

# HTML CSS

## 1 – Nosso Editor Front-End

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código.

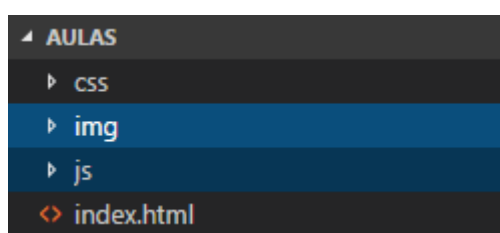
## 2 – Padrão de Projeto

Como todo tipo de projeto de software, existem algumas recomendações quanto à organização dos arquivos de um site.

Como um site é um conjunto de páginas Web sobre um assunto, empresa, produto ou qualquer outra coisa, é comum todos os arquivos de um site estarem dentro de uma só pasta e, assim como um livro, é recomendado que exista uma “capa”, uma página inicial que possa indicar para o visitante quais são as outras páginas que fazem parte desse projeto e como ele pode acessá-las, como se fosse o índice do site.

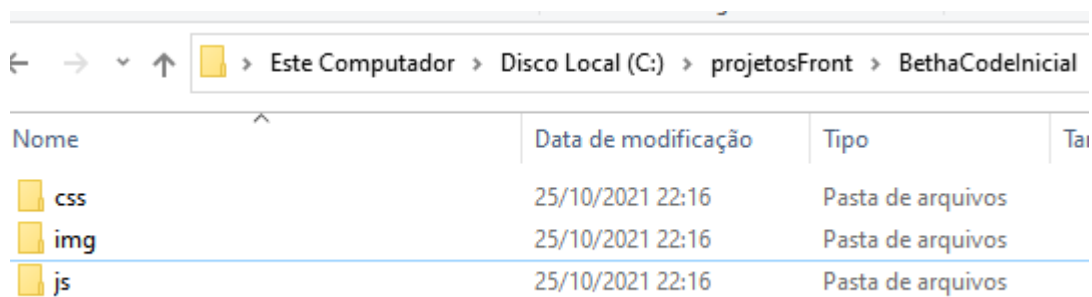
Esse índice, não por coincidência, é convenção adotada pelos servidores de páginas Web. Se desejamos que uma determinada pasta seja servida como um site e dentro dessa pasta existe um arquivo chamado `index.html`, esse arquivo será a página inicial a menos que alguma configuração determine outra página para esse fim.

Dentro da pasta do site, no mesmo nível que o `index.html`, é recomendado que sejam criadas mais algumas pastas para manter separados os arquivos de imagens, as folhas de estilo CSS e os scripts. Para iniciar um projeto, teríamos uma estrutura de pastas como a demonstrada na imagem a seguir:



Muitas vezes, um site é servido por meio de uma aplicação Web e, nesses casos, a estrutura dos arquivos depende de como a aplicação necessita dos recursos para funcionar corretamente. Porém, no geral, as aplicações também seguem um padrão bem parecido com o que estamos adotando para o nosso projeto.

Para nosso primeiro projeto Front-End, vamos criar a pasta projetosFront >> BethaCodeInicial e nesta pasta iremos criar a estrutura indicada acima com as pastas css, img e js, ficando da seguinte forma:

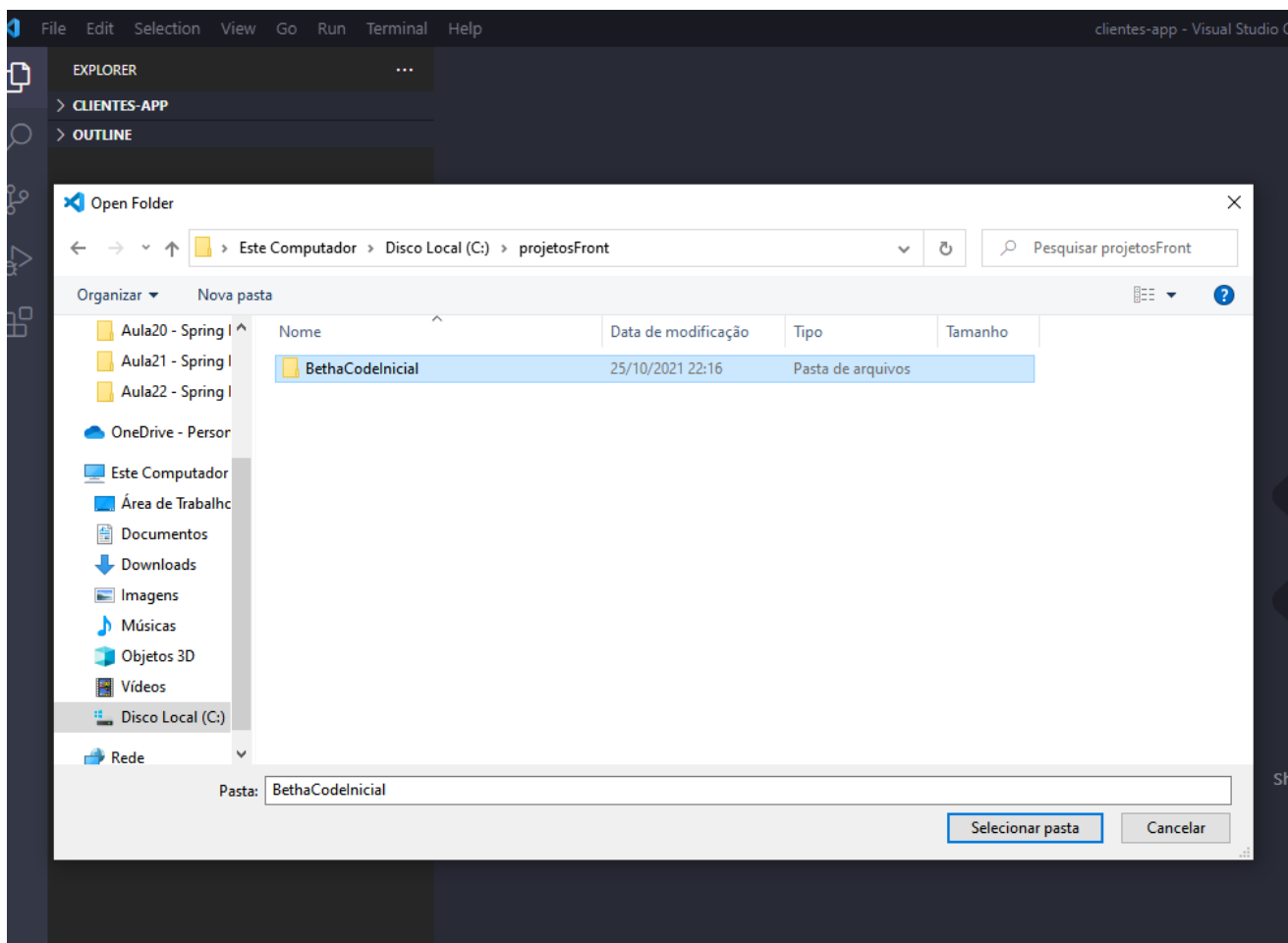


Nome	Data de modificação	Tipo	Tai
css	25/10/2021 22:16	Pasta de arquivos	
img	25/10/2021 22:16	Pasta de arquivos	
js	25/10/2021 22:16	Pasta de arquivos	

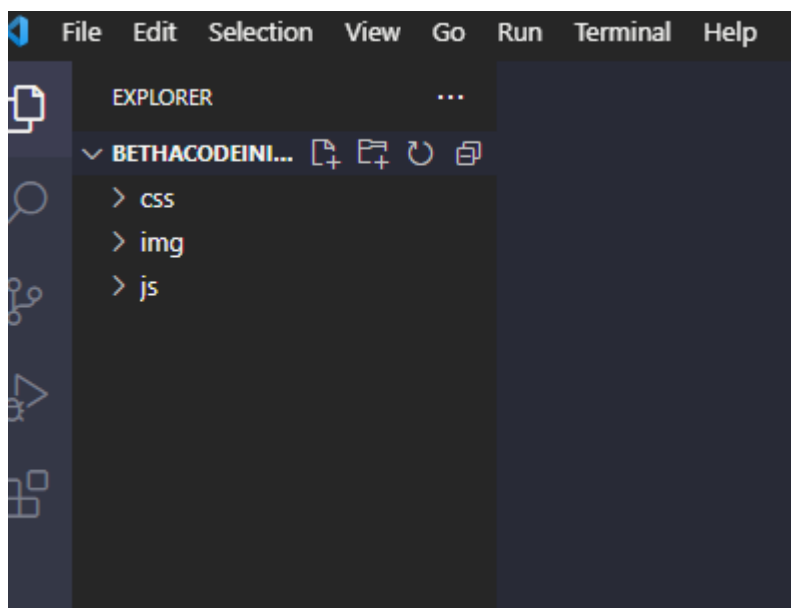
### 3 – Nosso Projeto

Para conhecermos o HTML, iremos criar uma pequena amostra de um site de nosso curso. Vamos criar o BethaCodeLovers.

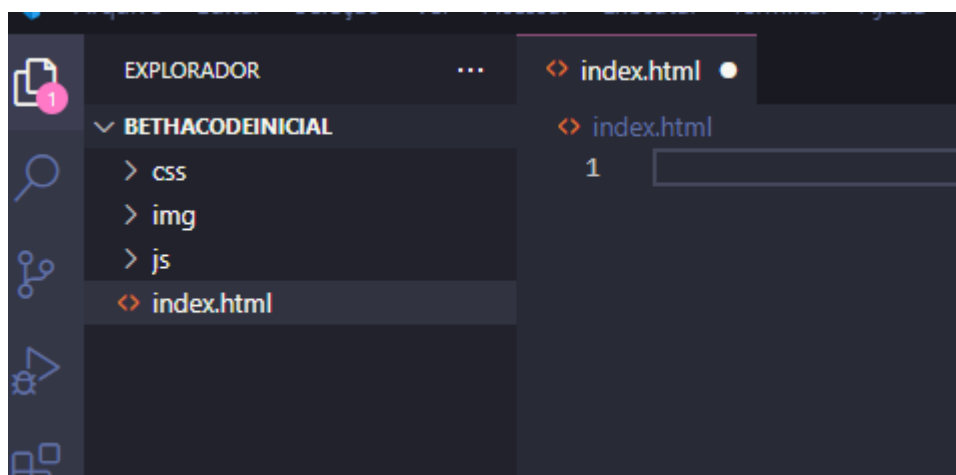
Começaremos abrindo o Visual Studio Code. Nosso primeiro passo é abrirmos a pasta que criamos com nossa estrutura de projeto, então clicamos em File >> Open Folder e selecionamos a nossa pasta BethaCodeInicial.



Será demonstrada na estrutura:



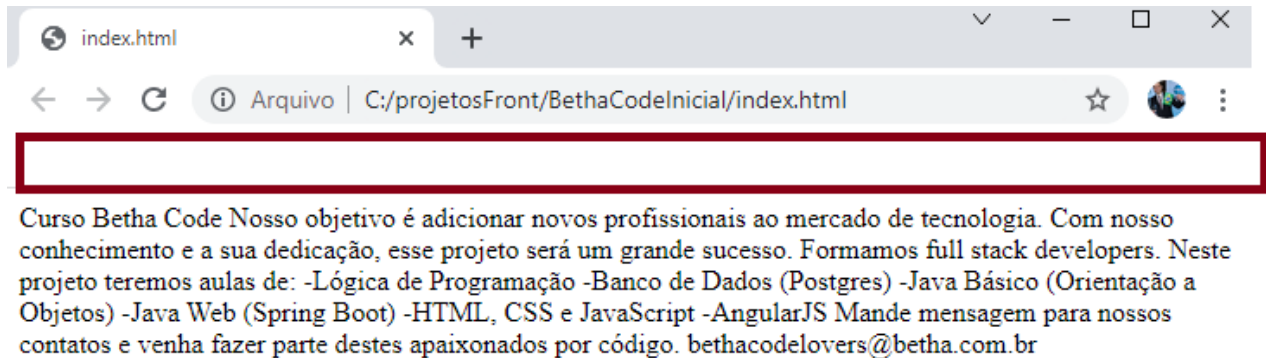
Então, vamos criar o nosso arquivo principal do projeto. Aquele que será executado inicialmente o index.html. Clicar em new File e informar index.html.



Em nosso arquivo index.html, vamos adicionar um texto, para que seja demonstrado para nossos usuários e futuros betha coders lovers.

```
<> index.html X
<> index.html
1  Curso Betha Code
2
3  Nosso objetivo é adicionar novos profissionais ao mercado de tecnologia.
4
5  Com nosso conhecimento e a sua dedicação, esse projeto será um grande sucesso.
6
7  Formamos full stack developers.
8
9  Neste projeto teremos aulas de:
10
11 -Lógica de Programação
12 -Banco de Dados (Postgres)
13 -Java Básico (Orientação a Objetos)
14 -Java Web (Spring Boot)
15 -HTML, CSS e JavaScript
16 -AngularJS
17
18
19 Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código.
20
21 bethacodelovers@betha.com.br
```

Então vamos abrir o nosso arquivo HTML, para isso vamos na pasta onde temos o index.html e iremos indicar para o mesmo ser aberto com Google Chrome. Então teremos o resultado abaixo



Isso porque adicionamos apenas informações como um texto, então nosso navegador, interpretou desta forma.

HTML(Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias além do HTML geralmente são usadas para descrever a aparência/apresentação (CSS) ou a funcionalidade/comportamento (JavaScript) de uma página da web.

"Hipertexto" refere-se aos links que conectam páginas da Web entre si, seja dentro de um único site ou entre sites. Links são um aspecto fundamental da web. Ao carregar conteúdo na Internet e vinculá-lo a páginas criadas por outras pessoas, você se torna um participante ativo na world wide web.

O HTML usa "Marcação" para anotar texto, imagem e outros conteúdos para exibição em um navegador da Web. A marcação HTML inclui "elementos" especiais, como `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, `<span>`, `<img>`, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>`, `<ul>`, `<ol>`, `<li>` e muitos outros.

Um elemento HTML é separado de outro texto em um documento por "tags", que consistem no nome do elemento entre "<" e ">". O nome de um elemento dentro de uma tag é insensível a maiúsculas e minúsculas. Isto é, pode ser escrito em maiúsculas, minúsculas ou uma mistura. Por exemplo, a tag `<title>` pode ser escrita como `<Title>`, `<TITLE>` ou de qualquer outra forma.

## 4 – Estrutura Básica de HTML

A estrutura de um documento HTML apresenta os seguintes componentes básicos:

```
<html>
  <head>
    <meta charset="utf-8">
    <title> Título do Documento </title>
  </head>
  <body>
    textos,
    imagens,
    links, etc...
  </body>
</html>
```

**<HEAD>** contém as informações sobre o documento. A etiqueta HTML `<meta charset="utf-8">` significa que: meta é usado para declarar metadados (informações sobre a própria página, como resumo do conteúdo, palavras chaves, indicações e robôs de busca, entre outros), charset é usado para indicar o formato de codificação de caracteres usado no document e utf-8 é um formato de codificação, ou seja, é usado para mostrar ao navegador o tipo de codificação que será utilizada no site. O elemento `<Title>` define um título que é mostrado no alto da janela do browser.

**<BODY>** tudo que estiver contida nesta tag, será mostrada na janela principal do browser, sendo apresentado ao leitor. Em `<BODY>` pode conter cabeçalhos, parágrafos, listas, tabelas, links para outros documentos, imagens, formulários, animações, videos, sons e scripts embutidos.

Iremos adicionar ao nosso projeto as TAGS básicas de um documento html. Adicionamos a tag html principal. Dentro de html, definimos o head e body. No head temos as informações gerais do documento, no caso com informação da codificação do documento utf-8 e o título que deve aparecer no browser para nossa página. No body temos todo o conteúdo de nosso site, mas ainda sem as tags específicas.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Betha Code Lovers</title>
  </head>

  <body>
    Curso Betha Code

    Nosso objetivo é adicionar novos profissionais ao mercado de tecnologia.

    Com nosso conhecimento e a sua dedicação, esse projeto será um grande sucesso.

    Formamos full stack developers.

    Neste projeto teremos aulas de:

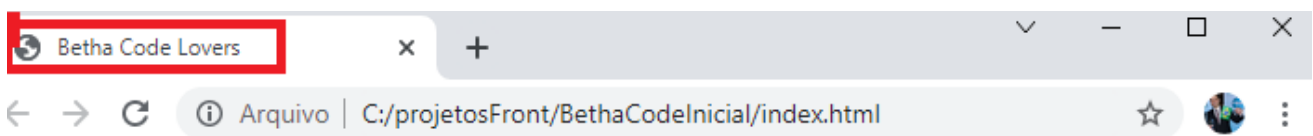
    -Lógica de Programação
    -Banco de Dados (Postgres)
    -Java Básico (Orientação a Objetos)
    -Java Web (Spring Boot)
    -HTML, CSS e JavaScript
    -AngularJS

    Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código.

    bethacodelovers@betha.com.br
  </body>
</html>

```

Executando o nosso arquivo teremos agora o título do site, mas ainda não teremos a formatação do documento.



Curso Betha Code Nosso objetivo é adicionar novos profissionais ao mercado de tecnologia. Com nosso conhecimento e a sua dedicação, esse projeto será um grande sucesso. Formamos full stack developers. Neste projeto teremos aulas de: -Lógica de Programação -Banco de Dados (Postgres) -Java Básico (Orientação a Objetos) -Java Web (Spring Boot) -HTML, CSS e JavaScript -AngularJS Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código. bethacodelovers@betha.com.br

## 5 – Elementos de Texto HTML

As Tags HTML possuem uma função muito importante: informar ao navegador a estrutura do site, quais são os elementos, como os textos vão estar dispostos, e de que maneira tudo isso vai se encaixar. Basicamente você vai montar uma estrutura que vai ser interpretada pelo navegador, e que

por sua vez vai disponibilizar todas essas informações para que qualquer pessoa que acesse a página possa visualizar.

## 5.1 - Tags de Título

Elaborar um título de um artigo, seja título principal ou secundário, parte de um trabalho feito com Tags HTML, e por isso também é uma forma de você conseguir explorar todas as possibilidades de como organizar o conteúdo. Elas são apresentadas da seguinte maneira:

```
<h1></h1> - Título mais importante da página  
<h2></h2>  
<h3></h3>  
<h4></h4>  
<h5></h5>  
<h6></h6> - Título menos importante da página
```

A variação entre os títulos já podem fazer um texto de muitas palavras. Primeiro é pensado um título para o h1, o título principal. Ele é muito importante, e demanda inclusive técnicas de **SEO** para a definição das palavras que serão utilizadas.

O h2 ou h3 é reservado para títulos secundários. Como um desenvolvimento do texto. O primeiro subtítulo importante. Já o h4, h5 e h6 são pouco utilizados e possuem baixo valor hierárquico em relação aos demais. Essa divisão é fundamental para o trabalho de **SEO** e consequentemente para o alcance de sua página.

## 5.2 - Tags de Texto

Entendendo então como você começa a construir um conteúdo escrito, é preciso também pensar em estratégias para o estabelecimento de textos. Eles servem para ornamentar o texto ou para indicar alguns detalhes que precisam ser de fato destacados. Alguns deles são:



`<p></p>` – É a tag que vai definir o início de um parágrafo. Onde o seu texto vai começar.

`<span></span>` – São tags muito próximas da `<p>` mas possuem uma função diferente. Ela é utilizada para outros elementos como imagens, legendas ou informações que vão somar ao texto. Formam uma espécie de container.

`<pre></pre>` – Utilizada principalmente para a utilização de textos pré formatados, como códigos.

`<b></b>` – Transforma todo o texto dentro desse intervalo em negrito.

`<i></i>` – Transforma todo o texto dentro desse intervalo em itálico.

`<br/>` – Tag de HTML utilizada para quebra de linha. Não necessita de fechamento.

`<hr/>` – Formação de linha horizontal no texto.

Conhecendo algumas tags de formatação de texto, iremos realizar adaptações em nosso texto. Utilizamos as tags `<h1>` e `<h3>` para indicação de títulos. A tag `<p>` serve para definir paragrafos. A tag `<i>` indica que a palavra deve ficar em itálico e a tag `<strong>` que a mesma deve ficar em negrito. Ainda temos as tags `<ul>` e `<li>` que permite a criação de uma lista em nosso site.

```
<body>
  <h1>Curso Betha Code</h1>

  <h3>Nosso objetivo é adicionar novos profissionais ao mercado de tecnologia.</h3>

  <p>Com nosso conhecimento e a sua dedicação, esse projeto será um grande sucesso.</p>

  <p>Formamos <strong>full stack</strong> developers.</p>

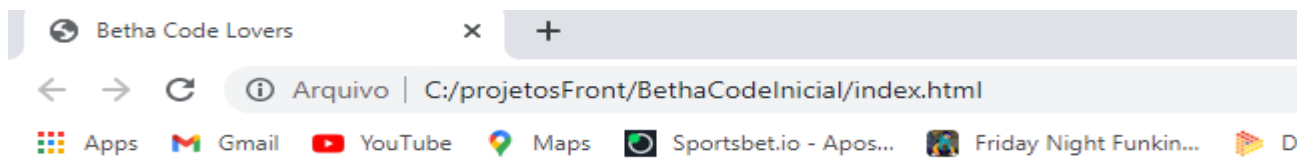
  <br>
  <br>
  <br>
  <br>

  <p>Neste projeto teremos aulas de:</p>

  <ul>
    <li>Lógica de Programação</li>
    <li>Banco de Dados (Postgres)</li>
    <li>Java Básico (Orientação a Objetos)</li>
    <li>Java Web (Spring Boot)</li>
    <li>HTML, CSS e JavaScript</li>
    <li>AngularJS</li>
  </ul>

  <p>Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código.</p>

  <p><i>bethacodelovers@betha.com.br</i></p>
  <p>BETHA SISTEMAS</p>
</body>
```



# Curso Betha Code

**Nosso objetivo é adicionar novos profissionais ao mercado de tecnologia.**

Com nosso conhecimento e a sua dedicação, esse projeto será um grande sucesso.

Formamos **full stack** developers.

Neste projeto teremos aulas de:

- Lógica de Programação
- Banco de Dados (Postgres)
- Java Básico (Orientação a Objetos)
- Java Web (Spring Boot)
- HTML, CSS e JavaScript
- AngularJS

Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código.

*[bethacodelovers@betha.com.br](mailto:bethacodelovers@betha.com.br)*

## 5.3 – Imagens

A tag `<img>` define uma imagem em uma página HTML e necessita de dois atributos preenchidos: `src` e `alt`. O primeiro aponta para o local da imagem e o segundo, um texto alternativo para a imagem caso essa não possa ser carregada ou visualizada.

O HTML 5 introduziu duas novas tags específicas para imagem: `<figure>` e `<figcaption>`. A tag `<figure>` define uma imagem com a conhecida tag `<img>`. Além disso, permite adicionar uma legenda para a imagem por meio da tag `<figcaption>`.

Estamos adicionando duas imagens ao nosso projeto. No caso o `src` indica o caminho e a imagem, enquanto `alt`, será o texto demonstrado, quando não carregada a imagem pelo site.

```
<h1>Curso Betha Code</h1>

<figure>
  
</figure>
```

```
<figure>
  
</figure>

<p>Neste projeto teremos aulas de:</p>
```

## 6 – CSS

CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS foi desenvolvido pelo W3C (World Wide Web Consortium) em 1996, por uma razão bem simples. O HTML não foi projetado para ter tags que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site. A relação entre HTML e CSS é bem forte. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

CSS não é tecnicamente uma necessidade, mas provavelmente você não gostaria de olhar para um site que usa apenas HTML, pois isso pareceria completamente abandonado.

### 6.1 – Vantagens do CSS

A diferença entre um site que implementa CSS e outro que não o usa é gigantesca e notável. Você já deve ter visto um site que não carrega completamente ou tem um plano de fundo branco com texto azul e preto. Isso significa que a parte CSS do site não foi carregada corretamente ou não existe.

Antes de usar CSS, toda a estilização tinha que ser incluída na marcação HTML. Isso significa que você deveria descrever separadamente todo o plano de fundo, as cores das fontes, os alinhamentos, etc. Mas o CSS permite que você estilize tudo em um arquivo diferente, criando assim o estilo separadamente. E, mais tarde, faça integração do arquivo CSS na parte superior da marcação HTML. Isso mantém a marcação HTML limpa e fácil de manter.

Resumindo, com o CSS você não precisa mais escrever repetidamente como os elementos individuais se parecem. Isso economiza tempo, encurta o código e diminui a chance de erros. O CSS permite que você tenha vários estilos em uma página HTML, tornando as possibilidades de personalização quase infinitas. Hoje em dia, isso está se tornando mais uma necessidade do que um simples recurso.

## 6.2 – Como CSS Funciona

O CSS usa uma sintaxe simples baseada em inglês com um conjunto de regras que o governam. Como mencionamos anteriormente, o HTML nunca teve a intenção de usar elementos de estilo, apenas a marcação da página. Foi criado para descrever apenas o conteúdo. Por exemplo: `<p>Este é um parágrafo.</p>`.

A estrutura da sintaxe CSS é bem simples. Tem um seletor e um bloco de declaração. Você seleciona um elemento e depois declara o que deseja fazer com ele. O seletor aponta para o elemento HTML que você deseja estilizar. O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula.

Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos. Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves.

No exemplo estamos adicionando CSS diretamente no HTML, para isso colocamos a `<style>`, depois basta informarmos o elemento a ser estilizado e adicionar as suas propriedades.

```
<style>
  p{
    color: ■crimson;
    background-color: ■cyan;
  }
</style>

<style>
  h1{
    color: ■blueviolet;
    background-color: ■coral;
  }
</style>
```

## 6.3 – Criando arquivo CSS

O correto é criarmos um arquivo CSS com as estilizações e vincular ao arquivo HTML. Desta forma organizando nosso projeto. Para isso iremos criar na pasta css o arquivo estilo.css, com as estilizações que estão na tag <style>.



No arquivo HTML, iremos retirar as tags <style> e no <header> do arquivo iremos vincular o arquivo de CSS. Adicionamos a tag <link>, no atributo rel temos a informação stylesheet, no type adicionamos text/css e no href temos a informação do arquivo css/estilo.css.

```
<head>
  <meta charset="utf-8">
  <title>Beta Code Lovers</title>
  <link rel="stylesheet" type="text/css" href="css/estilo.css">
</head>
```

## 6.4 – Elementos Básicos do CSS

Podemos adicionar as estilizações de CSS de três formas (Tags HTML, classe e id).

No caso de Tag HTML é a forma que estamos fazendo até o momento, adicionamos o elemento HTML no arquivo CSS, adicionamos a estilização e todos os elementos deste tipo receberão as propriedades no HTML.

Por classe, será utilizando quando queremos que apenas alguns elementos com a tag, recebam determinada estilização, para isso adicionamos uma classe no HTML e adicionamos a sua estilização no arquivo CSS. No HTML temos o atributo class, onde indicamos o nome da classe. No CSS adicionamos o .(ponto) e o nome da classe adicionada, então pode adicionar estilizações para as tags que utilizam a classe.

```
<p class="testeDeClasse">Neste projeto teremos aulas de:</p>
```

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

Por id, será utilizando quando queremos que apenas alguns elementos com a tag, recebam determinada estilização, para isso adicionamos uma classe no HTML e adicionamos a sua estilização no arquivo CSS. No HTML temos o atributo id, onde indicamos o nome do elemento. No CSS adicionamos #.(cerquilha) e o nome do id adicionado.

