HTML e CSS – Cursos em Vídeo

Recomendações de bibliografias para estudar HTML e CSS.

* Referência MDN
* 3wc Standards
* WHATWG Living standard

HTML: Linguagem de Marcação de Hipertexto (Hypertext marckup language)

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 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 da world Wide Web (www).

A história da Internet.

Origem da internet

A internet é fruto da guerra fria.

Quando a União soviética criou o potnic pra fazer a orbita da terra e cair no mar, os estados unidos viram isso como uma possível espionagem que a união soviética estava desenvolvendo, sendo assim com receio de deixar seus dados alocados em um único ponto, eles precisavam transmitir esses dados como uma espécie de Backup para outros computadores ao redor dos estados unidos.

Com essa intenção os Estados Unidos criaram uma pequena rede, chamada de **ARPANET**. Essa que por sua vez começou com 4 pontos nos Estados Unidos, os locais e computadores da época eram os seguintes:

SDS Sigma 7 – Universidade da California

SDS 90 – Universidade de Stanford

IBM 350/75 – Centro da Matemática de Los Angeles

DEC PDP-10 – Base militar de UTA

Como pode ser observado, os computadores são modelos diferentes, alguns de Marcas diferentes. Por conta disso esses computadores apenas conseguiam com aqueles semelhantes, nunca uns com os outros.

Na tentativa de resolver este problema, a Universidade da California criou o protocolo **NCP** (*network control protocol*), que para a época funcionava, por se tratar de uma rede menor com menos computadores se comunicando.

Vale mencionar: A comunicação dos quatro pontos precisava parar para que dois desses pudessem se comunicar e fazer transferência de dados.

Com o crescimento de dispositivos, se tornou impraticável usar o protocolo NCP, pois esse precisava bloquear a comunicação dos demais locais para que dois deles pudessem se comunicar entre si. Foi então que em 1972 o pesquisador Bob Kahn criou o protocolo **TCP** (*transfer control protocol*), porém esse tinha um problema, como a rede estava crescendo de mais, então esse protocolo tinha dificuldade na identificação nesses pontos. Foi então que outro pesquisador *Vint Cerf* que criou um protocolo de identificação de máquinas, o **IP** (*internetwork protocol*). Sendo esse o protocolo que usamos até os dias atuais **TCP/IP*.*** Claro, esses protocolos sofreram atualizações desde sua criação até os tempos atuais.

Por conta da ascensão da rede, os militares não queriam mais se responsabilizar pela ARPANET sendo assim, foram dividias em redes menores. Os militares ficaram com uma parte sendo essa a MILNET, e as universidades ou partes cientificas ficaram com a rede que foi chamada de NSFNET (*national Science fundation*).

Analogia para explicar como funciona a internet.

Quando você (o cliente) faz uma solicitação, como por exemplo pra acessar uma imagem ou video do servidor (aquele que há de fornecer o conteúdo solicitado) ele envia o solicitado em forma de vários pacotes.

É facil observar esse processo ocorrendo, quando a internet está lenta, e a imagem vai chegando aos poucos, isso são os pacotes preenchendo a imagem solicitada.

**DOCUMENTÁRIO DO YOUTUBE**

https://www.youtube.com/watch?v=TNQsmPf24go&t=162s

Como a internet funciona

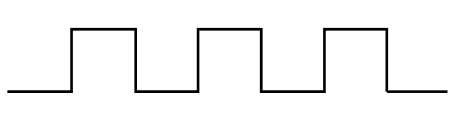
Para entender como funciona a internet, primeiro precisamos entender como funciona a máquina, ou computador. Primeiro precisamos entender que o computador não é tão inteligente quando faz parecer. Ele apenas é rapido na interpretação de sinais e tradução desses sinais em um formado entendível, o que faz dos programadores o verdadeiro cérebro por traz da operação. Esses códigos assim como o átomo para a gente é a forma menor que compõe um corpo, o bit é o que compõe um dado, representado por 0 e 1, sendo, sem sinal representa o 0, e com sinal representa o 1, assim que as informações são transmitidas, isso vale pra qualquer dispositivo eletrônico hoje em dia, de celulares, tablets, smarthwatchs, consoles, etc.

Porém fato é que 2 bits somente, é algo bem limitado, não da pra representar muita coisa com isso, então na computação reúne um conjunto de oito(8) bits (exemplo **01000001**) que é a porção mínima para representar um dado, e a essa sequencia de 8 bits também temos um nome que é o **byte**.

A título de curiosidade esse código utilizado acima **01000001** ele representa a letra A, sendo assim na memória do computador ele não entende pela letra, ela fica armazenada como essa sequencia binária, todo vez que usamos essa letra.

Para saber toda a sequência basta procurar por: **código multibyte UTF-8**

**Na verdade, o computador não vai entender como 0 ou 1, porque esses são estados como visto anteriormente “com sinal” e “sem sinal” ao que chamamos de onda quadrada que pode ser vista na imagem a seguir:**



Então sabendo que esses números apresentados anteriormente representam 1 byte existem também os múltiplos de byte (exemplo: na pesagem a gente utiliza o grama, o quilograma, a tonelada) nos bytes temos o equivalente a isso. Nos bytes a referencia é sempre 1024 bytes. Sabendo que um byte tem 8 números 0 e 1, então com 1024 **bytes** temos 1 **KB**

**Curiosidade: no nosso sistema de numeração temos como base o 10 como em (1, 2, 3, 4, 5, 6, 7, 8 , 9, 10) sendo assim sempre pra subir uma casa decimal temo que utilizar toda essa sequência.**

**Com a computação é diferente, ela tem por base o número 2 (de 0, e 1) então a referência dela é sendo assim 1024 bytes, que é o equivalente a 1 KB, então a tabela de dados fica da seguinte maneira:**

**8 bits = 1 byte**

**1024 bytes = 1KB**

**1024KB = 1MB**

**1024 MB = 1GB**

**1024 GB = 1 TB**

**1024 TB = 1 PB (petabyte)**

**1024 PB = EB (exabyte)**

**1024 EB = 1 ZB**

**1024 ZB = 1YB**

**MB (megabytes) ≠ Mb (megabits)**

O **Megabyte** ele é comumente utilizado quando se fala de armazenamento como 1 Gibabyte de memoria, um Megabyte de arquivo.

O **MegaBITS** é utilizado quando se fala em transmissão de dados como a internet, a velocidade de transmissão de dados de uma internet X é 60 megabits, gibabits, terabits etc.

Quando vamos acessar a internet, primeiro fazendo uma solicitação de arquivos ou de acesso a internet, sendo assim isso faz do usuário co cliente. Porém alguns processos precisam ser feitos para poder chegar os dados solicitados, primeiro, é preciso um dispositivo que conecta a rede de telefone (senoidal) a rede do computador (rede quadrada) o que faz essa conversão de ondas é o dispositivo Modem, tanto pra transmitir dados, quando para receber.

Quando uma onda quadrada é convertida em u a onda senoidal é o que chamamos de **modulação**

E o contrário também é verdade, convertendo a senoidal para onda quadrada é chamado de **demodulação**

Entendido isso, quando acessamos a internet seja para acessar um site, o que exatamente estamos acessando para ter acesso a esses dados?

Isso é o que chamamos de servidores, toda máquina que pode transmitir dados, pode ser considerado um servidor, até mesmo máquinas de uso doméstico, como computador, desde que você crie de maneira adequada uma codificação, como usando serviços de nuvem conectados a funções.

[www.iplocation.net](http://www.iplocation.net)

através desde link é possivel saber o ip local, no caso o ip da máquina que esta sendo usado para acessar ao site.

Além desse existe o IP público, que seria o do roteador que seria outro número de identificação.

Mas além de tudo, neste mesmo link, é possível acessar e saber o link de outros sites, bastando apenas pesquisar pela URL do site de interesse.

É de se notar que como os IPs de site estão constantemente mudando, que necessite haver uma agenda para registrar os sites pelo domínio e não pelo IP (lembrando que o domínio de um site é como ele foi registrado, podendo haver somente um site com aquele nome em específico como em **Instagram**) Então essa “agenda” que salva o site por domínio e não por IP, é o que chamamos de **servidor DNS** (Domain name System – sistema de nome de domínio). Esses servidores DNS são locais especiais na internet ou servidores onde é solicitado o IP atual dos servidores que é de interesse acessar.

Exemplo: Quando solicitado o domínio do instagram, ao invez de acessar diretamente o site, ao que seria improvável já que o ip esta constantemente mudando. É solicitado ao servidor DNS qual o IP atual do domínio Instagram naquele momento, sendo assim ele devolve o número atual do IP solicitado e acessa o site. Por mais que pareça algo complexo, a máquina já realiza todo esse processo de maneira automática, sem precisar que façamos isso manualmente.

E uma vez no site do Instagram, ele envia as imagens, os feeds, os comentários, os conteúdos como um todo nele presente, através de pequenos pacotes em diferentes tempos e toras para ser interpretados pela nossa máquina.

Com isso surge outra questão, o que são essas rotas?

As rotas tal qual uma rodovia são os caminhos que os dados vão percorrer pra chegar até o servidor, e pra fazer os dados para nosso dispositivo.

Exemplo: O servidor do Instagram esta localizado na Europa, na Inglaterra [exemplo]

Estando eu em Curitiba, os dados solicitados vão passar até São Paulo, onde vão para salvador onde passaram por cabos submarinos até Portugal, e então para a Inglaterra, porém pode haver congestionamento em algum desses pontos, sendo assim os dados podem passar por outros caminhos, podendo ser pelos estados Unidos, ou pelo Pacifico chegando na china ou Rússia, pra então chegar no servidor.

O que é domínio e hospedagem?

Criando uma problematização: Uma vez que eu crie um site, e queira compartilhar para que outras pessoas vejam minha criação, eu não posso fazer isso sem ter uma hospedagem de servidor, pois o site uma vez criado ele vai estar os arquivos tanto HTML quando CSS na minha máquina, sendo assim é preciso de um domínio que dê acesso ao servidor onde estará localizado o site.

Para que seja possivel entregar o endereço do site criado para outrem, é preciso de um domínio além também de um servidor, afinal **hospedar** dados em uma máquina doméstica pode ser perigoso

Reforçando conhecimento:

Domínio é o nome que será dado ao site;

* O nome tem que ser único
* Para versões profissionais de domínio, você vai pagar anualmente uma taxa para manter o nome de domínio
* Na hora de escolher, tem que ser escolhido um TLDs (Top level domain’s)

Hospedagem o nome dado ao local onde será guardado o site;

* Espaço para armazenar arquivos
* Pago mensalmente
* Hospedagem você escolhe por: espaço, memória, recursos (se tem e-mail, se suporta banco de dados, se suporta php, hub, python).

URL

*Uniform Resource Locator –* Localizador de recurso Único/Uniforme

Usando como exemplo do curso que estou fazendo:

[www.github.com/gustavoguanabara](http://www.github.com/gustavoguanabara)

**github** é o nome do domínio

**.com**  é o TLD – outros exemplos de TLD’s pode ser (.com, .net, .io, .info, .online, .store)

**Catalogando os TLD’s**

**GTLDs (*Generic top level domain’s)***

.com – instituição comercial

.edu – instituições educacionais

.store – lojas

.gov – instituições governamentais

**CCTLDs (Country Code)**

Como em: .com.br, .us (estados unidos), .uk (Inglaterra), .ar (argentina)

Uma vez explicado sobre o TLD, e sobre o domínio temos agora o **WWW** que é um sub-domínio. No caso o www é o sub-domínio principal do servidor web que esta sendo utilizado.

Problematização

[www.github.com](http://www.github.com) - é um domínio de um site; Ou seja pesquisando por essa URL será acessado a página inicial do site.

Porém quando o domínio é sucedido por uma **/ [barra]** seguido de um Nome como em

[www.github.com/gustavoguanabara](http://www.github.com/gustavoguanabara) - esse nome é o Caminho

Curiosidade sobre protocolo:

Http – protocolo do navegador

https – Esse S representa o SSL (secure sockets Layer)

Diferença entre HTML e CSS

Antes de aprender sobre essa linguagem de programação primeiro é preciso entender a seguinte frase:

“eu programo em **Html** e **CSS**”

Não é correto afirmar que se programa em HTML e CSS, essas tecnologias não são efetivamente Linguagens de programação.

JavaScript é uma linguagem de programação, PHP, Python, Ruby, Kotlin, Swift, C sharp etc.

