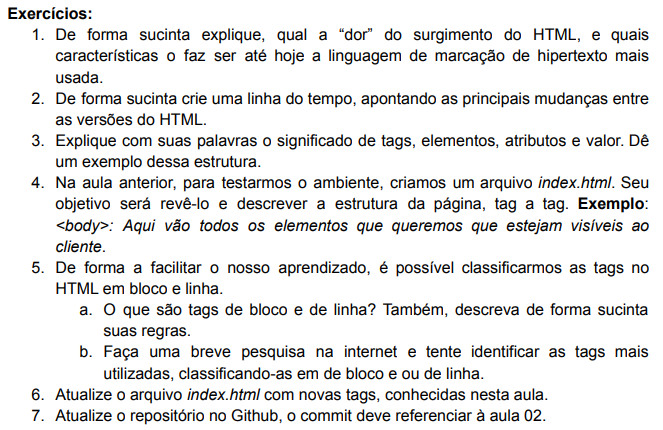
ATIVIDADE 02



**Q1)**

Conversando com alguns colegas, chegamos no consenso de que a primeira parte da questão se refere ao problema existente na época que teve como consequência a necessidade de uma mudança. Mudança essa que ocasionou na criação do HTML para sanar o problema. Levando isso em consideração, em 1991 o HTML foi criado por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na suíça. Com o objetivo de interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Posteriormente com o desenvolvido da rede de alcance mundial o HTML foi se popularizando a nível global. É importante comentar que ela foi a primeira e, portanto, acabou sendo o padrão nos navegadores décadas atrás, possuindo uma equipe voltada em melhorar a linguagem adicionando mais recursos a cada nova versão. Atualmente está na versão HTML5 e sendo a mais empregada, possui diversos recursos, é fácil de aprender e usar, é leve e rápido de carregar, não exige nenhum software extra, pois já vem como padrão em cada navegador e sintaxe livre, apesar de diversas boas práticas presentes na comunidade de desenvolvedores.

.

**Q2)**

Linha do tempo do HTML:

1991: HTML 🡪 1995: HTML 2.0 🡪1997: HTML 3.2 🡪 1999: HTML 4.01 🡪 2014: HTML5.

O criador do Hypertext Markup Language (HTML) chama-se Tim Berners-Lee, utilizando o ambiente de desenvolvimento NeXTSTEP. Na época, a linguagem não era uma especificação, mas uma coleção de ferramentas para resolver um problema: a comunicação e disseminação das pesquisas entre Tim e o seu grupo de colegas.

O HTML 2.0 foi desenvolvido pelo IETF. Eles definiram o padrão para os principais recursos HTML.

O HTML 3.2 adicionou recursos, como tabelas, applets, fluxo de texto em torno de imagens, sobrescritos e subscritos, ao mesmo tempo em que fornece compatibilidade com o padrão HTML 2.0.

O HTML 4.01 adicionou suporte para mais opções multimídia, linguagens de script, folhas de estilo (style sheets), melhores recursos de impressão e documentos mais acessíveis a usuários com deficiências.

O HTML5 tem uma série de recursos adicionais em relação aos seus antecessores, como um suporte a armazenamento de mídias offline, elementos de conteúdo mais específicos (como rodapé, cabeçalho, navegação, etc.), doctype inline mais simples, suporte para gráficos vetoriais, áudio e suporte à incorporação de vídeo.

Q3)

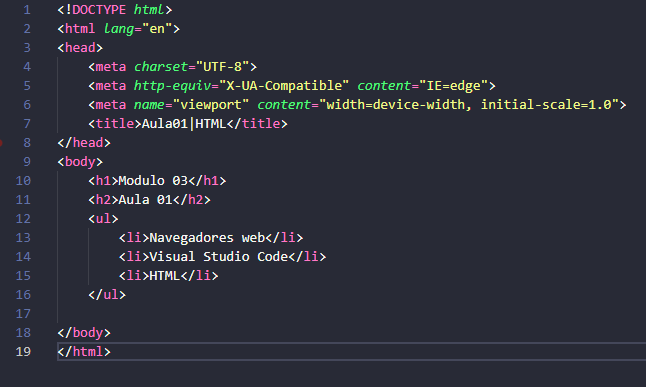
TAG, ELEMENTO, ATRIBUTO E VALOR

**Tag** é um conjunto de caracteres que ficam entre “<” e “>” ou “</” e “>”, por exemplo, <p>, </h1>, <a>, etc. Existem dois tipos de Tags, as que necessitam de fechamento e as que não necessitam de fechamento.

