**Introdução Parte 1**

Inicialmente precisamos entender que todo software é um agrupamento de pastas (diretórios) e arquivos. Pastas a nível de organização e arquivos a nível de execução de comandos e tarefas. Nesse módulo estudaremos dois tipos de arquivos: .html e .css

HTML (Hyper Text Markup Language) faz a estrutura do website e organiza as informações.

CSS (Cascading Style Sheet) adiciona estilo ao website.

Para criar e trabalhar com esses arquivos, nós devemos usar um editor de texto. No nosso caso usaremos o Vs-Code.

**Linha de Comando no Prompt:**

mkdir 🡪 cria um diretório

cd 🡪 para acessar diretórios

ls 🡪 lista o que tem no diretório

code . 🡪 abre o vscode no diretório

**Atributos HTML:**

Atributos servem para personalizar e adicionar funcionalidades a uma tag HTML. Existem muitos atributos e muitas tags tem seus atributos próprios.

**🡨 Atributos em Comum entre as Tags 🡪**

class=“” 🡪atribui uma classe para o elemento (uma classe pode ser utilizada para um ou mais elementos);

id=“” 🡪 Atribui um id para o elemento (um id deve ser único, ou seja, atribuído a um único elemento);

style=“” 🡪 Permite incluir elementos CSS (estilo) dentro da tag;

lang=“” 🡪 Define o idioma principal do elemento;

title=“” 🡪 Define o título do elemento;

alt=“” 🡪 Define um texto alternativo e, por isso, é muito ulitizado em imagens e auxilia no SEO;

hidden 🡪 Oculta o elemento;

align=“” 🡪 Permite definir o alinhamento do item. Ex: right, center, left e justify;

width=“” 🡪 Define uma largura para o elemento;

height=“” 🡪 Define uma altura para o elemento.

**Tags HTML:**

**🡨 Tags Estruturais 🡪**

<header></header> 🡪 Definem um cabeçalho;

<main></main> 🡪 Representam o conteúdo principal do seu corpo, ou seja, o conteúdo relacionado diretamente com o tópico central da página.

<footer></footer> 🡪 Definem um rodapé para a página;

<section></section> 🡪 Definem uma sessão para a página;

<article></article> 🡪 Definem um artigo da página. Separam o conteúdo da página. São usadas especialmente por blogs;

<aside></aside> 🡪 Representam uma seção de uma página cujo conteúdo é tangelcialmente relacionado ao conteúdo de seu entorno;

<nav></nav> 🡪 Define a navegação do site. Geralmente fica no header;

<div></div> 🡪 Define uma divisão da página. Não possui um valor semântico.

**🡨 Tags de Títulos 🡪**

As tags de títulos vão do <h1></h1> até o <h6></h6>. A ordem de importância segue a ordem crescente. O <h1> é o mais importante e o <h6> o menos.

<span></span> 🡪 É um elemento genérico que é usado para um texto que não é nem um título e nem um parágrafo.

<p></p> 🡪 É o parágrafo propriamente dito.

**🡨 Formulários 🡪**

Basicamente servem para inserir dados. Ex: login, email, senha. A estrutura de um formulário é a seguinte:

<form action=“”>

<label for=“”>Título</label>

<input style=“” id=“” type=“” name=“” placeholder=“” />

</form>

**Entendendo os atributos:**

style=“” 🡪 serve para definir a disposição do formulário, por exemplo, se estarão em linha ou um embaixo do outro;

id=“” 🡪 serve para marcar o elemento e manda-lo para o for=“”;

type=“” 🡪 serve para definir o tipo do input (email, senha, etc.);

name=“” 🡪 serve para indicar qual a chave do input que aparece na url;

placeholder=“” 🡪 serve para auxiliar o input. Aparece o texto no input. Ex: placeholder=“Digite o seu email”.

Tipos de input:

type=“email” 🡪 verifica se a entrada é um email válido;

type=“password” 🡪 esconde a entrada;

type=“text” 🡪 serve apenas para escrever um texto qualquer;

type=“CheckBox” 🡪 é uma caixa de verificação

type=“radio” 🡪 serve para possibilitar a escolha de apenas um valor. O radio precisa contar com um value=“”. Ex: Masculino [x] Feminino [];

type=“range” 🡪 determina um valor com base em um alcance;

type=“number” 🡪 aceita apenas números;

type=“file” 🡪 serve para escolher um arquivo dentro da sua máquina;

type=“date” 🡪 serve para facilitar a escolha de uma data;

Também temos o **<select>**, que não é um input, mas também serve para inserir dados. Ele possibilita a escolha de apenas um valor dentre alguns valores. Assim como o type=“radio” ele terá que contar com um value=“”. Estrutura:

<select name=“”>

<option value=“”>Texto</option>

<option value=“”>Texto</option>

</select>

Além disso, também existe o **<textarea>**, que serve basicamente para inserir um texto maior. Estrutura:

<textarea name=“” id=“”> </textarea>

**🡨 Tags de Mídia 🡪**

As tags de mídia são basicamente as tags de inserção de áudio, imagem e vídeos no site. Tags:

**<img src=“” alt=“” />** 🡪 em src=“” você passa o caminho da imagem.

**<video controls>**

**<source src=“” type=“”>**

**Seu navegador não possui suporte para vídeos.**

**</video>**

Na tag vídeo você usa o atributo controls para controlar todas as funcionalidades do vídeo. E na tag source você coloca o caminho e o tipo do vídeo (Ex: vídeo/mp4).

**<audio controls>**

**<source src=“” type=“”>**

**Seu navegador não possui suporte para áudio.**

**</áudio>**

**<iframe src=“”> </iframe> 🡪** A tag iframe serve para exibir o conteúdo de outra página.

**🡨 Style, Script e Âncora 🡪**

**<style> </style> 🡪** É usada no header e permite escrever CSS dentro do documento HTML.

**<script> </script> 🡪** É usado no header e permite escrever Javascript dentro do documento HTML.

**<a href=“”> </a> 🡪** É a tag de âncora e serve para permitir a navegação entre páginas. O atributo href=“” coloca o caminho para a página a ser navegada. Todo conteúdo dentro da âncora é clicável.

**Keyframes**

É a maneira de construir animações com o CSS. A primeira coisa a fazer é criar um keyframe. Para isso, faça:

@keyframes nome {

from{

border-radius: 0

}

to{

border-radius: 50%

}

}

Para adicionar a animação em um elemento, faça:

.minhadiv{

width: 100px;

height: 100px;

background-color: black;

animation: nome;

animation-duration: 5s;

}

**Media Query**

Para prosseguirmos é importante entendermos sobre a **Responsividade**. Um site responsivo é aquele que se ajusta aos diferentes tamanhos das telas (celular, notebook, PC). Para isso, é importante entender sobre o conceito de **Breakpoints**, que são pontos de quebra com relação as dimensões e o tamanho da tela que nosso site está sendo exibido. Por exemplo, imagine que você criou um site com imagens na horizontal e definiu que a partir de 768px de largura ele se ajuste e as imagens fiquem na vertical para caberem dentro da tela. Dessa forma, o breakpoint é 768px. Temos 3 breakpoints mais utilizados que são:

* 480px 🡪 para usar para dispositivos menores;
* 768px 🡪 pode ser usados para telas menores, mas geralmente é usado para tablets;
* 1280px 🡪 normalmente usado para computadores. Tem gente que usa 1024px também.

Para colocar em prática utilizaremos o **Media Query**:

@media(min-width: 480px){

.minhadiv{

background-color: red

}

}

A maioria dos acessos hoje em dia acontecem em dispositivos móveis. Por isso atualmente o desenvolvimento foca em desenvolver primeiro para mobile. Isso é conhecido de Mobile First. No exemplo feito no vscode (Responsividade e Responsividade-2.0) note que a primeira estrutura usada foi para a menor tela e a partir dela fomos crescendo usando o min-width.

Para um projeto Desktop First usaríamos do max-width, para iniciar com a maior tela e ir diminuindo e adaptando.