HTML e CSS não são linguagens de programação, elas não linguagens, mas não de programação

O termo mais correto para empregar na frase apresentada anteriormente seria;

“Eu desenvolvo em HTML e CSS”

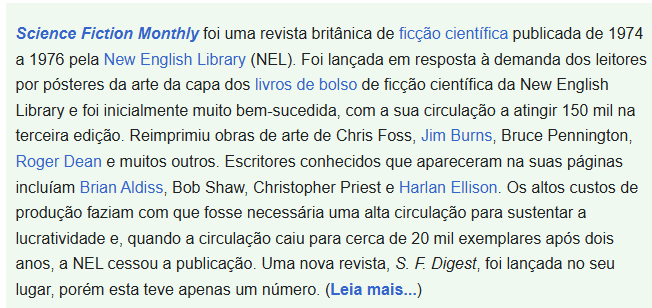
“Eu crio sites em HTML e CSS”.

Por não serem linguagens de programação é errôneo afirmar que se programa em alguma dessas linguagens.

Explicando de uma vez por todas os termos

**HTML: Hyper Text Marckup Languagem**

Explicando de uma maneira simples, **Hyper textos**, são textos clicáveis, que são destacáveis dos demais textos



Como nessa imagem tirada do site da Wikipédia, todos os destaques em azul são Hyper Textos. Mas apesar do nome **TEXT** ela também pode funcionar para imagens, caixas etc, levando para outro link é uma HTML

**Marckup:**

A HTML é fundamentada em Marcas, ela não tem estruturas que linguagens de programação tem, como: **variável, repetição, condição, vetores, objeto, acesso a banco de dados.** A HTML não tem isso, e por conta disso ela não é considerada uma linguagem de programação, e sim uma linguagem de marcação

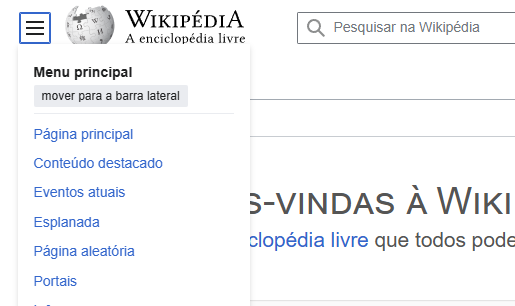
**Sobre as CSSs: Cascading style Sheets (*folhas de estilo em cascata*)**

**“ A linguagem HTML é uma linguagem focada em conteúdo”.**

Então tudo aquilo que esta relacionado a conteúdo vai ter relação com HTML, texto, imagem, vídeos, planilha, tabela, lista.

Por outro lado, tudo aquilo relacionado a Design, a atrativos visuais vai ser provido pela linguagem CSS, como cores, sombras, tamanhos, posicionamento.

Ainda há uma terceira linguagem que complementa a HTML e as CSSs, O JavaScript, que é responsável pela interatividade de um site. Por exemplo



Clicando no “menu hamburguinho – as três linhas paralelas horizontais” vai abrir uma lista de opções, isso na mesma página, ou seja sem gerar links, sem gerar **hyper textos** separados, essa interação, é feita através da linguagem de **programação** JavaScript.

Resumidamente, em JavaScript, é possível criar menus interativos, animações, pop-ups, mensagens avulsas que interagem com o leitor.

Conteúdo em HTML

**< > Esse sinal pode ser chamado de colchete angular**

Para transformar em título basta colocar tags

Sendo assim o título ficaria da seguinte maneira:

**<h1>**Exemplo de título – é o que se chama de conteúdo **</h1>**

**Para fazer um paragrafo em HTML utilizamos a tag <p>**

**<P>**Exemplo de Parágrafo**</P>**

**Algumas tags podem ser mais complexas como o caso da tag de imagem em uma página**

**<img src**=“foto.png**” alt= “**Exemplo de foto**”>**

**Analisando a tag acima. Alguns valores como src e alt, são conhecidos como parâmetros**

A tag **src** significa **source ou origem**

Sendo assim estou apontando que a origem da imagem é o arquivo foto.png

A tag **alt** de **alternative** é o texto alternativo que será exibido caso a imagem não seja carregada. Sendo assim ao invez de só deixar de aparecer a imagem, aparecera escrito “exemplo de foto” como escrito na tag.

**E a foto.png, e Exemplo de foto são chamados de valores**

Esse exemplo de tag traz algumas informações interessantes, por exemplo o fato de quem nem todas as tags tem fechamento, existem outros casos como as tags: **hr, meta, mg** mas são casos mais pontuais, ainda sim a maioria das tags possui fechamento.

Estilo CSS

Um estilo CSS esta geralmente dentro de uma tag Style de HTML mas ao invez de conchetes angulares, ela utiliza seletores, isso para mudar o estilo do título do código anterior que observamos:

h1{

font-family: Arial; [essa linha se chama declaração]

font-size:20pt; [outra declaração]

color: blue; [“…….”]

}

Explicando a tag:

**Font-family** a família da letra que por padrão a font no navegador é **times,** deixara de ser times, e passara a ser **Arial** que foi a escolhida na tag

**Font-size:** tamanho da font utilizada vai ser 20 pontos

**Color:**  a cor da fonte vai ser azul

**Color:** é o que chamamos de propriedade

**Blue:** é o que chamamos de valor

IMPORTANTE! Toda declaração em CSS tem que terminar em **; [ponto e vírgula].**

**Entendido isso vamos ver a estrutura básica de um documento HTML**

A primeira linha tem que indicar que o documento será escrito em HTML 5 sendo assim será utilizado !DOCTYPE

Explicando as tags.

<Head> é onde vai ficar os menus, as configurações, é o cabeçalho

<body> é onde vai ficar o corpo do site, as informações textos, noticias etc.

UTF-8 isso fara com que o site seja compatível com caracteres de UTF-8, exemplo: na língua portuguesa temos alguns caracteres especiais, como **ç** como acentos **^, ~, ´** coisa que não tem no inglês, sendo assim essa tag UTF-8 ira indicar para o site que estará sendo usado palavras com acentuações. Essa tag UTF-8 é OBRIGATÓRIA no caso de desenvolver sites em português.

A tag meta name= “viewport”… está dizendo que o site irá ocupar todo o espaço em branco da tela, além de ser visualizado em uma escala de 100%, sendo assim, não vai abrir com muito zoom, nem sem zoom, vai abrir padrão.

<!DOCTYPE Html>

<Html lang=“pt-br”>

<head>

<meta charset=“UTF-8” >

<**meta name=“viewport”**

**Content=“width=device-width,**

**Initial-scale=1.0”>**

**<title>**Document**</title>**

</head>

<body>

<h1>**Olá, Mundo!**</h1>

</body>

</html>

Essa é a estrutura básica de todo documento HTML!

Front-end, Back-end, Full Stack

Primeiramente explicando que um site estático, quando é feita a requisição de acesso a um site, o servidor que contém as informações daquele site em questão, vai enviar para a máquina de cada cliente os arquivos em HTML, CSS, e JavaScript para que esses possam ser interpretados pelo navegador.

Essas tecnologias levam o nome de Client-side, que estão do lado do cliente, pois o próprio navegador já conta com a interpretação desse tipo de tecnologia na sua execução.

Com isso temos nosso primeiro termo, desenvolvedor Front-End, que será um desenvolvedor que irá criar com as tecnologias HTML, CSS e JavaScript, tecnologias essas que vão rodar do lado do cliente.

Então esse desenvolvedor, além do conhecimento de desenvolvimento de sites, seria ideal ter outros conhecimentos, como photoshop, cores, tipografias.

O desginer para não confundir, por outro lado, vai imaginar, como vai ficar organizada a formatação do site, a fonte que será utilizada, ele vai decidir o layout que será empregado no site em questão, e vai entregar isso para o desenvolvedor, para que ele possa aplicar aquilo que o designer solicitou.

Tendo entendido que o desenvolvedor Front-End atua nas tecnologias para o lado do cliente *cliente-side*, aquele que vai atuar em *server-side* do lado do servidor, é o desenvolvedor Back-End.

Então entendemos que, o desenvolvedor Front-end está especializado em *cliente-side* ou seja, a parte interativa e visual do site, se preocupando em fornecer uma boa experiência para o usuário.

Por outro lado o desenvolvedor Back-end, estará mais interessado em como o código dele há de interagir com o servidor.

Algumas linguagens que funcionam do lado do servidor, pode ser citado o PHP, o JavaScript com a tecnologia do NodeJS, C sharp (C#), Python, Ruby, e linguagem Java

Problematização

Imaginando um cliente acessando um site de compras, ele vai acessar, se esse site for estático, em JavaScript, HTML, e CSS puro, o Cliente não vai ter como saber se um produto esta em estoque ou não, porque o que fundamenta um site estático, é aquele, que não pode ser feita alterações, ou caso seja feita, tenha de mexer diretamente no código do site. Sendo assim fazer uma loja online somente com tecnologias Client-side é inviável, porque se não teria que atualizar manualmente diretamente no código sobre promoções, estoque etc.

Sendo assim do lado *server-side* vai ter um banco de dados conectado ao servidor que vai ter essas informações e nuances que podem ocorrer, no exemplo em questão de uma loja de camisetas, quais produtos estão em estoque, tamanhos disponíveis, cores etc.

EXPLICAÇÃO:

O cliente faz uma requisição para o servidor, a tecnologia PHP vai consultar o banco de dados da loja em questão para ver fatores como estoque disponível, se há promoções, quais tamanhos e cores tem disponível no banco de dados, então essa linguagem PHP vai gerar uma página HTML puro, e enviar para o cliente claro com os atributos como JavaScript, HTML e CSS, será gerado.

Meu primeiro código HTML

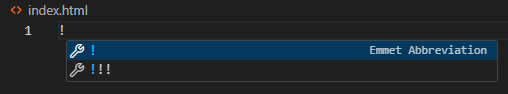
Primeiramente no visual Studio Code, é possível adicionar a quebra de linha pesquisando na barra de configurações por **word wrap**

O primeiro passo para começar um código HTML é criar onde ficaram os arquivos da linguagem.

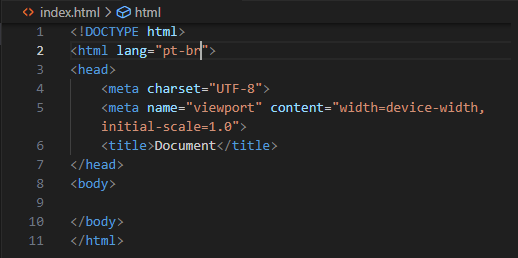
“*Para um bom programador, organização é fundamental!”*

PRIMEIRO CÓDIGO

Uma técnica que há de poupar um tempo no desenvolvimento do site é a utilização da letra ou símbolo inicial de cada código após o colchete angular (**<**) como pode ser visto a seguir, o Visual Studio Code (VS Code) fornece o escopo inicial em um código HTML.

****

O resultado obtido confirmando a escolha do ponto de exclamação será o seguinte.



Apenas com esse simples passo, com o apertar de 2 botões, é possivel ter um código funcional em HTML

Explicando algumas variáveis:

A tag **<head>** é a parte de configurações do site – ou seja alguns atributos, e funcionalidades que vão funcionar internamente no site, que não ficarão visíveis ficarão nesta parte da aplicação, exceto o título; Esse que por sua vez vai representar o nome da guia do navegador, como pode ser visto a seguir:





A tag **<body>** é a parte do conteúdo do site

A tag !DOCTYPE irá informar ao site que está sendo usado o HTML 5 para desenvolver a aplicação

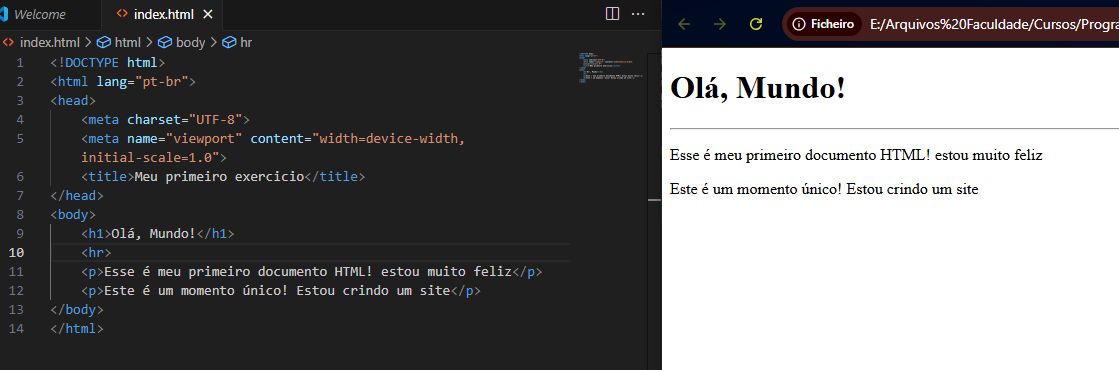
Na tag **meta** está sendo informado ao site que o conteúdo que será exibido em **<body>** estará ocupa dando 100% do espaço disponível da tela

**<title>** título do site

**<h1>** a escala de tamanho para a criação de site é sempre da maior para a menor, sendo assim h1 é a maior (normalmente usada para títulos, cabeçalhos etc) as demais numerações são escolhidas baseadas no grau de importância delas no site, nas respectiva ordem h2, h3, h4, h5, h6 etc.

**<hr>** (*Horizontal Roll*) linha horizontal, pode ser usada para separar melhor os recursos do site. A parte de design vai ser feita majoritariamente pelas CSSs, porém pra ter uma indentação que possa vir a ajudar neste primeiro momento cujo foco é a HTML fazer uso desse tipo de separador vai deixar mais facil a visualização.

Então até o presente momento o código, e o site ficou da seguinte maneira



Próxima Aula:

**NOTA: é importanto quando for escrever um arquivo, escrever sempre em letra minúsculo, pois o tamanho da fonte pode influenciar no código que há de ser escrito.**

**Outro ponto importante de ser comentado é, sempre começar um novo arquivo criando a pasta do zero, isso ajuda na memorização e a se acostumar com o ambiente de trabalho.**

**“a prática leva a perfeição”**

Como o HTML é uma linguagem de marcação de texto, caso você quebre a linha na codificação, esse não vai alterar em nada no resultado da página, caso se faça necessário quebrar a linha em algum momento, é preciso fazer uso de uma tag para que essa possa fazer esse procedimento.

Outra curiosidade, e de se tornar que o HTML faz uso de tags para demarcar a área, ou sinalizar que algo está acontecendo ali, porém, e se precisar usar o símbolo de colchete angular?

Para tal pode se utilizar a seguinte codificação, para **<** basta escrever **&lt;** e para escrever **>**  basta escrever **&gt;**

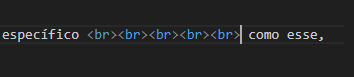
**Uma explicação do porque a tag usa lt, e gt é porque lt significa less than (menor que**

**E gt greater than (maior que) por isso lt, gt.**

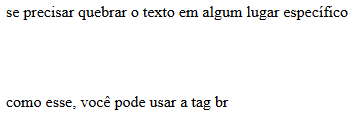
Símbolos e emojis no site

Dando continuidade com o aprendizado

É de se notar que caso precise quebrar a linha mais de uma vez, assim fornencendo um espaço maior, poderia ser feito utilizando a tag <br> de maneira repetida da seguinte maneira:



E o resultado é satisfatório, esse ficaria da seguinte maneira:



Porém, esse é o tipo de solução que um profissional da área adotaria para resolver esse problema.

É mais do que aprender a desenvolver um site, estamos aprendendo a ser bons profissionais.

Caso se faça necessário utilizar mais espaço, fazemos uso da CSS.

Agora como adicionar comentário no meio da tag, tanto para o desenvolvedor deixar uma informação importante para outros desenvolvedores, quando para ele mesmo se localizar a respeito de algumas coisas?

Isso pode ser feito coma seguinte tag <!-- --> o comentário deve ser feito no meio da figura da seguinte maneira

<!-- comentário -->

Para adicionar símbolos no HTML

**&reg; Simbolo de marca registrada**

**&copy; Simbolo de copyright**

**&trade; trademark/ marca registrada em inglês**

**&euro; também é possível usar símbolos para moedas**

**&pound; libra que é a moeda da Inglaterra, porém lá é chamado de pound**

**&yen;**

**&cent;**

**&delta; escrevendo Com d minúsculo o delta vai ser o d**

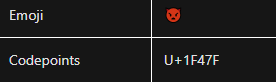
**&Delta; Para escrever o delta símbolo, precisa pôr o D maiúsculo**

**&uparrow; símbolo de seta para cima**

**&uarr; versão simplificada da uparrow, mas que funciona da mesma maneira**

Para adicionar emojis

Uma maneira simples de adicionar emojis seria consultando o site emojipedie.org onde é possível utilizar os emojis apenas copiando, e colando o arquivo dentro do parágrafo do código HTML. Porém essa solução acaba por não ser tão interessante tendo em vista que alguns navegadores não reconhecem esse tipo de simbologia. Sendo assim para se criar um site mais inclusivo, o ideal seria utilizar o código de cada emoji, que também é possivel encontrar no mesmo site.



Dessa maneira você sabe qual o código do emoji que você está tentando utilizar. Porém é de se notar que o código apresentando para representar o emoji esta da seguinte maneira **U+1F47F,** todavia, não é escrito os dois primeiros caracteres, sendo assim o correto para representar esse emoji, assim como os demais, seria o que sucede os dois primeiros caracteres **1F47F**

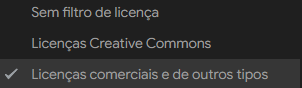
E para adiciona-lo no código é importante o uso de **&#x,** o **&** já é de entendimento que significa que haverá um símbolo e **#x** representa que o símbolo a seguir será em hexadecimal**.** Sendo assim o código vai ficar da seguinte maneira pra representar o seguinte emoji 👿 **&#x1F47F;**

IMPORTANTE SOBRE O EMOJI! Dentro da tag não vai aparecer em destaque indicando que deu certo, vai aparecer como um texto normal, porém na página web vai aparecer o emoji, então por mais que pareça que não deu certo, teste primeiro.

Como colocar imagens no site

Primeiramente, como a intenção é aprender a ser um profissional da área de desenvolvimento de sites, então algo precisa ficar claro, que são os direitos autorais de imagem.

Que não é correto utilizar imagens sem dar os devidos créditos é um fato, porém imagens também são extremamente uteis na hora de apresentar um trabalho, elas transmitem melhor a ideia central do que esta sendo passado. Então tem formas de se obter imagens, sem essas terem direitos autorais que é da seguinte maneira:



Sem filtro, o navegador busca qualquer imagem, seja essa com ou sem direitos autorais.

Licanças Criative Commons são um conjunto de licenças que permitem que os criadores compartilhem seus trabalho de forma mais flexível, definindo os termos de uso. Existem diversos tipos de licenças Criative Commons

Alguns sites que possuem imagens para serem usadas de maneira comercial sem direitos autorais:

Unsplash

Pexels

Free pickel

Howpixel

Pixabay

Libreshot

Wikimedia Commons

Mas mesmo nesses sites alguns deles oferecem sim bastante imagens gratuitas para utilizar de maneira livre, porem vários conteúdos de sites são pagos também, então mesmo diante dessa situação sempre tome cuidado com relação ao arquivo que esta sendo baixado.

Formato de arquivo:

No mundo existem vários formatos de arquivo para imagem, mas para aplicações web é basicamente utilizado o formato JPG. e PNG, algumas poucas versões pode ser utilizado o formato GIF, TIF

No geral depende de vários fatores para decidir qual o formato mais recomendado para colocar em um site, mas explicando a respeito do formado JPG, esse foi desenvolvido em 93, e ele é um formado que compacta imagens. Como na internet não pode ter arquivos pesados, trabalhar com formatos de arquivo JPG é muito interessante.

Agora o formato de arquivo de imagem PNG é mais recente datando de 96 e ele foi criado pela *World wide Web Consortion* ou W3C, mesmo órgão que criou e hoje mantem a linguagem HTML. Incialmente o arquivo PNG foi desenvolvido pela w3c para substituir o formato GIF.

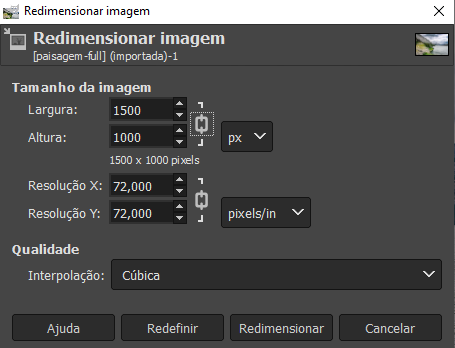
Embora o formato JPG seja um formato de alta compactação, ele não permite transparência. Por outro lado, o formato de arquivo PNG não é tão compacto quanto o JPG, mas permite transparência.

O GIF permite uma qualidade de imagem inferior porém ele permite transparência de animação, em troca da qualidade.

Uma vez entendido isso podemos pegar uma imagem de tamanho mais grande no caso JPEG, e redimensionar ela utilizando a ferramenta GIMP.

Vale lembrar que imagens muitos grandes ao serem adicionadas em sites podem deixar muito lento o processo de carregamento do site, o tamanho máximo recomendado para utilizar é 1500 pixels, dessa maneira ainda é de se manter certa qualidade de imagem, porém se pesar tanto o arquivo do site.

Apesar do foco ser o aprendizado em HTML, para melhorar a experiencia do usuário que esta acessando sua página, é manter essa sempre otimizada sem maiores problemas na hora de acessa-la para tal, utilizando o GIMP na aba IMAGEM em seguida REDIMENSIONAR IMAGEM é possível diminuir o tamanho do arquivo de imagem utilizado sem perder tanta qualidade como pode ser visto no exemplo a seguir:



Outro ponto interessante, é que dependendo da imagem a resolução vai estar mais alta, é possivel encontrar na faixa dos 150, 200, o que também aumenta o tamanho do arquivo, sendo assim deixar o arquivo em 72 a 50 de resolução é suficiente para não pesar a página HTML que está sendo desenvolvida, além de não perder tanta qualidade gráfica.

Feito isso, redimensionado a medida que for interessante no projeto, vale lembrar que não é regra, mas são pequenos passos que melhorar o desempenho geral da página que está sendo desenvolvida.

Inclusive em sites HTML é comum ver imagens de qualidade até mais baixa, 650, 720 etc, afinal não precisa ser um arquivo pesado no site, apenas um arquivo para complementar, ou ilustrar o que está sendo dito no texto, afinal é importante lembrar que HTML é uma linguagem de marcação de texto.

O ideal é sempre manter o arquivo original salvo em backup para caso seja necessário fazer alguma alteração. Todavia com as imagens redimensionadas, já obtermos um resultado satisfatório sem perder tanta qualidade, o que vai ser de grande ajuda na hora de utiliza-las em sites.

É de se comentar que no CSS é possivel fazer alteração de tamanho, porém não se altera o tamanho do arquivo, é importante fazer essa alteração em aplicativos externos para depois somente utilizados no site.

Essas pequenas coisas impactam diretamente na operação do site, um site pesado, é um site lento de carregar, um site lento de se carregar é um site em que o google não recomenda tanto para usuários em potencial, então é importante manter um padrão positivo para o site para que isso não prejudique ele futuramente.

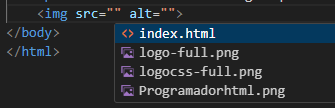
Tag de adição de imagem

Para adicionar uma imagem em HTML, basta dentro da tag body, escrever por **img** que o VS Code, puxa o código que será utilizado para realizar aquela função.

O código que será carregado será: **<img src= “” alt= “ ”>**

Explicando a tag **img** está informando ao navegador que será um arquivo de imagem **src** significa source ou **origem**, no caso aonda está localizado esse arquivo de imagem, qual pasta. E **alt** significa **alternative** que seria uma explicação breve da imagem que está sendo exibida, caso essa demore a carregar, você pode ter noção do que se trata através do texto, podendo oferecer até mesmo acessibilidade para leitores de tela, e deficientes visuais.

Porém é de entendimento que colocar o nome do arquivo, a pasta pode dar margem para erro, sendo assim existem atalhos no VS CODE que hão de facilitar a adição de imagens dentro do código, bastando utilizar o comando **Ctrl + backspace** para que o VS Code puxe os arquivos da pasta de origem, o resultado vai ficar da seguinte maneira:



Importa lembrar que esse comando deve ser realizado dentro das aspas de **Src.**

O resultado na página vai ser uma imagem muito desnecessariamente grande, sendo assim é preciso fazer um recorte na imagem para não pesar o arquivo da página.

O peso do arquivo é irrisório tendo em vista que se tratam de arquivos em KB, porém isso economiza tráfego de banda.

Este foi um exemplo quando os arquivos **index** e **Imagem** estão na mesma pasta, mas e quando não?

Neste caso é possivel visualizar a pasta que foi criada dentro do diretório também utilizando a tecla de atalho vista anteriormente, porém o código em si vai apresentar mudanças, ao invez de apenas aparecer o nome e formato da imagem, primeiramente vai aparecer o nome da pasta nomedoarquivo.png (dependendo do formato), exemplo:

Um arquivo de imagem que esteja na mesma pasta do arquivo index, vai ficar da seguinte maneira:

**Logocsss-200.png (nome do arquivo**

Por outro lado, se tiver uma sub-pasta dentro da pasta que está o index, neste caso ficaria assim:

**Imagem/logocss-200.png (imagem seria o nome da pasta e / indica que o arquivo está dentro dela)**

Outra maneira de adicionar imagens dentro de arquivos, seria através de links, porém essa prática não é recomendada, pois uma vez utilizando link, você está utilizando o servidor daquele que fornece o link e a imagem para uso próprio, além do fato que, uma vez que caia o servidor, você perde acesso a imagem consequentemente. Então o recomendado é sempre utilizar imagens do seu próprio banco de dados, ou da sua própria máquina.

Como mudar o favicon de um site

Explicando o que é Favicon.

Basicamente favicon são os ícones que ficam na aba de pesquisa do navegador, exemplo:



Esses estão representando respectivamente

Youtube Music;

WhatsApp;

Unifael;

Porém como adicionar favicon nas nossas aplicações?

Uma dica, para quando for criar um favicon, é utilizar imagens sem muito detalhe para não perder a qualidade, e ficar visível que se trata do seu site, existem outras coisas que são importantes de se comentar a respeitos dos favicons, que é o formato do arquivo, o ideal é que esse seja em .ICO não JPG nem PNG

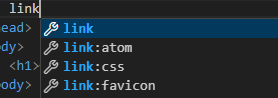
Sites que podem ajudar no desenvolvimento:

**Iconarchive** para encontrar um favicon para utilizar no site, nesse data base tem várias opções disponíveis.

Ou no caso de querer criar um favicon próprio, tem o site **favicon.cc**

Tem também a opção de criar um favicon do zero, ou converter um arquivo de imagem em .ICO, e para tal pode ser feito no site **favicon.io**

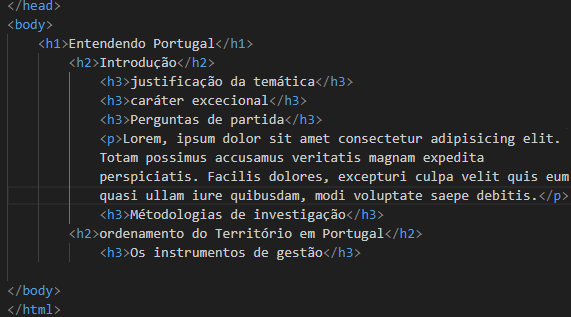
Tendo criado o favicon, para adiciona-lo no site, basta utilizar o código **link** dentro do VS CODE



Escolhendo a opção favicon, ele vai abrir o código, e entre aspas, onde estiver escrito “favicon.ico” você substitui pelo nome do arquivo de imagem que vai ser o ícone.

Hierarquia de Títulos

é de uma importância de na hora de usar a tag <h1> (heading), <h2>, <h3> até no máximo <h6>. entender a hierarquia que cada uma representa, h1 é usada para títulos de nível 1, ou para o assunto que você quer que seja de maior destaque na página que está sendo apresentada, inclusive uma página web pode sim ter mais de um h1, é muito comentado o fato de só poder haver um h1 na página web, porém isso não é verdade, uma página web pode sim ter mais de h1, tudo depende do nível de importância que determinado texto vai ter na página.

Uma vez indentado o resultado vai ficar parecido com isso: 

O paragrafo utilizado “Lorem Ipsum” é um texto fictício, sem sentido, que é utilizado em design gráfico e desenvolvimento web como um espaço reservado para o conteúdo real. Ou seja, seria como um test pra saber se o código em questão está funcionando corretamente.

Inclusive escrevendo **lorem** o resultado vai ser o texto que foi mostrado acima.

Vale lembrar que fazer uso de h1, h2, h3 etc somente com a intenção de diminuir o tamanho da fonte, é incorreto, essas são tags para ordenar a hierarquia dos títulos encontrados em um site; tamanho, cor, fonte, tudo isso é feito através do CSS.

Semântica na HTML5

Algumas tags foram que existiam no HTML4 passaram a não ser mais recomendadas para uso no HTML5, porém não deixaram de funcionar, mas o próprio órgão responsável por manter e atualizar o HTML recomenda não utiliza-las mais, isso porque essas tags por serem desatualizadas podem ser removidas a qualquer momento, então o ideal é se acostumar com o cenário sem elas, para não sentir sua falta.

O próprio VS Code indica em vermelho as tags que estão obsoletas como pode ser visto a seguir:

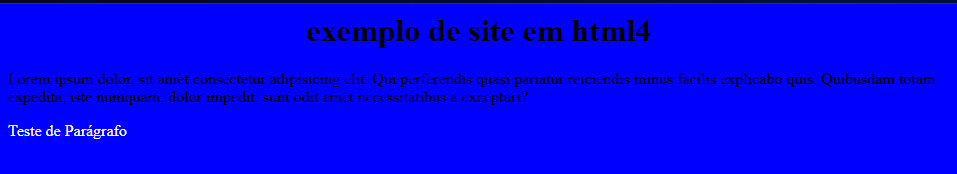


Exemplos de tags herdadas do HTML4 que funcionam, mas que a W3C recomenda deixar de utilizar

<body bgcolor=”blue”>

Ao que deixa o background do site azul

Para centralizar um título, entre uma tag h1, adicionar outra tag <center> </center>



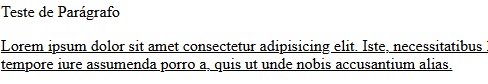
É de se notar que o resultado ainda funciona, tal como funcionava em sua versão anterior, porém isso não significa que é recomendado utilizar esta versão.

Isso vale para outras tags como: font:face, font:size nada mais funciona.

A HTML está focada em semântica em outras palavras, a HTML está focada no **significado** que cada tag pode trazer para a construção do site, uma tag de centralizar texto, ou uma tag de tamanho ou cor não traz significado, e sim **forma** para o texto. E para **tudo** que é **forma** se reservam as CSSs.

Então tudo relacionado a forma, seja o tamanho, a fonte, a cor (tanto da fonte quanto do fundo) isso será feito em CSS. O HTML5 é totalmente focado nas **tags semânticas**.

Outro exemplo dessa forma que antes era utilizada em HTML, porém que agora não é mais, é a tag <u></u>



É de se notar que o resultado ainda funciona, porém novamente, isso traz forma para o HTML, sendo assim é uma tag que não é recomendado o uso.

Um exemplo seria ao invez de utilizar a tag <u> seria utilizar uma tag como <address> que ainda mantém o destaque (no caso de se tratar de um endereço) porém, sem tornar o resultado em uma forma, mantendo a semântica.

O resultado de um tag de endereço escrita em HTML5 seria:

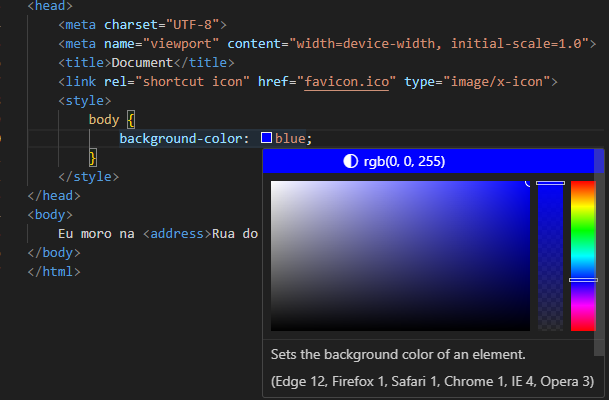


É de se notar que ainda tem certo destaque, porém agora esse destaque não está mais relacionado a forma, e sim a semântica que a tag traz. É possível que esse endereço caso o site seja acessado por dispositivos móveis, vire um link que leve para o Google Maps indicando o endereço marcado.

TAG DE ANIMAÇÃO (NÃO SEMÂNTICA)

Ainda existe uma tag, que é a tag **marquee** que basicamente cria uma animação para a palavra, que ela se move para uma direção. Porém novamente isso não pertence a HTML, falou animação então esse código deveria ser em JavaScript

Exemplo de código para substituir o código **bgcolor** porém em CSS



O resultado ainda é satisfatório, porém agora é feito através exclusivamente de CSS

De qualquer maneira, fica o link do site oficial da W3C para acompanhar possíveis alterações em códigos:

<https://www.w3.org/TR/2014/REC-html5-20141028/obsolete.html>

Negrito e itálico

É possível utilizar uma tag <b> para tornar um texto em negrito, porém isso é uma maneira não semântica de escrever, para escrever um texto, ou deixar uma palavra em **negrito** a tag utilizada seria <Strong> que além de trazer o mesmo negrito, agora traz a semântica, agora traz o significado do porque a palavra está em negrito, pois esta está em destaque.

NOTA: Quando você escreve um texto, mas precisa adicionar uma semântica, isso vai ser útil para outras coisas, mas como exemplo caso se faça necessário envelopar um texto, seja pra torna-lo negrito, ou itálica etc, é possível escrever o texto primeiro, depois selecionar uma parte do texto que se deseja envelopar, e com o atalho **Ctrl, shift + p** vai abrir uma barra de pesquisa, que procurando por abb (de abbreviation)a opção **wrap abbreviation** há de aparecer, você pode escolher o formato de envelopamento para o texto selecionado, bastando escrever qual a tag que vai envelopar o texto selecionado. Exemplo: escrevendo **Strong**, o texto que foi selecionado, vai ser envelopado por uma tag Strong, consequentemente deixando a tag selecionada em negrito.

Para escrever um texto em itálico tem a maneira obsoleta e não recomendada com a tag <i> (itálico) e a tag recomendada e atualizada <em> *emphasis*

Formatações adicionais

Como adicionar uma marcação de texto

Para escrever um texto marcado assim como esse trecho, basta escrever a tag <mark> entre o texto que se deseja que fique marcado

Da seguinte maneira:



É de se notar que a cor da marcação é por padrão amarela, porém e se o fundo, o background de uma página for de uma cor de ofusque essa marcação?

Quando se fala em formatação de cor, entra na questão da semântica; cor é uma tag de forma não de significado, porém dentro das tags HTML é possivel adicionar uma função “style” que é uma função CSS dentro do HTML, e isso é feito quando são alterações pequenas ou pontuais, normalmente criar uma style sheet css somente para isso, acaba não compensando, então é utilizado um recurso CSS para formatação em HTML, sem fazer o HTML perder a semântica; o resultado vai ficar semelhante ao que se pode ver a seguir:

VS CODE:



Página:



NOTA: Dentro do faixa de cores é possivel escolher outras cores utilizando o cursor do mouse, sem a necessidade de saber o nome ou o código hex de cada cor.

Vale lembrar que a tag <mark> não aparece no atalho do vs code, sendo assim tem que escrevê-la manualmente.

Ainda com relação a MARK temos a seguinte problematização; no caso de uma página possuir muitas tags MARK a solução para isso é criar uma tag **style** no cabeçalho, ou seja na tag **<head>** da página, você cria um seletor para definir a cor padrão de todas as tags mark que vão ter no site que está sendo desenvolvido.

Letras grande e letras pequenas

Em HTML tem um código que ele é ultrapassado muito por conta da semântica, porém sua versão oposta não tem se tornou obsoleta. Essa tag é a **big** e a **small.** A tag <big> ela ainda funciona como a maior parte das tags obsoletas, porém ela não envolve mais significado, e sim formatação, sendo assim o correto é o fazer em CSS. Porém a tag small ainda apresenta certo significado, e o motivo para tal é pura especulação, mas que muito por conta de existir letras miúdas em contratos, o significado que a tag small pode trazer, é esse destaque inverso mas ainda importante, diferente da tag big que não há nenhum tipo de significado.

Texto deletado e texto inserido

Nada maias que do que o traço no meio do texto tal qual ~~esse,~~ e texto inserido seria um texto escrito da seguinte maneira, respectivamente escritos com a tag <del> e tag <ins>

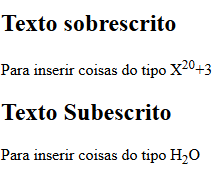
Tem outra maneira de adicionar o sublinhado a um texto que seria com a tag <u> porém isso é defasado, não é recomendado fazer desta maneira, pois entra na regra de semântica

Texto sobrescrito texto subscrito

Um exemplo que vai ficar claro do como se utiliza seria esse exemplo se refere ao texto sobrescrito, e a tag utilizada para tal seria a tag <sup>

Agora o texto sub escrito seria da seguinte maneira e a tag utilizada para o texto sub escrito seria a tag <sub>

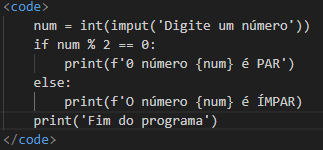
O resultado de ambas vai ficar da seguinte maneira:



Citações e códigos

Normalmente, quando há de se escrever um texto em HTML alguns textos tem larguras diferentes para cada letra, neste caso o ideal é utilizar algo chamado *font mono espaçada* que seria uma font que deixa o tamanho de cada letra distribuído de maneira igualitária. E para fazer isso o código utilizado é o <code> vale lembrar que é possível utilizar a opção envelopamento para selecionar a parte do texto que já foi escrita, para adicionar direto a tag <code>.

