**Aula 1 – O iframe ainda pode ser usado? – Capítulo 22**

Ex024 – iframe001.html

Iframe = frame inline (quadros em linha)

Existe alguns sites que não funcionam no iframe.

**Aula 2 – Configurando iframes – Capítulo 22**

Ex024 – iframe001.html

Nem todo navegador de celular é compatível com iframe, então, ainda dentro dessa tag, abra uma tag <a> e direcione para o site que você deseja.

Por padrão ele é **inline-block**, para usar **margin auto** 🡪 use **display: block;**

O tamanho padrão do iframe é de 300x150 pixels

No CSS do iframe você pode configurar um **height** e **widht** específicos. (**CSS > config das Tag**)

Dentro da tag iframe você pode não permitir o scroll com o parâmetro **scrolling=”no”;**

**Aula 4 – Navegação no iframe – Capítulo 22**

Ex024 – iframe003.html

Para abrir uma página dentro do frame, dê um **name=””** para o **<iframe>** e esse name será usado no **target=“nome que você colocou”** do link **(tag <a>)**

**Aula 5 – Conteúdo no iframe por código – Capítulo 22**

Ex024 – iframe003.html

No **<iframe>** ele apaga o parâmetro **src=””** e adiciona o **srcdoc=””;** dentro da “” ele abre tags h1 e p

Também é possível colocar img.

**Aula 6 – Inconvenientes do iframe– Capítulo 22**

Iframes tem problemas de acessibilidade 🡪 Seu conteúdo não é lido pelos leitores de tela para cegos

Iframes tem problemas de usabilidade

**Aula 7 – Tornando iframes mais seguros – Capítulo 22**

Ex024 – pagina004.html, pagina005.html, iframe004.html e iframe005.html

Use na tag iframe os parâmetros **sandbox=”allow-same-origin”;** **referrerpolicy=”no-referrer”;** Isso irá evitar que outros sites peguem dados de algum formulário ou cadastro do seu site OU executar scripts dentro do seu site.

Ainda dentro do **sandbox=””,** dentro das aspas você deve colocar **allow-forms e allow-scripts.**

**Aula 8 – Dicas para iframes melhores – Capítulo 22**

Ex024 – iframe006.html

Você consegue incorporar uma localização no google maps, abra ela no site, vá em compartilhar e use a incorporação.

Para arquivos no google docs (textos, ppt ou excel), abra ele 🡪 arquivo 🡪 compartilhar 🡪 publicar na web 🡪 Incorporar

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

**Aula 3 – Criando a página inicial do projeto – Capítulo 23**

Desafio 13

**Main** ficou com **position:relative** e a **section#telefone** com **position: absolute;**

Use no **section#telefone{top: 50%; left: 50%;**

**transform: translate(-50%, -50%);** 🡪 Para centralizar

**width: 311px;** 🡪 Que é a largura da imagem do celular (veja nas propriedades)

**heigh: 627px;** 🡪 Que é a altura da imagem do celular (veja nas propriedades)

**background-size: contain;** 🡪 Para aparecer todo o telefone

**Aula 4 – Posicionando o conteúdo do site – Capítulo 23**

Cara tela que aparece dentro do celular é um html diferente

**IMPORTANTE: O CSS NÃO VAI PARA ESSAS TELAS DIFERENTES, ENTÃO ABRA A TAG STYLE E FAÇA \*{padding: 0px; border: 0px; em cada HTML**

O **body** de todos deve ter {

**Background-color: black url (‘endereço da imagem’) no-repeat** e o alinhamento da imagem;

Ele cria um iframe dentro da section#tela, nele ele adiciona src=”home.html” name=”tela” id=”tela”

No css:

**iframe#tela {position: relative;**

**TODOS OS VALORES ABAIXO ELE FOI CHUTANDO ATÉ ENCAIXAR**

**top: 80px;**

**left: 22px;**

**width: 269px;**

**height: 471px;**

Depois ele voltou no **home.html** e no CSS do body ele colocou **background-size: cover;**

**Aula 5 – Criando os botões para redes sociais – Capítulo 23**

Transformando imagens quadradas em ícones redondos

Os links da seção das redes sociais ficaram com **target: “tela”;** 🡪 name dado ao bg do cel

Ele usou **border-radius: 50%;** 🡪 Transformou img quadrada em redonda.