```
<p id="testeDeID">Neste projeto teremos aulas de:</p>
```

```
#testeDeID{  
  color: blue;  
}
```

## 6.5 – Nosso CSS

Em nosso HTML, iremos adicionar o id (imagemPrincipal) na primeira imagem e na segunda imagem o id (imagemSecundaria). Nos parágrafos para a tag <p>, iremos adicionar a classe projeto para os itens indicados na imagem abaixo.

```
<figure>
  
</figure>
```

```
<figure>
  
</figure>

<p class="projeto">Neste projeto teremos aulas de:</p>

<ul>
  <li>Lógica de Programação</li>
  <li>Banco de Dados (Postgres)</li>
  <li>Java Básico (Orientação a Objetos)</li>
  <li>Java Web (Spring Boot)</li>
  <li>HTML, CSS e JavaScript</li>
  <li>AngularJS</li>
</ul>

<p class="projeto">Mande mensagem para nossos contatos e venha fazer parte destes apaixonados por código.</p>

<p class="projeto"><i>bethacodelovers@betha.com.br</i></p>
<p>BETHA SISTEMAS</p>
```

Em nosso CSS, temos o elemento body que equivale a toda a página. Definimos a cor de fundo, do texto e que o tamanho da página equivale a 80% do layout disponível. Nos elementos <h1>, <h3> e <p> adicionamos o alinhamento de texto centralizado.

No elemento HTML com o id (imagemPrincipal) indicamos que deve possuir 60% do layout disponível, deve utilizar FLOAT a esquerda, indicando que pode ter outros elementos na mesma posição e o padding indica que deve ter 4% de espaço entre a imagem e o texto.

No elemento HTML com o id (imagemSecundaria), temos o tamanho de 35% do layout disponível e o FLOAT a direita, indicando que a esquerda pode ter outros elementos.

Nos elementos HTML que utilizam a classe (projeto), terão o texto alinhado a esquerda.

```

body {
  background-color: ■ whitesmoke;
  width: 80%;
  color: ■ blue;
}

h1, h3, p{
  text-align: center;
}

#imagemPrincipal{
  width: 60%;
  float: left;
  padding-right: 4%;
}

#imagemSecundaria{
  width: 35%;
  float: right;
}

.projeto{
  text-align: left;
}

```

## 7 – Link

Desde a origem da internet, o que torna a web o que ela é hoje é justamente a possibilidade de vincular um documento a qualquer outro documento ou recurso. Essa função é chamada de hiperlink ou link.

O link HTML, nada mais é que uma função HTML que permite inserir os hiperlinks em diversos elementos, como textos e imagens. Um link precisa sempre apontar para uma URL (endereço) existente em seu site. Caso contrário, você poderá obter uma mensagem de erro 404. A tag link do HTML está presente desde a primeira versão, criada por Tim Berners-Lee.

No HTML, os links são definidos pela tag <a>. Dentro dessa tag incluímos o atributo href (Hypertext Reference), que é o endereço de destino do link. Dentro do conteúdo da tag <a>, incluímos então o texto ou elemento que servirá como redirecionador, ou seja, que ao ser clicado, executará a função de redirecionar para o endereço dentro do atributo href. Dessa forma, a sintaxe básica do HTML\_link é:

```
<a href="url">Exemplo</a>
```



A tag `<a>` pode ser utilizada dentro ou fora dos demais elementos do HTML, como no exemplo abaixo, apenas o texto Betha Sistemas contém um hiperlink para a página inicial da Betha:

```
<a class="projeto" href="https://www.betha.com.br/">Betha Sistemas</a>
```

Podemos ainda adicionar o href informando um endereço de e-mail, com a sugestão de assunto, quando clicar no link, será aberta a ferramenta de e-mail com os dados adicionados ao hiperlink.

```
<a class="projeto" href="mailto:bethacoders@betha.com.br?Subject=Sou um Coder">bethacodelovers@betha.com.br</a>
```

Neste ponto iremos criar uma nova página com as informações do Full Stack Developer, para linkarmos no texto de nossa página.

Vamos criar o arquivo fullstack.html e adicionarmos o seguinte conteúdo. Importante notar que no final temos a opção de Voltar, linkado com o nosso index.html.

```
<html>
  <head>
    <title>FULL STACK DEVELOPERS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/estilo.css">
  </head>

  <body>
    <h1>Este é o Coringa do desenvolvimento de sistemas!</h1>
    <h3>O desenvolvedor Full Stack é aquele que pode atuar em qualquer etapa do desenvolvimento de sistemas.</h3>
    <h3>Seja no Front-End ou no Back-End.</h3>

    <figure>
      
    </figure>

    <p><a href="index.html">Voltar</a></p>
  </body>
</html>
```

Estamos reaproveitando o nosso estilo.css, desta forma iremos apenas adicionar o detalhamento da nova imagem no arquivo.

```
#fullstack{
  width: 100%;
}
```

Em nosso index.html, iremos adicionar o link para a nova página criada, apenas indicando o arquivo no atributo href.

```
<p>Formamos <strong><a href="fullstack.html">Full stack</a></strong> developers.</p>
```

## 7 – Bootstrap

O Bootstrap é um framework front-end, atraente e muito intuitivo, para o desenvolvimento web, mais rápido e eficaz. Ele tem suporte para todos os navegadores populares.

Para utilizar o Bootstrap é necessário conhecimento em HTML e CSS. Seu design é totalmente responsivo, sendo ajustado automaticamente para desktops, tablets e celulares.

Basicamente foi desenvolvido um CSS pela equipe do twitter com vários componentes e alguns códigos JavaScript. Desta forma precisamos adiciona-los a nossa página HTML e usar as classes conforme definição dos codificadores deste framework.

Podemos verificar as informações no site oficial do Bootstrap. <https://getbootstrap.com.br/>. Para testarmos o Bootstrap iremos criar em nosso projeto o arquivo teste.html. Neste arquivo iremos copiar o conteúdo de Starter Template no site do Bootstrap e colar em nossa página.

Pode ser verificado que no HTML, temos a importação do CSS do Bootstrap e também a utilização de um JavaScript. Agora podemos utilizar as classes estilizadas do Bootstrap.

```

<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <!-- Meta tags Obrigatórias -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

    <title>Olá, mundo!</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>

    <!-- JavaScript (Opcional) -->
    <!-- jQuery primeiro, depois Popper.js, depois Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41J3" ></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384" ></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384" ></script>
  </body>
</html>

```

Neste iremos apenas adicionar a tag <div>, com a classe container do BootStrap. Na tag <h1> iremos adicionar o display-3 m5.

```

<div class="container">
  <h1 class="display-3 m-5">Olá, mundo!</h1>
</div>

```

---

# Olá, mundo!

Iremos alterar nosso código para adicionar uma tabela, apenas colocando a classe table.

```
<div class="container">
  <table class="table">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Primeiro</th>
        <th scope="col">Último</th>
        <th scope="col">Nickname</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row">1</th>
        <td>Tiago</td>
        <td>Valério</td>
        <td>tiagorv</td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td>João</td>
        <td>Silva</td>
        <td>Jhon</td>
      </tr>
      <tr>
        <th scope="row">3</th>
        <td>Manoel</td>
        <td>Oliveira</td>
        <td>chuck</td>
      </tr>
    </tbody>
  </table>
</div>
```

#	Primeiro	Último	Nickname
1	Tiago	Valério	tiagorv
2	João	Silva	Jhon
3	Manoel	Oliveira	chuck

Apenas alterar a class para table table-dark, teremos outro resultado.

```
<table class="table table-dark">
```

#	Primeiro	Último	Nickname
1	Tiago	Valério	tiagorv
2	João	Silva	Jhon
3	Manoel	Oliveira	chuck

Por última, iremos verificar o que mais nos interessa, os formulários. Neste criamos o formulário utilizando as classes do bootstrap. Temos a informação do e-mail, senha, checkbox e um botão do tipo primary.

```
<div class="container">
  <h1>Cadastro de Usuários</h1>
  <form>
    <div class="form-group">
      <label for="exampleInputEmail1">Endereço de email</label>
      <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Seu email">
      <small id="emailHelp" class="form-text text-muted">Nunca vamos compartilhar seu email, com ninguém.</small>
    </div>
    <div class="form-group">
      <label for="exampleInputPassword1">Senha</label>
      <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Senha">
    </div>
    <div class="form-group form-check">
      <input type="checkbox" class="form-check-input" id="exampleCheck1">
      <label class="form-check-label" for="exampleCheck1">Clique em mim</label>
    </div>
    <button type="submit" class="btn btn-primary">Enviar</button>
  </form>
</div>
```

## 8 – Material Complementar

<https://www.youtube.com/watch?v=SV7TL0hxmIQ>

<https://www.caelum.com.br/apostila/apostila-html-css-javascript.pdf>

<https://www.youtube.com/watch?v=wwqOJ2o84S4>

<https://www.youtube.com/watch?v=vHSYytrtUSw>

**Exercícios**

- 1 – Escolha um template do BooStrap e faça um pequeno site.
- 2 – Crie alguns formulários com BooStrap que representem seus cadastros de seu Projeto Final.