Como o nome sugere essa tag é ótima para envelopar códigos que normalmente são extensos, todavia essa tag tem um problema que, quando você vai escrever um código, é de se esperar que tenha toda a indentação do código, que é a organização do código da maneira que ele foi escrito, na imagem a seguir é possível ver um exemplo:

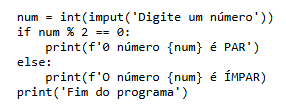


É de se esperar que esse código seja exibido neste formato, porém não é o que ocorre; O resultado obtivo se assemelha mais a imagem a seguir



Isso ocorre porque o HTML é uma linguagem de marcação de hiper texto, sendo assim pra deixar o formato do código que foi escrito anteriormente, da maneira que ele foi escrito, levando em consideração seu espaçamento, suas quebras de linha, para tal é utilizado o código <pre> que vai manter o código do jeito que ele também está no software de desenvolvimento.

Com essa pequena adição o código será exibido da seguinte maneira:



Que é mais correto se tratando de um código que foi escrito acima, então precisa ser exibido da maneira que foi solicitado.

Nota: quando se faz necessário como no exemplo acima, onde para o código ficar no canto da página, ou seja retirar uma tabulação (que nada mais é que o espaço entre o extremo do código, e código em si, ou seja o espaço vazio) é utilizado a tecla **shift + tab** e para aumentar uma tabulação, basta utilizar a tecla **tab** e pode ser bem útil corriqueiramente.

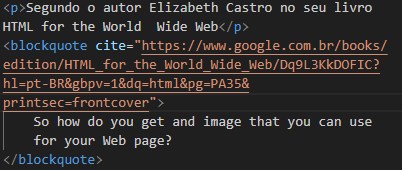
Citações

Quando você vai escrever um texto, e vai utilizar uma citação, por exemplo “penso logo existe – René Décarte” você pode utilizar um tag <q> de *quote* (citação) e o que essa tag faz, é adicionar aspas no trecho que foi selecionado como citação. E mais do que somente adicionar aspas na citação, essa tag traz o significado para o trecho, deixando claro que se trata de uma citação.

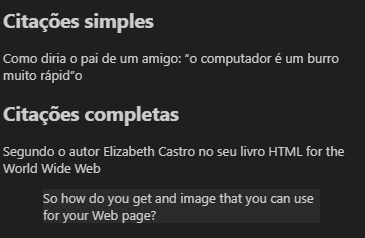
Essa citação vista acima, é uma citação simples, ou seja não demanda que detalhes ou sua origem além daquela já informada.

A tag **<q>** não é incompleta em comparação com a tag **<blockquote>** apenas se trata de uma escolha, se você quer ter o texto ou citação vinculado com autor ou obra, faz se uso da tag blockquote, é bastante interessante por conta de dar os devidos créditos, até para o buscador associar seu conteúdo com a obra, porém utilizar a tag **<q>** vai demonstrar que um texto é uma citação, ainda que não seja dado os devidos crédito, o fato de indicar ser uma citação já não faz disso um uso indevido do direito de autenticidade do conteúdo original.

Exemplo:



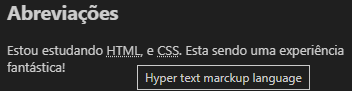
E o resultado dessa citação, quando for executado na página será uma quebra de linha, um destaque para o texto definido como citação.



Abreviações

Exemplo de abreviação, HTML que significa *Hyper text marckup language* ou seja quando é utilizado uma abreviação na página é possivel adicionar uma tag <abbr> e adicionar qual o significado da sigla que foi abreviada.

O resultado visual nessa tag vai ficar da seguinte maneira:



Quando é passado o Cursor por cima da tag definida como abreviação, é possível ver seu significado

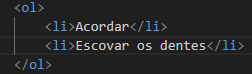
Tag para inverter o lado da escrita

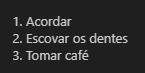
A tag em questão é mais um alivio cômico do que uma tag realmente útil, que é basicamente uma tag para inverter o lado do texto, é possível escrever da esquerda pra direita, ou o contrário tbm sendo possível.

<bdo dir= “ltr”>

Lista OL e UL

A respeito da lista ordenada, ou seja aquela que possui numeração, vamos imaginar a seguinte situação, você vai fazer uma lista de a fazeres em uma ordem que precisa ser seguida, porém quando você está bem avançado na lista você lembra de uma coisa do começo que precisava ser adicionada. Neste caso, você precisaria mudar cada um dos números dos a fazeres, para deixar de acordo com a numeração e ordem, sendo assim, existe uma tag par resolver, ela cria a numeração da lista baseado na ordem que está no código, isso facilita tanto pra não precisar colocar códigos, quanto para no caso de precisar adicionar outros elementos anteriores na lista. Essa tag seria a tag *order list* ou **<ol>.** Mas não basta somente isso; foi definido que haverá uma lista ordenada, porém precisa identificar os *itens da lista,* ou *listen itens* que seria na tag **<li>**, sendo assim o resultado vai ficar da seguinte maneira:

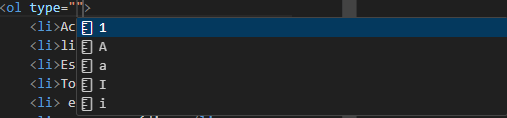




Além dessa função lista, que já da pra imaginar várias funções para utiliza-la ainda tem parâmetros que podem ser adicionados. Por exemplo a lista acima, é uma lista numérica, porém existem outros parâmetros para adicionar nela, e para o fazer, basta utilizar a tag **type=“”** para definir se vai ser uma tag ordenada numérica, ou utilizando letras, normalmente questões de múltipla alternativa, ao invez reposta: 1, 2, 3, 4, 5 etc é utilizado A, B, C, D, E etc. Porém não somente isso, vamos supor que você queira que sua lista comece em 4 neste caso o parâmetro seria outro, para esse caso se utiliza o parâmetro **start=“”** (IMPORTANTE: no parâmetro star não vai mudar o formato, então supondo que o **type** escolhido foi o Alfabeto, então escrever **B** no parâmetro **start** não vai surtir efeito, o que deve ser feito é, a letra B (nesse exemplo) é a segunda letra do alfabeto, então o parâmetro **Start** para começar em **B** vai ter que ser 2).

Detalhe, normalmente quando tem muitas tags semelhantes em uma aplicação web, é possivel alterar o estilo através do cabeçalho de todas simultaneamente, dispensando a necessidade de escrever uma tag de estilo pra cada uma, porém com relação a tag **<ol>** precisa escrever individualmente dentro de cada tag para mudar.

Enfim, esses seriam detalhes que podem ajudar na formatação de arquivos, agora falando sobre o parâmetro **type** que pode ser adicionado dentro da tag **<ol>** ficaria da seguinte maneira **<ol type=“1”>** mas existem outras opções como:



Explicando cada uma.

* 1 Sequência ordenada numérica
* A ordem alfabética
* a ordem alfabética com fonte pequena
* I Ordem numérica em romano
* I Ordem numérica em romano com fonte pequena

Listas não-ordenadas

Como o título sugere são listas onde a sequência não importa, por exemplo em uma lista de compras, não importa a ordem que vai ser feita, apenas o item que haverá nelas, isso também vale para, lista de pedidos, lista de presença, falta etc.

E a tag seria **<ul>** *unorder list* lista não ordenada.

E nesta tag é de se notar que também temos alguns parâmetros, como o símbolo que aparecerá antes de cada item; a tag utilizada segue sendo **type** assim como na lista ordenada, porém agora com símbolos, o símbolo padrão é o disc, porém existem ainda outras opções como:

**type=“circle” ○**

**type= “square” □**

**type= “disc”**  •

Listas mistas e de definição

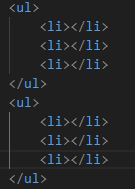
Explicando que o termo que eu MATEUS conheço como indentação, também pode ser conhecido como aninhamento de {ninho} de por uma tag dentro da outra.

Atalhos de formatação:

Quando você tem muitas tags semelhantes e precisa adicionar um parâmetro em cada uma delas, se não for forma não tem como adicionar em CSS então é possível adicionar em mais de uma tag ao mesmo tempo, para fazer isso basta escolher uma tag, colocar o cursos no final dela, segurar a tecla alt e apertar na mesma posição porém nas demais tags

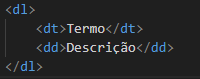
Como exemplo se tiver mais de uma tag **<ol>** ou **<ul>**

Exemplo prático:



Por exemplo nesta tag acima, para alterar as duas ao mesmo tempo para adicionar um parâmetro, dentro o VS CODE (pode ser que não funcione, dependendo do software que está sendo utilizado) basta apertar com o cursor do mouse na frente do **l** de um das tags **<ul>** e segurando a tecla **alt** apertar na frente do **l** da outra tag **<ul>** isso não vale somente para as tags **<ul> <ol>** isso vale para qualquer tag que precise adicionar uma função ou parâmetro, que seja semelhante.

Entendido todos esses pontoa acerca das listas anteriores, ainda existe um terceiro tipo de lista, que seria a lista de definição, que diferente das listas anteriores ela funciona a base de três tags, diferentemente das duas das listas anteriores, essa lista funciona semelhante a um dicionário, onde tem um termo, e embaixo do termo tem seu significado, é possivel ver como funciona na imagem a seguir:



Temos a tag **<dl>** *definition* *list* (lista de definição) e dentro dessa tag temos as tags que darão sentido para a tag como um todo, que seria **<dt>** *definition Term* (termo da definição) e **<dd>** *Definition Description* (Descrição da definição).

Esse tipo de lista ferramentas de busca valorizam bastante, pois quando alguém pesquisa por definição de um termo, como HTML, a ferramenta de busca via buscar nos sites que tenham a definição além de outros conteúdos relacionados ao conteúdo pesquisado.

Não significa colocar em todos os sites, com o objetivo de obter engajamento, mas nos que tiver sentido colocar, seja pelo conteúdo do site em si, pode sim favorecer.

Links e âncoras

Toda vez que a gente acessa uma página construída em HTML temos os pontos denominados **links** que nada mais são do que pontos clicáveis do site, normalmente destacados em azul ou roxo, dependendo se aquele link já foi acessado, esses links são chamados de **âncora** ou **hiper link**.

Como o tema abordado tem ralação com links e buscadores, uma curiosidade para enriquecer o conhecimento acerca desse assunto. O google é o maior buscador do mundo, o segundo maior é o YouTube, plataforma também do Google

Com esse tema surgem mais alguma dúvidas, quando um site for publicado, como faz para cadastrar o site para ele aparecer nas pesquisas do google. E a resposta é simples, não cadastra, contanto que tenham pessoas procurando pelo conteúdo do seu site, elas vão achar, o google vai recomendar, o google vai achar e partilhar com as pessoas, isso se o tema abordado for de interesse delas.

Tem uma sigla que pode ajudar de alguma forma, SEO (*search engine optimization*) ferramenta de otimização de busca.

Dadas as devidas apresentações sobre o tema, agora sobre os links

O primeiro link será o link externo, que nada mais é do que links que vão apontar para outros sites, ou seja levar o usuário do seu site, para site de terceiros.

Bom você pode fazer isso em qualquer palavra ou texto selecionado, bastando apenas envelopar o texto que vai virar um link, você faz isso envelopando com a tag **<a href=“”>** esse **a** é de *anchor* ou âncora, de imediato junto da tag a temos também uma tag **href** que é o parâmetro de referencia de hiper texto, que seria a URL que vai estar sendo utilizada.

Porém normalmente, quando você acessa um link externo, ele sobrepõe seu site, então dessa maneira você perde potenciais cliente, Então a solução para isso seria criar um parâmetro dentro da tag **<a>** que seria a tag **target=“\_blank”** isso fará com que ao invez de sobrepor seu site para abrir o site de terceiros através do link, agora quando alguém clicar no link, vai ser aberta outra guia, além disso é possível colocar outro parâmetro que seria o **rel=“external”** que significa *relative* (ou relativo a), significando que o link é relativo a um site **externo** ao atual.