Utilizando as tags é possível formar um **elemento**. O conteúdo do elemento fica entre as tags. Por exemplo, se quisermos exibir um parágrafo em um site utilizamos a Tag <p> que semanticamente quer dizer que o conteúdo que se espera nesse Elemento é um parágrafo. As Tags servem para marcar onde começam e terminam os Elementos. Já os Elementos são uma parte conceitual/semântica que têm um começo e fim determinados.

**Atributos** são informações que passamos na Tag para que ela se comporte da maneira esperada. Existem atributos globais (que funcionam em todas as Tags) e específicos (que são direcionados para cada Tag, através de especificação). Os Atributos possuem um “nome” e um “**valor**”, existem atributos que você vai usar praticamente sempre e existem outros que serão mais raros. Alguns exemplos de atributos: id, style, title, href, etc. Dependendo do tipo de atributo, o valor associado pode ser um endereço, um código de cor, um caminho, uma string, entre outras coisas.

Q4) Visitando o index.html da aula 01, temos a imagem abaixo:



Tags:

**<html>** Ela serve para definir o início e fim do documento.

**<head>** Serve para definir informações gerais (metadados) sobre o documento, como título, links para scripts, folhas de estilos (style sheets). O conteúdo dela não é visível no navegador.

**<meta>** define informação de metadados tais como base, link, script, style ou title.

**<title>** Define o título do documento, sendo mostrado na aba da página.

**<body>** Aqui colocamos todos os elementos que queremos que seja visível para o usuário.

**<h1>** Representa o nível mais alto de título de seção, é uma boa prática colocar apenas um desse na página inteira.

**<h2>** Representa o segundo nível mais alto de título de seção.

**<ul>** Representa uma lista de itens sem ordem, isto é, não formam uma ordem numérica e os elementos que estão dentro dela tem posição irrelevante.

**<li>** Representa um item que faz parte de uma lista, precisa está contido dentro de uma lista ordenada (ol) ou uma lista desordenada (ul).

Q5a) Um elemento em nível de **bloco** ocupa todo o espaço de seu elemento pai (container), criando assim um “bloco”. Navegadores normalmente mostram o elemento em nível de bloco com uma nova linha antes e depois do elemento. Os elementos **inline** podem ser exibidos em nível de bloco ou outros elementos inline. Eles ocupam somente a largura de seu conteúdo. Por padrão, os elementos no nível de bloco começam em novas linhas, mas, os elementos em linha, podem iniciar em qualquer lugar. Geralmente, os elementos no nível de bloco podem conter elementos em linha e, às vezes, outros elementos no nível de bloco. Inerente a essa distinção estrutural está a ideia de que elementos de bloco criam estruturas "maiores" que elementos em linha. Geralmente, elementos inline devem ter somente dados em outros elementos inline. Por padrão, os elementos inline não começam em uma nova linha.

Q5b)

**Tags bloco:**

<p> O conteúdo que fica entre as tag de abertura e fechamento normalmente é um parágrafo.

<section> Define uma seção da página

<div> Divisão de conteúdo.

<nav> Contém links de navegação, normalmente dentro dessa tag são colocados elementos que apresentam a tag <a>.

<header> Cabeçalho de seção ou página.

<main> Contém o conteúdo principal do documento.

<footer> Define o rodapé de uma página

<h1>, <h2>,...,<h6> Define títulos para a página. O primeiro é o título principal e é uma boa prática colocar apenas um desse por página, os demais são subtítulos.

<ul> Define lista não ordenada.

<ol> define lista ordenada, por exemplo, os elementos da lista recebem uma numeração, ou segue ordem alfabética.

<li> Representa um elemento da lista.

**Tags inline:**

<img> Utilizada para adicionar uma imagem

<a> Definir uma âncora, que contém um hyperlink para um outro local.

<script> Utilizado para adicionar um vínculo com um arquivo script, por exemplo, javascript. Elemento é usado para incorporar código executável ou dados.

<button> Define que o elemento é um botão, normalmente botões são associados a eventos.

