



**HTML5**  
**para iniciantes**

## SUMARIO

1. Introdução
2. O que precisamos para desenvolver na linguagem HTML?
3. Vamos iniciar o desenvolvimento?
  - 3.1. Exercícios
4. Criação de textos
  - 4.1. Exercícios
5. Atributos
6. Imagens
  - 6.1. Exercícios
7. Links
  - 7.1. Exercícios
8. Botões
  - 8.1. Exercícios
9. Listas
  - 9.1. Listas ordenadas
  - 9.2. Listas não ordenadas
  - 9.3. Exercícios
10. Criação de formulários
11. Atributos seletores
  - 10.1. Ids
  - 10.2. Classes
12. Iframes
  - 11.1. Exercícios
13. Tabelas
  - 12.1. Exercícios
14. Introduzindo css
15. Seletores CSS
16. Propriedades CSS de formatação de texto
17. Novas tag do HTML5
18. Edição de containers com CSS
19. Hover CSS
20. Animações com KeyFrames
21. Comentarios
22. Algumas propriedades CSS
23. Algumas tags do HTML5
24. Conclusão



## 1. INTRODUÇÃO

Neste E-book trataremos sobre uma das linguagens que comandam o mercado trabalho, onde se é requisito mínimo o domínio básico sobre esta linguagem para entrar no mercado de desenvolvimento web.

Assim, iniciamos com a sigla HTML que significa *HyperText Markup Language* (Linguagem de marcação de hipertexto), ou seja, ela está voltada, no desenvolvimento web, para criação de layouts e interações com usuários nos navegadores.

A linguagem HTML por ser uma linguagem de marcação de hipertexto, ela não se caracteriza com uma linguagem de programação, somente como uma linguagem de marcação, porque ela consegue executar algoritmos, pois ela é voltada para o desenvolvimento de layouts.

Como mencionado anteriormente, por ser uma linguagem de marcação de hipertexto, e ser a responsável por montar o layout das páginas webs nos navegadores. Ela funciona diretamente em seu navegador.

Você pode observar o poder desta linguagem, quando você visita web sites de grande porte, como comércio online, e visualiza toda distribuição minimamente planejada utilizando a linguagem HTML.

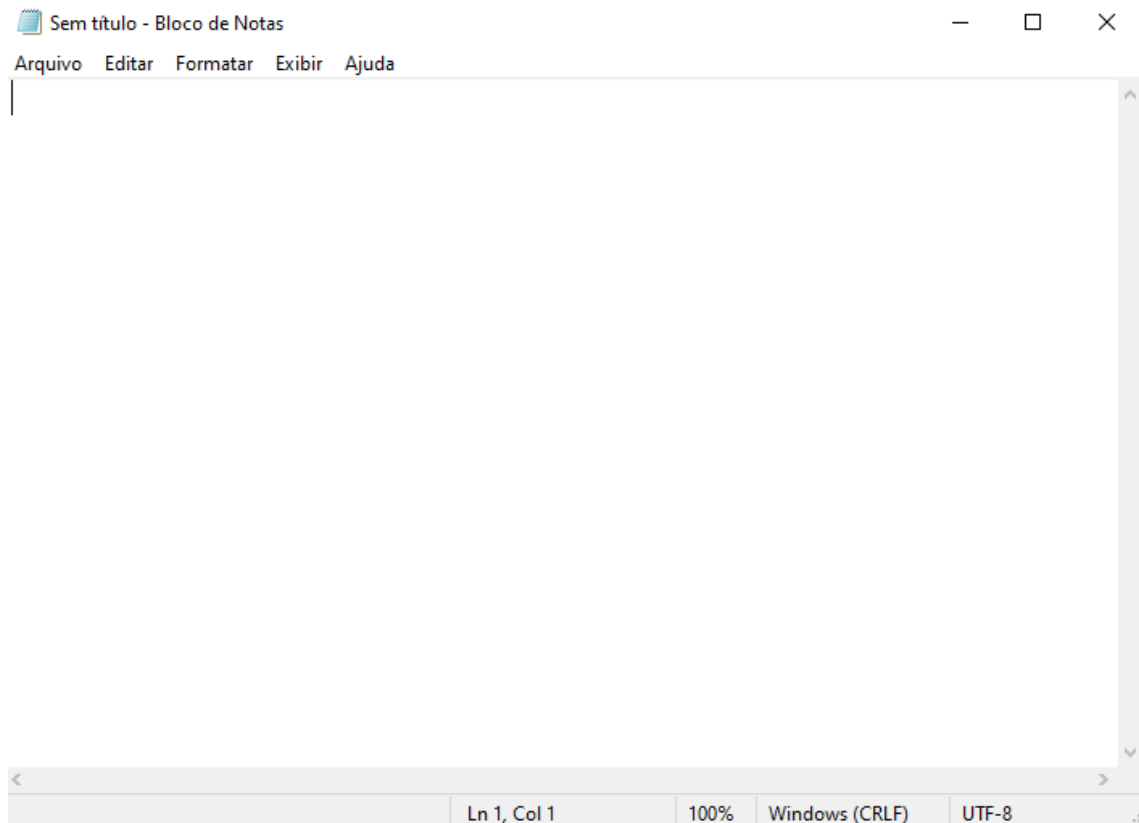
O HTML5 é uma atualização de sua última versão, o HTML4, assim trazendo inúmeras novas possibilidades na construção do web site. Com isto alguns atributos do HTML4 foram alterados, para facilitar o desenvolvimento e torna a interação das páginas web mais agradáveis.

## 2. O que precisamos para desenvolver na linguagem HTML?

Para o desenvolvimento nesta linguagem existe inúmeras ferramentas para o desenvolvimento. Inicialmente não utilizaremos nenhuma destas ferramentas, porque elas são feitas para agilizar o desenvolvimento, ou seja, compreende-se que quem a utiliza já se tem controle sobre a linguagem.

Assim vamos iniciar utilizando um editor de texto comum para nossos primeiros passos com esta nova linguagem. Utilizarei, e recomendo a utilização, do bloco de nota, por ser uma ferramenta gratuita e não estar voltada para o desenvolvimento, fixa-se melhor o conhecimento adquirido.

Esta será a tela que utilizaremos para dar nossos primeiros passos nesta nova linguagem:



### 3. Vamos iniciar o desenvolvimento?

Agora nós vamos “colocar a mão na massa!”, literalmente, agora nós vamos iniciar o desenvolvimento, iniciaremos nossos primeiros web sites!

Vamos iniciar desenvolvendo o que são tags, elas são elementos que seu navegador identifica e assim pode montar alguma coisa para você, dependendo da tag utilizada.

Na maioria das vezes, nós iremos encontrar as tag na seguinte estrutura:

**<nometag>conteudo</nometag>**

Normalmente toda tag que é iniciada deve ser finalizada, como mostrado no exemplo acima.

Então iniciando o a aprendizagem de nossos primeiros elementos, tags, desta linguagem de marcação extremamente poderosa, vamos falar sobre a tag:

**<html></html>**

Esta tag é fundamental, pois ela que informa que a página é uma página HTML, esta tag faz parte da estrutura básica da linguagem.

Assim, dentro desta tag, o conteúdo, podemos colocar outras tag filhas (tags filhas são tags que estão dentro de outras tag como conteúdo). Inicialmente colocamos a tag:

**<head></head>**

Esta tag é voltada para informa ao navegador que este é o cabeçalho, e que deve informar inúmeras configurações da página, sendo uma destas configurações muito importante o título da página que é dito para o navegador pela tag:

**<title>nome da pagina</title>**

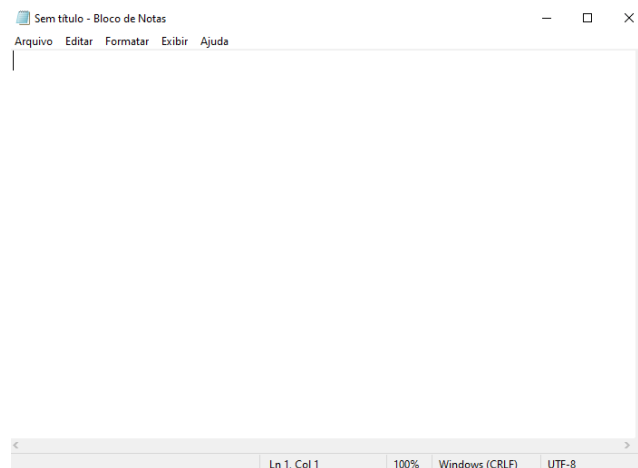
Esta tag fica dentro do head para informar ao navegador o título da pagina.

Logo em seguida devemos aprender sobre a tag:

**<body>conteúdo da pagina</body>**

Neste tag, localiza-se todo conteúdo da página HTML, ela diz ao navegador que ela é o corpo da página que ele deve montar, e todos os comando HTMLs devem estar dentro desta tag.

Assim, agora conhecemos as tag da estrutura do HTML, e iniciaremos nossa primeira página web agora. Vamos iniciar abrindo o bloco de notas:



Dentro do bloco de notas vamos informar que é uma página HTML inserindo a tag html:

```
*Sem título - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<html>

</html>|
```

Logo em seguida vamos declara o cabeçalho de nossa pagina colocando um título neste cabeçalho, para informamos ao navegador o título:

\*Sem título - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

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

Nossa pagina já está ganhando forma! Agora nós iremos declarar onde fica o corpo de nossa página, para que o navegador o monte para o usuário, e vamos colocar o famoso *hello world*:

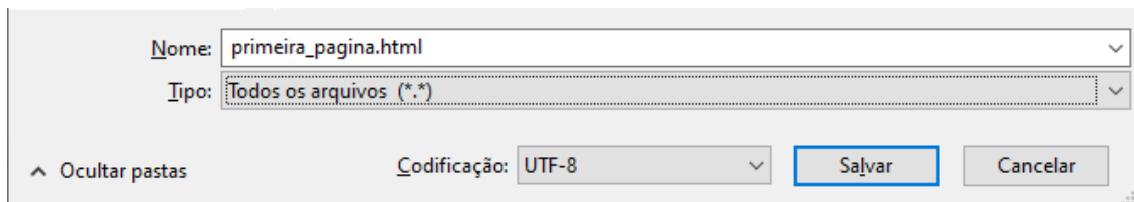
\*Sem título - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

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


  <body>
    Hello world!
  </body>
</html>
```

Pronto nossa primeira página está montada! Agora deve-se salva-la em algum local, então oriento que criem uma pasta chamada HTML na área de trabalho, quando nós for salvar a página devemos tomar cuidado com um detalhe, para que nossa pagina seja interpretada pelo nosso navegador, devemos salva-la com a extensão *.html* para que nosso navegador possa executa-la:



E a salve na pasta desejada, que oriento que siga minha orientação da criação da pasta.

Agora abra a pasta onde salvou o arquivo e abra-o no seu navegador:

Nome	Data de modificação
 primeira_pagina.html	21/12/2019 08:54

E teremos a nossa primeira página:



### 3.1 Exercícios

- 1) Crie uma página colocando seu nome como título e no corpo da página o que você mais gosta de fazer
- 2) Crie uma página falando sobre alguma atividade que você gosta

## 4. Criação de textos

Nas páginas web uma das coisas mais comuns de se encontrarem são textos de diversos tipos, tamanhos e estilos. Agora neste tópico iremos falar sobre tags de formatação de texto.

Iremos iniciar falando sobre parágrafos, os parágrafos são os mais comuns tipos de texto nas páginas web, principalmente nas páginas informativas (blogs). Nós já iniciamos a criação de texto nas nossas primeiras páginas, entretanto, nossos textos não possuíam nenhuma formatação! Para isto nós precisamos o domínio de algumas tags que fazem estas formatações.

Como mencionado, vamos iniciar com as tags de parágrafos. As tags de parágrafos podem ser criadas da seguinte maneira:

**<p>texto do paragrafo</p>**

Com esta estrutura podemos dizer inúmeros parágrafos:



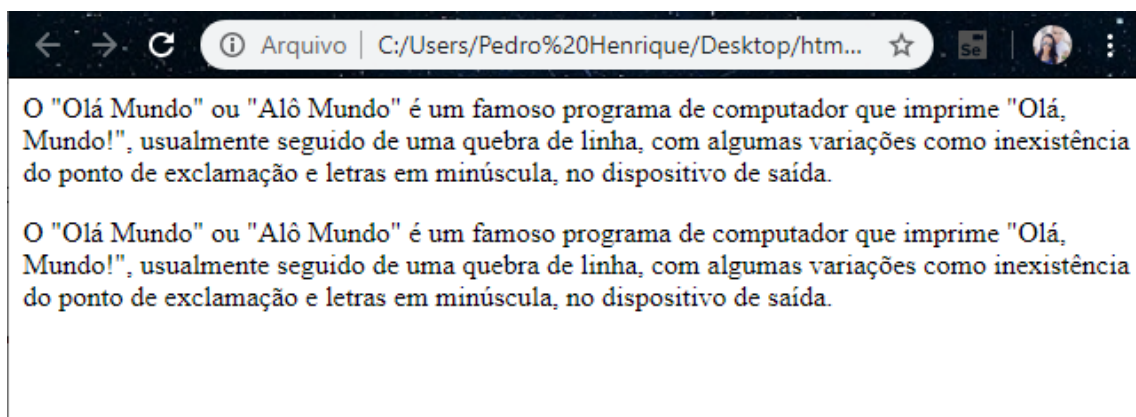
```

<body>
  <p>O "Olá Mundo" ou "Alô Mundo" é um
    famoso programa de computador que imprime
    "Olá, Mundo!", usualmente seguido de uma
    quebra de linha, com algumas variações
    como inexistência do ponto de exclamação
    e letras em minúscula, no dispositivo de saída.</p>

  <p>O "Olá Mundo" ou "Alô Mundo" é um
    famoso programa de computador que imprime
    "Olá, Mundo!", usualmente seguido de uma
    quebra de linha, com algumas variações
    como inexistência do ponto de exclamação
    e letras em minúscula, no dispositivo de saída.</p>
</body>

```

E este é o resultado:



Observe que as quebras de linhas dadas na digitação do código não são interpretadas pelo navegador, deve-se porque o navegador quando interpreta o código, ele faz em uma única linha, ou seja, essas quebras somem, e para que possamos colocar quebras, nós usamos a tag:

**<br>**

Também possui os famosos títulos de textos, que são bastante utilizados em páginas web, em diversos tamanhos. Para colocarmos estes títulos utilizamos a tag:

**<h1>Texto/palavra</h1>**

Resultado:

Texto/palavra

`<h2>Texto/palavra</h2>`

Resultado:

Texto/palavra

`<h3>Texto/palavra</h3>`

Resultado:

Texto/palavra

`<h4>Texto/palavra</h4>`

Resultado:

Texto/palavra

`<h5>Texto/palavra</h5>`

Resultado:

Texto/palavra

`<h6>Texto/palavra</h6>`

Resultado:

Texto/palavra

Também muito comum nas páginas web, são textos / palavras em negrito, e para podemos fazer isto, utilizamos a tag:

`<b>Texto/palavra</b>` ou `<strong>Texto/Palavra</strong>`

Resultado:

**Texto/palavra**

Como pode se observa a baixo:

O "Olá Mundo" ou "Alô Mundo" é um famoso programa de computador que imprime "Mundo!", usualmente seguido de uma quebra de linha, com algumas variações como incluir o ponto de exclamação e letras em minúscula, no dispositivo de saída.

primeira\_pagina.html - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
<html>
  <head>
    <title>Minha primeira página</title>
  </head>
  <body>
    <p>0
    "<b>Olá Mundo</b>" ou
    "<strong>Alô Mundo</strong>" é um
    famoso programa de computador que impri
```

Assim, também muito comum em páginas webs, são textos/palavras em *itálico* nestas páginas, e para fazermos isto nós utilizaremos a tag:

**<i>Texto/palavra</i>**

Resultado:

*Texto/palavra*

Para colocar um texto/palavra como marcado em uma página web, nós podemos utilizar a tag:

**<mark>Texto/palavra</mark>**

Resultado:

**Texto/palavra**

E para definimos um texto como riscado ou excluído, nós podemos utilizar a tag:

**<del>Texto/palavra</del>**

Resultado:

~~Texto/palavra~~

Para colocarmos o texto sublinhado nós podemos utilizar a tag:

**<ins>Texto/palavra</ins>**

Resultado:

Texto/palavra

## 4.1. Exercícios

- 1) Crie um arquivo que contenha um título, e subtítulos, e parágrafos, com todas as formatações mencionadas neste tópico.

## 5. Atributos

Agora que temos uma ótima base sobre tag, vamos nos aprofundar um pouco mais sobre algumas funcionalidades destas tags que tornam mais dinâmico o desenvolvimento em HTML.

Atributos são disponibilidade de recursos que nos permitem alterar atributos das tags, tornando-as bem mais dinâmicas, podemos torna essa alteração bem mais dinâmica com css que veremos mais na frente.

Para acessamos os atributos nós utilizaremos a seguinte estrutura:

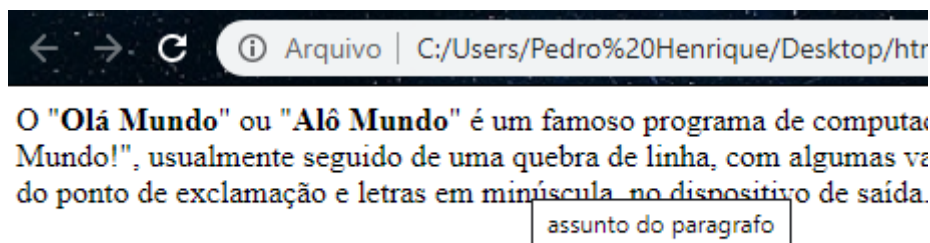
`<tagname atributo="valor">Texto/palavra</ tagname >`

Esta e a estrutura de acessa a estes atributos, muitas tags possuem atributos peculiares, e nós os demonstraremos no decorrer do E-book.

Um exemplo é o título de um parágrafo que podemos informar, como se contém qual assunto que este parágrafo trata, assim quando passar o mouse por cima do parágrafo aparece este título. EX.:

`<p title="assunto do paragrafo">Texto/palavra</ p >`

Aparecendo do seguinte modo:



## 6. Imagens

Nas páginas HTMLs é bastante comum encontrar imagens, para exemplificação, demonstração, beleza, etc. Assim, é notável a grande importância das imagens nestas páginas.

Para utilização de imagens em páginas, utilizamos a seguinte tag:

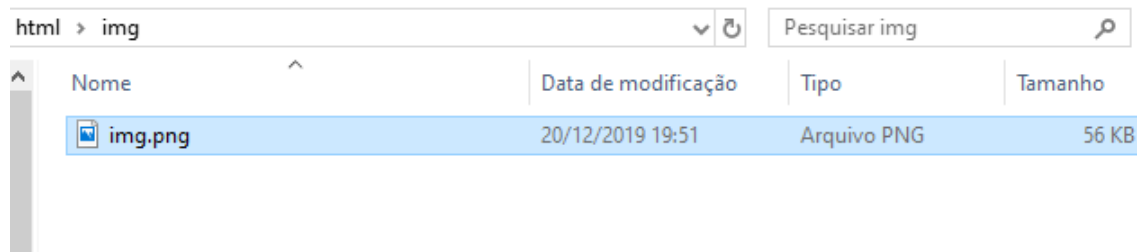
`<img>`

Para certamos a imagem neste tag utilizamos o atributo src:

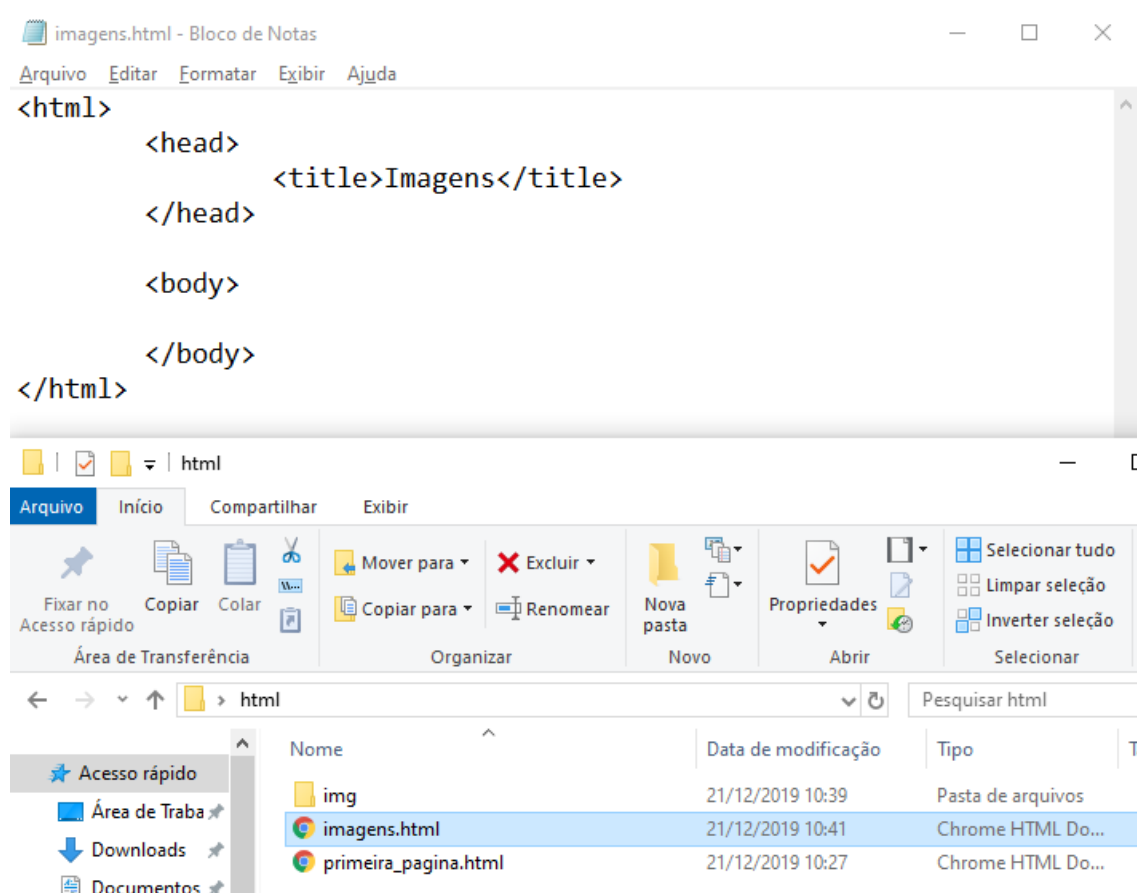
``

Neste atributo nós iremos informar onde está salva a imagem, normalmente no desenvolvimento de projetos, cria-se uma pasta denominada de img para salvar os arquivos de imagens nela, assim tornando-se mais fácil o acesso as imagens e organizado.

Vamos criar a pasta img em nossa pasta que estamos estudando HTML, e colocar uma imagem e renomear a imagem para img.extensão para a utilização:



Logo em seguida criaremos um novo arquivo no bloco de notas, e vamos nomeá-lo de *imagens.html* e criaremos a estrutura do HTML que já temos o domínio:



Logo em seguida, no corpo da página nós colocaremos a imagem que por nós foi selecionada, colocando o seguinte código:

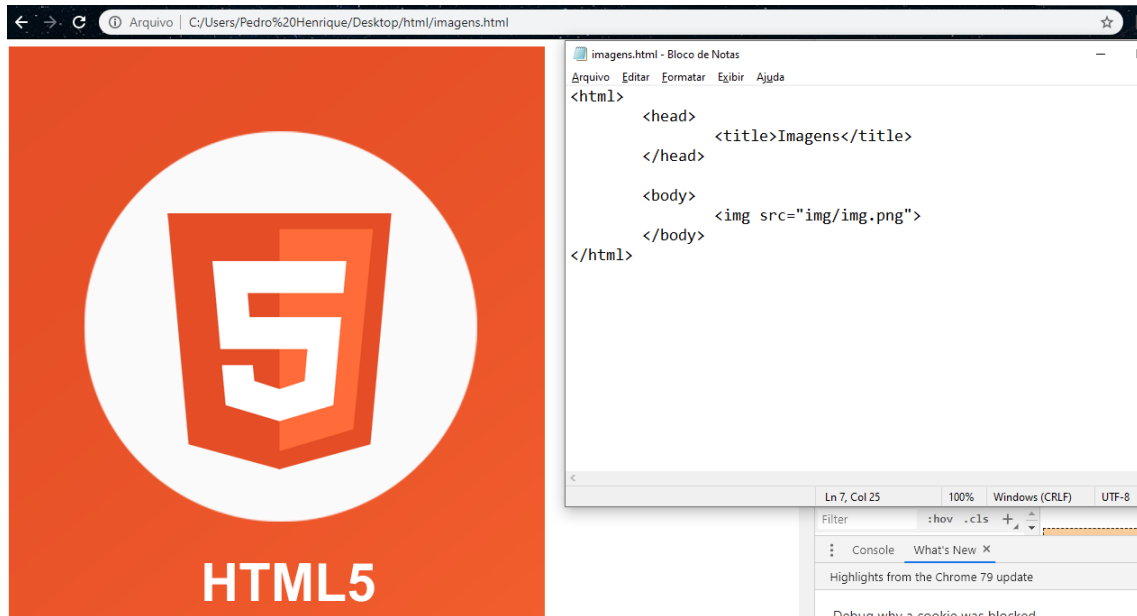
```

```

OBS.: Minha imagem está localizada em relação a nossa página, dentro da pasta img e o chamei de img.png, png é a extensão de minha imagem.

Caso nossa pagina estivesse dentro de outra pagina deveríamos colocar `../img/img.png`. `../` indica que ele vai voltar uma pasta.

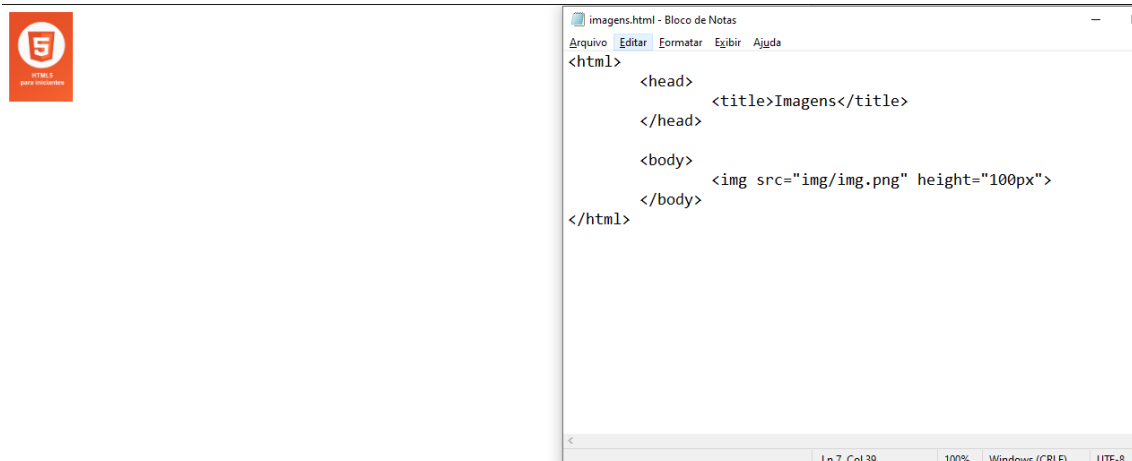
Quando executamos no navegador teremos o resultado:



Caso queremos alterar o tamanho da imagem, utilizamos o atributo *height*, informando quantos pixels queremos que esta imagem ocupe:

``

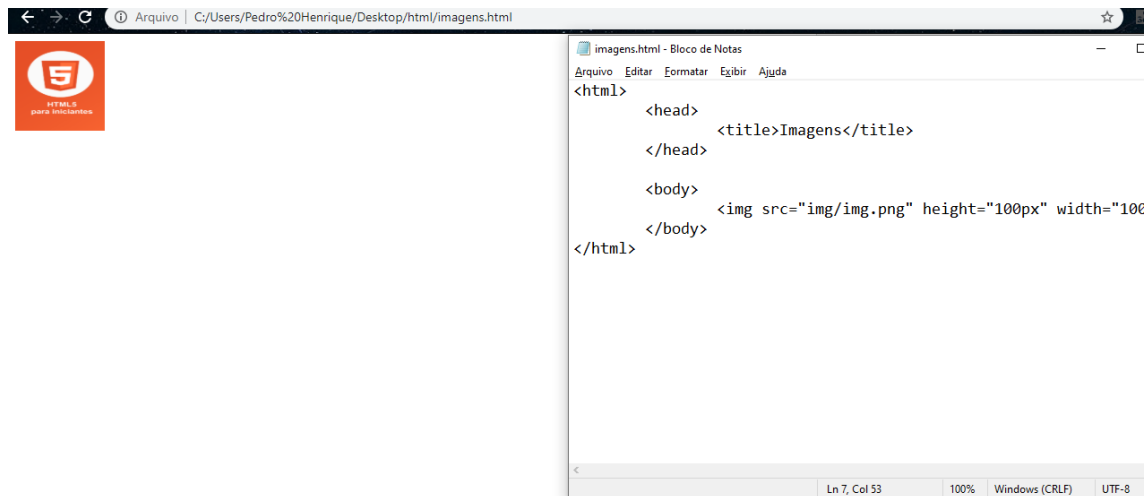
Com isto, essa imagem ocupará uma altura de 100 pixels da tela do computador. E teremos:



Caso queremos alterar a largura da imagem vamos utilizar o atributo *width* para alterar sua largura:

``

Assim a largura da imagem também ocupará 100 pixels da tela. Assim podemos observar:



## 6.1. Exercícios

- 1) Crie uma página que contenha uma imagem com as dimensões largura 300 pixels e altura 400 pixels

## 7. Links

Links nas páginas webs são elementos essenciais para uma ótima página web. Por meio destes links podemos navegar por estas páginas, e acessamos outras páginas disponíveis, além de acessar arquivos externos.

Os links são definidos pela tag:

**`<a>Texto do link</a>`**

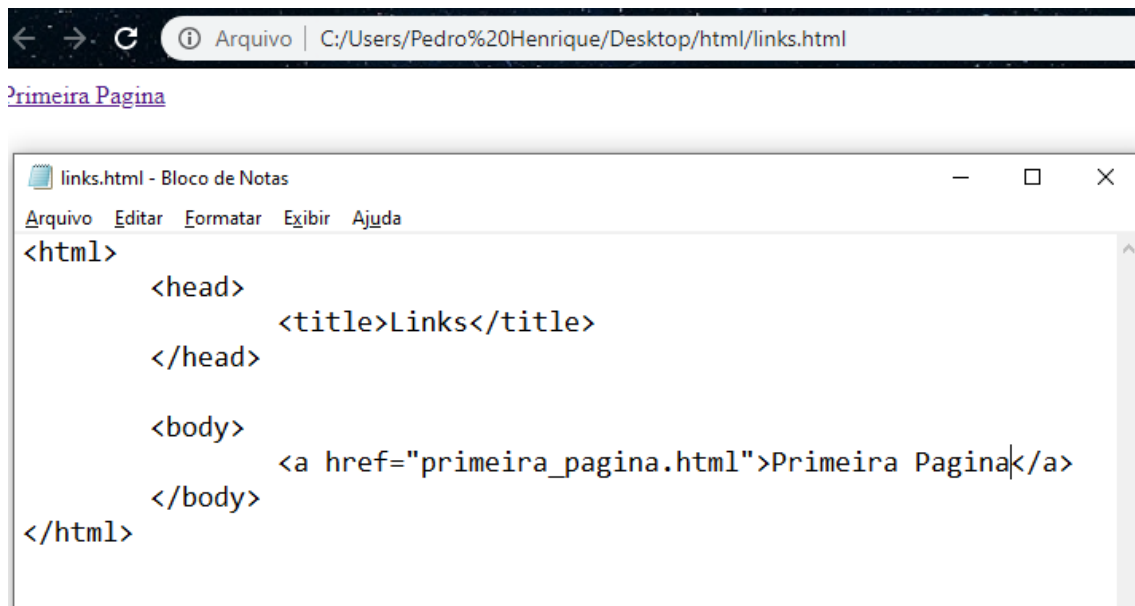
Para podermos redirecionar os usuários por meio dos links, nós utilizamos o atributo *href* onde nós inserimos um link:

**`<a href="https://google.com.br">Texto do link</a>`**

Assim, como neste exemplo acima, direcionamos o cliente para o site da Google, e assim como fizemos no exemplo acima, podemos direcionar para nosso próprio projeto, crie um novo arquivo chamado *links.html*, lembra de nossa primeira página? Caso não tenha mais crie outra para testes, neste arquivo *links.html* colocaremos o seguinte código:

**`<a href="primeira_pagina.html">Primeira pagina</a>`**

Neste exemplo teremos:



E quando clicamos no link somos direcionados para o link informado no *href* da tag *a*.