Os links externos, normalmente são os primeiros a serem estudados pois eles levam para sites que já estão prontos, então o único cuidado que precisa ter é em colocar a URL correta

Links internos

Explicando de maneira resumida, links internos são links que transferem o cliente para outra página, porém pagina essa do mesmo site, exemplo: Quando você acessa o Youtube você está na página inicial do Google, quando você aperta para abrir um vídeo, você permanece no Youtube, porém em um layout de página diferente, isso porque o Youtube tem várias páginas dentro dele, cada vídeo tem sua página, por isso cada vídeo tem uma URL diferente.

Quando é feito um link interno, não precisa indicar a URL completa, o VS code puxa pelo nome do arquivo, se esse claro estiver na pasta do diretório principal.

E o mesmo vale para fazer uma tag de link nesse segundo site, basta apenas colocar essa segunda página com o link da primeira da mesma maneira, o que vai dar mais perpectiva para uma página de qualidade, seria a formatação, ou texto utilizado para fazer a âncora.

Na tag de link externo, temos o parâmetro rel=“url site externo” de *relativo a* no link interno tem uma tag semelhante, porém ao mesmo tempo diferente essa tag seria a **rel=“next”** vale lembrar, isso é adicionado somente na primeira página, na pagina principal, pois ela esta indicando ao navegador que o link se trata da *próxima página* por isso next. Agora na segunda página também existem uma tag **rel** porém essa tem como valor **prev** de *previous* de previamente, isso não vai ser utilizado somente quando for a segunda página.

Ou seja página principal para as demais, a tag a ser utilizada será a tag **next** qualquer página que não seja a principal utiliza-se a tag **prev**

Esse detalhe vai ser utilizado para o buscador, que identificar qual o sistema de navegação utilizado na página, ao invés de deixar o buscador decidir qual vai ser a próxima página, você aponta para ele atreves desses códigos vistos.

Existem também parâmetros para identificar links que são de matérias de sites não oficiais, sites ao qual você não tem vínculo, você não quer que o buscador perpetue o link da página como se essa tivesse alguma relação com a sua, essa tag **rel** seria **rel=“nofollow”.** Essa tag pode ser usada justamente em caso de patrocínio, ou link de referência de matéria.

Entendendo o parâmetro target=“\_self”

Pode parecer redundante utilizar o parâmetro **target=“\_self”** em uma tag âncora, já que o comportamento padrão é, de fato, abrir a página de destino na mesma aba.

Clareza e manutenção:

Utilizando esta tag, você deixa explícita a intenção de que o link deve ser aberto na mesma aba. Isso facilita a leitura e a compreensão do código, especialmente para outros desenvolvedores.

A importância de aprender Linux para o desenvolvimento de páginas web.

É importante aprender sobre o Linux, ainda que a máquina utilizada seja Windows, pois o servidor que vai armazenar a página web vai ser Linux, por isso a importância de aprender sobre Linux, próximo curso que há de ser feito, terei de adiar o curso de JavaScript 😢.

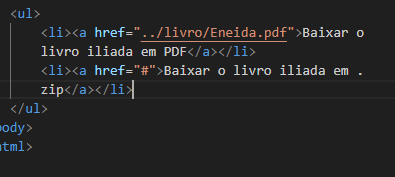
Continuando o tema de desenvolvimento web já entendemos que quando tem uma pasta dentro do diretório principal, o próprio VS CODE consegue puxar o arquivo da parta, porém e para voltar uma pasta, quando você está no Index da pasta, mas precisa voltar?

Neste caso basta se utilizar do seguinte comando **../** antes do nome do arquivo da página, no exemplo utilizado, **index.html**

Links para Download

Já conseguimos aprender bastante coisa a respeito de várias tags em HTML, e se tratando de tags e quando tiver algum arquivo para download no site, como fazer este link?

Até chegarmos o complexo processo de gerar um link de download, mais uma dica de formatação que pode ser útil, quando estiver gerando alguma tag de ancora <a href= “”> sempre lembre de utilizar uma # dentro da tag, isso vai indicar que é um link vazio.



A imagem acima representa tanto a # para representar o link que será adicionado futuramente, quando o arquivo livro já adicionado na tag anterior, PORÉM esse link ainda não está realizar download, apenas está abrindo o arquivo em uma nova guia do navegador.

Para fazer um download além de adicionar o nome do arquivo, é fundamental adicionar um parâmetro para identificar que seria para fazer o download sendo esse parâmetro o **download** adicionado dentro da tag *anchor* sendo assim o resultado ficaria:

**<a href=“#” download=“#” type=“media-type do formato do arquivo aqui” >**

Ainda é possível adicionar uma tag **rel** para indicar que vai levar para um link externo, ou uma tag **target** para abrir em uma guia separada. Mas fato é que a tag **download** e **type** indicam respectivamente, que é para fazer download do arquivo, e indicar o formato do arquivo que será baixado.

Em **type** para poder fazer o download de um arquivo em HTML5 é preciso informar o **media type** do arquivo, por exemplo em um arquivo pdf o media type seria **application/pdf** de um arquivo zip seria **application/zip** mas isso pode ter variações, então para saber qual o formato de download do arquivo que você deseja que o usuário baixa seria consultando a IANA que é um repositório completo, e o órgão responsável por cuidar de Media Types’s ou MIME type, depende do formato do arquivo, vale lembrar que esses dois são um tipo de informação que identifica o formato de um arquivo, permitindo que os programas (como navegadores web) saibam como interpretar e exibir esses arquivos corretamente.

Vale lembrar que a funcionalidade desse download depende do navegador, pois tentando baixar o arquivo pdf em minha máquina infelizmente não foi possível, porém o arquivo **.zip** deu certo, então ou depende de mudar o formato do arquivo, ou de repente de alguma outra maneira procurar um formato para o navegador conseguir fazer o download do arquivo em pdf.

Imagens dinâmicas áudio e vídeo.

Nos tempos atuais é importante que determinados matérias se adaptem a multiplataformas, como por exemplo uma imagem que será exibida em um computador, ou em um dispositivo móveis.

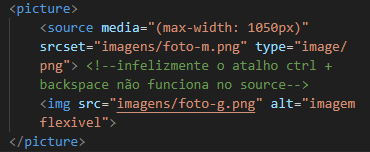
Como visto anteriormente no que dizia respeito a imagens, quando temos imagens muito grande em alta resolução essas costumam deixar o site pesado e lento, algo que o buscador não favorece, então é importante criar imagens adaptáveis ao formato de tela que essa vai estar sendo exibida, desta forma não vai prejudicar a qualidade do site, nem sua performance.

Para criar imagens dinâmicas que se adaptem ao tamanho da tela, é utilizada a tag **<Picture>** essa tag seria como se você estivesse informando que vai ter uma imagem a seguir, porém ela permite utilizar várias fontes de imagem, vai ser possível ver melhor isso funcionando em breve.

A importância de utilizar imagens que se adaptem; o google tem um termo para isso que se chama **bouce rate** que nada mais é do que *taxa de rejeição*, e isso tem muito haver com o site, se você utiliza imagens grandes. Isso faz com que os usuários tenham que usar barras laterais para visualizar todo o conteúdo de uma página, pra alguns usuários isso pode até passar desapercebido, porém é em dispositivos móveis onde se encontra o maior problema, como em dispositivos móveis não tem barra lateral então deixar o site neste formato faz com que o cliente não queira voltar no site, por conta de ser uma experiência ruim para ele próprio.

Explicando um pouco mais sobre o **bounce rate** quando alguém acessa um site, e esse site não prende por muito a atenção de um cliente, e nisso é feita uma média de todos os acessos que aquele site teve, a ferramenta de busca passa a não recomendar mais tanto o site tendo em vista que o site tem uma bounce rate alto.

Por conta disso é sempre importante se preocupar com a experiência do usuário, para saber como melhorar cada vez mais seu site para reter a maior quantidade de usuários possível, assim aumentando a popularidade do site.



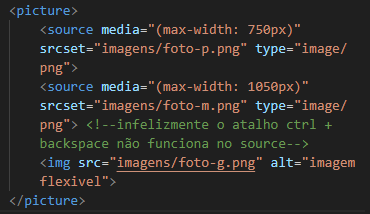
Como é de se notar infelizmente o atalho não funciona na tag source, mas o restante está funcionando perfeitamente.

Agora explicando sobre a tag acima;

**<source media>** seria a tag utilizada para adicionar essa função de adaptar a imagem dependendo do tamanho da tela, **max-width** seria o tamanho máximo que a tela tem que atingir pra substituir a imagem média para a imagem grande. No editor de imagem foi feito a imagem com o tamanho 1000px esses 50 a mais seriam por conta da barra bateral que ocupa um tamanho também, então para deixar a experiência ainda melhor sempre 50 pixels a mais em relação ao tamanho da imagem.

Importante! Nesta codificação a ordem importa então a imagem maior tem que ficar por último no código, pois foi definido que com o maior tamanho possível tem que consequentemente ser exibida a maior imagem possível, se colocar a menor imagem “sendo a mais baixa” no código o processo vai acontecer de maneira reversa, quanto mais pixels, menor o tamanho da imagem, então para ficar tudo correto, sempre a maior imagem fica por último, a menor imagem fica em primeiro, e assim por diante

O resultado vai ficar da seguinte maneira, pois foi criado em um formato de 3 tamanhos de imagens diferentes:



A ordem esta sendo levada em consideração o tamanho, como é de se notar: menor, média e maior. Outro ponto que eu notei mexendo com esse código, tem um parâmetro **type**, esse parâmetro indica qual o formato do arquivo, então é importante colocar que tipo de arquivo é bem como é mostrado, e qual seu formado, se não, a imagem não será exibida no site! Se atente a isso.

Áudio

Primeira coisa que é importante comentar sobre áudios, seria onde baixar de maneira adequada, pois assim como imagens, é de suma importância ter conteúdo original em seu site para não ter nenhum tipo de problema, sendo assim onde baixar áudios?

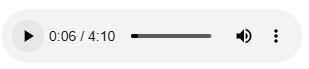
No próprio Youtube na ferramenta **Youtube Studio** temos uma opção de **biblioteca de áudio** onde possui algumas músicas que é possível usar de maneira livre, e outras que é possível usar, mas deixando os devidos créditos “créditos esses que aparecem como deve ser escrito no próprio arquivo de download da música”.

Outro cuidado que é importante ter com relação a áudio, e sobre tamanho, arquivos maiores são mais pesados, o que assim como imagens, pode sobrecarregar o site, sendo assim o recomendado é se utilizar de faixas de audio menores, ou criar um loop para elas em determinados pontos, tudo com a intenção de minimizar o peso da pasta, o que vai ajudar bastante no acesso com mais qualidade.

Agora como fazer para adicionar um áudio dentro do site?

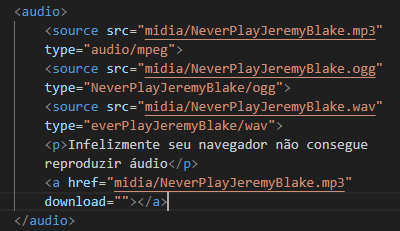
A resposta é simples com o seguinte código: 

Esse source aceita o comando **Ctrl + backspace** então para achar o arquivo de áudio na pasta é muito simples. O parâmetro controle que gera o controlador de volume, play áudio, assim como podemos ver na imagem a seguir:



Porém, alguns navegadores não tem compatibilidade com o formato de áudio MP3, então caso realmente queira utilizar essa opção de áudio no seu site, para manter a inclusividade, seria utilizar outros formatos de arquivo, como WAV e OGG.

Vale comentar que o exemplo acima seria a versão mais simples de se adicionar áudio em um site, temos outras versões mais complexas.



Nesse exemplo, assim como arquivos de imagens, o navegador vai reproduzir em ordem assim como no código, do primeiro até o último, até o áudio funcionar no caso. Nesse exemplo acima o áudio não foi convertido para outros formatos, então não funcionária de qualquer maneira, porém assim como em imagem e seu tamanho, é possível salvar o áudio em diferentes formatos ai o navegador vai tentando qual formato funciona.

Nesse caso o ideal ainda, e caso o navegador não suporte nenhum dos formatos, ter uma mensagem informando o usuários que o navegador dele não suporta o formato de arquivo de áudio, sendo assim colocar, se desejar, um link de download do arquivo para o usuário

Dentro da tag audio temos alguns parâmetros um desses seria o **preload** como **auto, metadata, none**.

**Auto**; ele só vai considerar que o site terminou de carregar quando o arquivo de audio terminar de carregar também, essa opção é controversa pois arquivos de audio extensos podem demorar bastante para carregar, o que pode gerar lentidão.

**Metadata**; essa opção vai carregar algumas poucas informações do áudio antes de acessar o site como data de publicação, tamanho do arquivo, tempo de reprodução, e só vai acessar o conteúdo completo do arquivo depois que a página já carregou.