Ele usou br após todo link de redes sociais e aí 1 ícone ficou abaixo do outro.

**section#redes-sociais {text-align: right;** 🡪 Jogou todos os ícones para a direita.

**Section#redes-sociais img:hover {border: 2px solid White;** 🡪 Animação ao passar o mouse nos ícones

Para a borda fazer parte do objeto (ícones) ele fez:

**Section#redes-sociais img {box-sizing: border-box;** 🡪 Aí a borda faz parte do objeto e não fica esquisito ao aparecer a borda branca quando você passa o mouse em cima da imagem

Animação ao passar o mouse nas imagens:

**section#redes-sociais img:hover {border: 2px solid rgba(255, 255, 255, 0.623);**

**transform: translate(-3px, -3px);**

**box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.63);**

**transition: transform 0.3s;**

**Aula 6 – Criando as páginas de cada rede social – Capítulo 23**

Ele criou o youtube.html

No body ele fez uma **<img>** com a tela do youtube e em baixo uma **<a>** levando pro link das playlists do curso em vídeo.

Para que o Acesse (do link) apareça embaixo da imagem ele fez na **<style>:**

**a {display: block;**

Para esconder a barra de scroll do youtube.html, faça na **<style>:**

**::-webkit-scrollbar {**

**width: 0px;**

**height: 0px;**

**Ele criou um CSS diferente do padrão que será como base para todo html de cada rede social.**

**Aula 7 – Personalizando as imagens do projeto – Capítulo 23**

Ele acessa cada rede social e tira um print, aí ele desce até o final do print anterior e tira outro. Isso gera vários prints de cada rede social.

Total de 4 prints por rede social

No GIMP:

Ele seleciona os 4 prints e arrasta pra lá

Menu superior 🡪 Imagem 🡪 Tamanho da tela de pintura 🡪 Altura 10.000 (X e Y 0) 🡪 Redimensionar

Ele ajusta o posicionamento da primeira imagem 🡪 vai na segunda imagem 🡪 nas opções da esquerda escolha seleção retangular 🡪 Selecione a parte da 2 imagem que você quer, copia e cola na primeira imagem comprida. 🡪 mova a segunda imagem para alinhar com a primeira.

Após finalizar a imagem total:

La embaixo ele deixa 12,5% para enxergar toda a imagem, selecione a ferramenta de crop (última da primeira linha) 🡪 marque só a parte que está com foto e dê enter.

Depois:

Menu superior 🡪 Imagem 🡪 Redimensionar imagem 🡪 600 largura e redimensione.

Menu superior 🡪 Arquivo 🡪 Exportar como 🡪 selecione os tipos de arquivos e escolha jpg 🡪 ele salvou como instagram.jpg 🡪 Exporte

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

**Aula 1 – Criando formulários com HTML5 – Capítulo 24**

Ex025 – form001.html

Todo formulário deve estar na tag **<form>**

Para criar uma caixinha de texto onde a pessoa irá digitar use **<input:text> (toda tag dessa tem name e id.**

No fim do formulário e dentro de uma nova <p>:

**<input type=”submit” value=”enviar”>**

**Aula 2 – Usar Label vai melhorar seus formulários – Capítulo 24**

Ex025 – form001.html

As info dos formulários serão enviadas para alguma linguagem de programação

Dentro do <form> ele adiciona action=”cadastro.php”

**Diferença entre label e id (presente em todo input):**

Name 🡪 Isso é mais útil para HTML e PHP

ID 🡪 Mais usado para Javascript.

Ele seleciona o “nome” dentro do <p> e envelopa com a tag **<label>**, dentro dela tem um **for=””,** coloque entre as aspas o **ID do input**.

Todo LABEL tem o seu INPUT

**Aula 3 – Métodos GET e POST para formulários – Capítulo 24**

Ex025 – form001.html

Você precisa usar o método para evitar que as informações fiquem na url que é carregada após clicar no botão submit

GET é o método padrão que é usado mesmo sem você configurar o form.

Use o parâmetro **autocomplete=”OFF”** na tag **<form>** para desativar as sugestões.

Use o parâmetro **method=””** na tag **<form>**

Use o **post** quando for o formulário enviar dados sensíveis que precisam de segurança

GET 🡪 Use quando for dados não sensíveis, for algo em que a URL pode ser compartilhada. PS: Tamanho máximo de 3.000 bytes.

Para proteger 100% os dados é o método https

**Aula 4 – Criando caixas de texto e de senha – Capítulo 24**

Ex025 – form002.html

Use **input:password** quando pedir senha

Use o **input:reset** logo após o sumit para a pessoa poder limpar oq foi digitado.

**Input:number** 🡪 Apenas para números

Quando for um **input** obrigatório 🡪 Adicione o parâmetro **required**

Caso a senha seja de no mín 5 digitos use **required minlenght=”5”**

Se houver um máximo de caracteres use também **maxlenght=””**

Você também pode adicionar o parâmetro **size=””** e oq vier dentro das aspas é a quantidade de caracteres que poderá aparecer

Você também pode adicionar o parâmetro **placeholder=””** e oq vier dentro das aspas é o texto que irá aparecer DENTRO da caixa, sem a pessoa ter digitado nada.

Você também pode adicionar o parâmetro **autocomplete=”” 🡪** Isso dirá para o seu navegador do que se trata as informações que serão inseridas (no **<form>** o **autocomplete=”ON”** é obrigatório)

**“Current-password”** 🡪 Quando for pra usar uma senha que já foi cadastrada.

**“New-password”** 🡪 Quando for para cadastrar nova senha.

**Aula 5 – Elementos number, month, date e time em formulários HTML – Capítulo 24**

Ex025 – form003.html

O parâmetro **min=””** e **max=””** dentro do **<input>** dirá qual o valor mínimo e máximo que será aceito.

Ele adicionou o parâmetro **step=”0.5”** no **<input>** da média para dizer que a nota vai mudando a cada 0.1

**Input: Month** 🡪 para o período letivo, ele usou o parâmetro **value="2020-07"** para já aparecer marcado o mês de julho de 2020

Quando for inserir o value, siga a ordem 🡪 Ano-Mês-Dia

**Input: date** 🡪 Para inserir dia, mês e ano.

**Input: time** 🡪 Para inserir a hora

**Aula 6 – Compatibilidade com navegadores – Capítulo 24**

Em outros navegadores o visual dos inputs do formulário pode ser diferente e muitos parâmetros (como o month) pode dar quebrado.

**Aula 7 – Formulários com telefone e e-mail – Capítulo 24**

Ex025 – form004.html

Estude sobre **RegEx** 🡪 Expressões regulares

Limitação do input email 🡪 Basta ter @, um email inválido acaba sendo aceito

Limitação do input tel 🡪 Ao digitar qualquer número será aceito

Use no **<input>** o parâmetro **pattern”^$”**

O pattern sempre começa com ^ e termina com $

No exemplo: **pattern=”^[0-9]{4,5}-[0-9]{4}$”**

[0-9] 🡪 Números de 0 a 9

{4,5} 🡪 De 4 a 5 dígitos

{4} 🡪 Total de 4 dígitos

**Coloque um placeholder no telefone para facilitar o usuário**

Ele envelopou todos os dados com a tag **<fieldset>** e embaixo abriu **<legend>** explicando sobre oq seria esse grupo de dados.

**Aula 8 – Checkbox e radio button – Capítulo 24**

Ex025 – form005.html

**Input: Checkbox** 🡪 Para ficar de marcar; não esqueça do **<label>**, pode ter **vários inputs marcados.**

**Input: Radio** 🡪 O **name** dos inputs deve ser **igual** para permitir a desmarcação! Use radio para permitir apenas uma marcação e não esqueça dos bizu do name igual. **O ID continua necessitando ser diferente.**

Para já aparecer marcado, adicione o parâmetro **checked** no input.

**Aula 9 – Elementos color, range e file em HTML – Capítulo 24**

Ex025 – form006.html

**Input: color** (cuidado com navegador) 🡪 Seleciona a cor, aqui você pode adicionar o parâmetro value=”” e dentro das aspas o # código hexadecimal da cor que virá como padrão.

**Input: range** 🡪 Seleciona um ponto dentro de um range, você pode usar min e max.

**Input: file** 🡪 Para que o usuário faça upload de algum arquivo. AQUI O FORM DEVE SER **POST e cuidado para não passar o limite de 3.000 bytes**

**Aula 9 – Select, datalist e textarea em HTML5 – Capítulo 24**

Ex025 – form007.html

Elementos fora da tag <input>

Para a pessoa selecionar 1 item dentro de uma caixa que desce várias opções:

Abra a tag **<select>** e dentro dela faça as tags **<options>** para cada elemento que aparecerá

Você pode envelopar várias **<options>** com a tag **<optgroup>** e dar um **label=””** para ela.

Se não há a opção que o usuário deseja nas sugestões, use o **<datalist>**:

Ele abriu uma label e um <**input:text** com o parâmetro **list=”lstprof”**

Esse parâmetro também foi para **<datalist** **id=” lstprof”>,** dentro dela ele fez várias **<options>** e depois **</datalist>**

Para adicionar um input que o usuário queira escrever uma mensagem**, IMPORTANTE O FORM SER POST.** faça:

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

**<textarea** name="msg" id="imsg" cols="30" rows="10"></textarea>

**Aula 11 – Elemento output – Capítulo 24**

Ex025 – form008.html e form009 e form010

Ele usa **<output e adiciona um ID=””>**

No input ele adiciona o parâmetro:

**Oninput= “id do output.innerHTML = Numer(id do input.value)”**

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

**Aula 1 – O que são Media Queries – Capítulo 25**

Usado para exibir o conteúdo do site em diferentes tamanhos e proporções de telas

O media querry gera uma versão para telas e outras para impressoras, nessa última não tem links, banners e nem rodapé, é focada apenas na matéria em si.

O media querry também é o menu enxuto e rodapé enxuto no celular.

Diferentes tamanhos geram diferentes formatos do site.

**Aula 2 – Criando um site com versão para impressora – Capítulo 25**

Ex026 – pasta mq001

**Media types**

Ele criou um link:css e um **parâmetro media=”screen”**

Ele criou outro um link:css e um **parâmetro media=”print”**

**Também existe a media=”all”**

Ele cria CSS diferentes com media diferentes para cada formato.

Para o menu não abrir na impressão, ele vai no CSS print e faz:

**menu {display: none;**

**Aula 3 – Múltiplas Media Features com CSS – Capítulo 25**

Ex026 – pasta mq002

**Media Querry = Media types + media features**

Media Features: Engloba os diferentes tamanhos de telas (características de tela)

Media features são escritas dentro da media do CSS e após o screen and **()**

Ex:

media="screen **and (orientation: landscape)"**

**Aula 4 – Seguindo a orientação do dispositivo – Capítulo 25**

Ex026 – pasta mq002

As configurações gerais (all) ficaram pra tela em pé e deitada.

As configurações do backgroud ele fez no body geral e nos específicos ele só fez o **backgroud-image= url.**

A tela paisagem ficava com a imagem flutuando, então ele fez no CSS paisagem:

**background-position: left bottom;**

Ele também fez bg position no retrato, mas fez **center bottom.**

Então, no CSS específico do formato você só faz oq for diferente pra aquele formato.