## 7.1. Exercícios

- 1) Crie uma página que contenha um link que direcione para a página do Facebook (<https://facebook.com>).

## 8. Botões

Os botões podem ser constantemente encontrados nas páginas web, assim o domínio sobre estas tag se faz fundamental.

Assim, para criar um botão usando HTML, utiliza-se a tag:

**<button>Texto do botão</button>**

Os botões estão distribuídos em 3 tipos principais: submit (para envio de formulário), reset (para reiniciar o formulário) e button (este é o padrão para a tag).

**<button type="submit">Texto do botão</button>**

**<button type="reset">Texto do botão</button>**

**<button type="button">Texto do botão</button>**

## 8.1. Exercícios

- 1) Crie os 3 tipos principais de botões



## 9. Listas

As listas e diversos casos são utilizados, principalmente na exposição de dados, que devem estar distribuídos em uma ordem que pode ser utilizado listagem.

As listas nós podemos subdividilas em 2 tipos, as listas ordenada e as listas não ordenadas. As listas ordenadas são listas que mantem uma ordem crescente.

### 9.1. Listas ordenadas

As listas não ordenadas são colocadas pela tag *ol*:

```
<ol>
```

...

```
</ol>
```

Nas listas ordenadas vem por padrão uma ordem numérica, entretanto isto pode ser alterado pelo atributo *type*:

```
<ol type="A">
```

...

```
</ol>
```

Para serem adicionados itens a lista, nós utilizaremos a tag *li*:

```
<ol>
```

```
    <li>Item 1</li>
```

```
    <li>Item 2</li>
```

```
    <li>Item 3</li>
```

```
</ol>
```

### 9.2. Listas não ordenadas

As listas não ordenadas não seguiram uma ordem, assim apenas aparecendo um ícone em sua frente. Estas listas podem ser declaradas pela tag *ul*:

```
<ul>
```

```
    <li>Item 1</li>
```

```
    <li>Item 2</li>
```

```
    <li>Item 3</li>
```

```
</ul>
```

Para alteração de seu ícone, será utilizado css, assim mostraremos isto mais a diante.

### 9.3. Exercícios

- 1) Crie um exemplo de lista ordenada com nome de pessoas.
- 2) Crie um exemplo de lista não ordenada com E-mails de pessoas.

## 10. Criação de formulários

Os formulários são bastante utilizados nas páginas webs em diversas situações, desde um simples cadastro a uma pesquisa, até mesmo pode ser utilizado para exposição de dados.

Os formulários são caracterizados pelo envio de informações para outras páginas ou *scripts*.

Para criação de formulários, vamos utilizar a tag *form* para criação, colocando alguns atributos.

```
<form action="localpagina.html" method="post">
...
</form>
```

Segue-se está estrutura para criação dos formulários. Observe que utilizamos 2 atributos, o *action* e *method*, estes atributos são de configuração de seu formulário. O *action* informa para onde serão enviado os dados do formulário, e o *method* informa qual será o método de envio destes dados.

**Method** valores:

**Post** - envia dados diretamente de modo mais seguro;

**Get** – envia os dados pela URL, menos seguro;

Dentro do nosso formulário, podemos informar os campos que ele deve conter, como uma caixa de texto, uma seleção, entre outros. Segue a seguinte estrutura:

```
<form action="localpagina.html" method="post">
    <input type="text" name="nomedocampo" value="">
</form>
```

Observe que nesta nova tag, *input*, temos 3 novos atributos. O atributo *type* diz qual tipo de informação aquele *input* deve conter ou receber, o *name* diz o nome do campo para que a página que recebe-lo poder trabalha-lo, e o *value* informa o valor padrão dele, caso não tenha pode deixado em branco.

Valores para o **type**:

**Text** – recebe valor de texto;

**Number** – recebe valor numérico;

**Checkbox** – recebe valor de seleção;

**Radio** – recebe valor de seleção (este pode ter vários com mesmo nome, e será somente enviado o selecionado)

**File** – recebe arquivos

**Search** - recebe valor de pesquisa

**Email** – recebe valores de E-mail

**url** – recebe valores de URL

**date** – recebe valores de datas

**color** – recebe valores hexadecimal das cores

**submit** – cria botão de enviar o formulário

Outra tag bastante utilizada em formulários, são as tags de seleções, como quando selecionamos um estado ou país, ela segue a estrutura:

```
<form action="localpagina.html" method="post">
  <select name="nome">
    <option value="valor">Nome da opção</option>
  </select>
</form>
```

Este cria uma caixa de seleção, o atributo *name* é utilizado pela outra página para trabalhar esta seleção, e envia o valor do atributo *value* do *option* selecionado, podendo haver vários, envia o que o usuário seleciona.

## 11. Atributos seletores

Atributos seletores, são atributos que colocamos nos elementos para poder selecioná-los em algum momento do desenvolvimento, além de auxiliar nos mecanismos de busca.

### 11.1. Ids

Ids são atributos seletores que utilizam a seguinte estrutura:

```
<tagname id="nomeid">...</tagname>
```

### 11.2. Classes

Classes são atributos seletores que utilizam a seguinte estrutura:

```
<tagname class="nomeclass">...</tagname>
```

## 12. Iframes

Iframes é uma tag útil para utilização em casos de demonstração de outras páginas em sua página, como o youtube, ou para sites de marketing.

A estrutura de um Iframe é:

```
<iframe> </iframe>
```

Para mostrar uma outra página neste Iframe, é necessário a utilização do atributo *src*:

```
<iframe class="url"> </iframe>
```

### 12.1. Exercícios

- 1) Crie um iframe que mostre um vídeo do youtube

## 13. Tabelas

As tabelas são comuns em sites webs na amostra de dados me modo organizado, para assim, torna mais acessível e organizado alguns dados para os usuários.

As tabelas são declaradas pela tag:

```
<table>
```

...

```
</table>
```

Assim, nós declaramos para o navegador que ele deve montar uma tabela para o usuário.

Porém, ainda não possuímos nenhuma informação nessa tabela, assim, nós podemos declarar o cabeçalho desta tabela pela seguinte tag:

```
<table>
```

```
    <thead>
```

```
    </thead>
```

```
</table>
```

Assim, nós declaramos o cabeçalho da tabela, entretanto, sabemos que tabelas possuem linhas e colunas que contém as informações, assim nós podemos declarar estas linhas pela tag:

```
<table>
```

```
    <thead>
```

```

        <tr>
      </tr>
    </thead>
  </table>

```

A tag *tr* declara que ali é uma nova linha, entretanto, nestas linhas devem ter colunas, assim, nós declaramos as linhas pela tag:

```

<table>
  <thead>
    <tr>
      <td>Informação</td>
    </tr>
  </thead>
</table>

```

Assim, nesta tabela acima, nós teremos uma tabela que possui um cabeçalho, que possui uma linha com uma coluna que possui “informação”. Mas nesta tabela somente se tem o cabeçalho, ainda nos falta inserir o corpo desta tabela.

Assim devemos inserir o corpo desta tabela pela seguinte tag:

```

<table>
  <thead>
    <tr>
      <td>Informação</td>
    </tr>
  </thead>

  <tbody>
</tbody>
</table>

```

A tag *tbody* é a tag que indica para o navegador que ali inicia-se o corpo desta tabela, e neste corpo devemos também inserir linhas e colunas, assim como no *thead*, ficando assim:

```

<table>
  <thead>

```

```

        <tr>
            <td>Nome</td>
            <td>Sobrenome</td>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td>Junior</td>
            <td>Souza</td>
        </tr>
    </tbody>
</table>

```

Neste Exemplo acima, teremos uma saída parecia com isto:

Nome	Sobrenome
Junior	Souza

Caso coloquemos mais linhas, podemos ter mais informações em nossa tabela:

```

<table>
    <thead>
        <tr>
            <td>Nome</td>
            <td>Sobrenome</td>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td>Junior</td>
            <td>Souza</td>

```

```

</tr>
<tr>
    <td>Mauro</td>
    <td>Braz</td>
</tr>

```

```

</tbody>

```

```

</table>

```

Neste exemplo teremos a seguinte saída:

Nome	Sobrenome
Junior	Souza
Mauro	Braz

Observe que neste exemplos, nossas tabelas estão sem nenhuma borda, assim, para colocamos uma borda em nossa tabela utilizaremos o atributo *border* da tabela

```

<table border="1px">
    <thead>
        <tr>
            <td>Nome</td>
            <td>Sobrenome</td>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td>Junior</td>
            <td>Souza</td>
        </tr>
        <tr>
            <td>Mauro</td>

```

```
<td>Braz</td>
</tr>
```

```
</tbody>
```

```
</table>
```

Devendo informar a espessura da borda da tabela, assim temos o seguinte resultado:

Nome	Sobrenome
Junior	Souza
Mauro	Braz

### 13.1. Exercícios

- 1) Crie uma tabela que possua no cabeçalho *nome*, *email*, *numero* e insira os dados desta tabela em seu corpo.

## 14. Introduzindo CSS

A Sigla CSS significa *Cascading Style Sheets* (folha de estilo em cascata). O CSS é um dos fatores de as páginas webs serem tão dinâmicas e práticas, com designs modernos que torna a experiência do usuário com a página bem mais agradável.

O CSS e o HTML5 devem sempre andarem juntos, eles são um conjunto que um completa o outro. O HTML está mais voltado a estrutura a página web, enquanto que o CSS ele é mais voltado por estilizar a página, tornando-a mais agradável.

O CSS podemos utiliza-lo de 3 maneiras diferentes, que possuem a mesma finalidade: o CSS embutido, o interno e o externo. Iremos trabalhar cada um dos três tipos.

### 14.1. CSS embutido

O CSS embutido é uma CSS que utilizamos mais em estilizações rápidas, que normalmente devem ser individuais a apenas a um elemento HTML.

A estrutura do CSS embutido é caracterizada como um atributo no elemento HTML, assim , seguindo a seguinte estrutura:

```
<tagname style="propriedade:valor">...</tagname>
```

Criando um exemplo simples, trabalharemos mais a fundo as possibilidades do CSS quando dominamos estes tipos. EX.:



```
<p style="color: red">texto</p>
```

Saida:

Texto

## 14.2. CSS interno

O CSS interno é localizado dentro de nossa página HTML, entretanto, não se é colocado como atributo nos elementos, mas sim, normalmente, no cabeçalho para ser trabalhado dentro da página, seguindo a seguinte estrutura:

```
<html>

    <head>

        <title></title>

        <style>

            p{

                color: red;

            }

        </style>

    </head>

    <body>

        <head>Texto</head>

    </body>

</html>
```

Saida:

Texto

## 14.3. CSS externo

O CSS externo, é o mais comum nas páginas web, por conta de sua organização, e seguiu padrões estabelecido por programadores para melhor aproveitamento de código.

Assim, o CSS externo é um arquivo criado fora da página HTML com a extensão .css que você deve na página HTML fazer referência a este arquivo, para que sua estilização seja feita na página.

Para fazemos esta referência utilizamos a tag *link*:

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

O arquivo *style.css* deve ser o arquivo que você criou para codificar seu CSS.

Assim como no exemplo do embutido é sua funcionalidade, porém, não coloca-se a tag *style*, pois o arquivo .css já é configurado para este tipo de codificação.

## 15. Seletores CSS

Os seletores CSS são modos de identificar elementos HTML pelo CSS para a modificação deles. Nós iremos utilizar 3 tipos principais.

O seletor pelo nome da tag já foi demonstrado neste E-book. Este seletor se torna meio genérico, pois este tipo de seleção atribui a todos os elementos sem seleção específica, apenas se adentrarmos no modo avançado, selecionando todas tag com aquele nome selecionado.

```
p{  
    color: red;  
}
```

Neste exemplo, fazemos uso deste tipo de seletor, fazendo com que todos os parágrafos, de tag *p*, fiquem de cor vermelha, assim tornando-se genérico.

Outro tipo de seletor, esta entre os mais usados, são seletores ID. Este seletor já foi ensinado a sua criação em “Atributos seletores”, porem, agora vamos aprender como utiliza-lo com CSS.

Ele serve para dizer ao navegador que todos os elementos com aquele ID, vão sofrer a estilização criada.

```
#meuid{  
    color: red;  
}
```

Seguindo esta syntax, deve-se sempre colocar # antes do nome do ID para se identificado. Assim, todos elementos que possuírem o ID “meuid” irão sofrer esta estilização.

Outro seletor bastante utilizado são as classes, que também já se foi ensinado como criar no capítulo “Atributos seletores”. Assim como os Ids, as classes são

funciona de mesmo modo, entretanto, por padrão utiliza-se mais os IDs por conta de algumas APIs utilizarem as classes.

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

Observando a sintax, é visível que a única divergências está que utiliza-se . antes do nome da classe, entretanto, ambos possuem a mesma função.

## 16. Propriedades CSS de formatação de texto

Observamos em capítulos anteriores, muitas formatações de texto podem ser feitas através do HTML, entretanto, também é possível fazer estas formatações através da estilização.

Primeira propriedade é a de alinhamento do texto dentro de seu container:

```
#meuid{  
    text-align: center;  
}
```

Os possíveis valores são: *right*, *justify*, *left* e *center*.

Segunda propriedade é a de alteração do tamanho do texto:

```
#meuid{  
    text-size: 20px;  
}
```

Outra propriedade é a de alteração de cor, como já foi mostrado anteriormente:

```
#meuid{  
    color: red;  
}
```

Outra propriedade e a de alteração do estilo de fonte do texto:

```
#meuid{  
    font-family: arial;  
}
```

Outra propriedade e a de alteração da sombra do texto:

```
#meuid{  
    text-shadow: 2px 2px black;
```

}

Na propriedade de sombra é notável vários valores, o primeiro é dizendo a distância do texto em relação ao eixo x, o segundo ao eixo y, e o terceiro é a cor da sombra do texto;.

## 17. Novas tags do HTML5

Como o HTML5 é uma atualização do HTML4, apareceram novas tags que facilitam o desenvolvimento e torna o código mais claro para o entendimento. Neste tópico falaremos de algumas destas facilidades que surgiram com esta atualização da linguagem HTML.

Uma das facilidade foi o surgimento da tag *canvas*, caso queira seguir na área de desenvolvimento de jogos você se deparará muito com esta tag, não falarei muito sobre está, porque deve ter uma conhecimento mais aprofundado de JavaScript.

```
<canvas id="mycanvas"></canvas>
```

Outro elemento de grande utilidade na reprodução de arquivos de mídia foi a tag *vídeo* que já é voltada para esta finalidade:

```
<video id="myvideo" controls="controls" src="myvideo.mp4"></video>
```

Também neste sentido, outra tag bastante útil foi a de reprodução de áudios:

```
<audio id="myaudio" controls="controls" src="myaudio.mp4"></audio>
```

Além de algumas tags voltadas para o desenvolvimento de layouts em HTML, temos todas as tags principais para criação de um layout, primeiramente a tag:

```
<header></header>
```

Esta tag já é voltada para criação do cabeçalho do corpo do site, fazer a cabeça de seu site, onde podemos colocar o banner, o menu, pesquisas, entre outras coisas.

Outra tag muito útil que o HTML5 trouxe foi a tag *main*:

```
<main></main>
```

Esta tag é voltada para criação do corpo do site, onde normalmente fica o conteúdo mais volátil, que normalmente são editados.

Outra tag que facilita o desenvolvimento em HTML, é a tag de menu, *nav*:

```
<nav></nav>
```

Está tag é voltada para conter o menu do seu site.

E para finalizar seu site, o HTML5 trouxe a tag *footer*, ela é voltada para criação de rodapé de seu site:

`<footer></footer>`

## 18. Edição de containers com CSS

Muitas vezes é necessário fazer ajustes nas tags, principalmente nas que contêm outras tags, assim precisamos ter o domínio sobre a edição destas tags;

Nossa primeira propriedade de edição destes elementos, é a propriedade de alteração de cor de fundo de um item:

```
header {  
    background-color: black;  
}
```

Outra propriedade de alteração de fundo é a propriedade de colocar uma imagem de fundo:

```
header {  
    background-image: url(imagem.png);  
}
```

Muitas vezes estas imagens são pequenas, e se repetem bastante, para evitar isso, usamos a propriedade de repetição de background:

```
header {  
    background-repeat: no-repeat;  
}
```

Normalmente quando fazemos esta alteração a imagem tende a ocupar um pequeno espaço no container, para isso nós usamos a propriedade de alteração de tamanho do background:

```
header {  
    background-size: 100%;  
}
```

Para nós alterar a largura de um container podemos utilizar uma propriedade CSS:

```
header {  
    width: 100%;  
}
```

E para alterarmos o tamanho deste container, nós podemos utilizar outra propriedade CSS:

```
header {  
    height: 300px;  
}
```

Muitas vezes precisamos alterar as posições dos objetos, para isto nós temos a propriedade:

```
main {  
    position: absolute;  
}
```

Podemos passar outros tipos de valores:

**absolute** - pega posições em relação ao início do navegador;

**relative** – pega posições em relação ao objeto que está acima dele;

**fixed** – Pega uma posição fixa em relação ao início do navegador

**static** – pega posições assim como o absolute.

Para alteração destas posições nós usamos outras 2 propriedades, assim podendo fazer a alteração das posições:

```
header {  
    position: absolute;  
    top: 100px;  
    left: 30px;  
}
```

Uma das propriedades comuns a utilização é a alteração do tipo de display:

```
main {  
    display: block;  
}
```

**Block** – tornamos aquele container como tipo bloco.

**Inline** – tornamos aquele item como linhas.

**Inline-block** – tornamos ele em block com distribuição em linha.

**None** – não está visível

Podemos alterar as propriedades de bordas dos elementos com CSS:

```
main {
```

```
border: 1px solid black;  
}
```

No primeiro parâmetro passamos a espessura da borda, depois passamos o tipo de borda, e o terceiro é a cor da borda.

Tipos de bordas:

**Solid** – sólida.

**Double** – borda dupla.

**Groove** – borda entalhada.

**Dotted** – borda pontilhada.

**None** – sem borda.

Podemos colocar uma margem de espaçamento, com a propriedade CSS:

```
main {  
    margin-left: 10px;  
    margin-right: 10px;  
    margin-top: 10px;  
    margin-bottom: 10px;  
    margin: 10px 10px 10px 10px;  
}
```

Um efeito de sombra sempre cai bem em objetos sólidos no HTML, assim temos a propriedade perfeita para isto:

```
main {  
    box-shadow: 1px 1px 1px 10px black;  
}
```

Note que passamos 5 valores, o primeiro é a distância da sombra do objeto no eixo x e o segundo o eixo y, o terceiro é o quanto ela estará borrada o quarto o seu tamanho, e o quinto valor sua cor.

Outra propriedade que podemos utilizar, e de tornar o objeto meio borrado, utilizando a propriedade *filter*:

```
main {  
    filter: blur(20px);  
}
```

Assim deixamos o objeto com um borrão em sua forma visual.

Outro efeito fornecido pelo CSS3 é a propriedade *transform* que podemos fazer diversas alterações da forma normal do objeto:

```
main {  
    transform: rotate(20deg);  
}
```

A propriedade *transform* possui muito valores possíveis como listado a baixo:

none	Defines that there should be no transformation
matrix( <i>n,n,n,n,n,n</i> )	Defines a 2D transformation, using a matrix of six values
matrix3d( <i>n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n</i> )	Defines a 3D transformation, using a 4x4 matrix of 16 values
translate( <i>x,y</i> )	Defines a 2D translation
translate3d( <i>x,y,z</i> )	Defines a 3D translation
translateX( <i>x</i> )	Defines a translation, using only the value for the X-axis
translateY( <i>y</i> )	Defines a translation, using only the value for the Y-axis
translateZ( <i>z</i> )	Defines a 3D translation, using only the value for the Z-axis



<code>scale(x,y)</code>	Defines a 2D scale transformation
<code>scale3d(x,y,z)</code>	Defines a 3D scale transformation
<code>scaleX(x)</code>	Defines a scale transformation by giving a value for the X-axis
<code>scaleY(y)</code>	Defines a scale transformation by giving a value for the Y-axis
<code>scaleZ(z)</code>	Defines a 3D scale transformation by giving a value for the Z-axis
<code>rotate(<i>angle</i>)</code>	Defines a 2D rotation, the angle is specified in the parameter
<code>rotate3d(x,y,z,<i>angle</i>)</code>	Defines a 3D rotation
<code>rotateX(<i>angle</i>)</code>	Defines a 3D rotation along the X-axis
<code>rotateY(<i>angle</i>)</code>	Defines a 3D rotation along the Y-axis
<code>rotateZ(<i>angle</i>)</code>	Defines a 3D rotation along the Z-axis

<code>skew(<i>x-angle</i>,<i>y-angle</i>)</code>	Defines a 2D skew transformation along the X- and the Y-axis
<code>skewX(<i>angle</i>)</code>	Defines a 2D skew transformation along the X-axis
<code>skewY(<i>angle</i>)</code>	Defines a 2D skew transformation along the Y-axis
<code>perspective(<i>n</i>)</code>	Defines a perspective view for a 3D transformed element
<code>initial</code>	Sets this property to its default value.
<code>inherit</code>	Inherits this property from its parent element

Muito utilizado são efeitos de transições para itens, para isto nós utilizamos a propriedade CSS *transition*, já voltada para trabalhar transições de modo mais suave possível:

```
main {
    transition: 1s;
}
```

Muito utilizado nas paginas, são itens que sobressair sobre outros itens, como o menu fixo no topo que deve ser sempre em primeiro plano, para isto nós utilizamos a propriedade *z-index* que indica qual camada o item está:

```
main {
    z-index: 9999;
}
```

## 19. Hover CSS

O hover no CSS é um dos que mais são utilizados, funcionando quando o mouse do usuário passa por cima do objeto selecionado, ele faz alguma alteração no objeto:

```
main {  
    Background-color: red;  
}  
  
Main:hover {  
    Background-color: blue;  
}
```

## 20. Animações com KeyFrames

As animações são bastante utilizadas nas páginas, para tornar a experiência do usuário mais amigável e surpreendente, observando coisas novas e nossa criatividade.

Para criar nossa animação usamos o *keyframe* para animação dinâmica. Obedecendo a estrutura:

```
@keyframes nomedaanimacao {  
    0%{  
        top: 10px;  
    }  
    50%{  
        top: 100px;  
    }  
    100%{  
        top: 10px;  
    }  
}
```

Nossa animação já está criada, porém ainda não é utilizada, assim, precisamos dizer que aquele objeto deve ter essa animação, fazendo uso da propriedade *animation*:

```
main {  
    position: absolute;
```

```

top: 10px;

animation: nome da animacao 1s;

}

```

Assim dizemos que o *main* fará esta animação, passando em primeiro parâmetro o nome da animação que nós criamos, e depois quanto tempo é para levar fazendo essa animação.

## 21. Comentários

Os comentários são extremamente importantes para manutenção do código, pois com o passar do tempo nós podemos esquecer algumas funcionalidades já escritas e levar algum tempo para deciframos novamente o que faz aquele código, e como tempo é dinheiro, é melhor nós lê uma linha de comentário do que perder 2 horas entendendo código.

Estrutura para comentários em HTML:

```
<!-- comentário -->
```

## 22. Algumas propriedades CSS

As propriedades CSS mais utilizadas no desenvolvimento:

### PROPRIEDADES DE COR

PROPRIEDADE	DESCRIÇÃO	CSS
<b>color</b>	Aplica cor em texto	1
<b>opacity</b>	Configura o nível de opacidade de um elemento	3

### PROPRIEDADES DE FUNDO E BORDA

PROPRIEDADE	DESCRIÇÃO	CSS
<b>background</b>	Configura todas as propriedades de fundo em uma linha	1
<b>background-attachment</b>	Configura quando uma imagem de fundo é fixa ou rola com a página	1
<b>background-color</b>	Configura a cor do fundo	1
<b>background-image</b>	Configura uma imagem de fundo para um elemento	1

<b>background-position</b>	Configura a posição de uma imagem de fundo	1
<b>background-repeat</b>	Configura como uma imagem de fundo deve se repetir	1
<b>background-clip</b>	Configura a área de pintura de um fundo	3
<b>background-origin</b>	Configura a área de posicionamento da imagem de fundo	3
<b>background-size</b>	Configura o tamanho de uma imagem de fundo	3
<b>border</b>	Configura todas as propriedades de borda em uma linha	1
<b>border-bottom</b>	Configura as propriedades da borda abaixo do elemento	1
<b>border-bottom-color</b>	Configura a cor da borda abaixo do elemento	1
<b>border-bottom-left-radius</b>	Configura o formato da borda inferior esquerda	3
<b>border-bottom-right-radius</b>	Configura o formato da borda inferior direita	3
<b>border-bottom-style</b>	Configura o estilo da borda abaixo do elemento	1
<b>border-bottom-width</b>	Configura a largura da borda abaixo do elemento	1
<b>border-color</b>	Configura a cor das quatro bordas	1
<b>border-image</b>	Propriedade abreviada para definir todos as propriedades de border-image-*	3
<b>border-image-outset</b>	Especifica a quantidade que a área da imagem da borda se estende além da caixa da borda	3
<b>border-image-repeat</b>	Especifica se a borda da imagem deve ser repetida, arredondada ou alongada	3
<b>border-image-slice</b>	Especifica os deslocamentos interiores da imagem da borda	3
<b>border-image-source</b>	Especifica uma imagem para usar como borda	3

<b>border-image-width</b>	Especifica a largura da imagem da borda	3
<b>border-left</b>	Configura todas as propriedades da borda esquerda	1
<b>border-left-color</b>	Configura a cor da borda da esquerda	1
<b>border-left-style</b>	Configura o estilo da borda da esquerda	1
<b>border-left-width</b>	Configura a largura da borda da esquerda	1
<b>border-radius</b>	Configura o estilo de todas as bordas	3
<b>border-right</b>	Configura a borda da direita	1
<b>border-right-color</b>	Configura a cor da borda da direita	1
<b>border-right-style</b>	Configura o estilo da borda da direita	1
<b>border-right-width</b>	Configura a largura da borda da direita	1
<b>border-style</b>	Configura o estilo de todas as bordas	1
<b>border-top</b>	Configura a borda do topo	1
<b>border-top-color</b>	Configura a cor da borda do topo	1
<b>border-top-left-radius</b>	Configura o formato da borda superior esquerda	3
<b>border-top-right-radius</b>	Configura o formato da borda superior direita	3
<b>border-top-style</b>	Configura o estilo da borda do topo	1
<b>border-top-width</b>	Configura a largura da borda do topo	1
<b>border-width</b>	Configura a largura da borda	1
<b>box-decoration-break</b>	Define o comportamento do fundo e borda de um elemento em uma quebra de página, ou, para os elementos em linha, na quebra de linha.	3
<b>box-shadow</b>	Cria uma sombra em um elemento	3

## PROPRIEDADES BÁSICAS PARA CAIXAS

PROPRIEDADE	DESCRIÇÃO	CSS
-------------	-----------	-----

<b>bottom</b>	Especifica a posição de um elemento posicionado relacionada à parte inferior do elemento que o sustenta	2
<b>clear</b>	Especifica qual lado do elemento não permitirá elementos flutuantes	1
<b>clip</b>	Recorta um elemento posicionado como “absolute” ou “fixed”	2
<b>display</b>	Especifica como alguns elementos devem ser mostrados	1
<b>float</b>	Especifica um elemento flutuante	1
<b>height</b>	Configura a altura de um elemento	1
<b>left</b>	Especifica a posição de um elemento relacionado à parte da esquerda do elemento que o sustenta	2
<b>overflow</b>	Especifica o que acontece se um elemento passar do tamanho do elemento que o sustenta	2
<b>overflow-x</b>	A mesma coisa de overflow, mas só controla direita e esquerda	3
<b>overflow-y</b>	A mesma coisa de overflow, mas só controla a parte superior e inferior	3
<b>padding</b>	Especifica o “enchimento” de um elemento	1
<b>padding-bottom</b>	Especifica o “enchimento” inferior de um elemento	1
<b>padding-left</b>	Especifica o “enchimento” da esquerda de um elemento	1
<b>padding-right</b>	Especifica o “enchimento” da direita de um elemento	1
<b>padding-top</b>	Especifica o “enchimento” superior de um elemento	1
<b>position</b>	Especifica a posição de um elemento (static, relative, absolute ou fixed)	2
<b>right</b>	Especifica a posição de um elemento relacionado à parte da direita do elemento que o sustenta	2
<b>top</b>	Especifica a posição de um elemento relacionado à parte superior do elemento que o sustenta	2
<b>visibility</b>	Especifica a visibilidade de um elemento	2
<b>width</b>	Configura a largura de um elemento	1

<b>vertical-align</b>	Configura o alinhamento vertical do elemento	1
<b>z-index</b>	Envia um elemento posicionado para frente ou para trás de outros elementos	2

## MARGEM E ESTILO

PROPRIEDADE	DESCRIÇÃO	CSS
<b>display</b>	Configura como o elemento será exibido	1
<b>margin</b>	Configura a margem de um elemento	1
<b>margin-bottom</b>	Configura a margem inferior do elemento	1
<b>margin-left</b>	Configura a margem da esquerda de um elemento	1
<b>margin-right</b>	Configura a margem da direita de um elemento	1
<b>margin-top</b>	Configura a margem do topo de um elemento	1
<b>max-height</b>	Configura a altura máxima de um elemento	2
<b>max-width</b>	Configura a largura máxima de um elemento	2
<b>min-height</b>	Configura a altura mínima de um elemento	2
<b>min-width</b>	Configura a largura mínima de um elemento	2

## DECORAÇÃO DE TEXTOS

PROPRIEDADE	DESCRIÇÃO	CSS
<b>text-decoration</b>	Especifica a decoração de um texto	1

## PROPRIEDADES DE FONTE

PROPRIEDADE	DESCRIÇÃO	CSS
<b>font</b>	Configura todas as propriedades de fonte	1
<b>font-family</b>	Configura a família da fonte utilizada	1



<b>font-size</b>	Configura o tamanho da fonte	1
<b>font-style</b>	Configura o estilo da fonte	1
<b>font-weight</b>	Configura a espessura da fonte	1

## PROPRIEDADES DE LISTAS

PROPRIEDADE	DESCRIÇÃO	CSS
<b>list-style</b>	Especifica todas as propriedades de elementos de lista	1
<b>list-style-image</b>	Especifica uma imagem para marcar uma lista	1
<b>list-style-position</b>	Configura a posição dos elementos de uma lista	1
<b>list-style-type</b>	Configura o estilo de um elemento de uma lista	1

## 23. Algumas tags do HTML5

Algumas tags que precisaremos no desenvolvimento:

<b>&lt;article&gt;</b>	Representa um pedaço independente do conteúdo de um documento, como um texto, notícias, blog, fórum
<b>&lt;aside&gt;</b>	Representa uma parte de conteúdo relacionado com a página
<b>&lt;command&gt;</b>	Um botão, um radiobutton ou checkbox
<b>&lt;details&gt;</b>	Descreve detalhes de um documento ou partes
<b>&lt;summary&gt;</b>	Um caption ou sumário, dentro da tag <details>
<b>&lt;figure&gt;</b>	Pode ser usado para associar uma legenda, como de um gráfico ou de vídeo
<b>&lt;figcaption&gt;</b>	Legenda da figura de seção
<b>&lt;footer&gt;</b>	Usado para indicação de um rodapé de documento ou seção, podendo incluir: data, autor, contatos e direitos autorais.
<b>&lt;header&gt;</b>	Usado para introduzir um documento ou seção. Pode incluir o elemento <nav>
<b>&lt;hgroup&gt;</b>	Usado em uma seção de títulos, usando <b>&lt;h1&gt;</b> para <b>&lt;h6&gt;</b> , onde o maior é o título principal da seção, e os outros são subtítulos
<b>&lt;mark&gt;</b>	Destacar um texto
<b>&lt;meter&gt;</b>	Medição. Usado somente quando há valor máximo e mínimo conhecido

<nav>	Usado para uma seção de navegação (menu)
<progress>	Progresso de uma tarefa.
<ruby>	Anotações em Ruby
<rt>	Explicação da anotação em Ruby
<rp>	Para ser exibido em navegadores que não suportam Ruby
<section>	Seção de um documento, como capítulos, cabeçalhos, rodapés ou outras seções do documento
<time>	Definição de tempo ou data
<wbr>	Quebra de palavra. Para definir uma oportunidade de quebra de linha

## 24. Conclusão

Finalizamos este E-book com a capacitação de desenvolvimento de Web sites em HTML5 e CSS3, e que este seja apenas o começo para a carreira na área de desenvolvimento. Esta é uma das áreas que mais crescem no mercado de trabalho. Oriento que desenvolva um portfólio com as habilidades de desenvolvimento adquiridas neste E-book.