**None**; não vai carregar o arquivo

**Alguns parâmetros da tag audio**



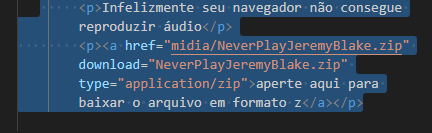
Como é de se notar acima, tem outros parâmetros importantes de se destacar;

Autoplay: opção para tocar o áudio dentro do site

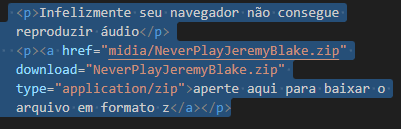
Sempre ficar atento com relação as tags, pois se uma tag não estiver funcionando é possível que qualquer item adicionado a mais dentro dela passe a não funcionar também, então sempre separe as tags devidamente para não ter problema na hora de testar

DICA! A tecla tab joga os itens da lista para frente, para fazer o oposto para melhorar a indentação da uma página utilize **shift + tab** para “voltar uma casa. Exemplo prática:

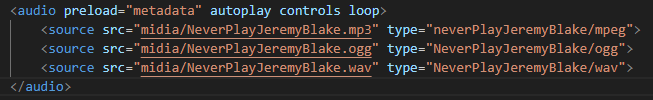
Apertando tab:



Apertando shift tab:



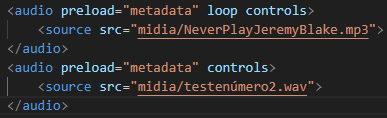
Referente ainda ao tema áudio



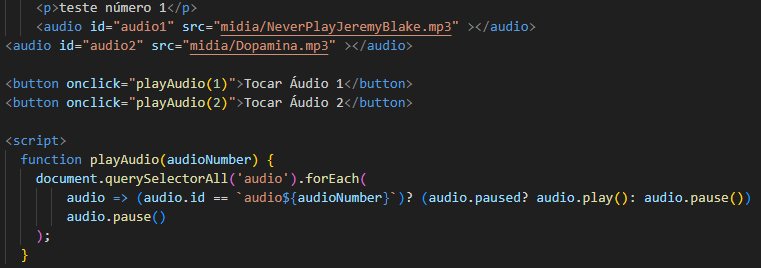
O resultado final vai ficar semelhante a imagem acima.

Dentro da tag **audio** vão ficar os parâmetros que vão influenciar dentro de todos os arquivos de áudio, no exemplo acima é o mesmo arquivo de áudio porém em formatos diferentes, isso porque nesse formato acima a tag vai se basear em testar os formatos até encontrar algum compatível para exibir na página, lembrando que o navegador vai testar na ordem que aparece de cima para baixo, até achar aquele compatível.

Outro ponto que, dentro da tag audio, é possível colocar um **mesmo audio** porém com formatos diferentes para justamente testar individualmente qual funciona no navegador, porém para colocar mais de um áudio, teria de criar outra tag audio individual, como no exemplo a seguir:



NÃO é possível colocar dois arquivos diferentes de áudio dentro da mesma tag, é preciso criar uma tag de audio separada para cada arquivo de audio porém isso traz certos problemas, como no caso de ficar um audio sobrepondo outro, e nesse caso, colocar cada áudio com variados tipos de formatos de arquivo para testar a compatibilidade com o navegador, porém tem certas nuances, como pausar uma música quando começar a tocar outra, que só pode ser adicionado com facilidade em Javascript.



Acima temos um código escrito em JavaScript, onde é possível adicionar dois arquivos de audio que interagem entre si, porém, como dito anteriormente é feito em JavaScript devinindo variáveis para cada audio apontando seus ids, para que uma interaja com a outra dependendo da ação que é solicitada. Esse conhecimento fica para quando for aprender JavaScript, por hora o básico é como adicionar audio utilizando HTML5

Videos

Tendo aprendido um pouco a respeito de imagem e audio e como adicionados em sites HTML, vamos ver a respeito de vídeos e como coloca-los em sites de maneira eficiente. Sem comprometer a velocidade do site que não vai comprometer o orçamento para ter de gastar mais com trafego de dados.

Os arquivos de vídeo, importante frisar sobre direitos autorais assim como nos arquivos de audio e imagem, podem ser baixados também através do site do pexels.

Agora falando sobre compatibilidade de arquivos de vídeo nos padrões usados no HTML5 para navegadores, seriam os formatos mp4 (que incluem os arquivos **.mp4 e m4v**) tem também o formato **webm**  que seria um formato de arquivo extremamente leve, mas claro com uma qualidade de imagem bem inferior por conta disso tem também o formato **ogg** e no caso dos vídeos seria **ogv**. E assim como nos arquivos de áudio o ideal seria ter vários formatos do mesmo arquivo para utilizar em sites que precisem de vídeo para comportar o maior número de formatos consequentemente clientes possível.

O ideal é sempre evitar de colocar vídeos em 1080p ou qualidade superior a isso, pois o custo de hospedagem de conteúdos assim é extremamente caro, sendo assim não é indicado, principalmente se for hospedagem local, neste caso o custo vai ser a lentidão que vai deixar para os usuários que tentarem acessar a página.

Através do programa **handbrake** é possível converter um vídeo de **mp4** para outros formatos como **mkv** e **webM,** porém além desses ainda tem outro formato de arquivo que pode ser utilizado que é a variando do formato **ogg** de imagem que seria o **ogv** e para tal é possível utilizar um conversos direto no navegador para fazer essa alteração.

O site utilizado para fazer a conversam de formato: <https://video.online-convert.com/pt/converter-para-ogv>

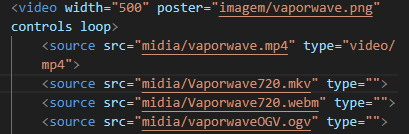
Agora como adicionar vídeos

A maneira mais simples, seria utilizando a tag **<video>** Uma maneira simples de se lembrar de como adicionar video seria associando a tag video com a tag **<áudio>** e assim como na tag áudio é possivel adicionar alguns parâmetros, um desses pode ser o **width** que vai determinar o tamanho que esse vídeo vai ocupar na tela. Porém por mais que diminua o tamanho da exibição, o tamanho do arquivo segue sendo o mesmo, por isso é importante analisar para saber se o tamanho do arquivo precisa ser grande ou isso pode ser dispensável. Porém diferente do arquivo de áudio que alguns formatos não estavam funcionando, o arquivo de vídeo funcionou no navegador **Chrome** nos 3 formatos disponíveis **mp4, mkv, webM.**

O ideal ao criar páginas assim como em áudios seria manter todos os formatos de arquivo, pois diferentes navegadores têm diferentes acessos a formatos. Com relação a como organizar os arquivos, assim como em áudios, ele segue a ordem que o código é escrito, então o navegador tenta acessar o arquivo de video até conseguir um em formato compatível, porém se não der certo, o ideal é gerar ume mensagem de erro informando que não há compatibilidade com o navegador, assim como os arquivos de áudio também.

Essa mensagem pode ser feita em uma tag **<p>**

O resultado vai ficar da seguinte maneira:



É de se notar que tem alguns parâmetros presentes na tag video todos direcionados com a formatação do arquivo, como tamanho, controle, e deixar o arquivo em loop sempre que terminar.

É de se notar que na imagem acima, não tem media type informado, mas a título de curiosidade o media type quase sempre vai ser o definição do arquivo, nesse exemplo **video/** formato do arquivo podendo variar dependendo de qual está sendo usado, nesse exemplo do vídeo seria **mp4, mkv, webm** ou **ogg** (esse último ogg acontece porque o padrão ogg é open source então o formato pra audio seria ogg e pra video ogv, mas a nomenclatura seria ogg porque inclui os dois formatos tanto para video quanto para audio)

Temos outros parâmetros interessantes como o **poster** que seria a thumbnail que será exibida antes de dar play no vídeo, algo que deixa mais esteticamente completo um arquivo de vídeo.

Porém é de se comentar que quando tem arquivos pesados como o caso dos vídeos, se tem muitos acessos de usuários por dia isso vai custar muita hospedagem para o cliente, sendo assim o ideal é procurar soluções para ter um video de boa qualidade sem perder nada de dados, que seria utilizando serviços de hospedagem de vídeo como o **youtube**. No site que eu estou estudando por exemplo *cursos em vídeo*, o vídeo não está no site, ele tem um link que abre o vídeo no youtube, inclusive aparecendo no histórico de vídeos do próprio youtube mesmo estando em outro site. É essa é a solução mais eficiente e barata para resolver o problema acerta da utilização de vídeos.

E para fazer isso, basta acessar um vídeo do Youtube que se deseja adicionar no site, na aba **compartilhar** tem uma opção de incorporar, que já gera um código **Html** para utilizar no site, sendo assim basta copiar o código, dessa maneira será exibido o vídeo do youtube em seu site, e isso é bom para todos, pois, dessa maneira não há consumo de dados do site, mas sim do servidor do youtube, e para o youtube também é bom, pois é uma maneira de conseguir ainda mais engajamento, como se os provedores do site ou servidor estivessem promovendo o video do servidor do youtube. Porém com relação ao youtube, nesse cenário, o video teria de ser upado no youtube para poder se utilizado no site o que torna o vídeo público, sendo assim em caso de querer deixar um vídeo privado (como por exemplo fazer vídeos de cursos, ou conteúdos exclusivos para assinantes etc), infelizmente não seria possível no youtube, mas tem outros canais de hospedagem de video que poderia ser úteis neste caso, como o **vimeo.**

Falando um pouco sobre essa plataforma, que é muito utilizada por criadores que vendem conteúdos exclusivos à parte, o serviço em sí tem uma qualidade ótima, em contrapartida, seu algoritmo não é tão eficiente quanto o do Youtube, porém como cada um tem propósitos diferentes, basta ver qual se enquadra melhor de acordo com a necessidade do cliente.

Com relação a como se utilizar da ferramenta do **vimeo**

Um atributo que é possível adicionar em arquivos de vídeo é **<frameborder>** que gera uma borda em volta do vídeo, por padrão vídeos com o atributo **incorporar** do youtube vem com frameborder igual a 0, porém fica da preferencia de cada um se é necessário manter ou tirar.

CSS’s

A proposta é tentar centralizar alguns vídeos (4) em uma página Html adicionando links em cada um que leve até o vídeo em uma página separada e outro botão isso dentro dessa outra página que leve diretamente no site que equivale a onde o vídeo está hospedado além de um botão de voltar para a página principal

Css

Para adicionar alguns atributos é mais simples realizados pelo CSS, primeira questão a ser resolvida, a posição do título da página, para centraliza-lo basta adicionar no arquivo css qual a tag utilizada seguida por

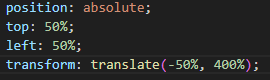
**#id** {

**text-align: center**

{

desta maneira o arquivo dentro do HTML será modificado, vale lembrar que, para que haja alteração no arquivo HTML é necessário que o arquivo css esteja conectado via **link** ID.

Falando agora sobre a posição de atributos dentro do HTML sendo imagens, vídeos, áudios, botões, isso é feito se utilizando de CSS porém, provavelmente um site não vai ter somente um arquivo editável, então o ideal é utilizar dentro do código do arquivo que se ser editar em HTML um **id=“nome\_do\_arquivo\_sem\_espaço”** neste caso você definiu que aquele atributo tem um id, e para localizar o id dentro do css basta utilizar **#nome\_do\_arquivo\_sem\_espaço** depois se tratando de uma css basta colocar dentro de colchetes os atributos que queira modificar, como o tema é posição o atributo vai ficar da seguinte maneira:



**Position: absolute** Determina que será modificado 100% da posição do arquivo, o que significa mudar a posição tanto no eixo cartesiano **X** e **Y** então talvez nem sempre ter poder absoluta vale a penas se é necessário apenas alterar a posição do eixo **X** por exemplo

As posições **top** e **left** em 50% indicam que inicialmente o arquivo vai começar centralizado, e as mudanças da posição serão feitas levando em consideração esse ponto como sendo o ponto 0, então a tag **transform** vai alterar a posição, vale lembrar que o valor está em porcentagem, então quanto menor for o valor inicial, menor será a diferença que as porcentagens vão impactar, nesse sentido pode ser em porcentagem a mudança, em pixels, em polegadas etc.

GIT e Github

No link a seguir é possível saber como pegar um projeto que já esta sendo trabalhando, como um curso e upar ele no github como repositório:

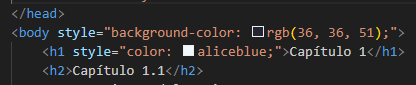
<https://www.youtube.com/watch?v=P0Hvrf8T3zo&t=5s>

de maneira resumida, você se utilizada do **github desktop** segue os passos que são passados dentro do software, que isso já é realizado.

