# Caderno de estudos de Front-end

**Anotações dia 1 – 22/02/24:**

**HTML:**

Toda página executa **HTML**, **CSS** e **Javascript**

HTML é escrito em **tags**:

Algumas tags **tem** **corpo**:

<tag> <!—Essa é a tag de abertura-->

\*corpo da tag

</tag><!—Essa é a tag de fechamento-->

Algumas **não tem** corpo:

<tag>

Tags normalmente tem **atributos** (principalmente as sem corpo). Atributos são colocados dentro da abertura da tag:

<tag atributo=”valor1 valor 2”>

\*corpo da tag

</tag>

Como é possível ver, o atributo se chama ‘atributo’ e seu **valor** é **composto**, ou seja, tem dois valores, isso é possível em alguns atributos como **classe**

HTML é **case** **insensitive**, ou seja, não liga para letras maiúsculas e minúsculas

A primeira coisa necessária em um documento HTML é a **especificação** **para o navegador** que um arquivo HTML é **de fato** um arquivo **HTML**:

<!DOCTYPE html>

Essa é a informação que vai na **primeira linha** do arquivo HTML e a palavra **DOCTYPE** **pode** ser escrita em **minúsculo** também (**mas** é convenção ser em **maiúsculo**).

A **segunda** **coisa** que vai ser feita num arquivo HTML é a **tag html**, também chamada de **tag raiz** ou **tag root**, essa tag é a **mãe de todas** as outras, portanto, **todas** as outras tags **devem** ser colocadas **dentro** dela.

<!DOCTYPE html>

<html>

</html>

A **principal** tag é a **tag HTML**, mas **dentro dela** temos outras **duas** tags essenciais: a **tag head** e a **tag body.**

<!DOCTYPE html>

<html>

<head>

</head>

<body>

</body>

</html>

A tag **head** é a tag responsável pelos **metadados** da página (metadados são **dados** **sobre** outros **dados**).

Já a tag **body** terá todo **conteúdo** **exibido** na página.

Algumas tags **importantes** da tag **head** são:

**<title></title>**: Tag **obrigatória** para definir o **nome** que será **exibido** no **topo** do **navegador**

**<meta charset=”utf-8”>**: Essa **tag** em **conjunto** **com** seu **atributo** **define** o **conjunto** de **caracteres** que está sendo **utilizado** no documento HTML, sendo o **mais** **usado** o **utf-8**.

Por incrível que pareça a tag **body** **não é** exatamente **obrigatória** para um arquivo HTML funcionar, **mas** é **EXTREMAMENTE** **recomendado.**

**Anotações dia 2 – 01/03/24:**

O Safari da Apple introduziu uma atributo da tag meta sobre tela para celulares, com essa tag a largura do site se alinha ao dispositivo, fazendo com que fique muito melhor de ler no celular:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

A forma ideal de escrever id’s e afins em HTML é sem sinais gráficos, todo em minúsculo e separando as palavras por kebab-case (traço).

Quando você utiliza o CSS, alguns estilizadores possuem mais prioridade que outros, então, caso você os sobreponha, o com maior prioridade prevalecerá. Caso você utilize dois iguais, o que foi dito por último prevalecerá.

Para identificar **id’s**, usa-se o ‘**#**’, para **classes**, o ‘**.**’

**Id’s** tem **especificidade maior**, ou seja, modificar algo em CSS neles prevalece mesmo que você mude na classe. Também existe o ‘**!important**’ para forçar a estilização da classe mas isso NÃO É RECOMENDADO

Tags <h1> ao <h6> são chamadas tags headings (cabeçalhos) e são títulos e subtítulos de texto ordenados por **RELÊVANCIA** e **não TAMANHO.**

Ao acessar o inspecionar do navegador, é possível ver a configuração ‘**display**’ de alguma tag HTML e é importante entender quando uma tag tem o **display block** ou **inline**, já que tags block ocupam toda a largura da página na porção onde estão e as inline ocupam apenas a parte que elas aparecem.

Tags HTML podem receber o atributo style que aplica CSS apenas a ele (entre aspas). Esse atributo também pode ser usado como tag dentro da tag <head> e nela o CSS é escrito para o documento todo da mesma forma que num arquivo CSS normal.

**Tag <b>** vs **Tag <strong>** e **Tag <em>** vs **Tag <i>**

Tanto as tags <b> quando <strong> deixam o texto em negrito, mas a tag <b> foca unicamente na parte visual e a tag <strong> foca na semântica, fazendo com que o texto seja interpretado como mais importante, inclusive o destaque do texto na tag <strong> pode ser editado para não ser apenas ‘deixar a palavra em negrito’.

Já as tags **<em>** e **<i>** têm a mesma relação, mas <i> deixa itálico e <em> deixa itálico e dá destaque semântico. Num geral é recomendado que sejam usadas as tags <strong> e <em>, mas da forma correta.

É possível usar tags de imagem dentro de tags de parágrafo, por exemplo, assim como tags de imagem dentro de tags de link (<a>) dentro de tags de parágrafo.

As tags **<s>** e **<u >** servem para riscar e sublinhar as palavras respectivamente, já as tags **<del>** e **<ins>** servem para fazer isso com o extra da semântica, assim como nas tags mostradas antes.

A tag **<small>** deixa o texto menor, a tag **<sup>** deixa o texto menor e alto (tipo uma potenciação) e a tag **<sub>** deixa baixo (tipo na química).

A tag **<br>** é uma tag de quebra de linha

A tag **<q>** representa uma citação e é usada junto com o atributo ‘cite=’ que leva a um local externo que mostra a origem da citação, essa tag **pode vir dentro do <p>.**

A tag **<blockquote>** faz o mesmo, porém ao invés de aspas, replica o formato de citação longa (espaçamento lateral maior), essa tag **não pode ir dentro do <p>.**

A tag **<hr>** faz uma linha lateral, ela não tem corpo **e não pode ir dentro da tag <p>**.

A tag **<code>** serve para colocar texto que fica com uma **fonte que indica código** (de programação), mas ela **não pode ir dentro da tag <p>** e também **não fica formatada** corretamente se for, para formatar corretamente, ela deve ir dentro da tag **<pre>**, que é para textos **pré-formatados**, com isso a o código colado dentro o HTML **fica formatado** igual na página.

As tags **<span>** e **<div>** são contâiners genéricos (divisões estruturais e não semânticas), a diferença é que a tag **<span>** é **inline** e a tag **<div>** é **block**.

# Recomeço dos estudos, agora com curso

**Anotações dia 3 – 17/05/24:**

**A HISTÓRIA DA INTERNET**

1982 – Nesse ano a internet começou a ser usada para fins de distribuição de textos acadêmicos, já nessa época o conceito de hipertexto surgiu. Uma forma bem simples de entender hipertexto é a Wikipedia.Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Final dos 90’: As empresas começaram a usar a internet para outras coisas, como blogs e sites de notícias. Como da pra imaginar, quando se tenta usar algo para uma finalidade que não era a original, surgem gambiarras. Nessa época os códigos eram mais ou menos dessa forma:

Texto

Descrição gerada automaticamente com confiança média Se quiser mudar a cor, tinha que mudar manualmente de item pra item.

Além disso, nessa época existia uma disputa entre os navegadores de internet netscape e internet explorer. Cada um entendia o HTML de uma forma e era necessário fazer duas páginas HTML, uma para cada navegador.

Após isso surgiu o W3C, que veio para padronizar todo o HTML e evitar problemas como esse.

Com essa padronização, cada tecnologia adquiriu sua função:

HTML: Responsável pelo conteúdo da página.

CSS: Pela estilização da página.

JavaScript: Usada para garantir boa usabilidade/interação do site.

**O QUE É UM ARQUIVO HTML**

Um arquivo HTML é um arquivo de texto que contém marcações de HTML, essas marcações dão significado ao conteúdo se lidas de forma correta. Para que o computador entenda o que está sendo dito, em HTML são usadas tags <nome da tag>conteudo da tag</nome da tag>.

Um arquivo HTML é construido em um editor de código e executado em um navegador. Uma página nada mais é que um arquivo HTML e quaisquer outros arquivos vinculados à ele (CSS, JS, imagens, etc.), um site é uma pasta com vários desses arquivos.

Comunicação cliente-servidor é uma comunicação entre cliente (usuário) e servidor (máquina conectada à internet 24/7), onde o cliente faz uma requisição (pedido ou request) ao servidor e este responde. Para o servidor receber a mensagem e depois enviar a resposta (response), há todo um caminho complexo onde é encontrada a localização do servidor/cliente e ambos formam uma conexão. Para que esta conexão funcione, precisamos de três coisas:

URL – *Uniform Resource Locator* é um localizador de recursos e cada arquivo HTML precisa ter seu próprio endereço (URL).

HTTP – *HyperText Transfer Protocol* é um protocolo (padrão) de comunicação entre servidores/clientes para que diferentes máquinas possam se comunicar.

HTML – Linguagem de marcação de hipertexto usada para inserir elementos na tela através do documento HTML.

**COMEÇANDO A ESCREVER EM HTML**

Se escrevermos conteúdo no arquivo html ele será tratado apenas como texto comum e não terá quebra de linha. Para isso servem as tags que comentei antes. Algumas tags (chamadas de tags container) tem abertura <> e fechamento </>, já outras possuem apenas abertura.

Atalho para ativar o salvamento automático do VS Code

CTRL + , Escrever Files: Auto Save Selecinar afterdelay

**PRIMEIRAS TAGS**

**<h1> até <h6>:** Essas tags servem para títulos de textos, tópicos, etc. São divididas pela ordem de relevância na página. OBS: Embora o browser mude o tamanho delas, elas não devem ser usadas com base no tamanho, mas sim na importância.

**Destaques em textos:**

**<i>:** Itálico sem semântica (na verdade tem um pouco, mas serve para destacar uma parte específica do texto).

**<em>:** Itálico com semântica. <em> vêm de emphasis.

**<b>:** Negrito sem semântica. <b> vêm de bold

**<strong>:** Negrito com semântica.

**<mark>:** Como se fosse uma caneta marcador (amarela por padrão), não tem semântica.

Se fossemos ordenar por nível de influência na semântica da página, ficaria assim:

1º - <strong>

2º - <em>

3º - <i>

4º - <b> e <mark>

No fim vou usar mais <strong> e <em> mesmo.

Existem outras tags de modificador de texto, como:

<del> - Risco no meio da palavra.

<ins> - Sublinha o texto

<small> Deixa o texto menor

<big> Deixa o texto maior

<sup> Deixa o texto pequeno em cima (tipo expoente)

<sub> Deixa o texto pequeno embaixo (tipo química)

<abbr> Usado para abreviação, não sei se entendi

<ruby> Usado para fontes orientais

<cite> Usado para citações longas

<q> Usados para citações curtas

<samp> Formata para formato de código

<kdb> Formata para texto que parece ter sido digitado em teclado

Uma imagem contendo Diagrama

Descrição gerada automaticamente <var> Formata para parecer um termo técnico

**Listas:**

Listas não ordenadas - **<ul>:**

Dentro de uma tag <ul> (unordered list) só pode-se ter a tag **<li>** (list item).

<ul>

<li>

</li>

<li>

</li>

<li>

</li>

</ul>

Listas ordenadas - **<ol>:**

Dentro de uma tag <ol> (ordered list) só pode ter a tag **<li>** (list item).

<ol>

<li>

</li>

<li>

</li>

<li>

</li>

</ol>

**Links <a> (anchor):** Tag para redirecionar para outra página, essa tag precisa ter dois atributos importantes: href (hypertext reference) e target (alvo). href trata-se do caminho para o qual quero encaminhar o usuário caso ele clique no local com essa tag, já target escolhe se o usuário vai ser direcionado na mesma janela, em outra janela ou alguma outra particularidade.

Existem quatro tipos de href possíveis (aaa é só pra exemplo):

Para links externos: href=”https://aaa.com”

Para links internos: href=”aaa.html”

Para seções dentro da mesma página: href=”#aaa”

Para enviar email: href=”mailto:aaaa@gmail.com”

Não é mais recomendado usar o href para email, visto que o usuário precisaria ter um programa de email instalado, mas hoje a maioria das pessoas usa o email via web.

**Quebra de linha <br> (break roll):**

Pula uma linha, nada demais.

**Quebra de linha horizontal <hr> (horizontal roll):**

Pula uma linha e põe uma linha cinza na linha pulada para simbolizar o pulo, embora seja uma tag semântica.