**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 300x150 px;
* 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;

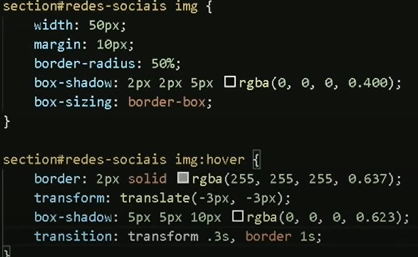
Texto

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 dendro 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 aprecem 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;