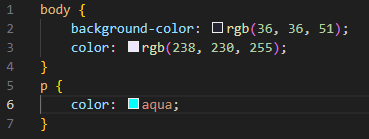
Css

Com relação as Css é mais separar um conhecimento previamente estudado anteriormente, onde são separados os estilos de edição no HTML agora fazem parte de uma página separada que será responsável por mexer em toda a parte estética de site.

As CSS’s podem ser aplicadas de três formas, a primeira forma seria com o estilo **Inline** que é a menos recomendada mas seria colocar a tag de estilo **<style>** na própria linha do código que se quer alterar, ficaria da seguinte maneira:



Não está incorreto a tag, por mais que elas estejam na mesma página, como essa tag de formato está dentro de uma tag estilo então fica implícito que se trata de uma CSS. Porém isso vai deixar muito bagunçado o código, por exemplo no caso de colocar parâmetros no código vai ficar muito lotado de conteúdos, fora que no caso de precisar alterar fica inviável, pra esse código ainda funciona pois é um código menor, mas em casos de códigos maiores de duas mil [2000] linhas fica inviável, além de que essa mudanças são locais ou seja se você quiser criar uma alteração por exemplo em todas as tags **<p>** é possível em CSS definindo a tag **<p>** como uma variável a parte, exemplo:

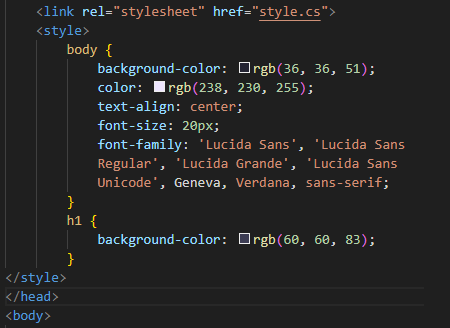


É possível notar que na variável **body** foi adicionada uma tag **color** que define por padrão toda a escrita dentro dela, então tudo dentro dessa tag vai ficar daquela cor, porém criando uma variável separada para a tag **<p>** essa vai se sobressair. Vale comentar que nesse exemplo tudo que está dentro da tag **body** porém que não faz parte da tag **<p>** segue da mesma cor definida na tag **<body>**

Estilos Locais / internos

Tem outra maneira de organizar os estilos CSS’s que seria colocando toda a função e declarações de estilo no cabeçalho da página. Como o cabeçalho é uma área dedicada a alterações no código da HTML para então entrar com o conteúdo escrito em body, é possível escrever o conteúdo nesta área, que não vai atrapalhar, ou deixar poluído as tags, já seria uma maneira de otimizar espaço

Veja a imagem a seguir para saber como fica o resultado:



Ainda não é a melhor solução, mas para projetos iniciais, já é muito mais bem formatado que o método anterior, é interessante utilizar esse método até como uma *blueprint* para saber como está ficando o site ou para ter uma noção de como está se desenvolvendo o projeto, sem ter que gerar links com páginas externas.

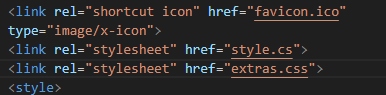
**Curiosidade** o nome da seguinte linha  é chamado de declaração; que é o conjunto de propriedade e valor.

Estilos CSS externos

Um fato interessante acerca das CSS são algumas **regras** (são códigos mas popularmente chamados assim, não significa que você tem que obrigatoriamente utilizar) que são colocadas no começo do código antes de começar a desenvolver os estilos, uma popularmente conhecida seria uma que também existe em HTML que seria **Meta Charset=“UTF-8”** para indicar que o estilo também faz utilização da regra de UFT para símbolos e caracteres especiais.

Inclusive com relação ainda aos links de conexão CSS é possível gerar mais de um, de tal maneira que você pode colocar um estilo geral para todas as páginas vinculadas ao arquivo da página principal, mas também adicionar outros links vinculados somente a algumas páginas, de tal forma que é possível adicionar estilos extras em algumas páginas especificas.

Veja o exemplo a seguir;



Dessa maneira, é possível fazer um estilo pra todas as páginas de tal maneira que elas conversem entre si, mas se precisar que uma página seja diferente das demais, ou algumas páginas basta adicionar outro link, as funções do primeiro link vão funcionar igualmente, porém o link extra pode trazer alguns estilos extras pra essas páginas especiais.

Também é possível misturar os estilos CSS, por exemplo adicionar estilos Internos e externos na mesma página, caso queira adicionar um estilo somente naquela página sem que isso seja distribuído para as demais, utilizar o estilo somente naquela página também se torna uma opção eficiente.

Agora vamos definir a seguinte problematização, e se colocar todos os estilos dedicados a uma mesma tag, qual vai prevalecer?

A ordem de preferencia do HTML vai ser sempre o estilo **inline** seguido por **interno** depois por **externo**

Então seria como se o estilo **inline** fosse o mais forte, aquele que vai sobressair os demais, e o **externo** o mais fraco, aquele que é sobreposto pelos demais.

Explicando sobre as tags **id** e **class**

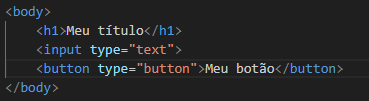
Ambas as tags tem um proposito semelhante, que é vincular uma tag ou um conjunto de tags em um arquivo CSS, a tag **id** ela é mais especifica, cada id é único e geralmente é associado quando quer ser feita uma alteração em uma única tag e não nas demais. E quando se quer fazer uma alteração em mais de uma tag é utilizado a tag class, que vai vincular o estilo empregado para todas as tags identificadas com o nome estipulado dentro da tag class, em suma, é uma tag assim como id, porém pode ser utilizada a mesma identificação em diferentes tags.

Como criar um botão de troca de tema para um site

O ícone de lâmpada em JS geralmente, indica uma sugestão ou possível correção para o seu código em um editor de código. Essa funcionalidade é cada vez mais comum em editores modernos e tem como objetivo ajudar o programador a escrever códigos mais limpo, eficiente e livre de erros.

Para tal, é preciso entender alguns conceitos básicos acerca do funcionamento da criação de site. A começar pela tag **input**, essa que é um dos elementos mais versáteis e importantes, utilizadas para criar campos interativos em formulários web. Esses campos permitem que os usuários insiram diversos tipos de dados, como texto, números, senhas, e-mail, datas, arquivos e muito mais.

O resultado dessa adição somada ao botão **button** vai ficar da seguinte maneira em código:

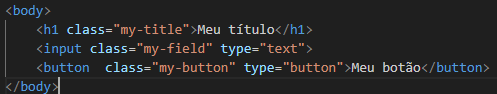


Em página vai ficar da seguinte maneira:

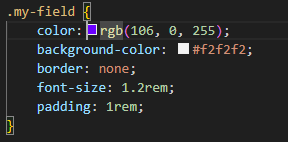


Porém este botão não tem nada atribuído a ele, sendo assim para poder prosseguir com o desenvolvimento do tema, é preciso adicionar classes em cada um para poder fazer edições externas nesses atributos.

O resultado vai ficar da seguinte maneira:

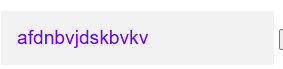


Basicamente o mesmo anterior somente com adição do atributo classe o nome que pode ser da preferência do desenvolvedor, porém ainda recomendado se utilizar de associações para ficar mais fácil memorizar e saber sobre o que cada um se trata.



Explicando os atributos do código acima, primeiro o nome **my-field** foi o nome atribuído a **class** do arquivo **input** sendo assim a barra de escrita.

O atributo color se trata da cor que a font dentro da barra vai ter, por exemplo:



O atributo **background** dentro da tag **input** representa a cor de fundo que essa barra vai ter, no caso o ideal é colocar tons claros, ou que destaque a barra do restante da página, caso se queira manter a cor padrão, utilize uma borda para deixar a barra destacada, ou se não utilize cores levemente mais claras em relação ao fundo.

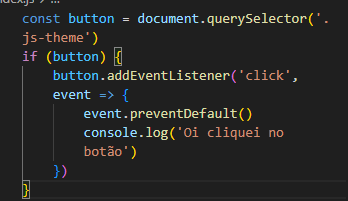
**Border** é o atributo da borda da barra, adicionando uma variável **none** você informa que não deseja ter borda, porém é possível adicionar outras variáveis, como **solid, dotted, dashed, double, groove, ridge, inset, outset.** Vale lembrar que esses atributos estão relacionados com a formatação da borda, por exemplo, **solid** (cor sólida), **dotted** (borda feita de pontos), etc.

DICA: tem um atalho muito funcional, no VS CODE que é para duplicar linhas de código, e isso pode ser feito utilizando a tecla **shift + alt + ↑**

Perfeito tendo feitas essas alterações, agora basta ir até a página JS para identificar qual o código equivalente ao botão para poder aplicar a ele uma interação de mudança de thema

DICA JS: É interessante quando for criar uma classe, colocar um prefixo como vai poder ser visto na imagem a seguir, **js-theme** pois quando for utilizar essa classe para mexer em JS ainda que mude o nome dos atributos anteriores, a classe não vai ser alterada tendo em vista que o prefixo segue o mesmo.

Em JS o código vai ficar da seguinte maneira até o momento:



Explicando as variáveis:

1ª linha:

**Const** é o mesmo que **let** ou que **var**, ele declara uma variáveil constante, neste exemplo, **btn** é uma cicla pra **button** sendo assim é uma maneira de facilitar a indentação já que os resultados em sí são os mesmos.

**Document.querySelector(‘js-theme’)** seleciona o primeiro elemento do DOM (Document Object Model) que possui a classe CSS **js-theme**. O DOM é uma representação em árvore da estrutura do seu HTML, e **querySelector** permite buscar elementos específicos nessa árvore.

Em outras palavras **document.querySelector()** é usado para selecionar elementos específicos do HTML, permitindo que você os manipule com JavaScript.

2ª linha

**If (button):** Verifica se o elemento com a classe **js-theme** foi encontrado no DOM. Se o elemento existir, o código dentro do bloco **{}** será executado. Essa verificação é importante para evitar erros caso o elemento não esteja presenta na página.

3ª linha

**Button.addEventListener(...):** Adiciona um “ouvindo de eventos” ao botão (**button**). Esse ouvinte fica “esperando” que um evento específico aconteça (no exemplo acima, um **clique**) que é o que veremos a seguir.

**‘Click’:** Especifica o tipo de evento que o ouvinte deve esperar um clique do mouse.

**(event) => {...}:** Define uma função anônima (arrow function) que será executada quando o evento **click** acontecer. O parâmetro **event** (opcional)representa o objetivo do evento, que contém informações sobre o evento ocorrido.

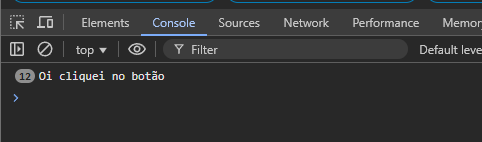
4ª linha

**Event.preventDefault():** Impede o comportamento padrão do evento. Por exemplo, se o botão estiver dentro de um formulário (**<form>**), essa linha impede que o formulário seja enviado quando o botão for clicado.

5ª

**Console.log(...)** exibe a mensagem “Oi cliquei no botão” no console do navegador. Essa função é útil para depurar o código e verificar se ele está funcionando como esperado.

O resultado disso vai fazer com que dentro do console (**Ctrl+ shift + i**) apareça a seguinte mensagem:



Sendo o 12 o número de cliques que ouve no botão anteriormente definido, isso serve mais para saber se está funcionando de acordo.

Tendo feito isso o console.log pode ser removido do código uma vez que ele funciona mais para a verificação.

Após esse teste verificando que a funcionalidade do botão está de acordo, será adicionado o método **setAttribute()** que em JS, é usado para adicionar ou modificar um atributo de um elemento HTML. Quando aplicado ao **document.body**, ele permite que você manipule diretamente os atributos do elemento **<body>** da sua página.

Veja a imagem a seguir contendo o código:



O que podemos tirar disso, é que dentro do documento <body> vai ser adicionado o seguinte atributo “data-theme” cujo valor vai ser “t-dark” quando apertar o botão. E para saber se está funcionando basta consultar novamente o console direto do navegador:



Apertando no botão:



Porém com isso surge outro problema, atualmente quando é apertado o botão de trocar de tema, ele gera o valor **t-dark** (*thema escuro*), porém o que deveria acontecer depois disso seria ele voltar para o thema padrão ou *light* assim que apertasse de novo o botão.

Então para resolver isso, é preciso fazer um **mapeamento de chave-valor**, o código que será utilizado em JS para resolver isso será o seguinte: