**Módulo 4: Aprofundando os conhecimentos**

Nesse 4° módulo aprenderemos sobre quadros em linha, formulários, media queries, mobile first.

**Capítulo 22 21:37 11/07/2022**

* Iframe = quadros em linha;
* O <iframe> são caixas inline que servem para colocação de links em um site. Porém ao invés de encaminhar para o site, ele mantém a página aberta dentro do site;
* A tag para inclusão do iframe é a seguinte:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

* Dentro do “src” colocamos o link desejado, e o “frameborder” já vem com o valor “0”, que retira a margem do iframe, mas caso queira manter a borda é só deixar “1”. Para personalizar a borda, basta ajustar nas CSS utilizando “border”, mas essa propriedade coloca borda em todos os iframes independentemente dos parâmetros;
* Alguns sites não aceitam que seu conteúdo seja colocado em um iframe por questão de proteção;
* O tamanho padrão do iframe é 300x150px;
* Normalmente os celulares não são compatíveis com o iframe. Para estes casos, dentro da tag do iframe podemos colocar o link do site ou uma frase, pois tudo que se escreve dentro dessa tag só aparece quando o iframe não for compatível com o navegador;

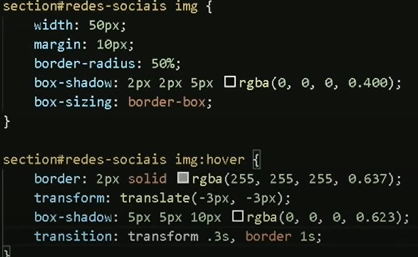
Texto

Descrição gerada automaticamente com confiança média

* Podemos ajustar a altura e largura do iframe no html, usando os parâmetros height ou width, ou então podemos ajustar nas CSS, lembrando que as configurações das CSS são priorizadas;
* Para retirar as barras de rolagem do iframe, podemos utilizar o parâmetro scolling. Para ele, temos algumas opções: auto (configuração padrão), yes (coloca barra de rolagem mesmo que o conteúdo caiba completamente no iframe) e no (retira a barra de rolagem, mas não funciona em todos os navegadores);
* Além de colocar links no iframe, também podemos colocar arquivos que estão dentro da mesma pasta do arquivo onde o iframe está localizado;
* Para carregar links de páginas que estão no site dentro do <iframe>, utilizamos o parâmetro “name=”””, colocando um nome dentro das aspas e no <a> colocamos “target=””” e o nome colocado dentro de “name”;
* Quando quisermos colocar informações dentro de um iframe que não sejam um link ou arquivo separado, no lugar do parâmetro “src=”, utilizamos “srcdoc=”””. Esse parâmetro permite inserir um código de html dentro do <iframe>, porém todas as tags precisam ser iniciadas uma do lado da outra e se alguma dessa tiver aspas, utilizaremos aspas simples ‘ para não ter conflitos com as aspas do “srcdoc=”””. É mais indicado fazer um arquivo separado, mas podemos utilizar esse formato para configurações estáticas simples;
* O uso do iframe não é tão recomendado para os mecanismos de buscas, pois eles têm dificuldade de indexar as informações que estão dentro dele. Então no caso de ter um conteúdo altamente indexável, não é ideal colocar em um iframe;
* Outro ponto importante é que para acessibilidade, o iframe dificulta os mecanismos de leituras de telas que pessoas cegas utilizam para utilizar os sites;
* Além dos pontos acima, ele também demonstra problemas de usabilidade, como por exemplo: se perde ao tentar voltar para tela principal do site do iframe, quebra o site todo no iframe quando o site que inserimos nele não for responsivo e no caso de ter alguma página dentro do site do iframe que utilize “target=”blank”, ele abre uma nova página por fora do nosso site ao invés de abrir dentro do iframe;
* Ao utilizar o iframe também é necessário a colocar sites que sejam seguros, pois se forem sites que foram invadidos ou que utilizem mecanismos de cópias de informações, abriremos portas para que esse site faça ataques em nosso site;
* Para deixar o iframe mais seguro, podemos utilizar 2 parâmetros dentro do iframe: sandbox que tem os atributos sandbox (totalmente seguro, impede do iframe pegar dados e é a configuração padrão) e alguns “allow” que liberam algumas permissões de acesso pontuais para o iframe, como por exemplo “allow-same-origin” que autoriza iframe que sejam da mesma origem do site. O outro é “referrerpolicy=no-referrer” que não permite que o iframe capture seus dados de navegação para lhe encaminhar anúncios;
* O iframe é muito útil, pois graças a ele podemos incorporar mapas, vídeos e documentos nos sites;

**Capítulo 23 21:18 18/07/2022**

* Para fazer com que a borda faça parte do objeto (o efeito coloca borda na imagem sem mexer nas imagens ao redor, como se fosse uma borda por dentro), utilizamos a função “box-sizing” e a propriedade “border-box”;
* Animação para botões ou imagens em geral:



* Para colocar um tempo para que algumas propriedades sejam aplicadas, podemos usar a propriedade “transition”, com o nome da propriedade que for alterar a velocidade da aplicação e o tempo, como no exemplo acima;
* “Display: block” preenche um espaço inteiro;
* “::-webkit-scrollbar” somados a “height: 0px” e “width: 0px” serve para retirar a barra de rolagem;

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Capítulo 24 21:50 20/07/2022**

* Todos os formulários precisam ser criados dentro da tag <form>;
* Para começar a criar o formulário, colocamos a tag <p>, por exemplo, escrevemos a informação que gostaríamos de receber e depois utilizamos a tag <input>. Quando digitamos esse input no VS aparecem várias sugestões. No caso de informações de texto, escolhemos “input:text”, para criar um botão de enviar, escolhemos “input:subimit” etc;

**Tela preta com letras brancas

Descrição gerada automaticamente**

* Para retirar o preenchimento automático dos campos dos formulários podemos utilizar o parâmetro “autocomplete=”off”” dentro de <form>;
* Para enviar os dados informados no formulário para um arquivo, primeiro colocaremos o parâmetro “action=”cadastro.php”” em <form>, por exemplo;
* É necessário classificar corretamente o name e o id nos inputs, pois o name é importante para a leitura do PHP e HTML e o ID é importante para as CSS e Java Script. Além disso, para a “label” funcionar corretamente, precisa ter o id identificado;
* Ao inserir a tag <label> ao redor do texto que precisa ser ligado ao bloco, aparecerá o parâmetro “for” “”, onde deverá ser colocado a informação do ID que está em input;
* O name e o ID podem ter os mesmos identificadores, porém para diferenciar e ficar mais fácil a visualização, o professor sugeriu colocar o i na frente de todo identificador referente ao ID;
* É importante inserir o label para criar uma relação entre a palavra e o objeto. Além disso, ao relacionar, caso o usuário clique na palavra, ela já o direciona para o objeto (a barra por exemplo). Isso ajuda bastante os usuários que acessam por celular por ampliar a área de sensibilidade;
* Ao inserir as informações dos formulários, os dados ficam aparecendo no link (1° vem a url e tudo que vem depois do ? são as informações enviadas pelo formulário). Isso ocorre porque é o método padrão dos formulários;
* Para ajustar essa questão temos o parâmetro “method=” ””, que tem os valores “get” que é o padrão dos formulário e “post” que retira os dados da URL;
* Quando utilizamos o método “post” os dados não aparecem na URL, mas isso não significa que não tenha como ter acesso essas informações e que a URL está segura, pois se abrirmos o dev tools na página, irmos na aba network, depois na sub aba headers, ao digitar as informações os dados serão capturados dentro dessa aba. Para a URL ficar segura, ela deverá ser criptografada, tendo o https;
* GET: deverá ser utilizado em caso de dados não sensíveis, arquivos compartilháveis e formulários com até 3000MB (3000 palavras em média), e não é possível enviar arquivos com ele;
* POST: deverá ser utilizado juntamente de https para dados sensíveis (senhas, cartão de crédito), arquivos maiores com mais de 3000MB e formulários com imagens;
* Input:password – para inserção de senhas;
* Input:reset – botão de limpar;
* Para tornar um campo obrigatório, utilizamos o parâmetro “required” dentro do <input>;
* Para colocar uma limitação de mínimo de caracteres, utilizamos dentro do <input> o parâmetro “minlength” e para o máximo “maxlength”
* Para ajustar o tamanho da caixa, podemos inserir o parâmetro “size” dentro do imput. Esse size ajusta o tamanho da barra para aparecer a quantidade de letras. Ou seja, se for colocado size=”10”, teoricamente vai mostrar 10 letras (sendo que ele erra um pouco a quantidade por não saber o tamanho de todas as letras;
* Para colocar mensagens de descrição nas caixas, podemos utilizar o parâmetro “placeholder”;
* Quando deixarmos o “autocomplete=”on”, é importante informar qual será a informação daquela caixa. Para isso, além de colocarmos um autocomplete em <form>, colocaremos um autocomplete no <input> também. Dessa forma, serão demonstradas diversas opções de tipo de informações das caixas para fazer a ligação;
* Autocomplete=”username” – nome do usuário;
* Autocomplete=”current-password” – senha atual;
* Ao inserir números, o type será “number”, e para delimitar o valor máximo e mínimo aceitos no campo, utiliza-se “min” ou “max”
* Para permitir números decimais nos campos, podemos utilizar a propriedade “step”, e classificar de quanto em quanto o valor decimal será aceito, como no exemplo abaixo:

**Texto

Descrição gerada automaticamente**

* Para inserção de datas, podemos utilizar o controle “month”;
* Para inserção de datas com dia, podemos utilizar o controle “date”;
* Para inserção de horário, podemos utilizar o controle “time”;
* Para deixar um valor de sugestão na label, podemos utilizar o parâmetro “value” e ao lado dele a sugestão desejada. Importante informar que o utilizar esse parâmetro, automaticamente o “placeholder” não aparecerá mais na caixa;
* Importante pontuar que os controles (inputs) não tem o mesmo funcionamento em todos os navegadores, podendo haver diferença nas visualizações;
* Para inserção de telefone, podemos utilizar o controle “tel;
* Para inserção de email, podemos utilizar o controle “email”;
* No caso do email, o campo só permite enviar quando tem um @, mas no caso do telefone não há nenhum bloqueio para informações invalidas, sendo assim podemos utilizar “Expressões regulares (RegEx)”, que devem ser estudas depois com mais calma;
* Para acrescentar uma RegEx, utilizamos o parâmetro “pattern” dentro da parte do input. Para definir uma quantidade mínima de dígitos para telefones e que tenham o DDD, podemos utilizar o seguinte atributo “^\(\d{2}\)\d{4,5}-\d{4}$”, da forma abaixo:

Texto

Descrição gerada automaticamente

* Os sinais ^$ indicam abertura de parênteses para inserção dos atributos;
* Os sinais \d significam dígitos e o {2} (ou qualquer outro número que esteja dentro) significa a quantidade de dígitos necessários para o telefone;
* Os () indicam os dígitos que devem ficar dentro dos parênteses, como o DDD;
* O – indica que deverá ter um traço entre os dígitos;
* Para agrupar formulários, podemos envelopar as informações que forem pertinentes ao mesmo assunto com a tag <fieldset> e acrescentar uma tag <legend> informando sobre o que são referentes aqueles campos;
* Para inserção de caixas para marcar, podemos utilizar o controle “checkbox”;
* O ID se conecta com o label;
* Para inserção de caixas para marcar uma opção ou outra, podemos utilizar o controle “radio” com “name” igual ao outro input para que eles permaneçam em grupos iguais e possa ser marcado uma opção por vez;
* Caso queira manter uma opção já marcada assim que acessem o site, podemos utilizar o parâmetro “cheked";
* No controle “radio” precisamos colocar o parâmetro “value” para levar a informação correta para o arquivo de cadastro;

Uma imagem contendo Texto

Descrição gerada automaticamente

* Para inserção de escolha de cores, podemos utilizar o controle “color”, incluindo “value” caso queira escolher a cor inicial da página (por padrão é preto);
* Para inserção de níveis (por exemplo nível de satisfação), podemos utilizar o controle “range”, incluindo alguns controles caso queira, como o “min” para escolher o valor mínimo da barra, “max” para escolher o valor máximo da barra e “value” para escolher de que ponto a barra deve aparecer no primeiro acesso ao site;
* Para inserir alguma foto no site, podemos utilizar o controle “file”, lembrando que sempre que o utilizar, o “method” precisa ser “post” por ser mais de 3000MB;
* Para acrescentar listas como opções de escolha, podemos utilizar o elemento <select>. Dentro de cada select, utilizamos um <option> para acrescentar as opções que aparecerão na lista e caso seja necessário agrupar essas informações, podemos utilizar a tag <optgroup> com um “label” de atributo para poder nomear os grupos;
* Para que uma opção apareça já como selecionada no site, podemos utilizar o atributo “selected” dentro do option desejado. Também é possível criar um option sem “value” e colocar esse elemento dentro apenas para que ao acessar o site não tenha nenhuma opção selecionada;

Texto

Descrição gerada automaticamente

* Para criar uma lista de opções e ainda poder digitar alguma informação diferente das escolhas sugeridas, utilizamos o controle “input:text” com o atributo “list” dentro dele. Depois disso, utilizamos o elemento <datalist> para descrever as opções que serão sugeridas. Lembrando que o “id” em <datalist> deve ter o mesmo nome que será indicado em “list”;

Texto

Descrição gerada automaticamente

* No exemplo acima o <option> ficou sem valor porque quando colocado nesse caso, as abreviações são levadas para a sugestão;
* Para inserir campos de mensagem no site, utilizamos a tag <textarea>. Ela traz vários atributos, mas os mais diferentes são “cols” e “rows” onde deveremos informar a quantidade mínima de linhas e colunas que aparecerão no primeiro momento de acesso ao site. Essa caixa de mensagem é moldável pelo usuário no site, podendo ficar maior ou menor;
* Os atributos citados acima não são delimitadores, pois a quantidade de texto nessa tag é infinito. Por este motivo, sempre que essa tag for inserida em um formulário deverá ser com “method=”post””;

Texto

Descrição gerada automaticamente

* O controle <output> serve para inserir comandos de programação;
* Um exemplo do que pode ser feito, é somar valores e demonstrá-los em tela apenas, pois essas somas não são levadas para os formulários, apenas aparecem em tela;

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* Output: nele serão colocados o name, o id e um valor;
* Oniput: nele serão colocados os comandos;
* Isoma se referência ao <output>;
* O innerHTML quer dizer que deverá ser puxado o valor dentro do HTML;
* Number levará em consideração o número inserido;
* (in1.value) traz o campo que deverá ser somado;
* Outro exemplo que se pode usar <output>:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* No exemplo acima foi utilizada uma função em JavaScript, onde foi criada uma função “calcIdade()”;
* O primeiro passo é abrir uma function calcIdade(), depois abrir chaves e iniciar com let atual = new Date() para trazer o ano atual e colocar .getFullYear() para trazer o ano inteiro. No segundo passo, informaremos que a idade trazida dentro do HTML iidade.innerHTML será o ano atual Number(atual) menos o ano inserido Number(iano.value);

**Capítulo 25 21:22 05/09/2022**

* Media Queries é uma tecnologia que trabalha com versões de estilos adaptáveis para diversas telas as tornando expansivas;
* Para criar arquivos Media Queries, deve-se criar vários <link:css> para cada tipo de visualização e adicionar ao final da tag o parâmetro “media”, sendo os valores dentro dele all (configuração geral), screen (configurações para telas) e print (configurações para impressões) referentes a media types;
* O ideal é fazer uma pasta para estilos para cada HTML;
* Media Queries é basicamente a soma de Media Type + Media Features, que são tipos de mídia mais características de mídias;
* Quando formos utilizar as Media Features, colocamos da seguinte forma: media=”screen and ()”, por exemplo;

Texto

Descrição gerada automaticamente

* Orientation: orientação da tela;
* Portrait: orientação para modo retrato;
* Landscape: orientação para modo paisagem;
* Para fazer as configurações das media queries, podemos fazer um arquivo de configurações gerais (all) e outro de configurações específicas, para não ficar repetitivo;
* É possível fazer a configuração de todos estilos dentro do próprio arquivo html, abrindo um seletor {style} e dentro dele colocando as 3 configurações, sendo necessário adicionar a regra @media screen and (orientation: ) { }, colocando as configurações dentro do seletor de media;

Texto

Descrição gerada automaticamente

* Também é possível fazer essa configuração com apenas 1 arquivo de estilo. Nesse formato as configurações serão iguais as de cima, só que dentro de um arquivo de estilo e não é necessário colocar o tipo de media no link de estilos;

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* É recomendável utilizar essas configurações apenas em projetos pequenos;
* Mobile First é uma abordagem que propõe ser feita primeiro a versão móvel do site (retirando somente o que poderia deixar o site móvel pesado) e depois ir reposicionando o site para monitores maiores;
* As vantagens de fazer um site com essa técnica são: maior divulgação do site no google, já que eles priorizam site que são feitos com essa técnica; melhoria na experiência do usuário já que o site será leve para celulares; maior credibilidade no site;
* Typical Device Breakpoints são medidas de resoluções para cada tela. Eles mudam de tempos em tempos, mas atualmente os recomendados pela W3C são:

Texto

Descrição gerada automaticamente

* Quando já tivermos feito o modelo do site baseado no mobile first, não precisamos fazer media queries para telas pequenas ou celulares, pois o primeiro modelo já é dedicado a eles;
* SEMPRE colocar a media query dentro dos parênteses;
* Mais um exemplo de media queries em um mesmo arquivo de estilo:

Texto

Descrição gerada automaticamente

* Para o modo impressão, não adianta colocar uma imagem de fundo (background), pois não imprime. Então as configurações de estilos serão mais pautadas em fontes, espaçamentos, bordas...;