<input> Há várias formas de input, define um espaço separado para receber algum tipo de dado de entrada do usuário. Costuma ser usado junto com a tag label, que descreve o tipo de dado que deve ser inserido

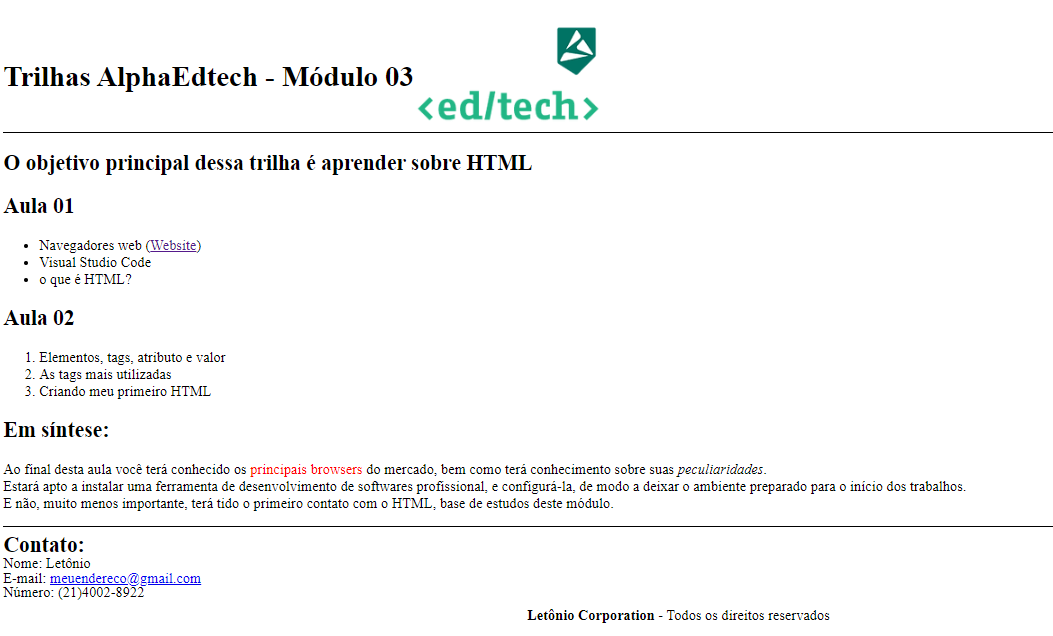
<span> é um contêiner inline para conteúdo, normalmente usado para agrupar elementos para fins de estilo.

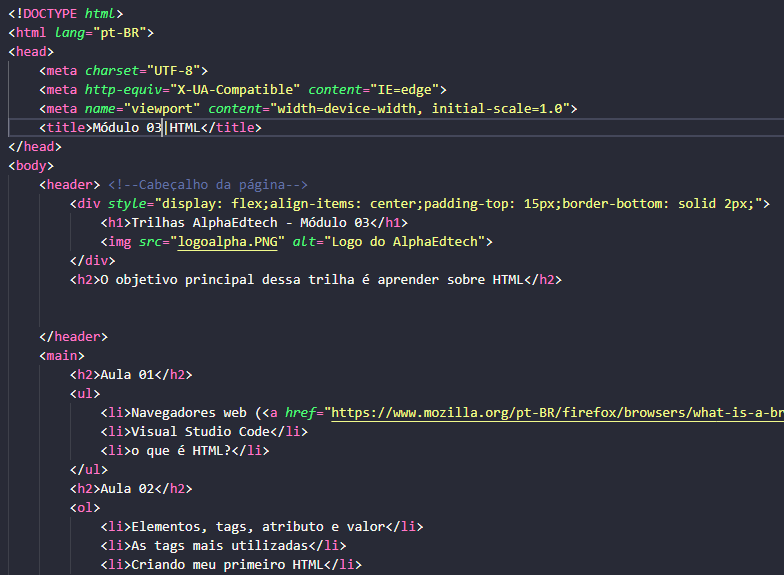
<label> O elemento representa uma legenda para um item em uma interface de usuário (input).

<br> Permite pular para a próxima linha, logo, o próximo conteúdo começará na linha seguinte.

<b> e <i> O conteúdo que fica entre a tag de abertura e fechamento ficará em negrito e itálico, respectivamente.

Q6) Atualizei o arquivo, ficou da seguinte forma:



Q7) Atualizei o repositório:

