MÓDULO 4 – CAÍTULO 25 – AULA 17: COMO CRIAR FORMULÁRIOS COM HTML5

O PROFESSOR INICIA CRIANDO UMA TAG PARA O FORMULÁRIO

Form => tag utilizada para criar formulário

O PROFESSOR FALA QUE PARA ESSA AULA, ESSE ESTILO DE FORMULÁRIO QUE ESTAMOS CRIANDO AINDA NÃO É O IDEAL, MAS É UTILIZADO PARA QUE POSSAMOS APRENDER

EM SEGUIDA, ELE CRIA DENTRO DA TAG form DOIS PARÁGRAMOS E DENTRO DOS PARÁGRAMOS UTILIZA INPUTS, PARA QUE SEJA POSSÍVEL ESCREVER OS DADOS SOLICITADOS.

LEMBRANDO QUE TODO INPUT DEVE TER type, name E id.

<form action="">

        <p>Nome: <input type="text" nome="nome" id="nome"></p>

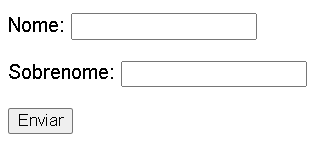
        <p>Sobrenome: <input type="text" nome="sobrenome" id="sobrenome"></p>

        <p><input type="submit" value="Enviar"></p>

    </form>

O PROFESSOR TAMBÉM ADICIONA NO FINAL UM INPUT PARA QUE SEJA POSSÍVEL SUBMETER OS DADOS DIGITADOS.

RESULTADO NO SITE:



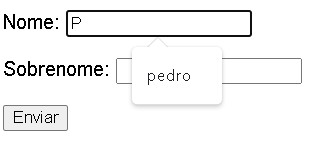
MÓDULO 4 – CAÍTULO 25 – AULA 18: USAR LABEL VAI MELHORAR SEUS FORMULÁRIOS

NESSA AULA O PROFESSOR ENSINARÁ COMO REALIZAR A CONEXÃO ENTRE O TEXTO ANTES DO INPUT, QUE SERÁ ONDE VOCÊ COLOCARÁ AS INFORMAÇÕES DO FORMULÁRIO. NO CASO DO EXEMPLO DA AULA ANTERIOR, É BASICAMENTE REALIZAR A CONEXÃO ENTRE “NOME” E O NOME DIGITADO NO INPUT.

NOME: PEDRO VITOR

SOBRENOME: CARTAXO TAVARES

PARA EVITAR QUE O SITE FIQUE MEMORIAZNDO OS DADOS COLCADOS NOS FORMULÁRIOS E FICAR SUGERINDO SEMPRE QUE FOR DIGITAR NOVOS DADOS, UTILIZE O SEGUINTE PARÂMETRO PARA DESLIGAR ESSAS SUGESTÕES:



<form autocomplete="off">

DURANTE O PROCESSO DE CRIAÇÃO DO FORMULÁRIO, É NECESSÁRIO QUE O MESMO SEJA LINKADO COM ALGUMA LINGUAGEM DE PROGRAÇÃO PARA QUE SEJA POSSÍVEL ARMAZENAR OS DADOS COLOCADOS DENTRO DO FORMULÁRIO.

NA AULA, O PROFESSOR UTILIZA DA LINGUAGEM PHP PARA FAZER ISSO.

É NECESSÁRIO QUE O SITE ENTENDA A LIGAÇÃO ENTRE A ETIQUETA E O SEU RESPECTIVO INPUT, COMO MOSTRADO NO EXEMPLO ACIMA, TEMOS UM ELEMENTO A SER PREENCHIDO E UM TÍTULO ANTERIOR, INDICANDO COM O QUE DEVE SER PREENCHIDO.

PARA ISSO, UTILIZAMOS O PARÂMETRO label .

DENTTRO DO INPUT COLOCAMOS OS PARÂMETROS name E id.

O PARÂMETRO name É MAIS IMPORTANTE PARA A LINGUAGEM PHP E O PARÂMETRO id É MAIS IMPORTANTE PARA A LINGUAGEM JAVASCRIPT. ENTAO É INTERESSANTE COLOCAR AMBOS PARÂMETROS DENTRO DO INPUT, COMO EXPLICAOD ANTERIORMENTE.

SENDO ASSIM, O PROFESSOR ENVELOPA AS ETIQUETAS (nome e sobrenome) UTILIZANDO A TAG label. A TAG label VEM ACOMPANHADA DO PARÂMETRO for, QUE DEVE SER PREENCHIDO COM O id REFERENTE AO INPUT A SER ASSOCIADO. ASSIM, SERÁ POSSÍVEL QUE OS BUSCADORES (COMO O GOOGLE) E O SEU SITE INDENTIQUE A LIGAÇÃO ENTRE A ENTIQUETA E O DADO ESPECÍFICO.

**OBS: USE SEMPRE O LABEL! É MUITO IMPORTANTE!**

**EXEMPLO DE AULA:**

<form action="cadastro.php" autocomplete="off">

        <p><label for="inome">Nome: </label>

            <input type="text" nome="nome" id="inome"></p>

        <p><label for="isobrenome">Sobrenome:</label>

            <input type="text" nome="sobrenome" id="isobrenome"></p>

        <p><input type="submit" value="Enviar"></p>

    </form>

MÓDULO 4 – CAÍTULO 25 – AULA 19: MÉTODOS get E post PARA FORMULÁRIOS

EXISTEM DOIS MÉTODOS PARA ENVIO DE FORMULÁRIOS.

O PRIMEIRO MÉTODO PADRÃO, É O MÉTODO get. JÁ VEM PADRONIZADO DENTRO DA TAG form.

O SEGUNDO MÉTODO É O POST

OBS: O PROFESSOR DEMONSTRA AMBOS OS MÉTODOS E MOSTRA QUE, QUANDO COLOCAMOS O FORMULÁRIO COM A CONFIGURAÇÃOO PADRÃO DE get, OS DADOS INSERIDO E ENVIADOS DENTRO DO SITE FICAM EXPOSTOS NA URL DO SITE. FIZ O TESTE DURANTE O EXEMPLO E VI QUE NO MEU NÃO APARECE, ACREDITO QUE JÁ DEVE TER OCORRIDO ALGUMA ATUALIZAÇÃO OU DO NAVEGADOR (google chrome) OU DO PRÓRPIO HTML5.

Ex: MÉTODO POST

<form method="post" action="cadastro.php" autocomplete="off">

DEPOIS DE UTILIZAR O MÉTODO POST, O PROFESSOR ENSINA OUTRA FORMA DE ENCONTRAR OS DADOS DIGITADOS DENTRO DO inspecionar DO GOOGLE CHROME. DESSA FORMA EU TAMBÉM NÃO CONSEGUI ENCONTRAR ESSAS DADOS.

MAS O PROFESSOR AFIRMA QUE EM NENHUM DOS FORMATOS É SEGURO E É NECESSÁRIO QUE SEJA CRIPTOGRAFADOS OS DADOS, ATRAVÉS DE OUTROS MÉTODOS, PARA ASSEGURAR OS DADOS DE QUEM PREENCHER O FORMULÁRIO.

EM QUE MOMENTO USO get OU USO post.

UTILIZA GET QUANDO OS DADOS QUE VOCÊ ESTÁ SOLICITANDO AO USUÁRIO NÃO SEJAM SENSÍVEIS, COMO POR EXEMPLO, NOME, ALTURA, IDADE, PESO.

MAS QUANDO FOR SOLICITAR DADOS COMO SENHA, CEP, ENDEREÇO, DADOS PESSOAIS, É NECESSÁRIO UTILIZAR O post DENTRO DO SEU FORMULÁRIO.

O get TAMBÉM É LIMITADO PELO NÚMERO DE BYTES (TEM QUE SER MENOS DE 3000 BYTES) DENTRO DO FORMULÁRIO E TAMBÉM É IMPOSSÍBILITADO DE ENVIAR FOTOS.

MÓDULO 4 – CAÍTULO 25 – AULA 20: CRIANDO CAIXAS DE TEXTO E DE SENHA

O PROFESSOR CRIA UM NOVO HTML PARA UTILIZAR NOVOS FORMATOS DE INPUT

<form action="cadastro.php" method="post" autocomplete="off">

        <p>

            <label for="iusu">Usuário</label>

            <input type="text" name="usu" id="iusu">

        </p>

        <p>

            <label for="isenha">Senha</label>

            <input type="password" name="senha" id="isenha">

        </p>

        <p>

            <input type="submit" value="Enviar">

            <input type="reset" value="Limpar">

        </p>

    </form>

FORMULÁRIO COM IDENTIFICADOR, SENHA, INPUT PARA SUBTMETER E TAMBÉM PARA LIMPAR O CONTEÚDO DIGITADO.

TODA AULA O PROFESSOR EXPLICA QUE PARA REALMENTE CRIAR UMA SEGURANÇA DENTRO DE SEU SITE, É NECESSÁRIO O ESTUDO DE HTTPS.

O PROFESSOR TAMBÉM ADICIONA DENTRO DO INPUT DE ID E SENHA O ATRIBUTO required, PARA QUE, CASO O USUÁRIO NÃO DIGITE NENHUM DADO DENTRO DAS CAIXAS, O SITE NÃO ENVIE OS DADOS VAZIOS.

TAMBÉM DENTRO DO required É POSSÍVEL DELIMITAR OS NOMES A SEREM DIGITADOS, BEM COMO SENHAS, COLOCANDO REQUISITOS MÍNIMOS E MÁXIMOS PARA QUE OS IDS E SENHAS SEJAM ACEITOS (EX: MINÍMO 5 DIGITOS OU NO MÁXIMO 15 DIGITOS, SENHA COM LETRA MAIUSCULA, MINUSCULA E CARACTERE.)

INPUT USUÁRIO

<label for="iusu">Usuário</label>

            <input type="text" name="usu" id="iusu" required minlength="5" maxlength="30" size="10" placeholder="nome do usuário" autocomplete="username">

Min length e Max lenght => delimita a quantidade de letras que podem ser colocadas dentro do input de usuário.

Size => tamanho da caixa do input

Placeholder => mensagem prévia dentro da caixa de usuário, mas que é apaga instantaneamente quando o usuário começa a digitar dentro da caixa.

Autocomplete = “username” => é necessário que o autocomplete também esteja ativado dentro do form. Utilizado para identificar que aquela caixa é do nome do usuário e realiza a sugestão baseada nos dados digitas apenas naquela caixa em específico.

INPUT SENHA

<label for="isenha">Senha</label>

            <input type="password" name="senha" id="isenha" required minlength="8" maxlength="20" size="8" placeholder="min. 8 letras" autocomplete="current-password">

BASICAMENTE OS MESMOS EXEMPLOS UTILIZADOS NO input USUÁRIO, MAS COM ALTERAÇÕES EM SUAS CARACTERÍSTICAS.

Autocomplete =”current-password” => autocomplete utilizado para senha.

MÓDULO 4 – CAÍTULO 25 – AULA 21: ELEMENTOS NUMBER, MONTH, DATE E TIME EM FORMULÁRIOS HTML.

OBS: O PARÂMETRO required PASSA A INFORMAÇÃO DA OBRIGATORIEDADE DAS LIMITAÇÕES DADAS.

CÓDIGO CRIADO NA AULA 21

<form action="cadastro.php" method="get" autocomplete="on">

        <p>

            <label for="iname">Nome</label>

            <input type="text" name="name" id="iname" required maxlength="30" placeholder="Nome completo" autocomplete="name" required>

        </p>

        <p>

            <label for="imedia">Média</label>

            <input type="number" name="media" id="imedia" placeholder="0 a 10" value="5" min="0" max="10" step="0.1" required>

        </p>

        <p>

            <label for="imes">Período letivo</label>

            <input type="month" name="mes" id="imes" value="2023-05">

        </p>

        <p>

            <label for="idia">Dia da prova</label>

            <input type="date" name="dia" id="idia" value="2023-05-09">

        </p>

        <p>

            <label for="ihorario">Horário da prova</label>

            <input type="time" name="horario" id="ihorario">

        </p>

        <p>

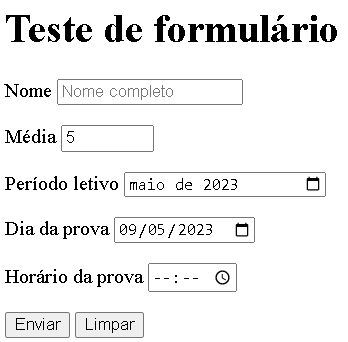
            <input type="submit" value="Enviar">

            <input type="reset" value="Limpar">

        </p>

    </form>

RESULTADO



MÓDULO 4 – CAÍTULO 25 – AULA 22: COMPATIBILIDADE COM NAVEGADORES

MÓDULO 4 – CAÍTULO 25 – AULA 23: FORMULÁRIOS COM TELEFONE E EMAIL

EXEMPLO DA AULA

<h1>Teste de formulário</h1>

    <form action="cadastro.php" method="get" autocomplete="on">

        <fieldset>

            <legend>Dados pessoais</legend>

            <p>

                <label for="inome">Nome</label>

                <input type="text" name="nome" id="inome" minlength="5" maxlength="20" required>

            </p>

            <p>

                <label for="imail">Email:</label>

                <input type="email" name="email" id="imail" autocomplete="email" required>

            </p>

            <p>

                <label for="itel">Telefone:</label>

                <input type="tel" name="tel" id="itel" autocomplete="tel" pattern="^\(\d{2}\)\d{4,5}-\d{4}$" required placeholder="(xx) xxxx-xxxx">

            </p>

        </fieldset>

        <!-- Estudar mais sobre RegEx -->

        <p>

            <input type="submit" value="Enviar">

            <input type="reset" value="Limpar">

        </p>

    </form>

RESULTADO



O PROFESSOR INTRODUZ DOIS NOVOS TIPOS DE INPUT, SENDO ELES PARA EMAIL E PARA TELEFONE.

EXERCITAMOS MAIS UMA VEZ VÁRIAS (NÃO TODAS, POIS EXISTEM MUITAS OUTRAS) CONFIGURAÇÕES PARA INPUTS.

GUANABARA TAMBÉM MOSTORU O PARÂMETRO pattern DENTRO DO INPUT PARA TELEFONE, ONDE ELE CONFIGUROU O PADRÃO DE DIGITAÇÃO NECESSÁRIO PARA QUE O FORMULÁRIO ACEITE O NÚMERO DE TELEFONE. CONFESSO QUE É BEM COMPLICADO E QUE NÃO ENTENDI MUITO BEM, MAS ELE EXPLICA QUE A MELHOR FORMA DE REALIZAR ESSE TIPO DE LIMITAÇÃO E CONFIGURAÇÃO ADEQUADA É ATRAVÉS DAS LINGUAGEM DE PROGRAMAÇÃO.

O PROFESSOR ADICIONOU TAMBÉM DUAS NOVAS TAGS, SENDO ELAS:

Fieldset => adiciona um quadro específico nas informações que deseja, por meio do envelopamento dentro do html.

Legend => adiciona uma legenda no fieldset, como mostrado no exemplo acima (Dados pessoais)

MUITO IMPORTANTE ESSAS TAGS AO MEU VER, PARA ORGANIZAÇÃO DO SEU FORMULÁRIO.

MÓDULO 4 – CAÍTULO 25 – AULA 24: CHECKBOX E RADIO BUTTON EM HTML

EXEMPLO DESSA AULA

<h1>Teste de formulário</h1>

    <form action="cadastro.php" method="get" autocomplete="on">

        <fieldset>

            <legend>Sexo</legend>

            <input type="radio" name="sx" id="isxmas" value="M"> <label for="isxmas">Masculino</label>

            <input type="radio" name="sx" id="isxfem" value="F" checked> <label for="isxfem">Feminino</label>

        </fieldset>

        <fieldset>

            <legend>Esportes favoritos</legend>

            <input type="checkbox" name="esfut" id="iesfut" checked> <label for="iesfut">Futebol</label> <br>

            <input type="checkbox" name="esbas" id="iesbas"> <label for="iesbas">Basquete</label> <br>

            <input type="checkbox" name="esnat" id="iesnat"> <label for="iesnat">Natação</label> <br>

            <input type="checkbox" name="esten" id="iesten"> <label for="iesten">Tenis</label> <br>

        </fieldset>

        <p>

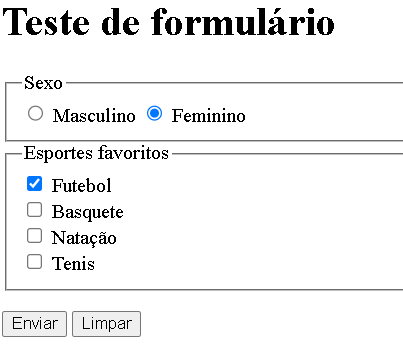
            <input type="submit" value="Enviar">

            <input type="reset" value="Limpar">

        </p>

    </form>

RESULTADO



NESSA AULA FOI INTRODUZIDO A UTILIZAÇÃO DOS INPUTS DO TIPO radio E checkbox. AMBOS OS TIPOS SÃO DE MARCAR, MAS O RADIO É MAIS UTILIZADO PARA SITUAÇÕES ONDE EXISTEM SÓ UMA RESPOSTA, COMO POR EXEMPLO O SEXO DA PESSOA. JÁ O checkbox PODE SER UTILIZADO EM SITUAÇÕES DE MULTIPLA ESCOLHA.

DENTRO DO radio É IMPORTANTE LEMBRAR DE MANTER AMBAS AS TAGS COM O MESMO name, NO CASO DO EXEMPLO FOI UTILIZADO O name = sx. SENDO ASSIM, FICA IMPOSSIBILITADO DO USUÁRIO MARCAR AMBAS AS OPÇÕES DE SEXO. SEM ESSA CONFIGURAÇÃO, O USUÁRIO PODE MARCAR AMBAS AS OPÇÕES, PROPORCIONANDO UM PROBLEMA DE IDENTIFICAÇÃO DO USUÁRIO.

NO checkbox A CONFIGURAÇÃO É MAIS PADRONIZADA COM OS PARÂMETROS ANTERIORES, SEM NOVIDADES.

É POSSÍVEL UTILIZAR UM OUTRO PARÂMETRO DENTRO DOS INPUTS, SENDO ELE O checkbox , PARÂMETRO E PERMITE QUE O SITE JÁ VENHA COM UMA SELEÇÃO PREDETERMINADA, APENSAR PARA QUE O USUÁRIO CONSIGA IDENTIFICAR QUE PODE SELECIONAR E MARCAR AQUELAS CONFIGURAÇÕES.

MÓDULO 4 – CAÍTULO 25 – AULA 25: ELEMENTOS, RANGE E FILE

EXEMPLO DA AULA

<h1>Teste de formulário</h1>

    <form action="cadastro.php" method="post" autocomplete="on">

        <p>

            <label for="icor">Cor:</label>

            <input type="color" name="cor" id="icor" value="#00ff00">

        </p>

        <p>

            <label for="inivel">Nível de satisfação</label>

            <input type="range" name="nivel" id="inivel" min="1" max="5" value="1">

        </p>

        <p>

            <label for="ifoto">Foto do perfil</label>

            <input type="file" name="foto" id="ifoto">

        </p>

        <p>

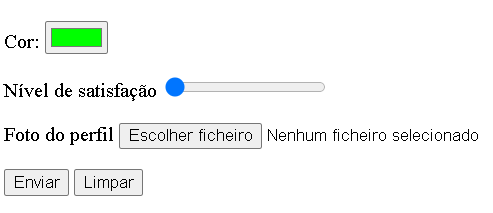
            <input type="submit" value="Enviar">

            <input type="reset" value="Limpar">

        </p>

    </form>

RESULTADO



NESTA AULA O PROFESSOR APRESENTA OS MÉTODOS DE INPUT DO TIPO file, range e color

BASICAMENTE, O INPUT COLOR NOS APRESENTA UMA INTERFACE NO SITE ONDE VOCÊ PODE ESCOLHER UMA COR ESPECÍFICA. É POSSIVEL ESPECIFICAR QUAL COR SERÁ APRESENTADA INICIALMENTE DENTRO DO INPUT, POR MEIO DO PARÂMETRO value.

O INPUT RANGE APRESENTA NO SITE UMA BARRA ONDE VOCÊ PODE SELECIONAR ALGUM NÍVEL PARA ESPECIFICAR ALGO, COMO O ATENDIMENTO OU SUA SATISFAÇÃO. (BOM PARA DAR NOTA AVALIATIVA PARA ALGO). AQUI É POSSÍVEL CONFIGURAR OS VALORES MÁXIMOS E MÍNIMOS TAMBÉM. (max. e min.)

JÁ O INPUT FILE SE RESUME A ENVIAR ALGUM TIPO DE ARQUIVO QUE SITE ESTÁ SOLICITANDO

QUANDO FOR UTILIZAR O CONTROLE DE ENVIO DE ARQUIVOS, É NECESSÁRIO UTILIZAR O metodh =post

MÓDULO 4 – CAÍTULO 25 – AULA 26: SELECT, DATALIST E TEXTAREA EM HTML

EXEMPLO DA AULA

<h1>Exemplo formulário</h1>

    <form action="cadastro.php" method="get" autocomplete="on">

        <p>

            <label for="iest">Estado</label>

            <select name="estado" id="iest">

                <option value="" selected>--- Escolha ---</option>

                <optgroup label="Região Sudeste">

                    <option value="SP">São Paulo</option>

                    <option value="RJ">Rio de Janeiro</option>

                    <option value="MG">Minas Gerais</option>

                </optgroup>

                <optgroup label="Região Nordeste">

                    <option value="CE">Ceará</option>

                    <option value="PB">Paraíba</option>

                    <option value="PE">Pernambuco</option>

                    <option value="RN">Rio Grande do Norte</option>

                </optgroup>

            </select>

        </p>

        <p>

            <label for="iprof">Profissão:</label>

            <input type="text" name="prof" id="iprof" list="lstprof">

            <datalist id="lstprof">

                <option value="ADM">Administrador</option>

                <option value="CONT">Contador</option>

                <option value="DEV">Desenvolvedor</option>

                <option value="PROF">Professor</option>

            </datalist>

        </p>

        <p>

            <label for="imsg">Mensagem</label> <br>

            <textarea placeholder="Sua mensagem aqui." name="msg" id="imsg" cols="30" rows="10"></textarea>

        </p>

        <p>

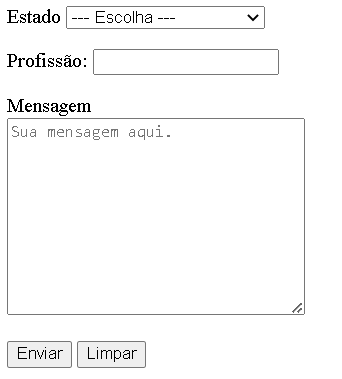
            <input type="submit" value="Enviar">

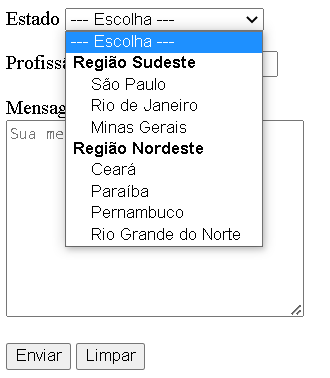
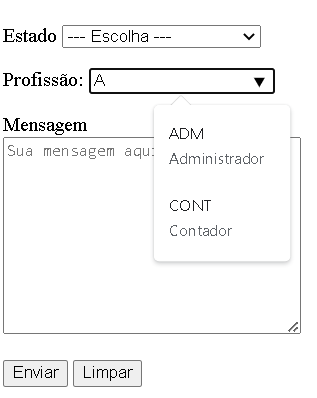
            <input type="reset" value="Limpar">

        </p>

    </form>

RESULTADO



O PROFESSOR APRESENTA NESTA AULA OUTROS TIPOS DE TAGS QUE SÃO POSSÍVEÍS UTILIZAR DENTRO DO form.

Select

Option

Optiongroup

Datalist

Textarea

O SELECT NADA MAIS É DO QUE UMA LISTA DIGITADA PELO PROGRAMADOR QUE SERÁ APRESENTADA DENTRO DO SEU SITE, ONDE O USUÁRIO PODERÁ ESCOLHER UMA OPÇÃO PARA DETERMINAR ALGO, COMO POR EXEMPLO O SEU ESTADO.

O PROFESSOR UTILIZAR JUNTO AO select A TAG option, PARA GERAR AS OPÇÕES APRESENTADAS DENTRO DA LISTA E TAMBÉM O optiongroup QUE NADA MAIS É DO QUE UM AGRUPAMENTO DE Options. ISSO É REPRESENTADO NO EXEMPLO COMO SENDO AS REGIÕES E, DENTRO DE CADA REGIÃO, TEM SEUS ESTADOS ESPECÍFICOS.

O datalist FUNCIONA COMO UM select, MAS AO INVÉS DE APARECER COMO UMA SELEÇÃO LISTADA, APARECE DE UMA FORMA PARECIDA COM QUANDO DEIXAMOS OS SITES GRAVAREM NOSSOS EMAILS PARA QUANDO RETORNARMOS NÃO PRECISARMOS DIGITAR TUDO NOVAMENTE, COMO MOSTRADO NA IMAGEM ACIMA.É POSSÍVEL TAMBÉM DIGITAR O QUE VOCÊ DESEJA, CASO NÃO POSSUA A OPÇÃO QUE REPRESENTE VOCÊ. NO CASO DAS PROFISSOES, A TAG datalist PERMITE QUE VOCÊ VEJA UMA LSITA PREDETERMINADA DE PROFISSÕES QUE O DESENVOLVEDOR COLOCOU NA LISTA E TAMBÉM PERMITE QUE VOCÊ ESCREVA SUA PROFISSÃO, CASO ELA NÃO ESTEJA LSITADA.

O textarea NADA MAIS É UMA ÁREA ONDE VOCÊ PODE DIGITAR DENTRO DO SITE, E QUE, ATRAVÉS DO ENVIO DE DADOS, PODE SER ARMAZENADA. MUITO UTILIZADO PARA QUE O USUÁRIO POSSA DEIXAR SUA OBSERVAÇÃO DO ATENDIMENTO OU DE QUALQUER OUTRA COISA. É POSSÍVEL SER DELIMITADO ATRAVÉS DE OUTROS PARÂMETROS, DETERMINANDO O NÚMERO DE CARACTERES QUE PODEM SER UTILIZADOS, POR EXEMPLO (max length) E TAMBÉM É POSSÍVEL CONFIGURAR O TAMANHO DA CAIXA E POR MEIO DOS PARÂMETROS cols (colunas) e row(linhas)

PARA UTILIZAR A TAG textarea É IMPORTANTE SE ATENTAR A DEIXAR O form COM methoh DO TIPO post.

MÓDULO 4 – CAÍTULO 25 – AULA 27: ELEMENTO OUTPUT EM FORMULÁRIO HTML

EXEMPLO 1 DA AULA:

<form action="cadastro.php" method="get" autocomplete="on">

    <p>

        <label for="in1">Number:</label>

        <input type="number" name="n1" id="in1" min="0" max="50" required oninput="isoma.innerHTML = Number(in1.value) + Number(in2.value)">

    </p>

    <p>

        <label for="in2">Number:</label>

        <input type="number" name="n2" id="in2" min="0" max="50" required oninput="isoma.innerHTML = Number(in1.value) + Number(in2.value)">

    </p>

    <p>

        <label for="isoma">Soma:</label>

        <output name="soma" id="isoma">0</output>

    </p>

    <p>

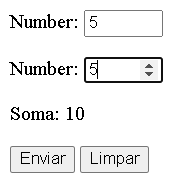
        <input type="submit" value="Enviar">

        <input type="reset" value="Limpar">

    </p>

    </form>

RESULTADO:



O PROFESSOR CIROU O CÓDIGO ACIMA UTILIZANDO AS TAGS output E oninput, SOMADOS AOS CONHECIMENTOS ESTUDADOS PREVIAMENTE, COMO input, label e form, CRIANDO ASSIM MAIS ALGUNS TIPOS DE FORMULÁRIOS.

DENTRO DAS TAGS input O PROFESSOR UTILIZA A PROPRIEDADE oninput PARA INFORMAR AO CÓDIGO QUE AS INFORMAÇÕES COLOCADAS DENTRO DO input SERÃO UTILIZADAS PARA RESOLVER ALGUMA QUESTÃO OU PARA QUE SEJA POSSÍVEL REALIZAR ALGUM CALCULO QUE SERÁ MOSTRADO UTILIZANDO O output. NO O EXEMPLO EM QUESTÃO, O PROFESSOR UTILIZA AS DUAS PROPRIEDADES MENCIONADAS PARA RESOLVER UMA SOMA ENTRE DOIS NÚMEROS. PRIMEIRO ADICIONA UM oninput DENTRO DO PRIMEIRO input, INFORMANDO PARA AQUELA LINHA DE COMANDO QUE O VALOR COLOCADO ALI SERVIRÁ PARA EXECUTAR UMA SOMA. EM SEGUIDA, REPRESENTA O MESMO RACIOCÍNIO DENTRO DO SEGUNDO input, ASSIM, UTILIZANDO O id = isoma DENTRO DO OUTPUT, id ESSE QUE SE REPETE PARA CARACTERIZAR A SAÍDA DO RESULTADO, É POSSÍVEL MOSTRAR O RESULTADO DA SOMA DENTRO DO output.

EXEMPLO 2

form action="cadastro.php" method="get" autocomplete="on">

    <p>

        <label for="inum">Número:</label>

        <input type="range" name="num" id="inum" min="0" max="10" value="5" oninput="ival.innerHTML = Number(inum.value)">

        <output id="ival">5</output>

    </p>

    <p>

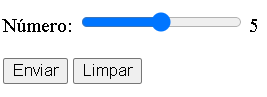
        <input type="submit" value="Enviar">

        <input type="reset" value="Limpar">

    </p>

    </form>

RESULTADO



NO SEGUNDO EXEMPLO, O PROFESSOR UTILIZA DO oninput DENTRO DO input DO TIPO range PARA EXIBIR A CLASSIFICAÇÃO (OU DO NÚMERO EM SÍ) AO LADO DO ELEMENTO, COMO MOSTRADO NA FIGURA. SEMPRE QUE A CLASSIFICAÇÃO DIMINUIR OU AUMENTAR, A EXIBIÇÃO AO LADO MUDARÁ O NÚMERO, SEGUINDO A ROLAGEM.

EXEMPLO 3:

 <form action="cadastro.php" method="get" autocomplete="on">

    <p>

        <label for="iano">Em que ano você nasceu</label>

        <input type="number" name="ano" id="iano" min="1900" max="2023" value="2000" required oninput="calcIdade()">

    </p>

    <p>

        <label for="iidade">Sua idade é:</label>

        <output id="iidade">0</output>

    </p>

    <p>

        <input type="submit" value="Enviar">

        <input type="reset" value="Limpar">

    </p>

    </form>

    <script>

        function calcIdade() {

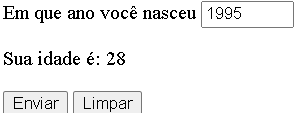
            let atual = new Date().getFullYear()

            iidade.innerHTML = Number(atual) - Number(iano.value)

        }

    </script>

RESULTADO:



O PROFESSOR UTILIZOU MAIS UMA VEZ A PROPRIEDADE oninput E A TAG output PARA EXECUTAR UM CÁLCULO SIMPLES PARA SABER A IDADE DO USUÁRIO PELA DATA DO SEU NASCIMENTO.

PRIMEIRO ELE SOLICITA O ANO DE NASCIMENTO DO USUÁRIO DENTRO DE UMA input E, DENTRO DESSE input ADICIONA UM oninput COM UMA FUNÇÃO.

EM SEGUIDA, UTILIZANDO JAVASCRIPTA, RESOLVE DENTRO DA FUNÇÃO O CALCULO PARA DETERMINAR A IDADE DO USUÁRIO E RETORNA ESSA RESPOSTA NO OUTPUT, UTILIZANDO A CONEXÃO POR MEIO DOS ids.

RESUMIDAMENTE O output NADA MAIS É DO QUE UMA EXIBIÇÃO EM TEMPO REAL UTILIZANDO OS DADOS ADICIONADOS DENTRO DOS inputs, POR MEIO DA UTILIZAÇÃO TAMBÉM DA PROPRIEDADE oninput.

MÓDULO 4 – CAÍTULO 25 – AULA 28: O QUE SÃO AS MEDIA QUERIES EM CSS

AS MEDIA QUERIES SERVEM PARA MELHORAR A RESPONSIVIDADE PARA DIVERSOS TAMANHOS DE TELAS, ONDE SEMPRE QUE O TAMANHO DA TELA FOR ALTERA, O CONTEÚDO DO SITE TAMBÉM SOFRERÁ ALTERAÇÃO DE FORMA AUTOMÁTICA E DE FORMA PROPORCIONAL, MANTENDO O CONTEÚDO VISIVEL E LEGIVEL.

MÓDULO 4 – CAÍTULO 25 – AULA 29: CRIANDO UM SITE COM VERSÃO PARA IMPRESSORA

NESTA AULA O PROFESSOR CRIA UM SITE DE NOTÍCIAS PARA DESENVOLVER UMA RESPONSIVIDADE PARA IMPRESSÃO.

INICIALMENTE O PROFESSOR CRIA O HTML BASE DE UM SITE DE NOTÍCIAS E ADICIONA DOIS LINKS PARA CSSs DIFERENTES, UM PARA O CSS DA TELA (WEB PAGE) E UM CSS PARA A IMPRESSÃO.

AQUI ELE JÁ INDICA QUE ESSES DOIS LINKS SÃO AS CHAMADAS PARA AS MEDIA QUERIES. ELE TAMBÉM INFORMA QUE NÃO NECESSARIAMENTE É PRECISO CRIAR DOIS CSSs PARA FAZER ESSAS MEDIA QUERIES, QUE É POSSÍVEL FAZER DE OUTRA FORMA, MAS QUE, PARA EXEMPLIFICAR PARA NÓS ALUNOS FICOU MAIS FÁCIL DESSA FORMA.

EU PARTICULAMENTE GOSTEI DESSA SEPARAÇÃO, PRA MIM AJUDA BASTANTE NA ORGANIZAÇÃO, MAS VAMOS ESPERAR PARA VER COMO QUE ELE IRÁ REALIZAR OS PRÓXIMOS PASSOS PARA OUTRAS MEDIA QUERIES.

DENTRO DOS CSSs TELA E PRINT, O PROFESSOR REALIZA CONFIGURAÇÕES PARA OS MESMOS ELEMENTOS MAS DE FORAM DIFERENTE, ASSIM, ALGUMAS CARACTERÍSTICAS DA WEB PAGE NÃO SÃO LEVADAS PARA A PÁGINA DE IMPRESSÃO, CRIANDO UMA IMPRESSÃO SIMPLES E ADEQUADA PARA O TAMANHO DA FOLHA DE PAPEL.

PARA A TELA DE IMPRESSÃO O PROFESSOR ALTEROU A FONTE UTILIZADA, BEM COMO AUMENTOU O TAMANHO DA FONTE. COLOCOU O PREENCHIMENTO DA TAG article PARA A IMPRESSÃO DE 100%, LOGO O TEXTO IRÁ PREENCHER TODA A PÁGINA DA FOLHA, AO ONTRÁRIO DO SITE, ONDE ESTA CENTRALIZADO E NÃO PREENCHE 100% DA TELA, FACILITANDO A LEITURA ONLINE. ELE TAMBÉM RETIRA A ÁREA DE MENU DO SITE, QUE POSSUE LINKS E ITENS DESNECESSÁRIOS PARA A IMPRESSÃO E LEITURA DO CONTEÚDO.

OUTRA ADIÇÃO INTERESSANTE PARA A PÁGINA DE IMPRESSÃO É UMA PSEUDOCLASSE COLCOADA NO ARTICLE.:

article::after {

    content: 'Esse artido foi impresso através do site www.cursoemvideo.com';

}

AQUI, O PROFESSOR ADICIONA UM TEXTO QUE SERÁ EXIBIDO APENAS NA PÁGINA DE IMPRESSÃO, MOSTRANDO DE ONDE O ARTIGO FOI RETIRADO.

CSS WEB PAGE

\*{

    margin: 0px;

    padding: 0px;

}

html {

    font-family: Arial, Helvetica, sans-serif;

    font-size: 1em;

}

header {

    background-color: lightgray;

}

header > h1 {

    text-align: center;

}

menu ul {

    list-style-type: none;

    background-color: gray;

}

menu li {

    background-color: darkgray;

    color: white;

    display: inline-block;

    padding: 10px;

}

article {

    width: 600px;

    display: block;

    margin: auto;

}

CSS PARA IMPRESSÃO

\*{

    margin: 10px;

    padding: 10px;

}

html {

    font-family: 'Courier New', Courier, monospace;

    font-size: 1.5em;

    line-height: 1.5em;

}

menu {

    display: none;

}

article {

    width: 100%;

}

article::after {

    content: 'Esse artido foi impresso através do site www.cursoemvideo.com';

}

VALE LEMBRAR QUE DENTRO DOS LINKS PARA OS CSSs DE TELA E PRINT (IMPRESSÃO) O PROFESSOR ADICIONOU OS TIPOS DE MEDIA QUERIES

<link rel="stylesheet" href="estilos/tela.css" media="screen">

<link rel="stylesheet" href="estilos/impressora.css" media="print">

FORAM UTILIZADOS OS TIPOS screen E print POR MEIO DA PROPRIEDADE media.

O PROFESSOR FALA TAMBÉM QUE O TIPO DE media screen É MTO GENERALISTA.

MÓDULO 4 – CAÍTULO 25 – AULA 30: MÚLTIPLAS MEDIA FEATURES COM CSS

MÉDIA QUERIE = MEDIA TYPE + MEDIA FEATURE

<link rel="stylesheet" href="estilos/style.css" media="all">

<link rel="stylesheet" href="estilos/retrato.css" media="screen and (orientation:portrait)">

<link rel="stylesheet" href="estilos/paisagem.css" media="screen and (orientation:landscape)">

O PROFESSOR ADICIONOU OS LINKS PARA OS CSS DE CADA MEDIA QUERY, NESSE CASO, UTILIZOU UM LINK PARA UM CSS ONDE AS CONFIGURAÇÕES DE CSS IRÃO SE REPETIR EM TODOS OS CONTEXTOS EM QUE A PAGINA FOR COLOCADA E CRIOU OUTROS DOIS LINKS PARA CONFIGURAÇÕES ESPECÍFICAS DO CONTEXTO, NO CASO DO EXEMPLO, UMA CONFIGURAÇÃO ESPECÍFICA PARA QUANDO A TELA ESTIVER EM MODO EM PÉ (PORTRAIT) E DEITADO (PAISAGEM/LANDSCAPE).

A TIPO DE MEDIA DO PRIMEIRO LINK É all, QUE FAZ REFERÊNCIA AS CONFIGURAÇÕES QUE SÃO REPETIDAS EM QUALQUER SITUAÇÃO. JÁ NOS DOIS OUTROS LINKS, O PROFESSOR ESPECÍFICA O MEDIA TYPE E TAMBÉM O MEDIA FEATURE. AMBOS OS media types SÃO DO TIPO screen, MAS COM O ADICIONAL DA media feature DO TIPO orientation, QUE NO CASO SÃO DIFERENCIADAS PELO POSICIONAMENTO DA TELA (DEITADO OU EM PÉ).

AS MEDIA FEATURES SÃO ESPECÍFICADAS SEMPRE DENTRO DE PARÊNTESES.

MÓDULO 4 – CAÍTULO 25 – AULA 31: SEGUINDO A ORIENTAÇÃO DO DISPOSITIVO

O PROFESSOR NESSA AULA CONFIGURA TODAS AS MEDIA QUERIES COLOCADAS NO EXERCÍCIO ANTERIORES

CONFIGURAÇÃO DO MEDIA QUERY TIPO ALL

\*{

    margin: 0px;

    padding: 0px;

}

html, body {

    width: 100vw;

    height: 100vh;

    background-color: #233eff;

    background-size: contain;

    background-repeat: no-repeat;

}

h1 {

    color: white;

    text-shadow: 2px 2px 0px #233dff71;

    padding: 10px;

}

CONFIGURAÇÃO DO MEDIA QUERY RETRATO (PORTRAIT)

body {

    background-image: url(../imagens/cev-portrait.jpg);

    background-position: center bottom;

}

CONFIGURAÇÃO DO MEDIA QUERY PAISAGEM (LANDSCAPE)

body {

   background-image: url(../imagens/cev-landscape.jpg);

   background-position: left bottom;

}

COMO EXPLICADO NA AULA ANTERIOR, O PROFESSOR CONFIGURAÇA O MEDIA QUERY DO TIPO ALL COM CONFIGURAÇÕES BASES QUE SE REPETIRAM EM TODAS OS TIPOS DE TELA, INDEPENDENTE DO TAMANHO OU POSICIONAMENTO. JÁ AS CONFIGURAÇÕES NAS OUTRAS DUAS MEDIA QUERIES DO EXEMPLO, SENDO ELAS PARA QUANDO A TELA ESTIVER DEITADA E EM PÉ, SÃO ESPECÍFICAS PARA ESSAS SITUAÇÕÇES.

MÓDULO 4 – CAÍTULO 25 – AULA 32: REUNINDO TUDO EM UM ÚNICO CSS

QUANDO O PROJETO É UM POUCO MENOR, É POSSÍVEL CRIAR APENAS UM ARQUIVO CSS, DIMINUINDO O NÚMERO DE ARQUIVOS DENTRO DO PROJETO. O EXEMPLO DA AULA ANTERIOR É BEM PEQUENO, SENDO ASSIM, É POSSÍVEL CRIAR APENAS UM ARQUIVO CSS QUE IRÁ SER CONFIGURADO PARA TOADS AS SITUAÇÕES.

<style>

        /\* Declarações gerais \*/

        \*{

            margin: 0px;

            padding: 0px;

        }

        html, body {

            width: 100vw;

            height: 100vh;

            background-color: #233eff;

            background-size: contain;

            background-repeat: no-repeat;

        }

        h1 {

            color: white;

            text-shadow: 2px 2px 0px #233dff71;

            padding: 10px;

        }

        /\* Declarações retrato \*/

        @media screen and (orientation: portrait) {

            body {

                background-image: url(../imagens/cev-portrait.jpg);

                background-position: center bottom;

            }

        }

        /\* Declarações paisagem \*/

        @media screen and (orientation: landscape) {

            body {

                background-image: url(../imagens/cev-landscape.jpg);

                background-position: left bottom;

            }

        }

    </style>

CONFIGURAÇÃO TODA EM UM ÚNICO STYLE.CSS

MÓDULO 4 – CAÍTULO 25 – AULA 33: MOBILE FIRST (PRIMEIRO O MÓVEL)

PRIMEIRA VANTAGEM: O ALGORITIMO DO GOOGLE FAVORECE SITES COM FOCO EM MOBILE FIRST.

SEGUNDA VANTAGEM: MELHORA A EXPERIÊNCIA DO USUÁRIO.

VAMOS INICIAR UM NOVO PROJETO.

MÓDULO 4 – CAÍTULO 25 – AULA 34: INICIANDO UM SITE MOBILE FIRST

O PROFESSOR CIRA O HTML JUNTO COM AS CONFIGURAÇÕES BASE DO MOBILE FIRST.

HTML

<body>

    <main>

        <h1>Testando Media Queries</h1>

        <img id="phone" src="imagens/icon-phone.png" alt="Acessando via smartphone">

        <img id="tablet" src="imagens/icon-tablet.png" alt="Acessando via tablet">

        <img id="print" src="imagens/icon-print.png" alt="Versão para impressão">

        <img id="pc" src="imagens/icon-pc.png" alt="Versão para desktop">

        <img id="tv" src="imagens/icon-tv.png" alt="Versão para TV">

    </main>

</body>

CSS MOBILE FIRST

@charset "UTF-8";

/\* Versão Mobile First \*/

\* {

    font-family: Arial, Helvetica, sans-serif;

    font-size: 1.3em;

    margin: 0px;

    padding: 0px;

    box-sizing: border-box;

}

html {

    height: 100vh;

}

body {

    background-color: black;

    background: url('../imagens/back-phone.jpg') no-repeat center center/cover;

}

main {

    background-color: rgba(255, 255, 255, 0.568);

    width: 90%;

    margin: auto;

    border-radius: 10px;

    padding: 10px;

    margin-top: 20px;

}

h1 {

    text-align: center;

    color: white;

    text-shadow: 2px 2px 2px rgba(0, 0, 0, 0.495);

}

img {

    display: block;

    margin: auto;

}

img#phone { display: block; }

img#tablet { display: none; }

img#print { display: none; }

img#pc { display: none; }

img#tv { display: none; }

MÓDULO 4 – CAÍTULO 25 – AULA 36: DEVICE BREAKPOINTS

TYPICAL DEVIDE BREAKPOINTS SEGUNDO A W3C

    Pesquise sobre Typical Device Breakpoints

-------------------------------------------

Pequenas telas: até 600px

Celular: 600px - 768px;

Tablet: 768px - 992px;

Desktop: 992px - 1200px;

Grandes telas: Ácima de 1200px;

LIMITES DADO PELO PROFESSOR NA DURANTE A AULA. SEMPRE É BOM PESQUISAR NO DIA QUE VOCÊ FOR REALIZAR O TRABALHO, POIS TALVEZ ESSE VALORES TENHAM MUDADO.

/\*

O site ja tomou como base o CSS para mobile first, logo, não existe a necessidade de criar essa configuração novamente.

@media screen and (max-width: 600px) {

}

@media screen and (min-width: 600px) and (max-width: 768px) {

}

\*/

HTML DO SITE

<body>

    <main>

        <h1>Testando Media Queries</h1>

        <img id="phone" src="imagens/icon-phone.png" alt="Acessando via smartphone">

        <img id="tablet" src="imagens/icon-tablet.png" alt="Acessando via tablet">

        <img id="print" src="imagens/icon-print.png" alt="Versão para impressão">

        <img id="pc" src="imagens/icon-pc.png" alt="Versão para desktop">

        <img id="tv" src="imagens/icon-tv.png" alt="Versão para TV">

    </main>

</body>

CSS BASE

\* {

    font-family: Arial, Helvetica, sans-serif;

    font-size: 1.3em;

    margin: 0px;

    padding: 0px;

    box-sizing: border-box;

}

html {

    height: 100vh;

}

body {

    background-color: black;

    background: url('../imagens/back-phone.jpg') no-repeat center center/cover;

}

main {

    background-color: rgba(255, 255, 255, 0.568);

    width: 90%;

    margin: auto;

    border-radius: 10px;

    padding: 10px;

    margin-top: 20px;

}

h1 {

    text-align: center;

    color: white;

    text-shadow: 2px 2px 2px rgba(0, 0, 0, 0.495);

}

img {

    display: block;

    margin: auto;

}

img#phone { display: block; }

img#tablet { display: none; }

img#print { display: none; }

img#pc { display: none; }

img#tv { display: none; }

CSS MEDIA QUERY

@media print {

    \*{

        font-family: 'Courier New', Courier, monospace;

    }

    body {

        background-image: url('../imagens/back-print.jpg'); /\* Desnecessário \*/

    }

    main h1 {

        text-shadow: none;

        color: black;

    }

    main {

        border: 2px solid black;

        width: 90vw;

    }

    main::after {

        content: 'Essa impressão foi feita através do site www.cursoemvideo.com';

        text-decoration: overline;

    }

    img#phone { display: none; }

    img#tablet { display: none; }

    img#print { display: block; }

    img#pc { display: none; }

    img#tv { display: none; }

}

@media screen and (min-width:768px) and (max-width:992px) /\* Tablet \*/ {

    body {

        background-image: url('../imagens/back-tablet.jpg');

    }

    img#phone { display: none; }

    img#tablet { display: block; }

    img#print { display: none; }

    img#pc { display: none; }

    img#tv { display: none; }

}

@media screen and (min-width:992px) and (max-width:1200px) /\* Desktop \*/ {

    body {

        background-image: url('../imagens/back-pc.jpg');

    }

    img#phone { display: none; }

    img#tablet { display: none; }

    img#print { display: none; }

    img#pc { display: block; }

    img#tv { display: none; }

}

@media screen and (min-width:1200px) /\* Grandes telas \*/ {

    body {

        background-image: url('../imagens/back-tv.jpg');

    }

    img#phone { display: none; }

    img#tablet { display: none; }

    img#print { display: none; }

    img#pc { display: none; }

    img#tv { display: block; }

}

MÓDULO 4 – CAÍTULO 25 – AULA 37: MENU RESPONSIVO COM HTML5 E CSS3

O PROFESSOR UTILIZOU ESSA AULA PARA EXPLICAR O PROJETO QUE IREMOS FAZER DAQUI PRA FRENTE.

MÓDULO 4 – CAÍTULO 25 – AULA 38: CONFIGURANDO O LAYOUT DO PROJETO

AULA PRÁTICA CRIANDO O LAYOUT DO PROJETO INICIADO NA AULA ANTERIOR.

PROCEDIMENTO ENSINANDO PELO PROFESSOR PARA INCIAR A CRIAÇÃO DE UM SITE, UTILIZANDO O MOBILE FIRST, OU SEJA, CRIAR O SITE INICIALMENTE PARA O USO EM CELULAR E EM SEGUIDA, UTILIZANDO AS MEDIA QUERYS, ADAPTA-LO PARA OS DEMAIS TAMANHOS DE TELAS, DEIXANDO O SITE O MAIS RESPONSIVO POSSÍVEL.

MÓDULO 4 – CAÍTULO 25 – AULA 39: CRIANDO UM MENU HAMBÚGUER

PARA CRIAR O MENU HAMBURGUER, O PROFESSOR VAI ATÉ O SITE DE ICONS DO GOOGLE PARA PEGAR O ICONE QUE SERÁ UTILIZADO DENTRO DO SITE. JÁ NO SITE, VOCÊ TERÁ UMA VARIADADE ABSURDADA DE ICONES PARA PODER USAR NO SEU SITE. CLICANDO NO ÍCONE HAMBURGUER, QUE SERÁ O UTILIZADO NESSA AULA, PODEMOS INSERI-LO DENTRO DO CÓDIGO HTML UTILIZANDO UM LINK DO PRÓPRIO SERVIDOR DO GOOGLE OU BAIXAR UMA FONTE ESPECÍFICA (ESSA PARTE O PROFESSOR NÃO ENSINA).

IMAGEM DO ÍCONE:



LINK DO ÍCONE (SERVIDORES DO GOOGLE):



ASSIM, BASTA COPIAR ESSE LINK E ADICIONAR DENTRO DO SEU HTML, PARA PODER UTILIZA-LO DENTRO DO SITE.

    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200" />

É NECESSÁRIO TAMBÉM COPIAR OUTRA LINHA DE CÓDIGO DENTRO DO SITE DO GOOGLE. ESSA LINHA REPRESENTA O ELEMENTO DO ÍCONE.

<span class="material-symbols-outlined">

            menu

        </span>

O PROFESSOR EXPLICA QUE EXISTE DIVERSAS OUTRAS MANEIRAS DE FAZER ISSO, MAS ESSA FORMA ENSINADA É A MAIS SIMPLES.

FEITO A ADIÇÃO DO ICONE, O PROFESSOR REMOVE DA PÁGINA A VISUALIZAÇÃO DA NAVEGAÇÃO, COM AS OPÇÕES DO MENU.

CSS nav

nav {

    display: none;

}

E EM SEGUIDA A INICIO A CRIAÇÃO DE UM CÓDIGO USANDO JAVASCRIPT PARA CRIAR UMA INTEREÇÃO ENTRE O USUÁRIO E O SITE.

ADICIONA PRIMEIRAMENTE UMA FUNÇÃO DENTRO DO span ONDE ESTÁ O ICONE DO MENU E UM id PARA O ELEMENTO nav (NAVEGAÇÃO) ONDE ESTÃO ALOCADOS AS OPÇÕES.

<span id="icon" class="material-symbols-outlined" onclick="clickMenu()">

            menu

        </span>

<nav id="itens">

ASSIM, É POSSÍVEL CRIAR O CÓDIGO EM JAVASCRIPT

<script>

        function clickMenu() {

            if (itens.style.display == 'block') {

                itens.style.display = 'none'

            } else {

                itens.style.display = 'block'

           }

        }

    </script>

A INTERAÇÃO FUNCIONA ATRAVÉS DO CLICK UTILIZANDO O CURSOS SOBRE O ELEMENTO MENU, QUANDO CLICA, O JAVASCRIPT FAZ A MUDANÇA NO display DO ELEMENTO PARA QUE ELE APAREÇA (display:block) OU DESAPAREÇA (display: none)

MÓDULO 4 – CAÍTULO 25 – AULA 40: MEDIA QUERIES PARA OUTROS DISPOSITIVOS

NESTA AULA CRIAMOS UMA MEDIA QUERY PARA TELAS COM TAMANHO MAIOR DO QUE CELULARES.

CSS

@media screen and (min-width: 768px) {

    #icon {

        display: none;

    }

    ul {

        display: flex;

    }

    li {

        margin-left: 10px;

        margin-bottom: 10px;

    }

    a {

        border-radius: 5px;

    }

    main {

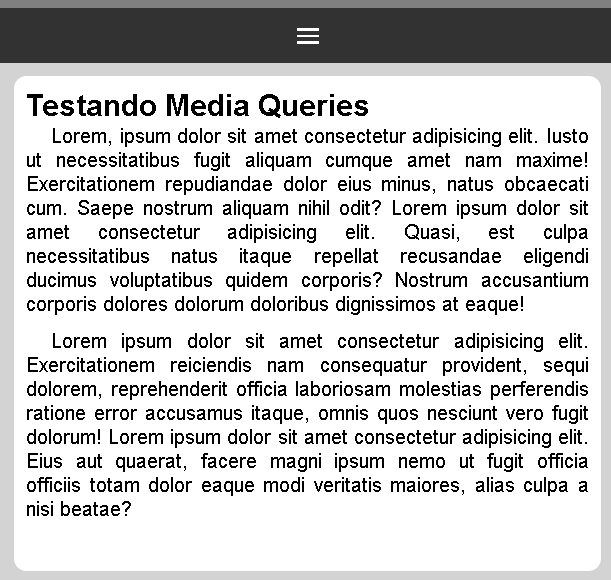
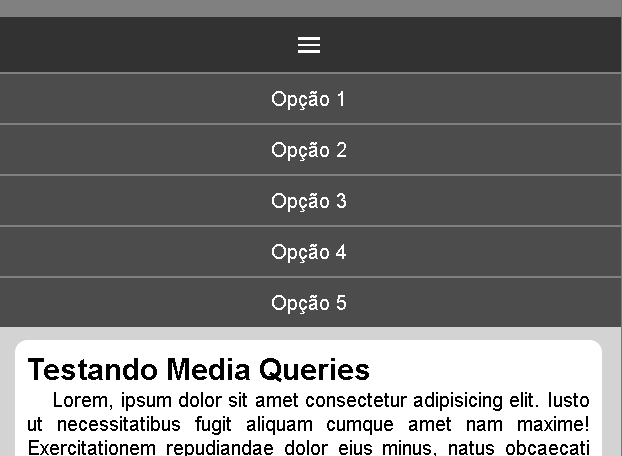
        width: 769px;

    }

}

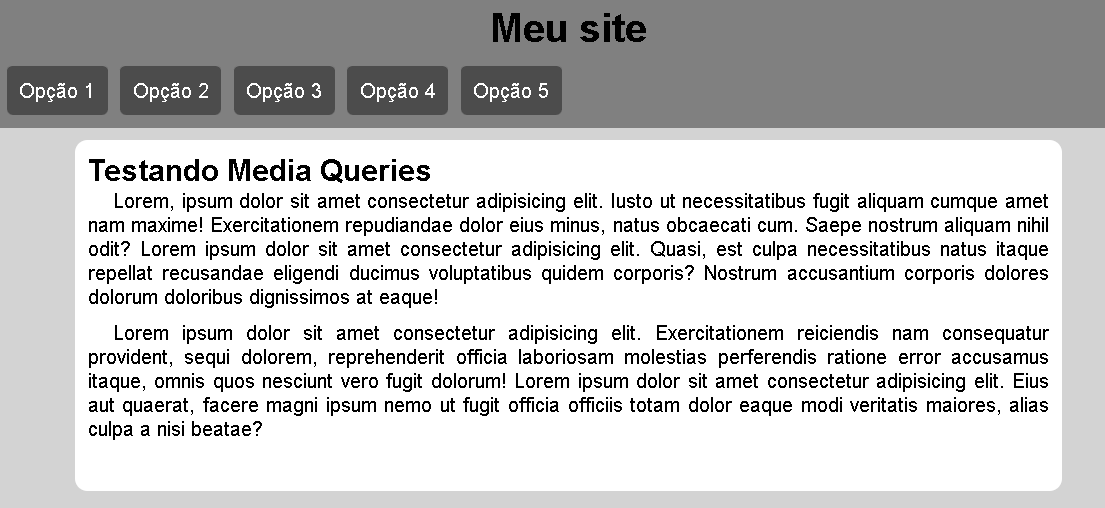
EU ESCREVI ESSE CÓDIGO CSS QUE FICOU UM POUCO DIFERENTE DAQUILO QUE O PROFESSOR FEZ, MAS O RESULTADO FOI O MESMO. NESSA CONFIGURAÇÃO FOI POSSÍVEL DIFERENCIAR O LAYOUT DAQUILO QUE FOI FEITO NA AULA ANTERIOR, OU SEJA, PARA TELAS ACIMA DE 768px FOI RETIRADO O MENU HAMBURGUER E APRESENTADO AS OPÇÕES EM TAMANHO NORMAL UMA AO LADO DA OUTRA.

MENU HAMBURGUER (PARA TELAS MENORES DO QUE 768px):

A PRIMEIRA IMAGEM SE REFERE AO MENU FECHADO E A SEGUNDA IMAGEM É O EFEITO QUE OCORRE QUANDO CLICAMOS EM CIMA DO ICONE MENU.

MENU PARA TELAS ACIMA DE 768 px:



NESSA AULA TAMBÉM FOI NECESSÁRIO CORRIGIR UM ERRO QUE ESTAVA OCORRENDO DENTRO DO SITE. QUANDO ABRIA O MENU HAMBURUGER, FECHAVA E AUMENTAVA A TELA, EM SEQUENCIA, O MENU COMO O DA IMAGEM ACIMA, NÃO APARECIA, DEVIA AO JAVASCRIPT, SENDO ASSIM, FOI NECESSÁRIO CRIAR UM NOVO SCRIPT PARA QUE O ERRO PARACE DE OCORRER.

<body onresize="mudouTamanho()">

 function mudouTamanho() {

            if (window.innerWidth >= 768) {

                itens.style.display = 'block'

            } else {

                itens.style.display = 'none'

            }

        }

ASSIM, FOI POSSÍVEL DEFINIR QUE PARA O TAMANHO DA TELA MAIOR QUE 768px, O DISPLAY DO ELEMENTO COM ID ITENS FOSSE IGUAL A BLOCK E, EM CASOS DE TELA MENOR QUE ISSO, O DISPLAY SERÁ NONE.