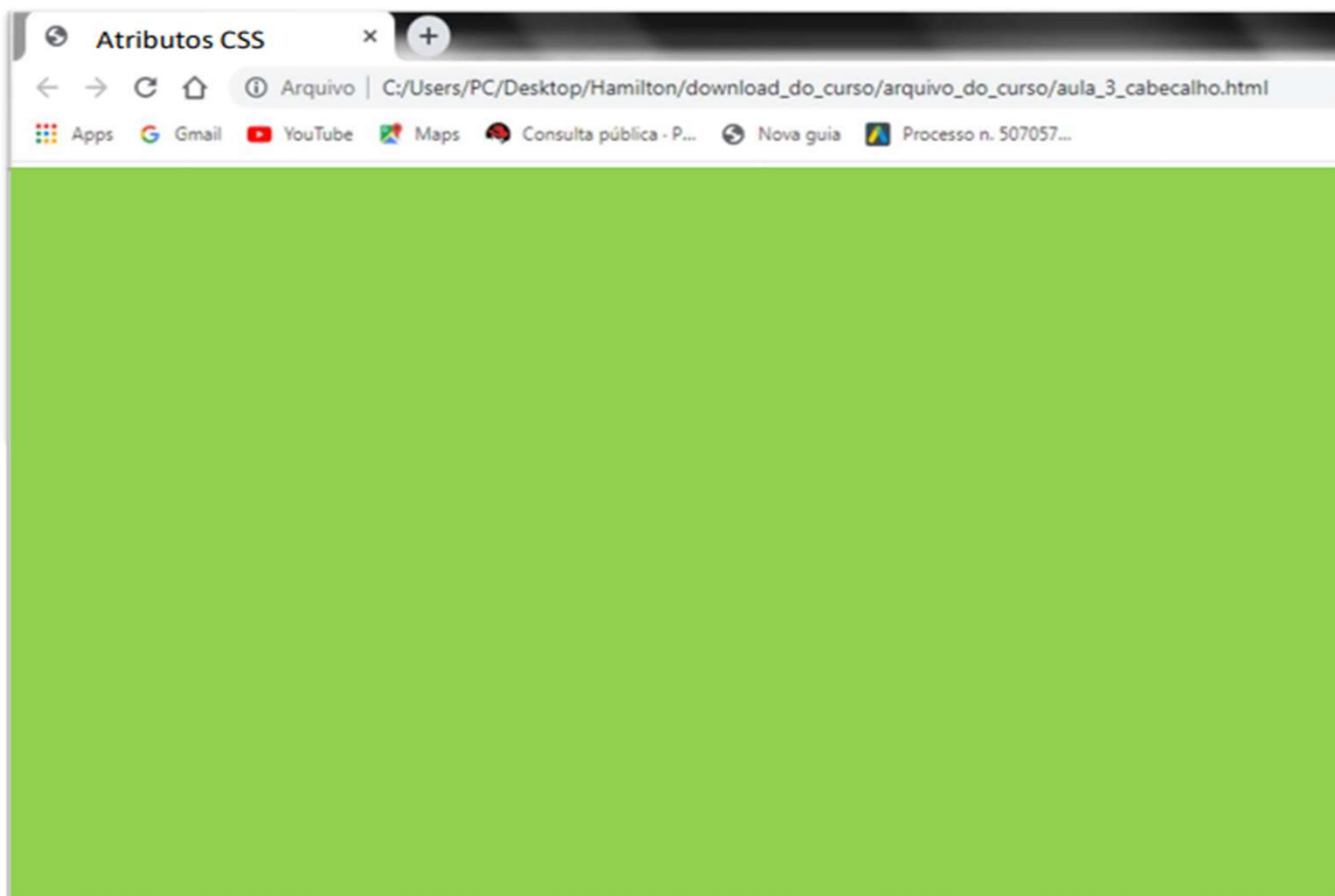
Mas para isso você vai precisar dos atributos CSS, para elaborar não só este projeto final do curso HTML para incitantes, mas para qualquer projeto que criar. Para isso elaborei os principais atributos CSS na Apostila Digital. Sendo assim, quando vier a dúvida, você pode consultar na apostila todas as vezes que precisar e quantas vezes quiser.

Principais Atributos CSS

- Background-color - Cor de fundo, imagem, transparência.

Exemplo:

```
body{  
background-color: green
```

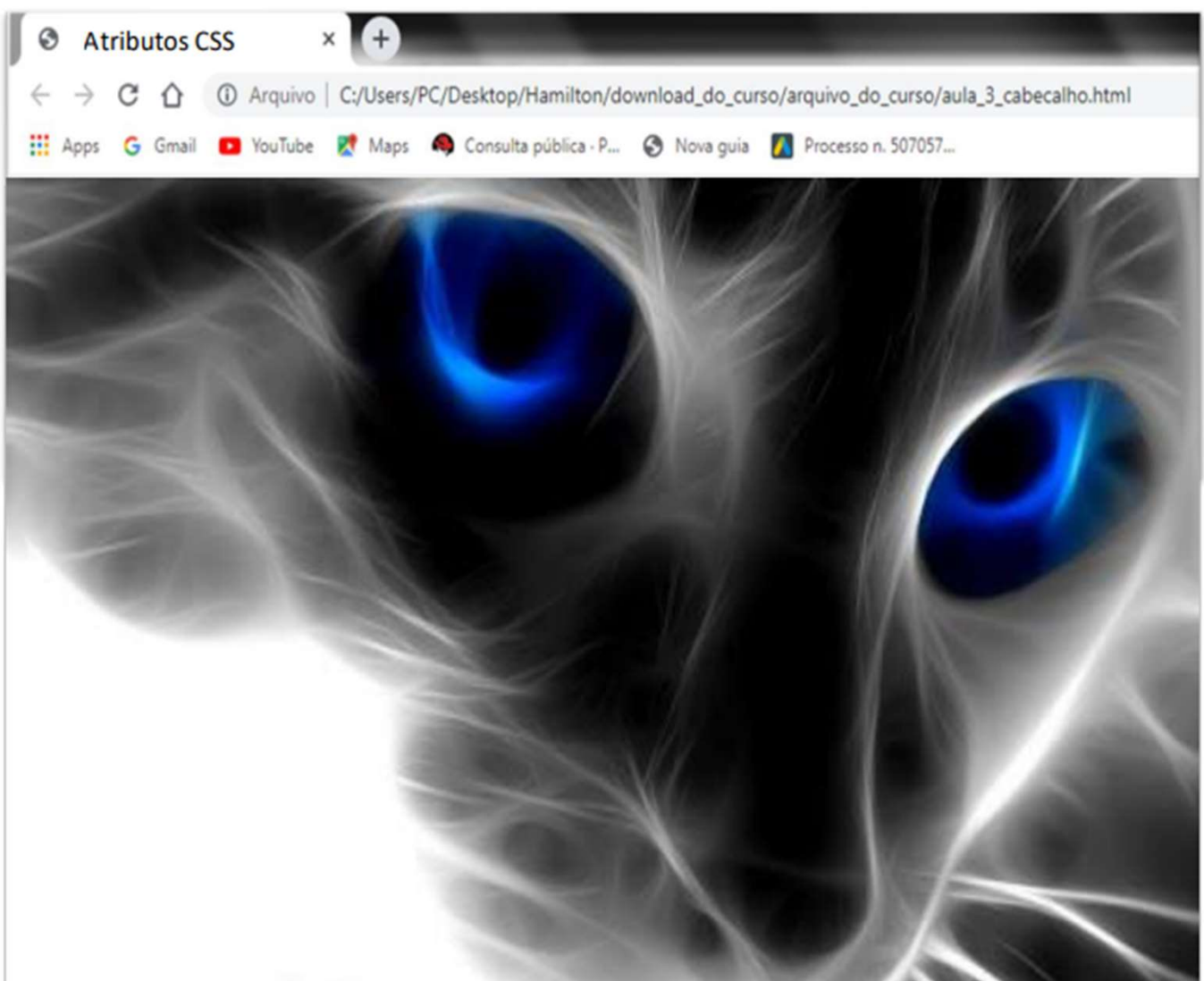


- **Background-image** - Imagem de fundo.

Ao utilizar uma imagem como background, é preciso indicar o seu endereço ao navegador para que ela seja carregada.

Exemplo:

```
body{  
background-image:url(..img/fundo-01.png);
```

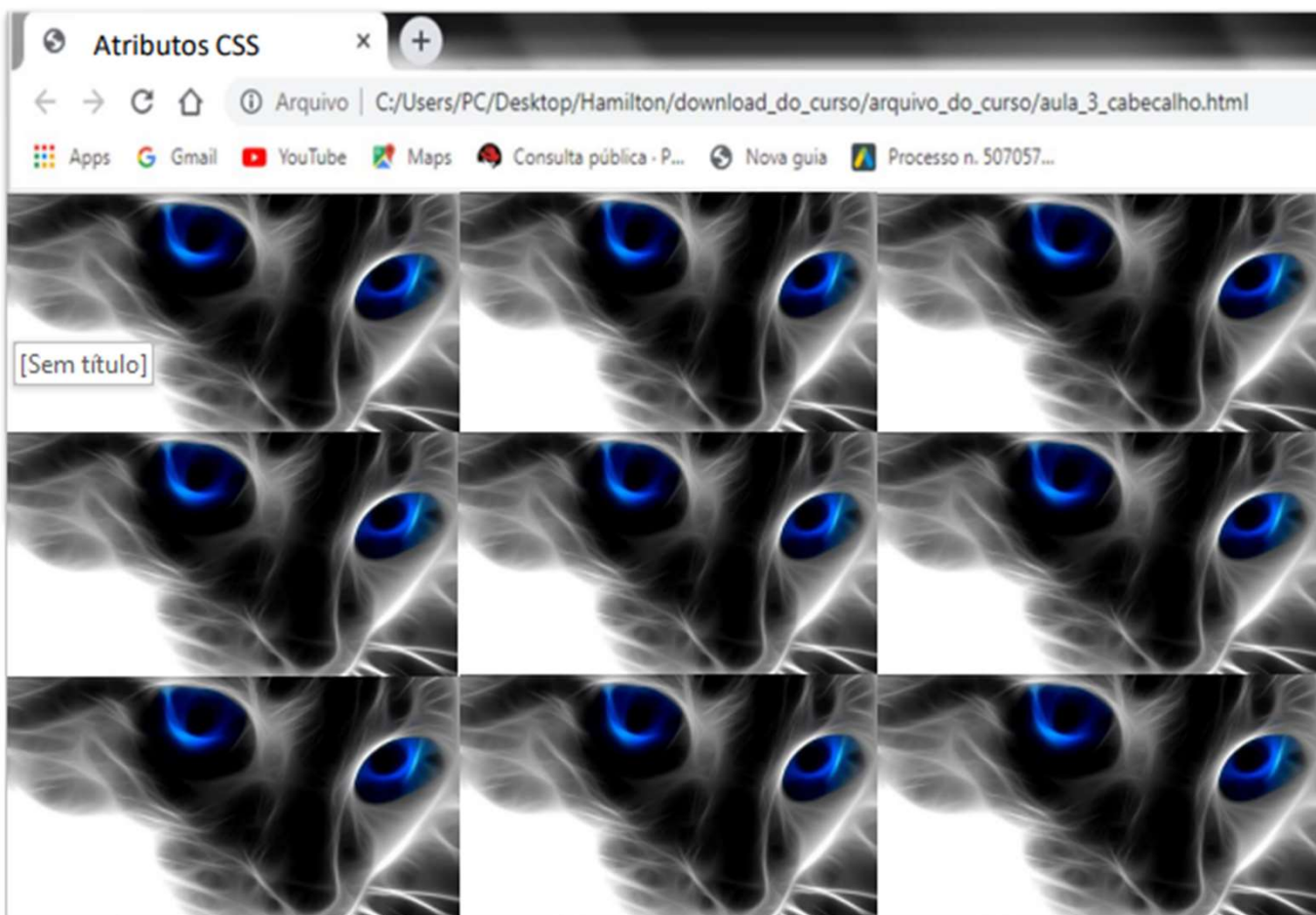


HTML BÁSICO



Por padrão, a imagem de background é posicionada no canto superior esquerdo do espaço definido para o elemento e se repete até que tudo seja preenchido. No exemplo acima a imagem de fundo ocupa toda a página porque área total de sua extensão é **1920 x 1080px**.

Veja este outro exemplo com a imagem reduzida: **512px x 200px**.



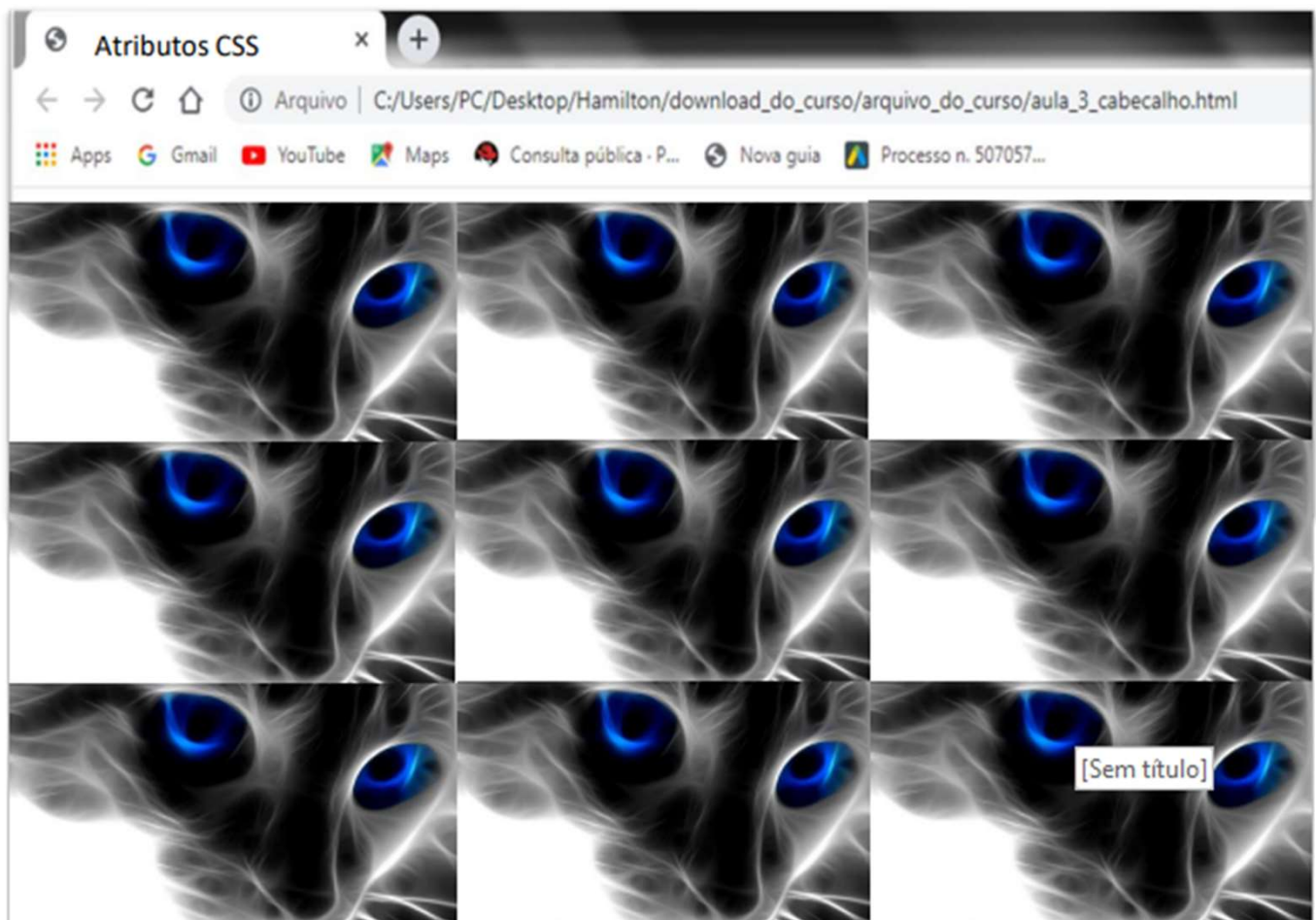
Para que a imagem ocupe um determinado espaço, fique centralizada ou se repita apenas em um sentido, por exemplo, é preciso utilizar as propriedades auxiliares, são elas:

- **Background-repeat** - Essa propriedade pode receber os seguintes valores para a repetição da imagem de fundo:

➤ **repeat**: repete a imagem na vertical e horizontal até o preenchimento da área do elemento;

Exemplo:

```
body{  
background-image:url(..img/fundo-02.png);  
background-repeat: repeat;  
}
```



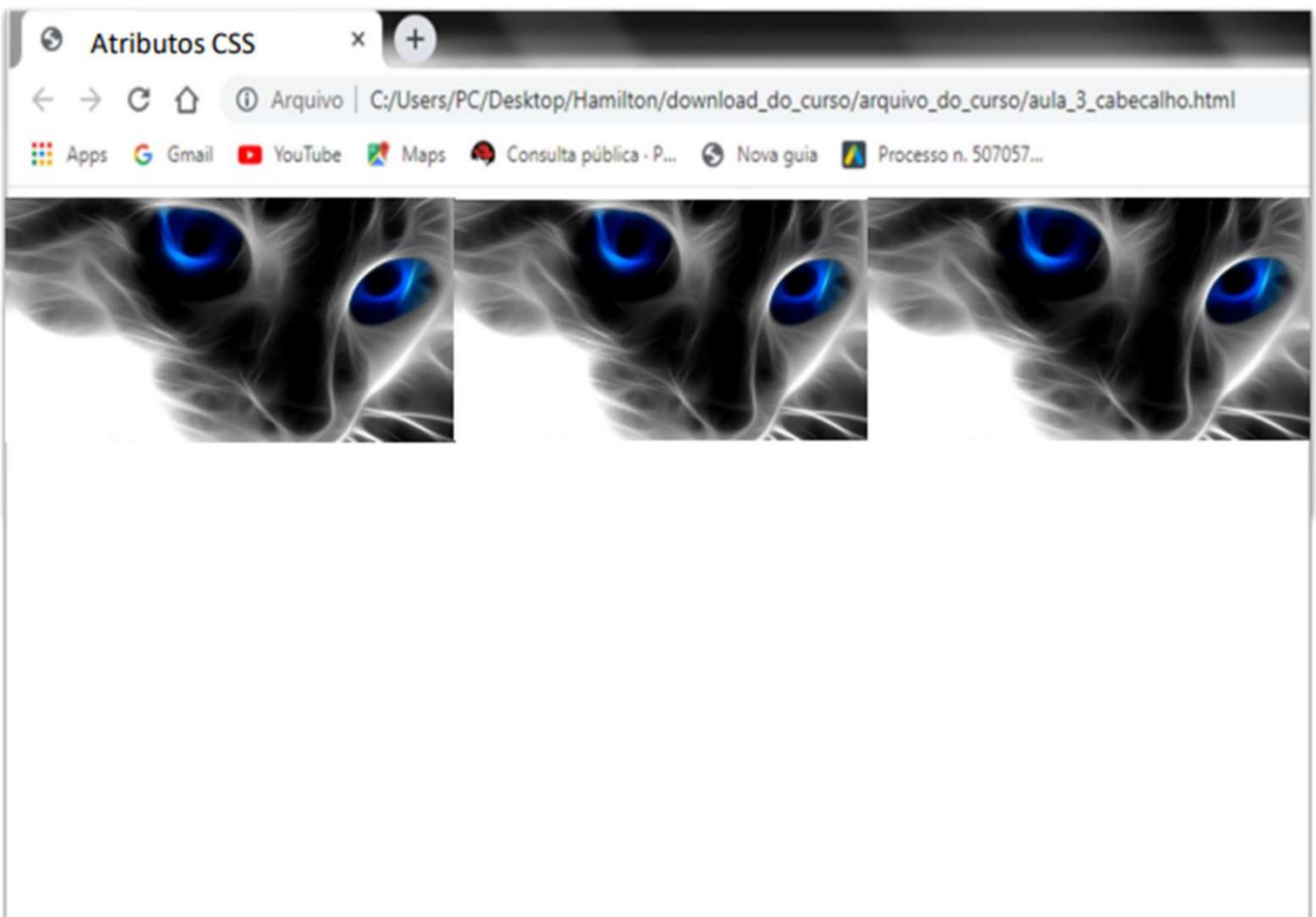
HTML BÁSICO



➤ **repeat-x:** repete a imagem apenas no sentido horizontal;

Exemplo:

```
body{  
background-image:url(..img/fundo-02.png);  
background-repeat: repeat-x;  
}
```



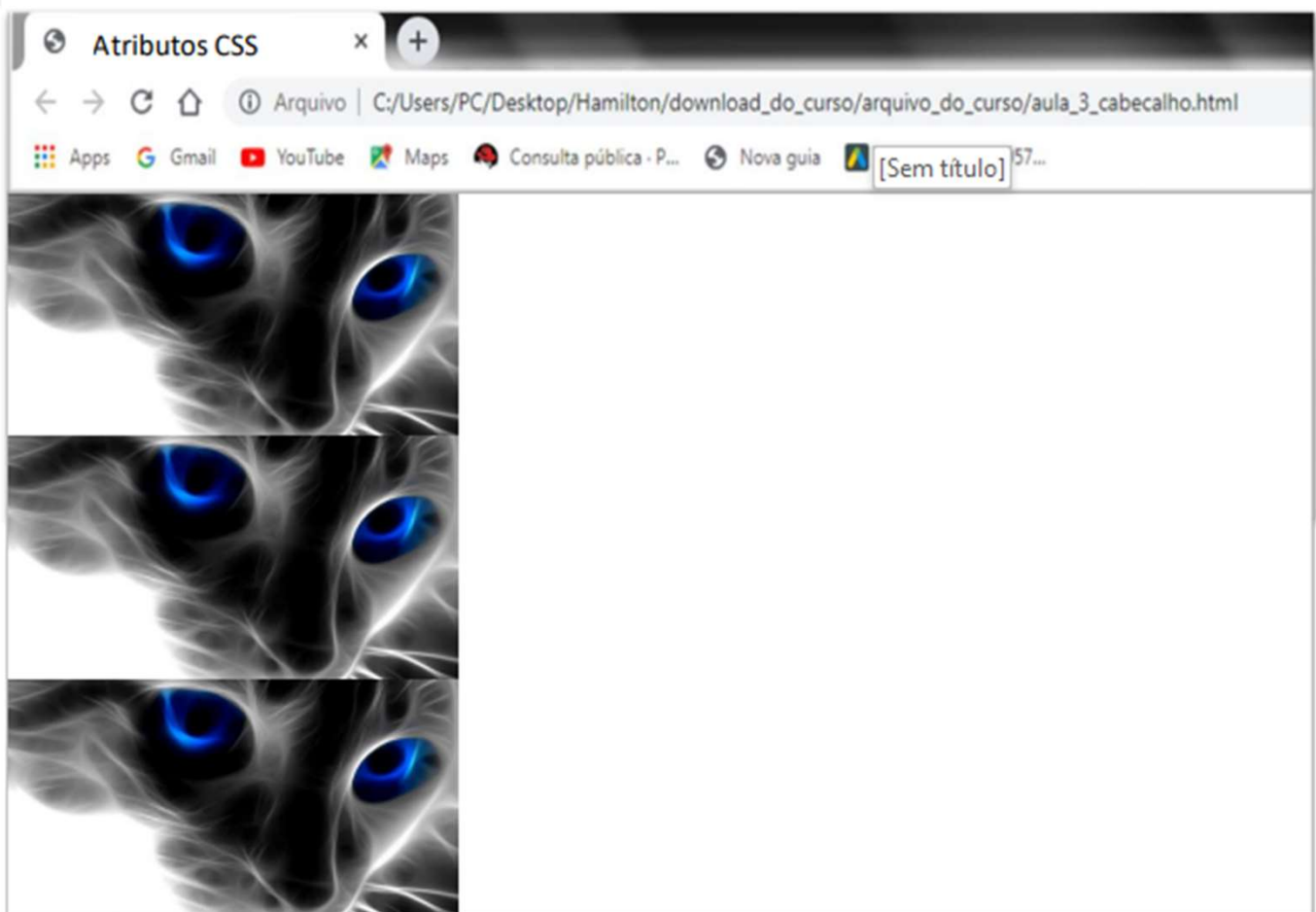
HTML BÁSICO



➤ **repeat-y:** repete a imagem apenas no sentido vertical;

Exemplo:

```
body{  
background-image:url(..../img/fundo-02.png);  
background-repeat: repeat-y;  
}
```



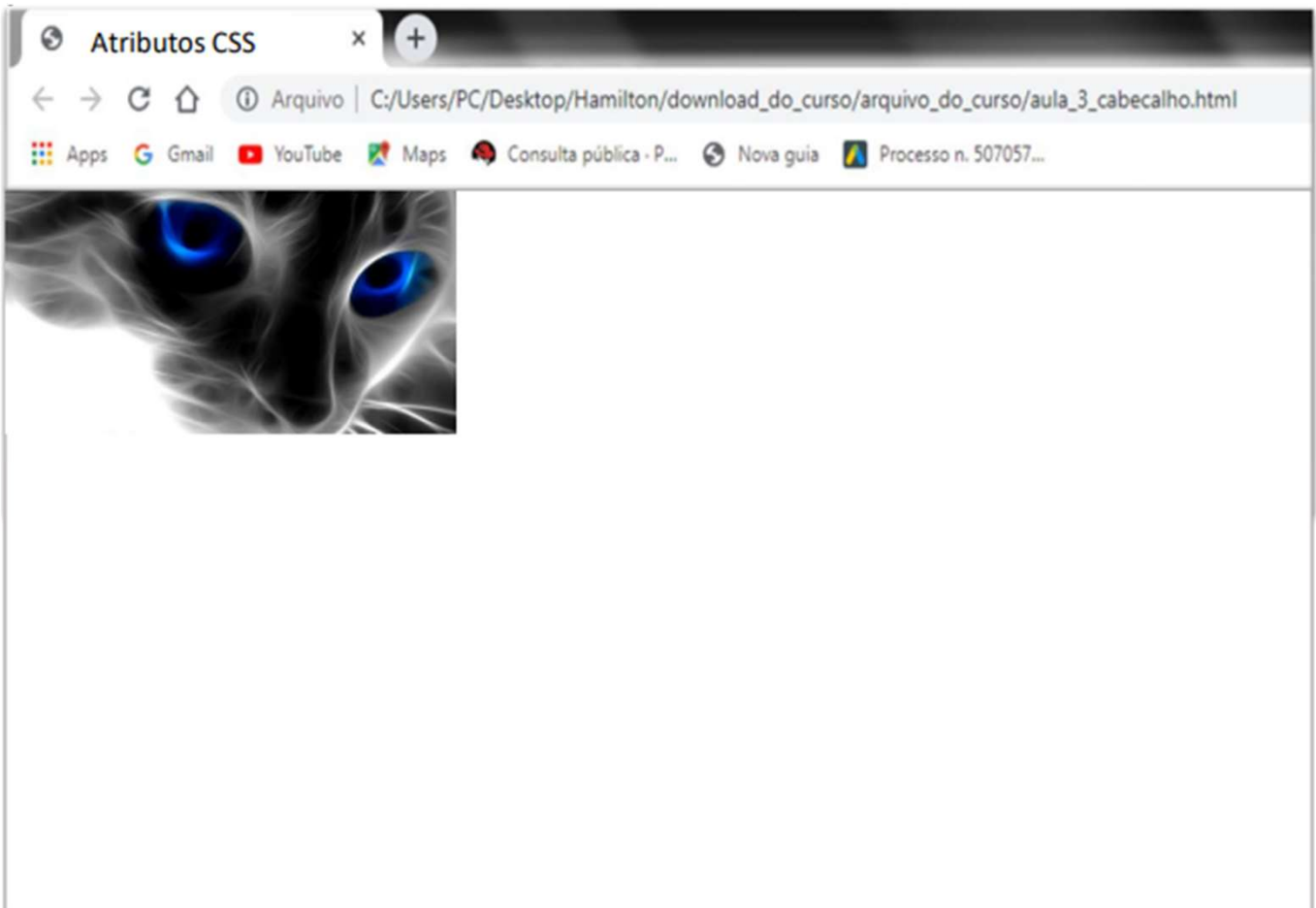
HTML BÁSICO



➤ **no-repeat:** a imagem é inserida apenas uma vez no plano de fundo.

Exemplo:

```
body{  
background-image:url(..img/fundo-02.png);  
background-repeat: no-repeat;  
}
```



HTML BÁSICO



- **Border** - Largura, estilo e cor de todas as 4 bordas.

Exemplo:

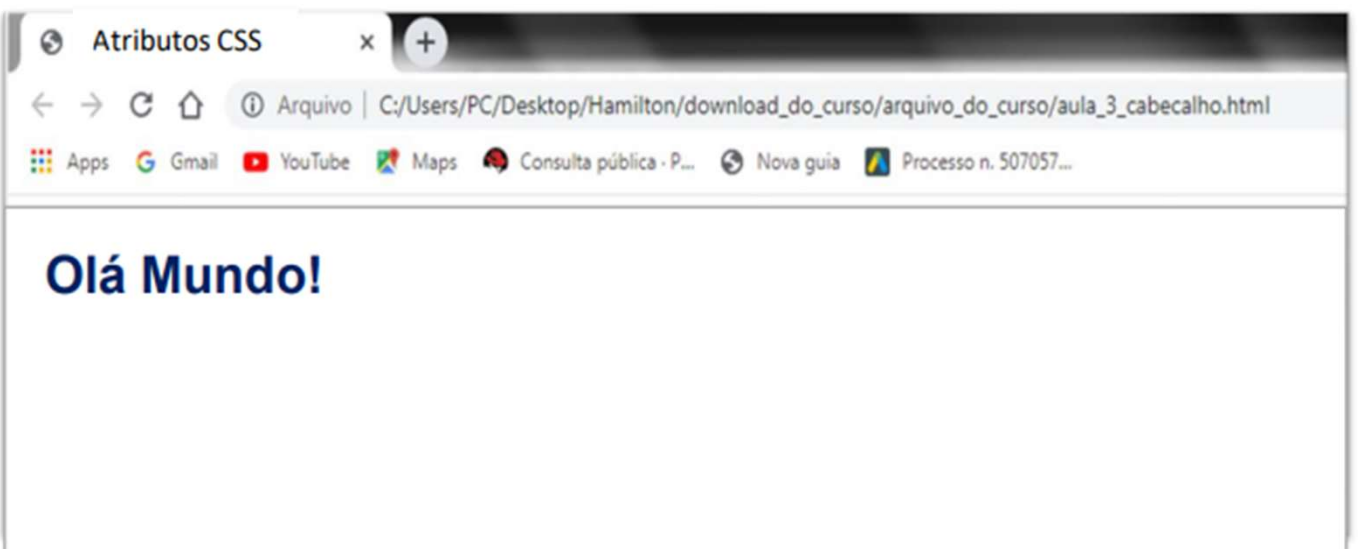
```
img{  
border:3px solid red;  
}
```



- **Color** - Cor de primeiro plano.

Exemplo:

```
h1 {  
color:blue;  
}
```



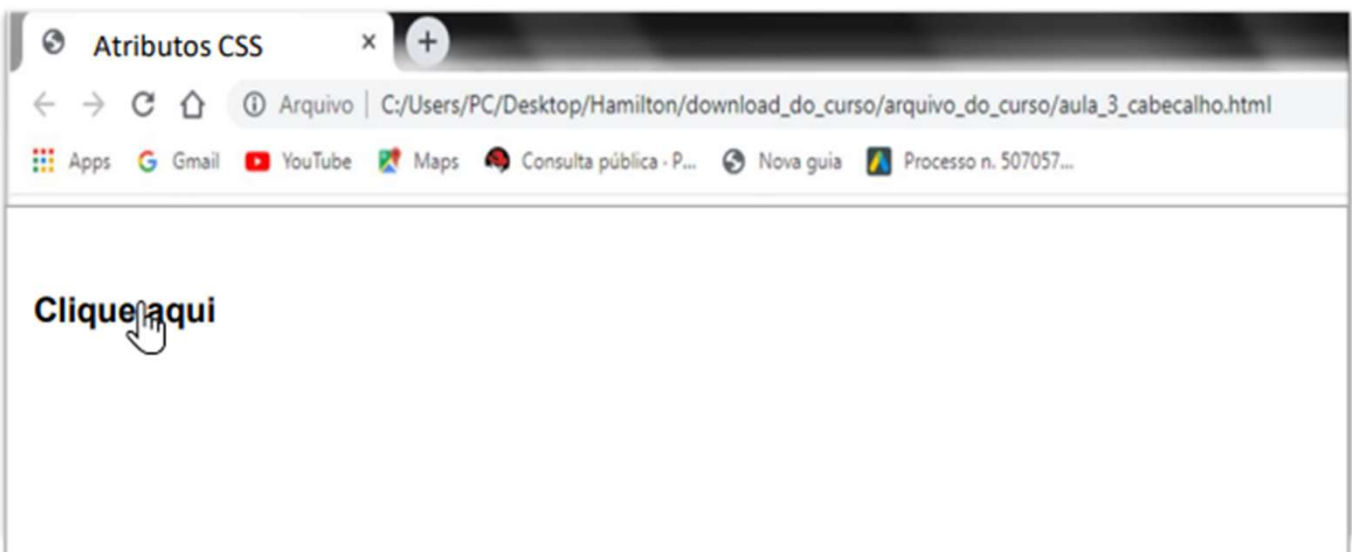
HTML BÁSICO



- **Cursor** - Tipo de ponteiro do mouse.

Exemplo:

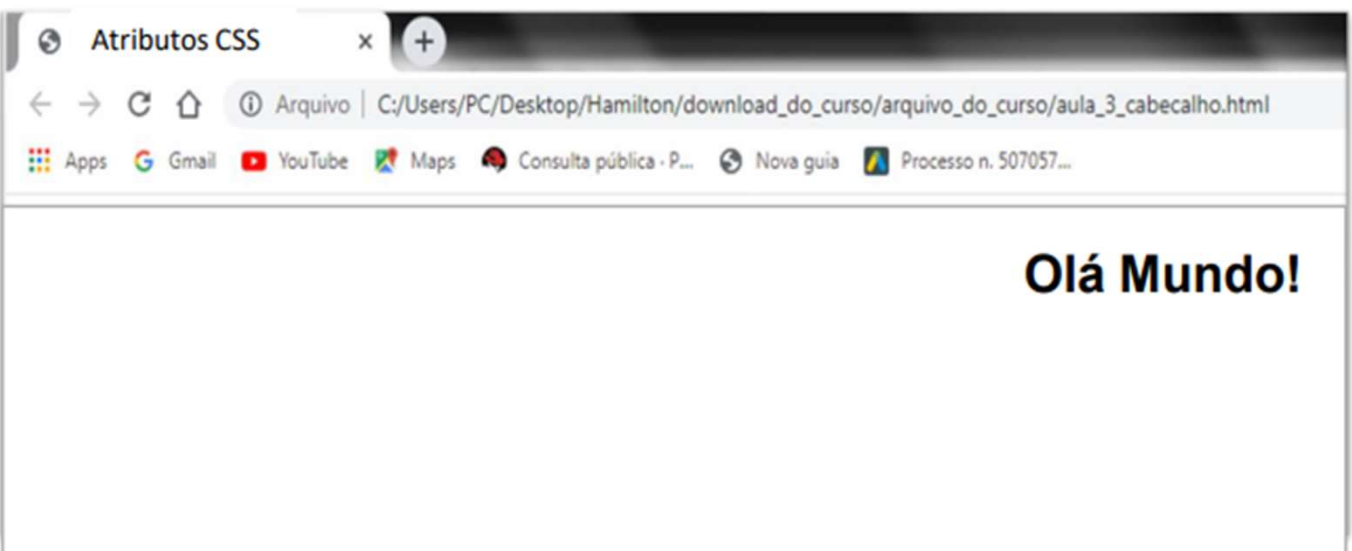
```
a{  
  cursor: pointer;  
}
```



- **Float** - Se o elemento flutua. (right, left)

Exemplo:

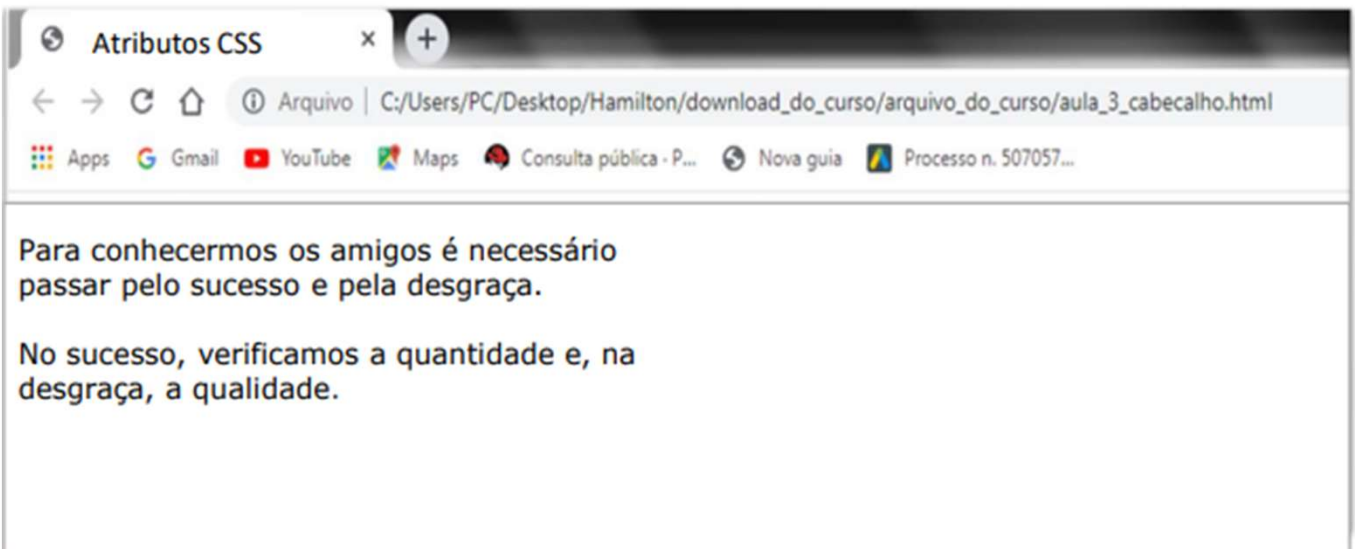
```
h1{  
  float:right;  
}
```



- **Font-Family** - Tipo de fonte.

Exemplo:

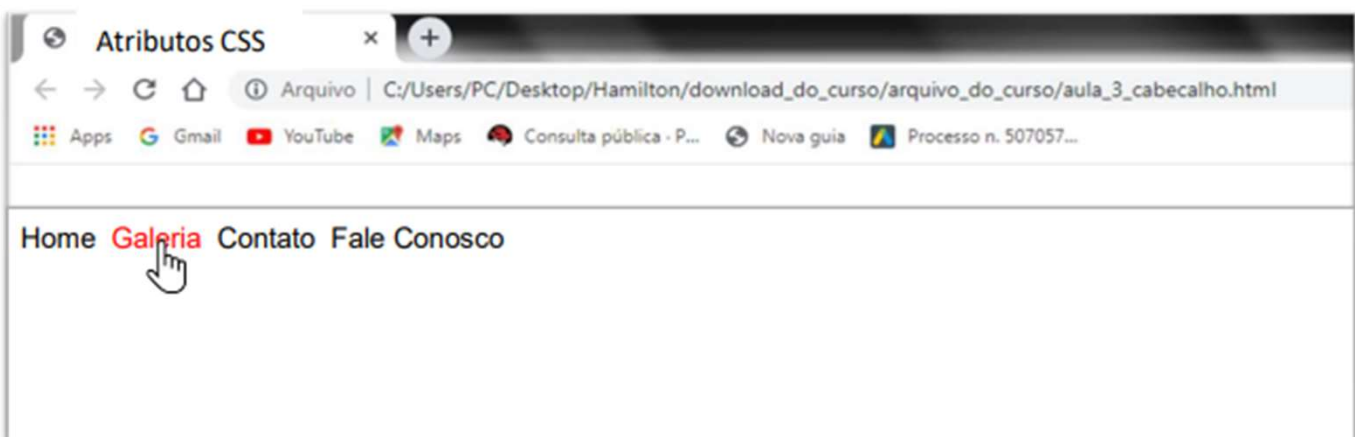
```
body {  
font-family:Verdana, Geneva, sans-serif;  
}
```



- **Hover** - Define um efeito de transição ao passar o mouse sobre o elemento

Exemplo:

```
a:hover {  
font-family:Verdana, Geneva, sans-serif;  
text-decoration:underline;  
color:red;  
cursor:pointer;
```



- **Font-Size** - Tamanho da fonte.

Exemplo:

```
h4{
font-size:30pt;
}
h5{
font-size:30px;
}
h6{
font-size:1.2em;
```



HTML BÁSICO



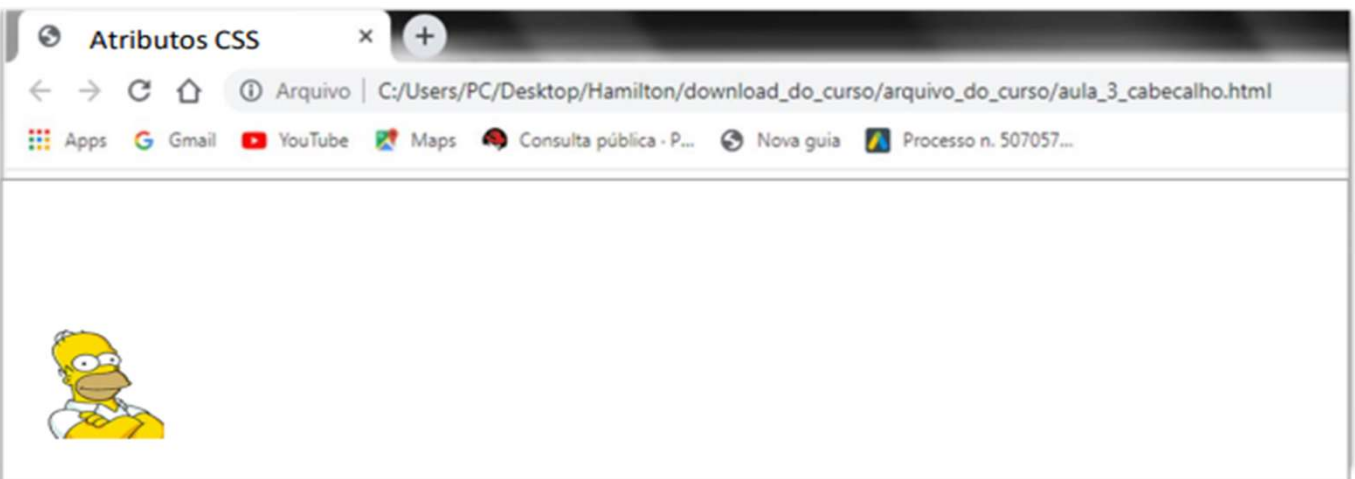
- **Margin** - É a margem do elemento, ou seja, o espaçamento externo do elemento.

Algumas formas de escrita:

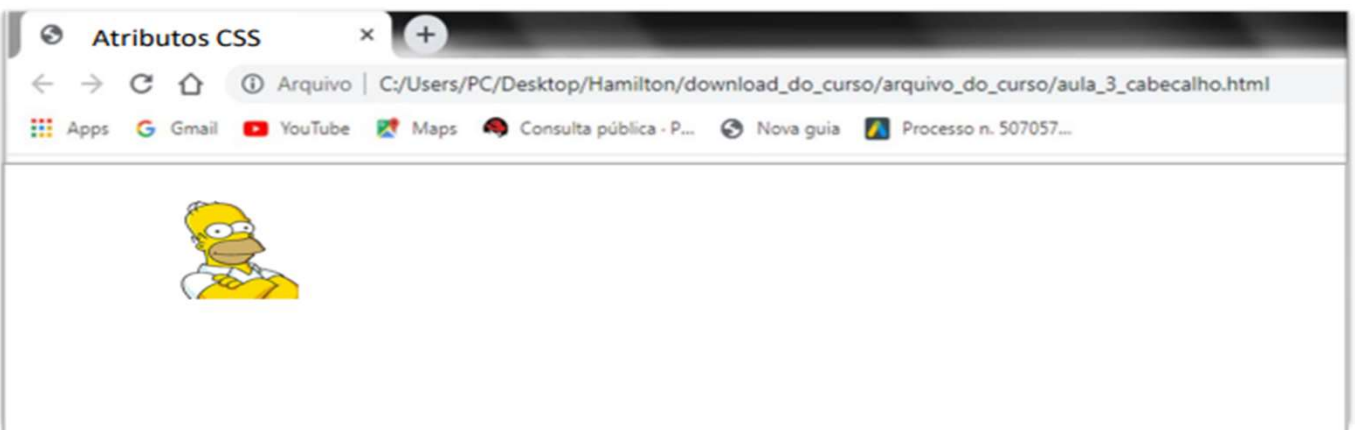
- **Margin-Top** - Define a margem superior do elemento.
- **Margin-Right** - Define a margem direita do elemento.
- **Margin-Botton** - Define a margem inferior do elemento.
- **Margin-left** - Define a margem esquerda do elemento.

Exemplo:

```
img{  
margin-top:50px;  
}
```



```
img{  
margin-left:50px;  
}
```



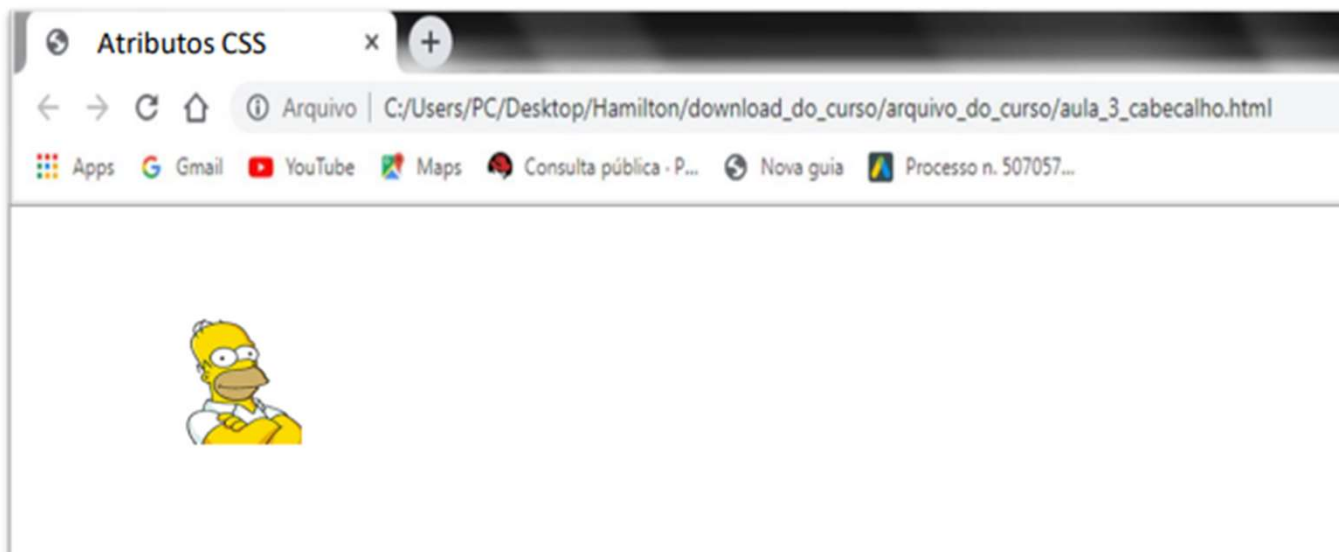
HTML BÁSICO



Nota: Pode-se atribuir mais de um elemento ao mesmo atributo. Por padrão a sequencia dos elementos é top, left, botton e right.

Exemplo:

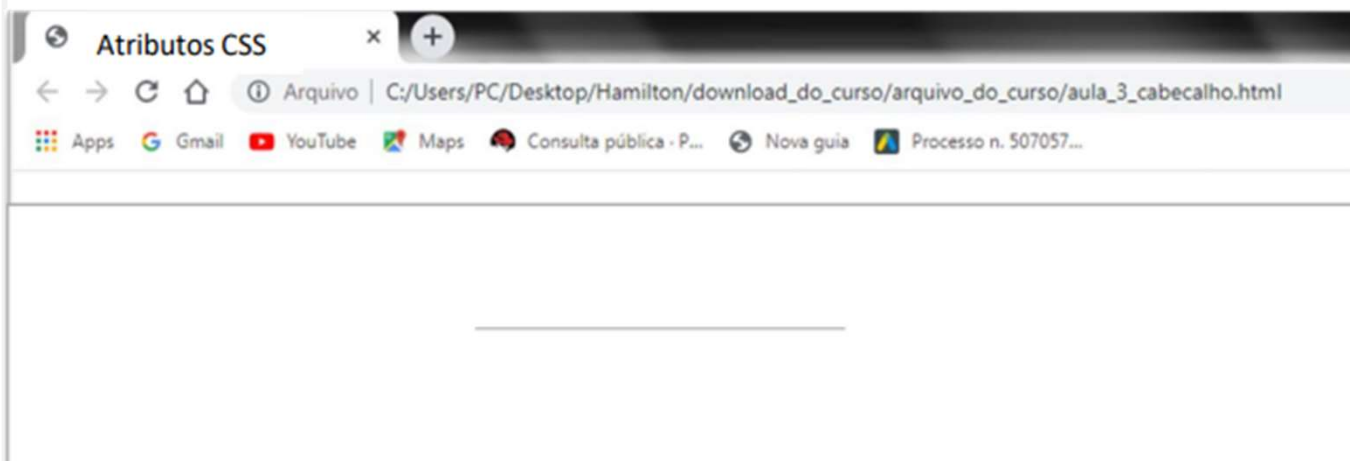
```
img{  
margin:50px 80px;  
}
```



- **Margin centralizada** – Deixa a margem centralizada no meio da página.

Exemplo:

```
hr{  
margin:auto; width:300px;  
}
```



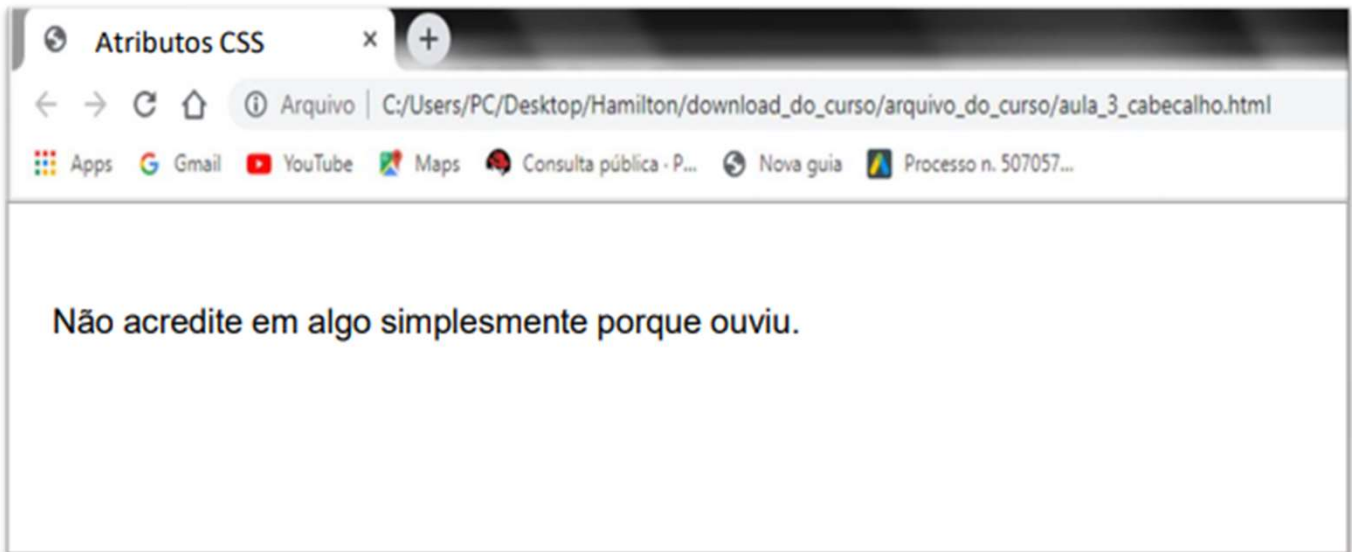
HTML BÁSICO



- **Padding** - Espaço em torno de um elemento em todos os lados. Top left, bottom e right

Exemplo:

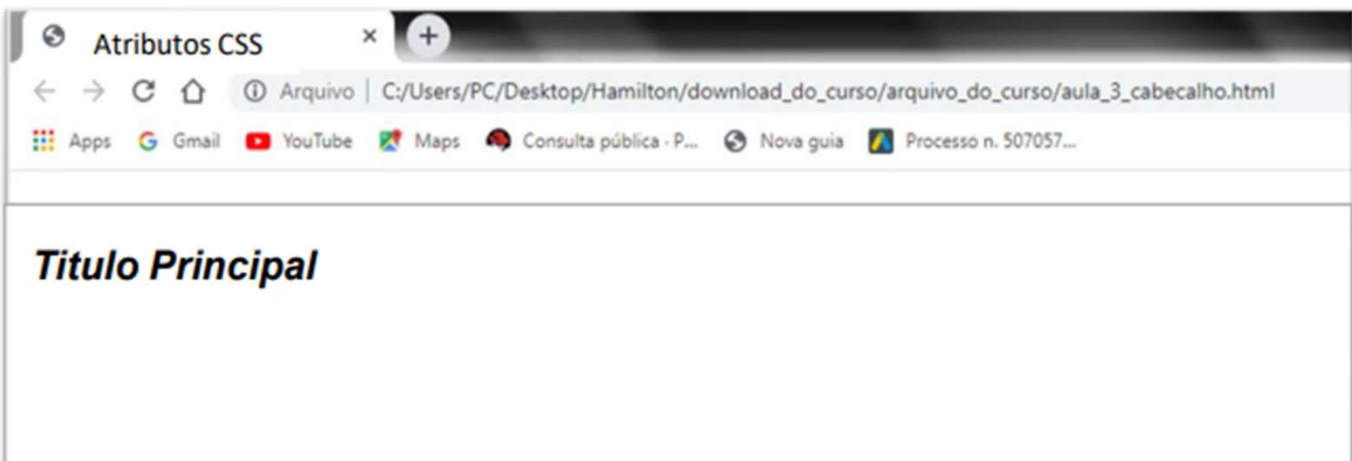
```
p {  
padding:50px 10px  
}
```



- **Font-style** - Fonte itálico.

Exemplo:

```
h3 {  
font-style: italic;  
}
```



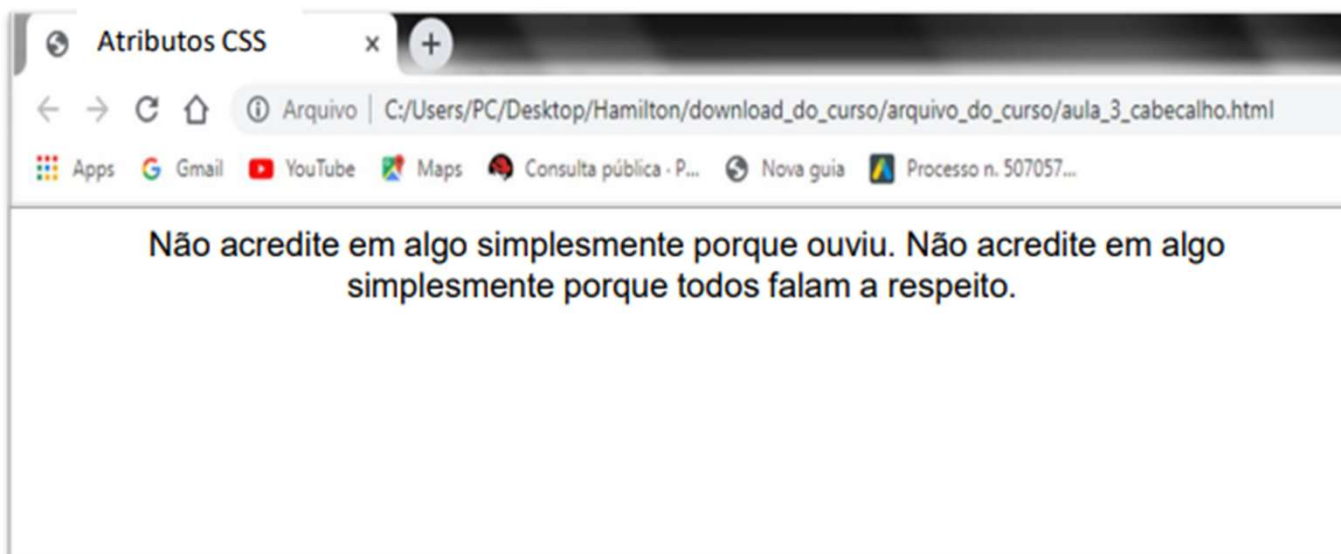
HTML BÁSICO



- Text-align - Alinhamento do texto: (center, left e right).

Exemplo:

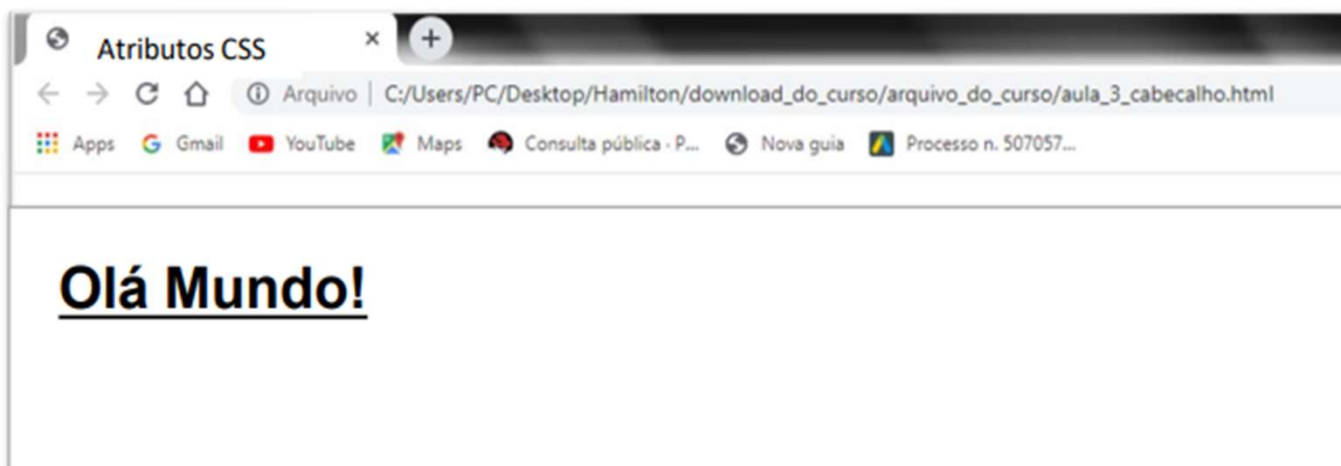
```
p {  
text-align:center;  
}
```



- Text-decoration - Sublinhado, sobrelinhado ou riscado.

Exemplo:

```
h1{  
text-decoration:underline;  
}
```



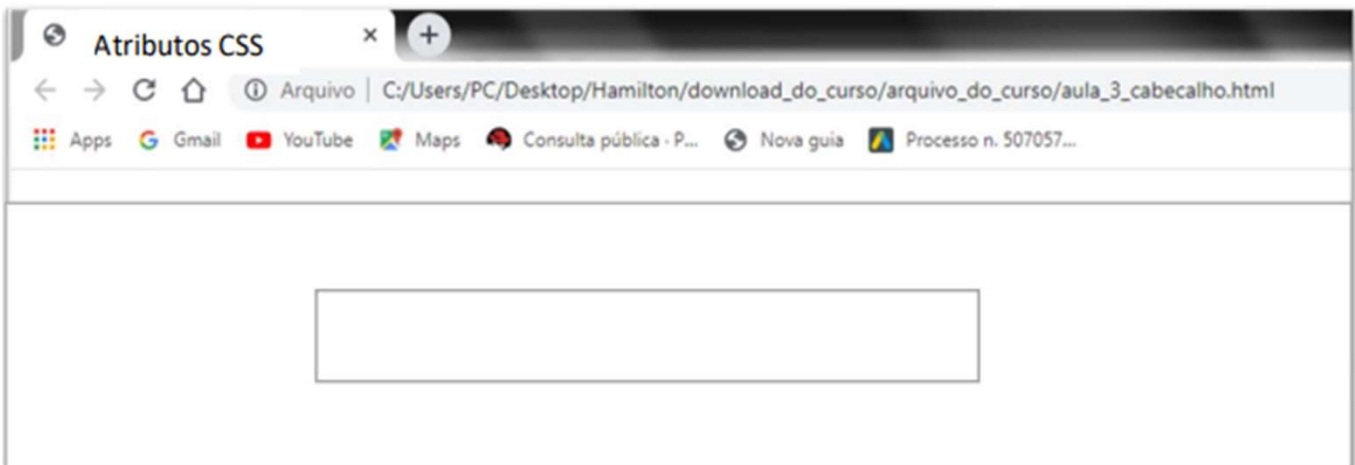
HTML BÁSICO



- **Width** - Largura do elemento.

Exemplo:

```
hr {  
width:250px;  
}
```



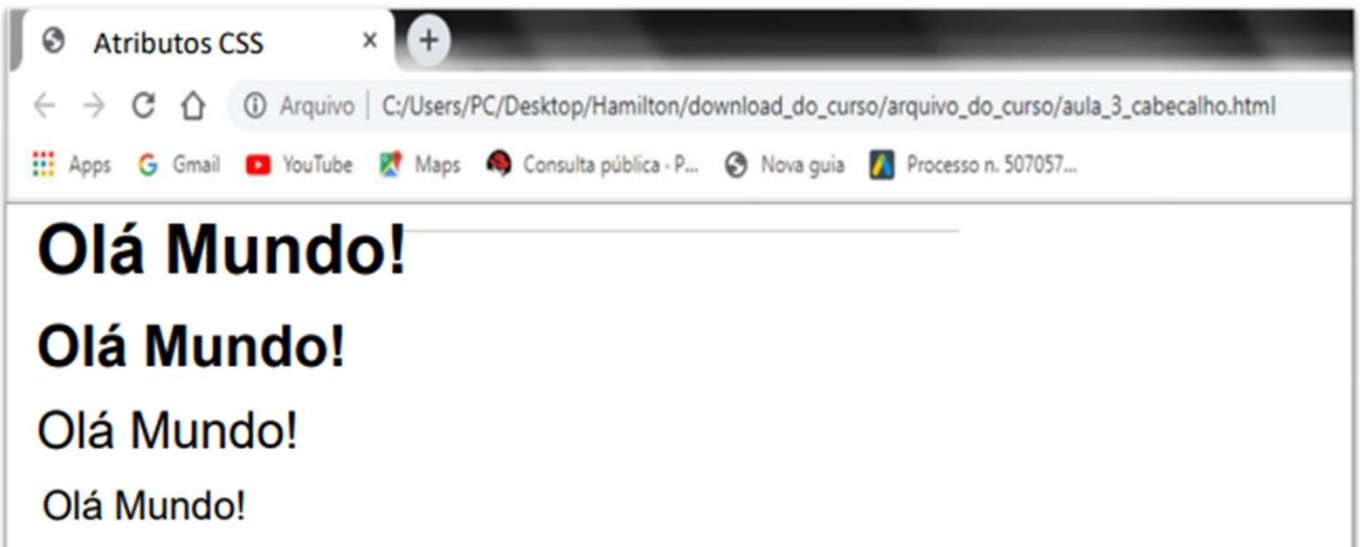
HTML BÁSICO



- **Font-weight** - Peso da fonte de claro a negrito.

Exemplo:

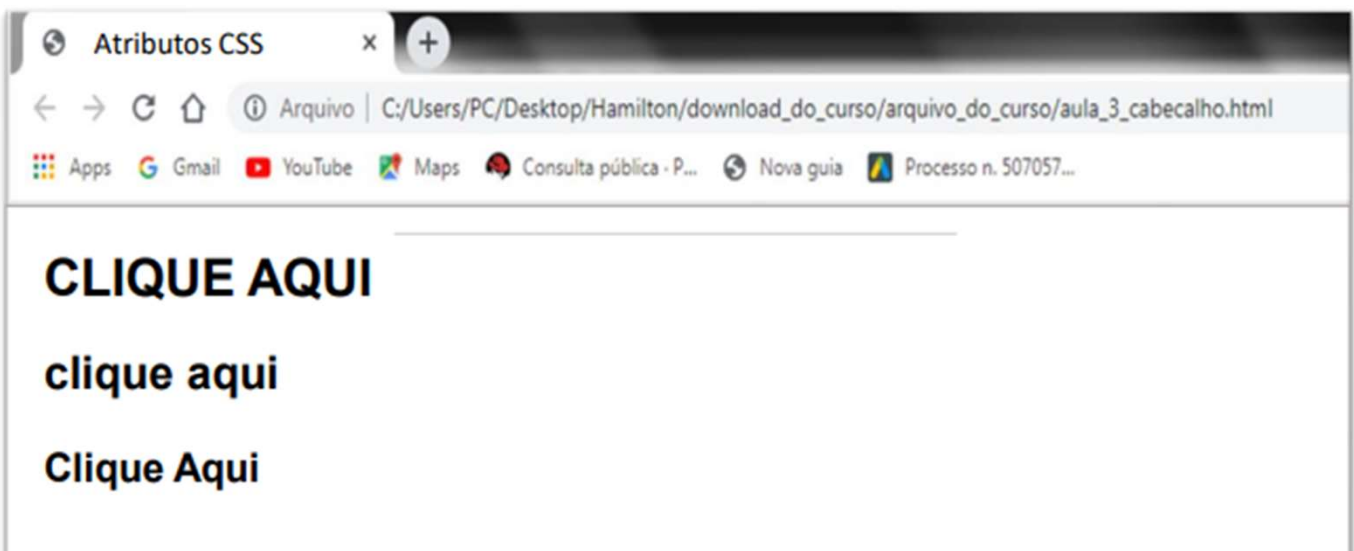
```
h1{
font-weight:bold;
}
h2 {
font-weight:bolder;
}
h3 {
font-weight:900;
}
h4 {
font-weight:500;
}
```



- **Text-transform** - Transformação para todas maiúsculas, minúsculas ou inicial maiúscula.

Exemplo:

```
h1{
text-transform: uppercase;
}
h2 {
text-transform: lowercase;
}
h3{
text-transform: capitalize;
}
```



Esses são os atributos mais usados em **HTML** e **CSS**, na próxima apostila, **CSS3** para Iniciantes, você vai ver todos os atributos disponíveis e atualizados segundo o padrão da **W3C**.

Div

- O que é DIV e para que serve?

O nome **div** vem de divisão, e esta tag vai lhe dar o simples poder de dividir qualquer trecho de seu código. Isso mesmo, você pode criar um bloco, uma divisão, e dentro deste bloco ter uma imagem, links, textos e o que mais você quiser.

- E qual a vantagem disso?

Você vai poder aplicar o **CSS** nesse bloco, e tudo que estiver dentro da div, seja imagem, texto, link ou o que mais for, vai receber aquelas regras de estilo. Se você fizer isso usando **CLASS**, por exemplo, vai ter que colocar esse seletor em cada tag. É bem comum dizer que a tag **<div>** é um container para armazenar diversos elementos. As divs são importantíssimas para estruturar, criar uma ordem lógica e organizar um site. Você cria uma div para o cabeçalho, outra para o conteúdo, outra div para os menus e uma última para o rodapé, então usa o CSS para estilizar cada uma dessas divs separadamente. Você também poderá criar outras divisões dentro destas divs. Vamos agora aprender a usar a tag div em HTML e CSS.

Exemplo:

```
<!doctype html>
<html lang = "pt-br">
<head>
<meta charset = "UTF-8">
<title>DIV</title>
<style>
div {
border:1px solid red;
}
</style>
</head>
<body>
<div>
<h1>Aqui fica todo o conjunto de elementos utilizando o HTML</h1>
</div>
</body>
</html>
```


Ao salvar o documento veja como fica:



- A sintaxe da tag <div>

Todo, qualquer e quantos elementos do HTML você queira colocar, obviamente, para identificar as mais variadas divs que podemos usar para estruturar um site, devemos usar os seletores **ID (IDENTIFICADOR)** ou **CLASS (CLASSES)**. Todo esse conteúdo ou seja, o conjunto de elementos utilizando o **HTML** deve estar organizado dentro de uma div única a qual por padrão nomeamos de **Container**.

- O que são Classes?

As classes são uma forma de identificar um grupo de elementos. Através delas, pode-se atribuir formatação a vários elementos de uma vez.

Exemplo:

Código HTML:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Class</title>
<meta charset="utf-8">
<link rel="stylesheet" type/text="css"
href="css/estilo_class.css"/>
</head>
```

HTML BÁSICO



```
<body>
<div class="classe1">Div 1</div>
<div class="classe1">Div 2</div>
<div class="classe1">Div 3</div>
<div class="classe1">Div 4</div>
<div class="classe1">Div 5</div>
</body>
</html>
```

Folha de Estilo CSS:

```
.classe1 {
background:blue;
width:50px;
height:50px;
margin:10px auto;
display:inline-block;
text-align:center;
line-height:50px;
color:white;
}
```

Ao salvar o documento veja como fica:



Sendo assim, todas as 'divs' que estiverem com a classe "**classe1**" estarão com um **background azul(blue)**.

- O que são IDS?

As ids são uma forma de identificar um elemento, e devem ser únicas para cada elemento. É como se fossem impressões digitais de nossos dedos ou RG. Através delas, pode-se atribuir formatação a um elemento em especial.

Exemplo:

Código HTML:

Div 1 Div 2 Div 3 Div 4 Div 5

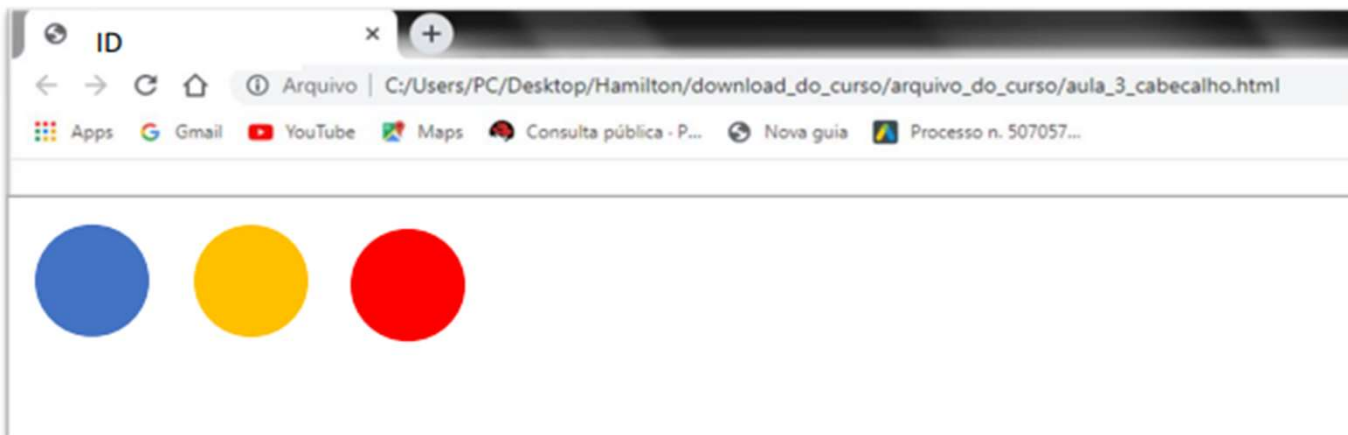
Código HTML:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>ID</title>
<meta charset="utf-8">
</head>
<body>
<div id="idUm"></div>
<div id="idDois"></div>
<div id="idTres"></div>
</body>
</html>
```

Estilo CSS:

```
#idUm {
background: blue; border-radius:100px;
width:50px; height:50px;
display:inline-block; margin:10px;
}
#idDois {
background: yellow; border-radius:100px;
width:50px; height:50px;
display:inline-block; margin:10px;
}
#idTres {
background: red; border-radius:100px;
width:50px; height:50px;
display:inline-block; margin:10px;
}
```

Ao salvar o documento veja como fica:



Então, como mostra o código acima, não podemos repetir uma 'id'. Deve ser especialmente única para cada elemento.

SPAN

O elemento HTML `` é um contêiner genérico em linha para conteúdo fraseado, que não representa nada por natureza. Ele pode ser usado para agrupar elementos para fins de estilo (usando os atributos `class` ou `id`), ou para compartilhar valores de atributos como `lang`. Ele deve ser usado somente quando nenhum outro elemento semântico for apropriado. O elemento `` é muito parecido com o elemento `<div>`, entretanto `<div>` é um elemento de nível de bloco enquanto `` é um elemento em linha.

Exemplo:

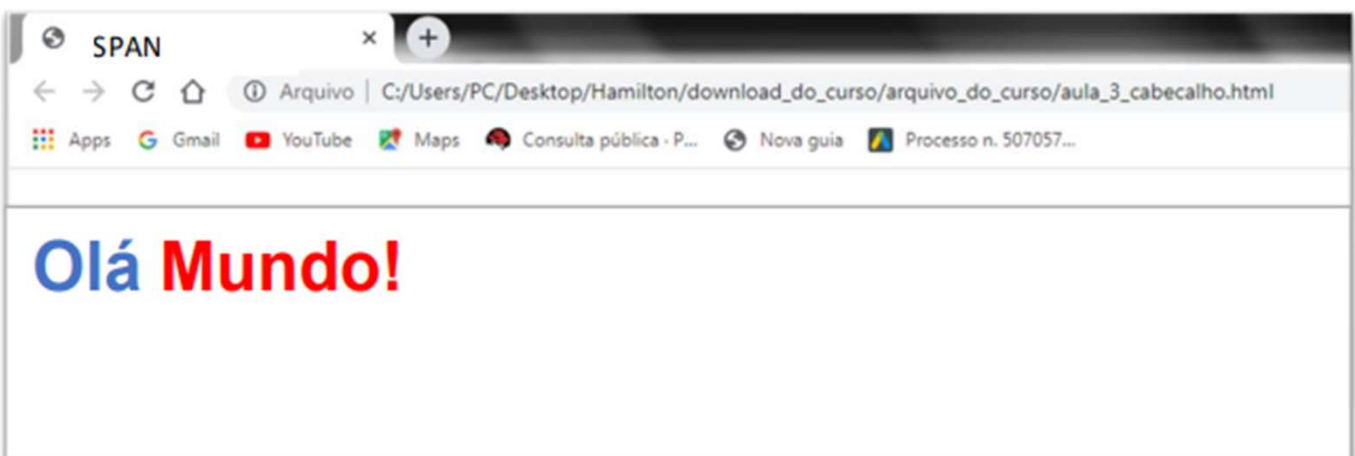
Código HTML:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>SPAN</title>
<meta charset="utf-8">
</head>
<body>
<span class="blue"><h1>Olá</span>
<span class="red">Mundo!</h1></span>
</body>
</html>
```

Estilo CSS:

```
.blue {  
color:blue;  
}  
.red{  
color:red;  
}
```

Ao salvar o documento veja como fica:



Formulários

Agora você vai aprender a criar “formulários eletrônicos” que permite um usuário submeter informações para o provedor de serviço. Existem vários tipos de campos de entrada como:

- Campos de entrada de textos
- Menus de múltipla escolha ou escolha única
- Botões sim ou não
- Botões para submissão ou limpeza de formulários

Cada um destes campos tem funcionalidade própria e você vai aprender em que situação utiliza-los em seus formulários. Para melhor compreender a notação utilizada nos formulários, você primeiro precisa se familiarizar com o **HTML**. Um formulário começa com a marcação **<form>** e termina com **</form>**.

• Elemento Input

O input é o elemento de formulário mais importante e pode ser exibido de várias maneiras, dependendo do atributo type .

Aqui estão alguns exemplos:

Tipo:	Descrição:
<code><input type = "text"></code>	Define um campo de entrada de texto de uma linha
<code><input type = "radio"></code>	Define um botão de opção (para selecionar uma das várias opções)
<code><input type = "submit"></code>	Define um botão de envio (para enviar o formulário)

• Entrada de Texto

`<input type="text">` - define um campo de entrada de uma linha para entrada de texto.

Exemplo:

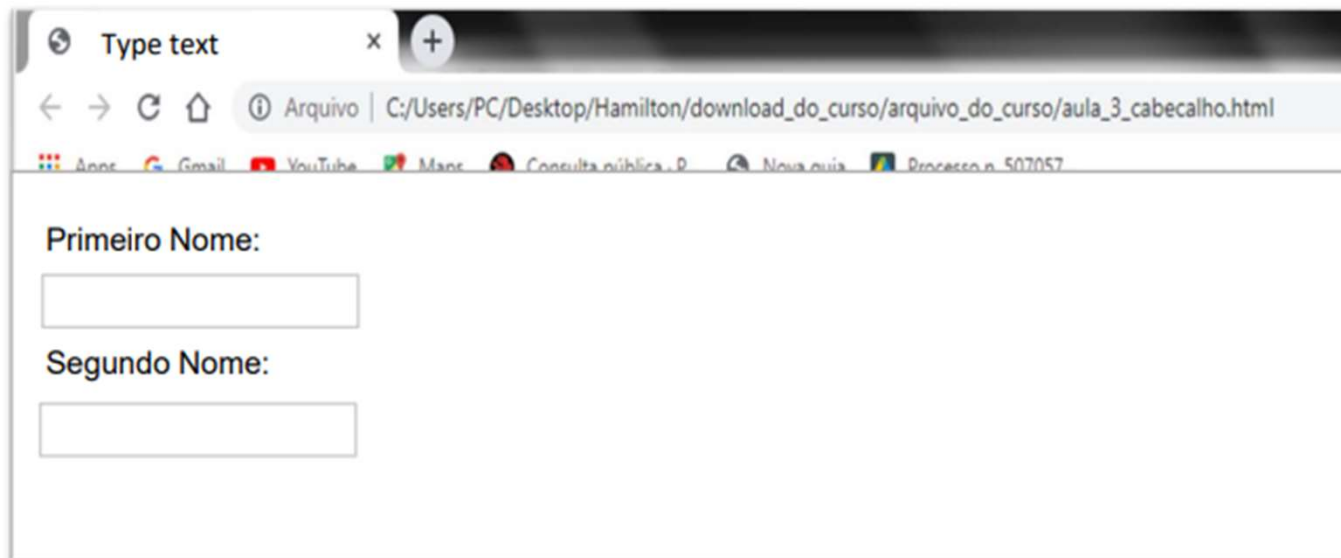
Código HTML:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Type text</title>
<meta charset="utf-8">
</head>
<body>
<form>
Primeiro Nome:<br/>
<input type="text" name="primeiroNome"><br/>
Segundo Nome<br/>
<input type="text" name="segundoNome">
</form>
</body>
</html>
```

HTML BÁSICO



Ao salvar o documento veja como fica:



Nota: O formulário em si não é visível. Observe também que a largura padrão de um campo de texto é de 20 caracteres.

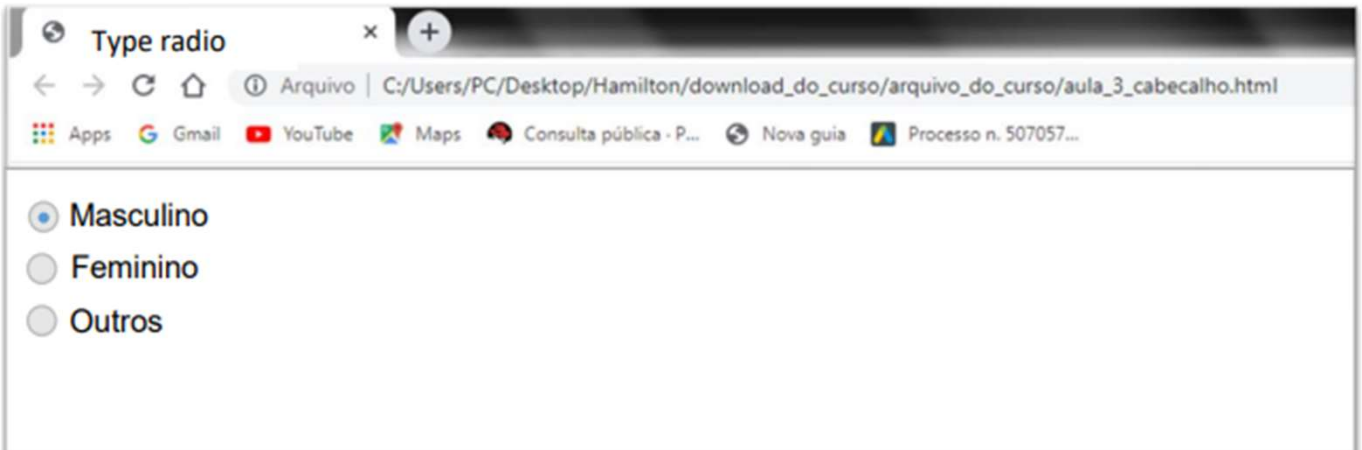
• Entrada de Botão de Radio

`<input type="radio">` - define um botão de opção. Os botões de opção permitem que um usuário selecione uma de um número limitado de opções.

Exemplo:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Type radio</title>
<meta charset="utf-8">
</head>
<body>
<form>
<input type="radio" checked> Masculino<br/>
<input type="radio"> Feminino<br/>
<input type="radio"> Outros
</form>
</body>
</html>
```

Ao salvar o documento veja como fica:



• Botão Enviar

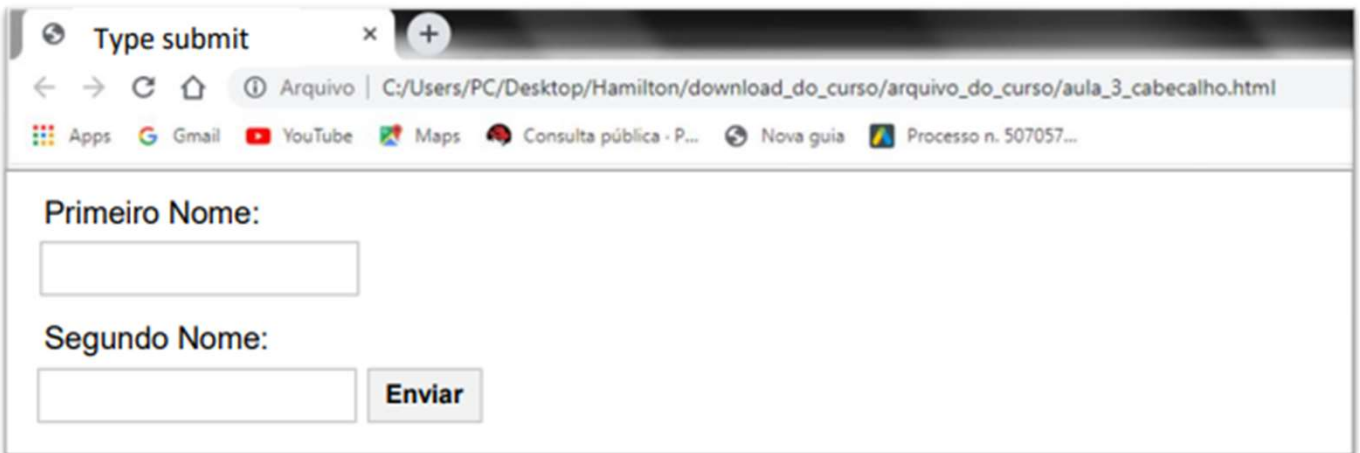
`<input type="submit">` - Define um botão para enviar os dados do formulário para um manipulador de formulários .

O manipulador de formulários geralmente é uma página do servidor com um script para processar dados de entrada. O manipulador de formulários é especificado no atributo de ação do formulário

Exemplo:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>type submit</title>
<meta charset="utf-8">
</head>
<body>
<form>
Primeiro Nome:<br/>
<input type="text" name="primeiroNome"><br/>
Segundo Nome<br/>
<input type="text" name="segundoNome">
<input type="submit" name="acao" value="Enviar"/>
</form>
</body>
</html>
```


Ao salvar o documento veja como fica:



The screenshot shows a web browser window titled "Type submit". The address bar displays the file path: "Arquivo | C:/Users/PC/Desktop/Hamilton/download_do_curso/arquivo_do_curso/aula_3_cabecalho.html". The browser's taskbar shows several open applications: Apps, Gmail, YouTube, Maps, Consulta pública · P..., Nova guia, and Processo n. 507057... The main content area of the browser contains a simple form with two text input fields. The first field is labeled "Primeiro Nome:" and the second is labeled "Segundo Nome:". To the right of the second input field is a button labeled "Enviar".

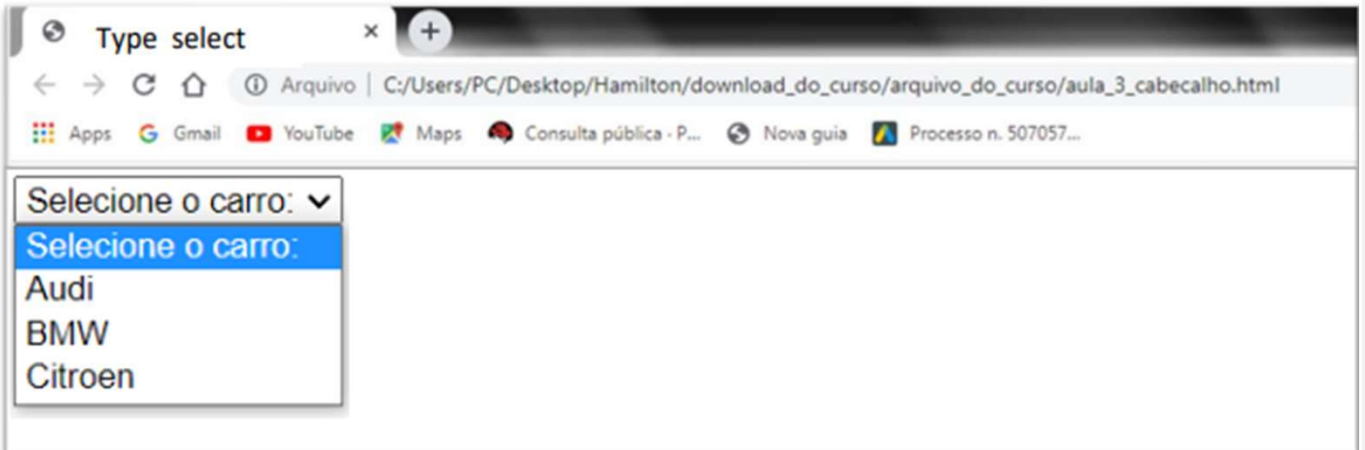
• Seleção dos Elementos

`<input type="select">` - representa um controle que apresenta um menu de opções. As opções dentro do menu são representadas pelo elemento `<option>` que podem ser agrupados por elementos . As opções podem ser pré-selecionadas para o usuário.

Exemplo:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Type select</title>
<meta charset="utf-8">
</head>
<body>
<form>
<select>
<option value="0">Selecione o carro:</option>
<option value="1">Audi</option>
<option value="2">BMW</option>
<option value="3">Citroen</option>
</select>
</form>
</body>
</html>
```

Ao salvar o documento veja como fica:



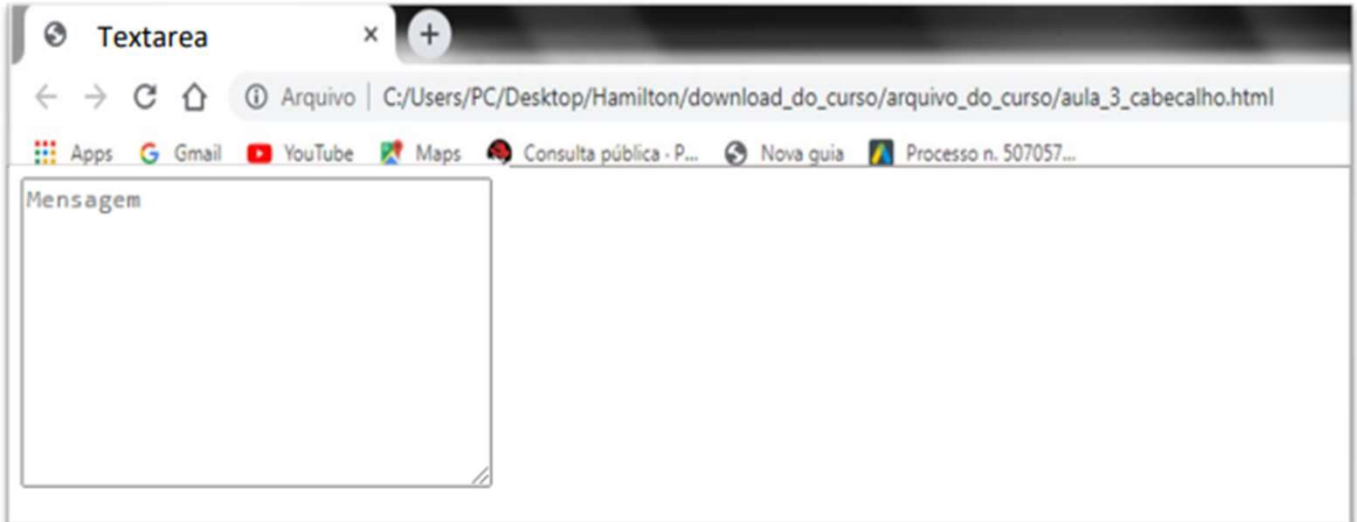
• Mensagem

`<textarea></textarea>` - define um campo de entrada de várias linhas (uma área de texto).

Exemplo:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Textarea</title>
<meta charset="utf-8">
</head>
<body>
<form>
<textarea rows="10" cols="30" placeholder="Mensagem"></textarea>
</form>
</body>
</html>
```

Ao salvar o documento veja como fica:

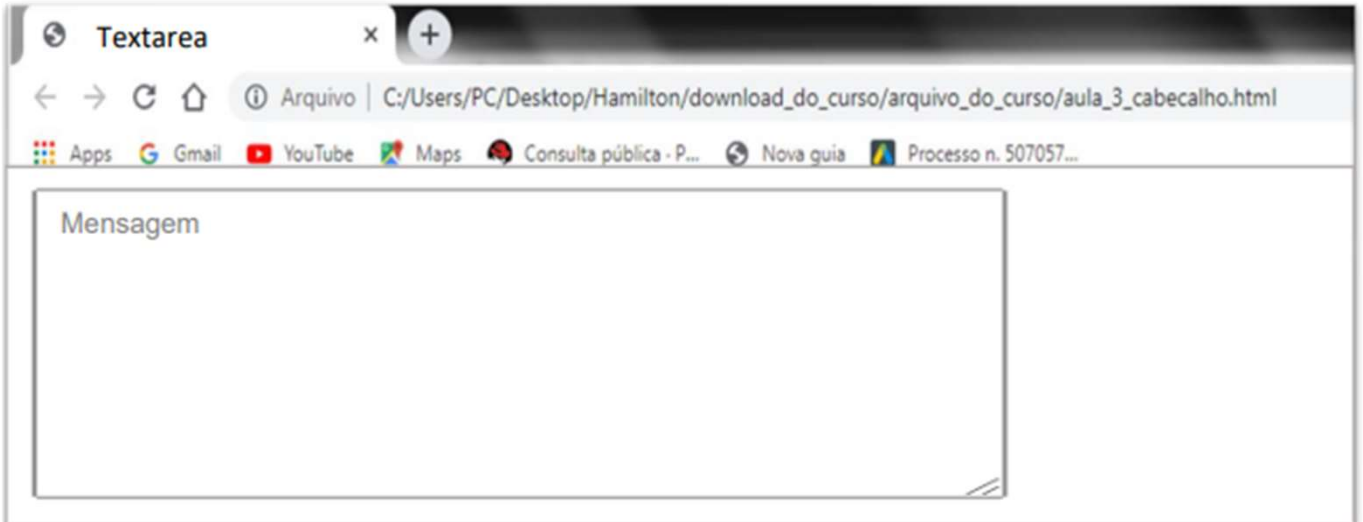


Você também pode definir o tamanho da área de texto usando **CSS**:

Exemplo

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Textarea</title>
<meta charset="utf-8">
</head>
<body>
<form>
<textarea style="width:600px; height:200px;" placeholder="Mensagem"></textarea>
</form>
</body>
</html>
```

Ao salvar o documento veja como fica:



- **Botão de Clicar**

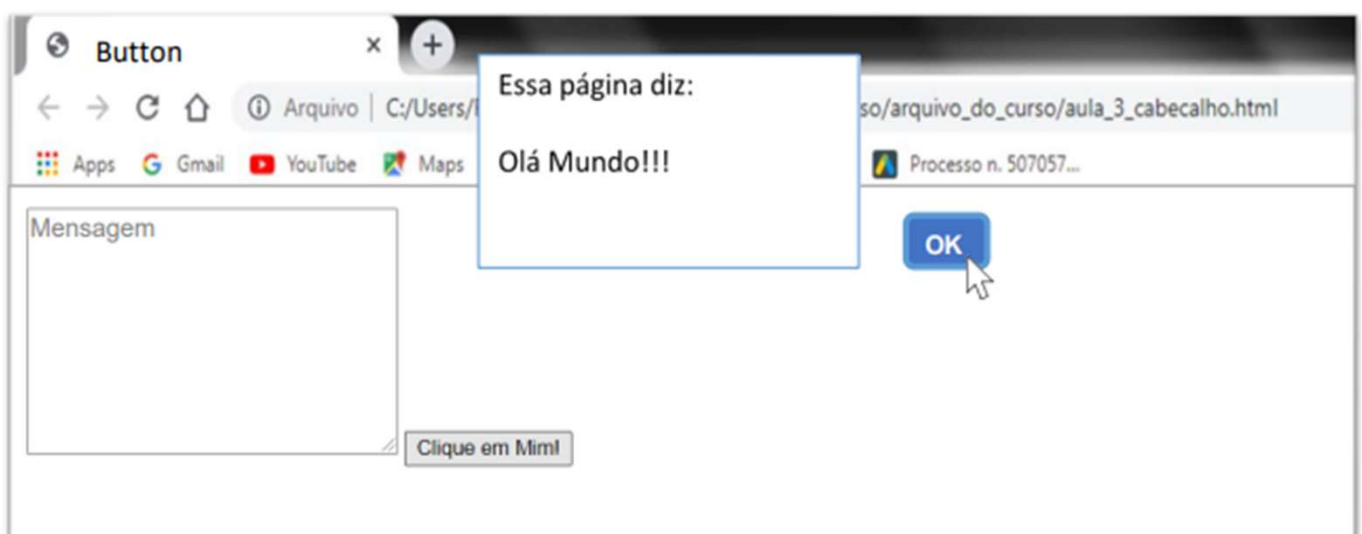
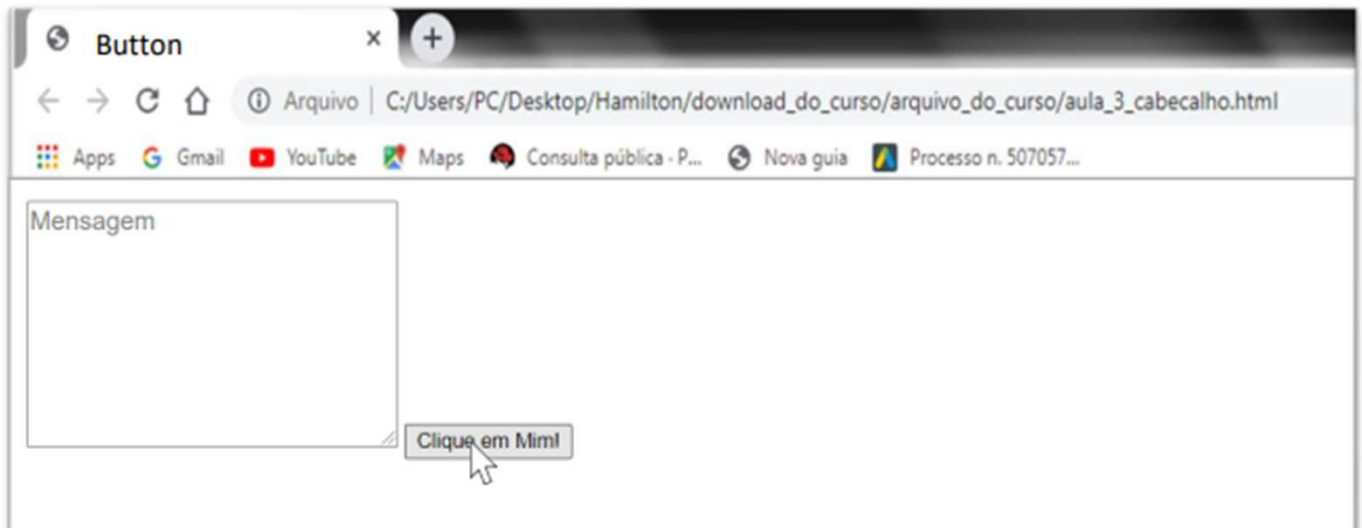
`<button></button>` - elemento define um botão de click.

Exemplo:

```
<!doctype html>
<html lang="pt-br">
<head>
<title>Textarea</title>
<meta charset="utf-8">
</head>
<body>
<form>
<textarea name="mensagem" rows="10" cols="30"
placeholder="Mensagem"></textarea>
<button type="button" onclick="alert ('Olá Mundo!')">
Clique em Mim!
</button>
</form>
</body>
</html>
```

HTML BÁSICO

Ao salvar o documento veja como fica:



Layout com Div

Vamos ver na prática como criar um layout com quatro divs: Container, Topo, Corpo e Rodapé.

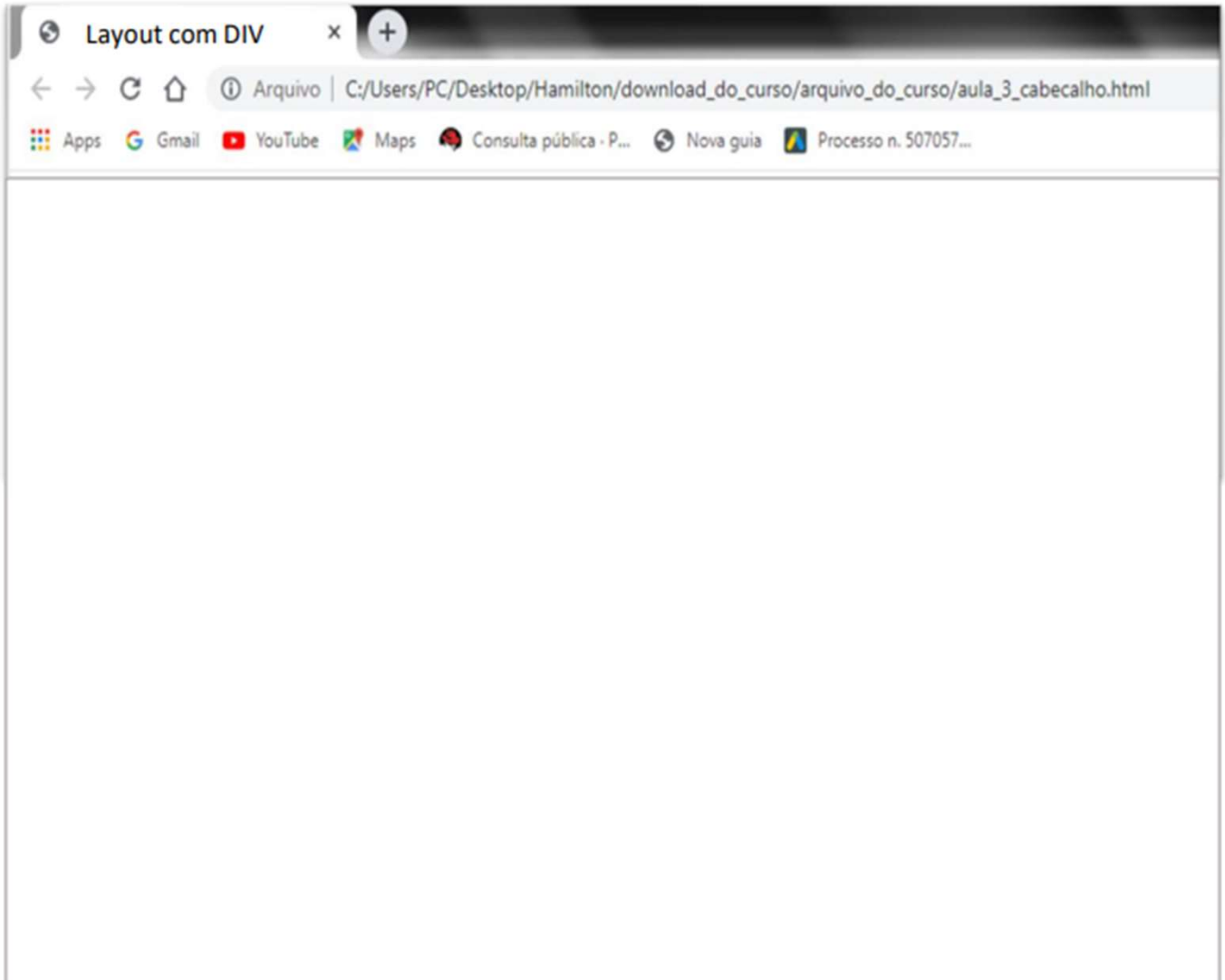
A estrutura vai ficar assim:

```
<!doctype html>
<html lang =”pt-br”>
<head>
<meta charset =”UTF-8”>
<title>Layout com Divs</title>
<link rel=”stylesheet” type=”text/css” href=”css/estilo.css”/>
</head>
<body>
<div id=”container”>
<div id=”topo”>
Topo
</div>
<div id=”corpo”>
Corpo
</div>
<div id=”rodape”>
Rodapé
</div>
</div>
</body>
</html>
```

HTML BÁSICO



Ao ver o resultado no Navegador/Browser vai ficar assim:



Isso porque não estilizamos o documento ainda utilizando o atributos **CSS**.

Como estamos usando **CSS externo**, digite assim na folha de estilo:

```
#container{
width:840px;
height:704px;
border:1px solid gray;
margin:auto;
}
#topo{
width:800px;
height:150px;
border:1px solid gray;
margin:auto;
padding:20px;
}
#corpo{
width:800px;
height:330px;
border:1px solid gray;
margin:auto;
padding:20px;
}
#rodape{
width:800px;
height:100px;
border:1px solid gray;
margin:auto;
padding:20px;
}
```

Antes de visualizar no navegador/browser, veja um detalhe no código fonte da página, observe bem no elemento `#container`.

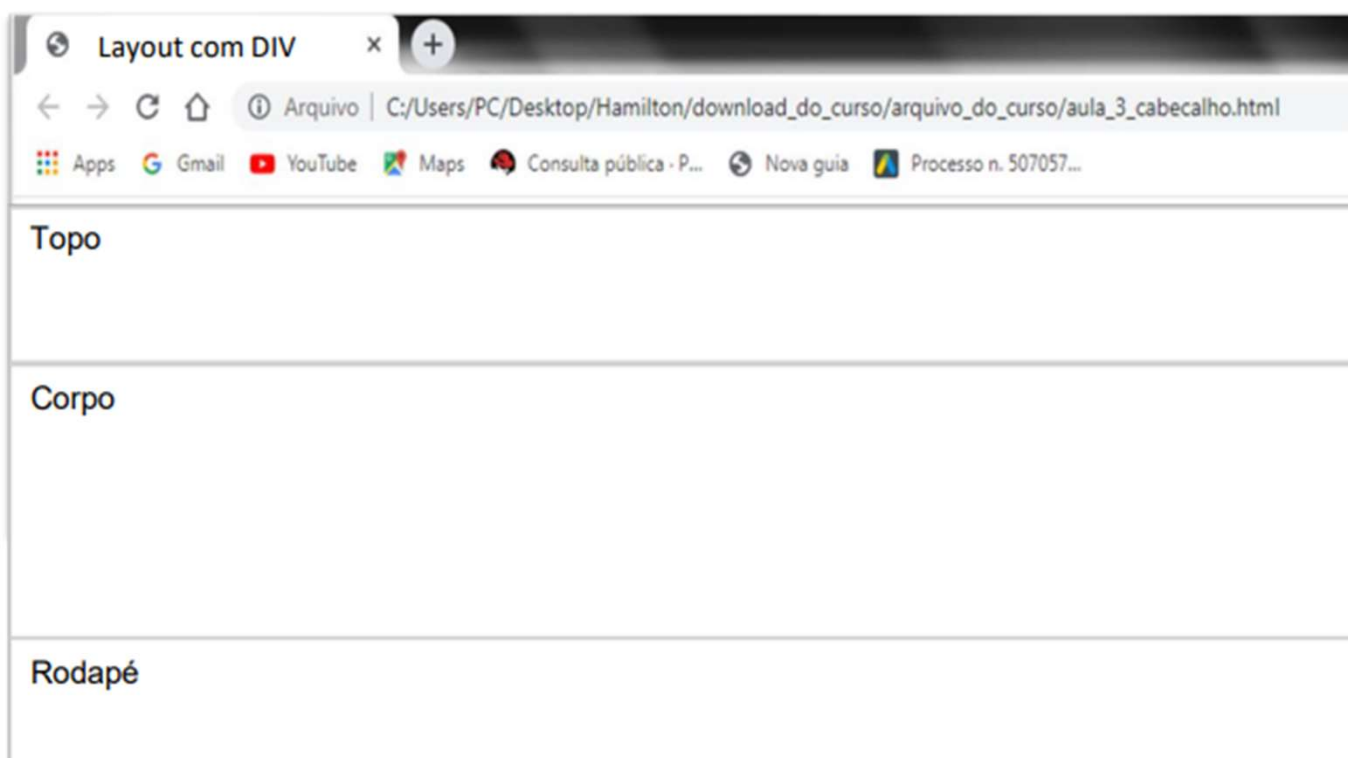
```
#container{
width:840px;
border: 1px solid gray;
margin:auto;
}
```


HTML BÁSICO



Perceba que a largura (**width**) está com **840px** e não **800px**, como nos demais elementos, **#corpo** e **#rodape**. Isso ocorre porque cada vez que acrescentamos o elemento padding em uma div, a largura é alterada devido a quantidade de espaçamento interno do elemento padding adicionado.

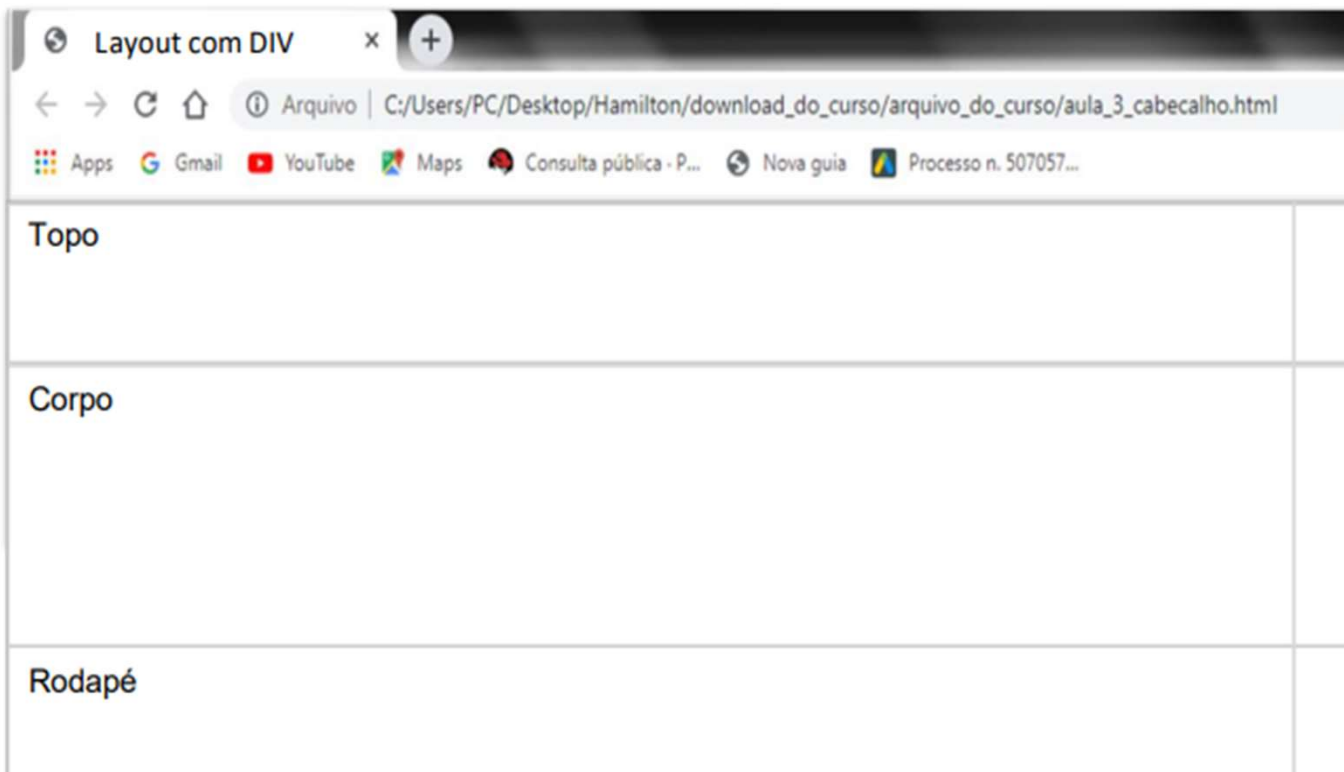
Salve o documento para ver o resultado:



Agora coloque 800px na largura do elemento **#container**, depois salve para ver como fica.

```
#container{
width:800px;
height:580px;
border: 1px solid gray;
margin:auto;
}
```

Salvando o documento e abrindo no navegador vai ficar assim:



Como estamos vendo, a largura da **div #container** sendo **800px**, a **div #topo** sofre alteração nas margens laterais, pois possui um padding de **20px**, ou seja **10 pixels** a mais

de espaçamento interno em cada margem lateral. Portanto as demais divs **#corpo** e **#rodape** que também possuem **padding**, também sofrem alteração em suas margens laterais.

Para resolvermos esse problema de uma maneira mais simples e prática vamos utilizar o atributo **box-sizing**. O tamanho agora levará em conta até a borda ou seja, o width será a soma do conteúdo com a borda e o padding. Portanto não importa a quantidade de espaçamento interno que se aplica através do padding. A sintaxe para utilizar o atributo **box-sizing** é: **box-sizing:border-box**. Sendo assim div nunca sofrerá alteração em suas bordas, ou seja, a div se ajustará automaticamente entre a quantidade de espaçamento e a largura. Na próxima etapa do curso comentaremos detalhadamente sobre o atributo **box_sizing**.

Veja este outro exemplo:

A estrutura HTML vai ficar assim:

```
<!doctype html>
<html lang = "pt-br">
<head
<meta charset = "UTF-8">
<link rel="stylesheet" type="text/css"
href="css/estilo-div.css"/>
<title>Layouts com Divs</title>
</head>
<body>
<div id="container">
<div id="topo">
Topo
</div>
<div id="corpo">
<div id="texto">
Texto
</div>
<div id="img">
<img src=img/fundo-02.png width=400px
height=330px/>
</div>
</div>
<div id="rodape">
Roadapé
</div>
</div>
</body>
</html>
```

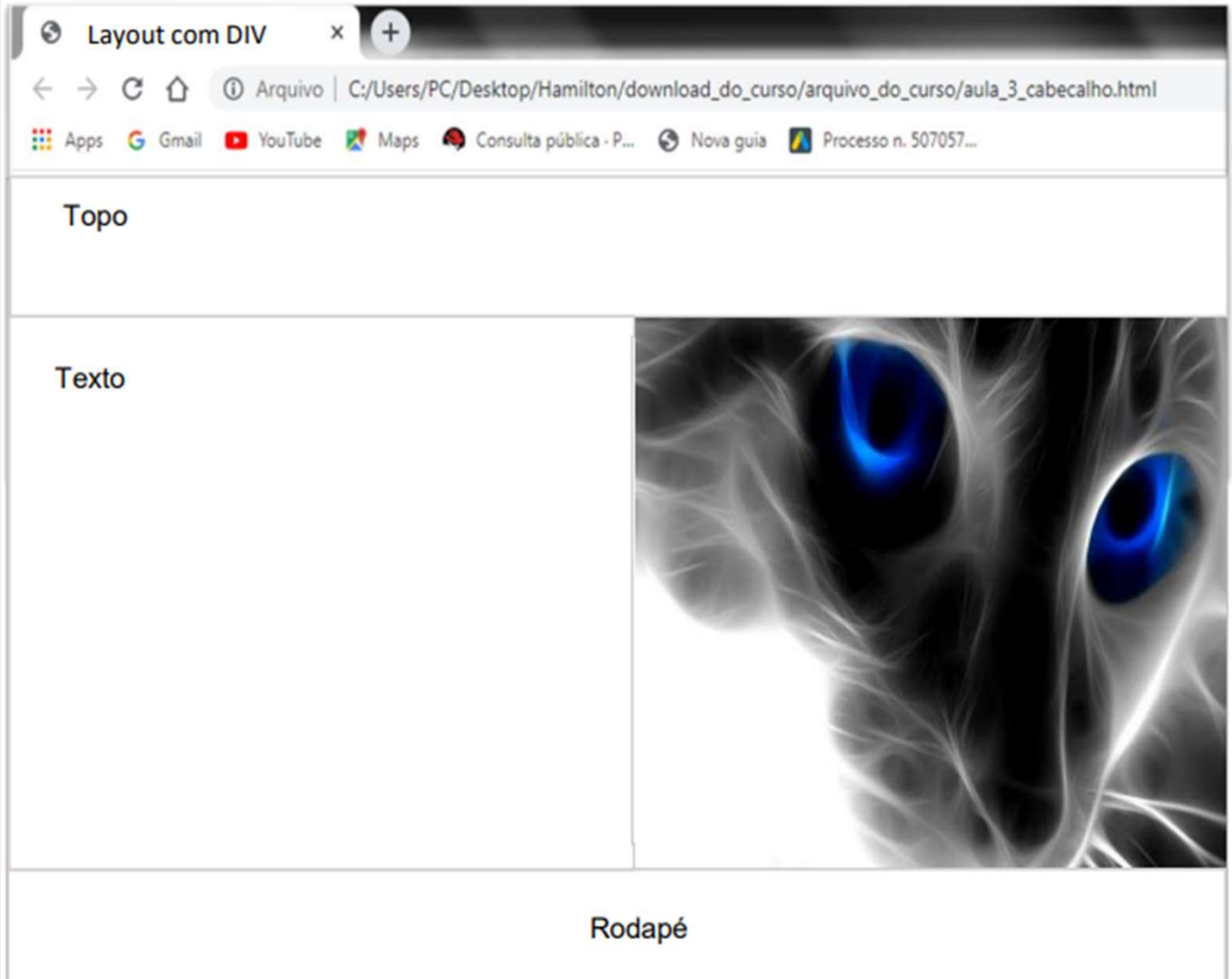
A estrutura CSS vai ficar assim:

```
#container{
width:800px;
border: 1px solid gray;
margin:auto;
}
#topo{
width:800px;
height:150px;
border:1px solid gray;
margin:auto;
padding:20px;
box-sizing: border-box;
}
#corpo{
width:800px;
height:330px;
border:1px solid gray;
margin:auto;
}
#texto{
width:400px;
height:330px;
border:1px solid gray;
float:left;
padding:10px;
box-sizing: border-box;
}
#img{
width:398px;
height:330px;
border:1px solid gray;
float:right;
}
#rodape{
width:800px;
height:100px;
border:1px solid gray;
line-height:100px;
text-align:center;
}
```

HTML BÁSICO



Salvando o documento e abrindo no navegador vai ficar assim:



Chegamos ao final desta etapa do curso. Estou muito feliz por ter chegado até a etapa final deste módulo. Terminar o momento, encontrar o final da jornada em cada passo do caminho, viver o maior número de boas horas, é sabedoria. Acreditar em si mesmo leva a um destino infinito. Agradeço mais uma vez pelo ótimo desempenho do curso e até a próxima! Te aguardo na próxima etapa.

Hamilton Staino

Porque é importante aprender HTML?

Segundo o site Mashable, ter conhecimentos básicos de **HTML** pode fazer uma grande diferença na sua carreira, ajudando você a crescer profissionalmente e oferecendo independência aos pequenos empresários.

O que exatamente o HTML faz?

Hyper Text Markup Language, ou **HTML**, é utilizado para definir o local de cada item em seu navegador. Através dele você consegue posicionar o cabeçalho, os menus, os botões, entre outros detalhes, como construir uma plataforma bem planejada e firme, que seja de fácil uso. Também você poderá criar uma interface profissional e compatível com a identidade da sua empresa.

Como está o Mercado para quem aprende HTML?

Se você já é um especialista em **Photoshop**, com certeza subirá de nível quando souber um pouco de **HTML**. O mercado tem procurado **designers** que saibam **transformar seus projetos em sites reais**, e você já saiu na frente. Conquiste seu espaço no mercado e ganhe mais, estas são só algumas das situações em que o conhecimento de programação fará diferença no seu dia a dia.



HTML