**Programa DIO**

**Formação HTML Web Developer**

**HTML II – Ampliando o Conhecimento**

HTML: Recursos Especiais

**Tag <datalist>**

Parecido com o select box, mas possui vantagem na **“busca sugerida”** de uma informação a partir de uma lista pré-determinada.

Obs. datalist abrira uma lista de sugestão de escolha ou busca, enquanto o select box são valores fixos.

Exemplo do select box:

<body>

<**select** name="browsers">

<option value="">Selecione o navegador</option>

<option value="chrome">Google Chrome</option>

<option value="firefox">Mozila Firefox</option>

<option value="edge">Microsoft Edge</option>

<option value="safari">Safari</option>

<option value="opera">Opera</option>

</select>

</body>

Exemplo do datalist:

<input **list="browsers**" type="search" name="searchh"/>

<datalist **id="browsers**">

<option>Google Chrome</option>

<option>Mozila Firefox</option>

<option>Microsoft Edge</option>

<option>Safari</option>

<option>Opera</option>

</datalist>

Obs. o atributo list nesse caso é necessário, pois, será informado ao input onde procurar essa lista, e, portanto, precisa de um id no datalist.

Obs. o datalist é um pouco limitado para o CSS

**Tag <code>, <kbd>, <pre>**

Existem tags específicas para mostrar no HTML comandos de teclado, ou seja, a **tag <kbd> (keyborad)**.

Assim, kbd fala que determinada parte do texto são teclas do teclado.

Obs. pode-se estilizar ele via CSS.

Essas tags pré-formatam texto no formato mono space.

Para **a tag code**, envolve parte do texto em mono space e através do CSS pode-se trocar a formatação.

Para a **tag pre**, permine pegar várias linhas e colocar no formato mono space.

Obs. code e o pre não há muita diferença de formatação. Mas semanticamente, fica melhor assim:

<p> Para copiar esta informação, pressione <kbd>CTRL</kbd> + <kbd>C</kbd></p>

<p>Para trocar o fundo de um elemento, utilize o comando <code>background-color</code></p>

<p>Comando completo para alterar cor de fundo:</p>

<pre>

code {

background-color: #f1f1f1;

color: red;

}

</pre>

<pre>

<code>

code {

background-color: #f1f1f1;

color: red;

}

</code>

</pre>

Fontes mono space: todos os caracteres ocupam a mesma largura.

Se colocar uma fonte serifada, os caracteres não ficaram com a mesma largura.

**Tag <details> e <summary>**

Botão de expandido e recolhido com a tag “**details**”. Ela irá falar basicamente sobre detalhes. Então, por exemplo, pode-se colocar tag “**p**” e fazer uma descrição longo se quiser, ou tendo uma pergunta e uma resposta.

Obs. por padrão ele trás um **“Saiba mais”** com uma seta e quando se clica na seta aparecerá o conteúdo que está descrito entre a tag “**details**”.

Obs. e quando quiser mudar o padrão “Saiba mais”, utiliza-se a tag “**summary**” dentro da tag details.

Esta tag details é própria do HTML, ou seja, não precisa de nenhum outro recurso para fazer essa funcionalidade. Agora se quiser deixar ele mais apresentável poderá utilizar o CSS.

Obs. sempre que for utilizar alguma tag específica, se atentar se os navegadores aceitam esse tipo de tag.

**Tag <meter> e <progress>**

Tag **progress**, semântica dele é para definir o progresso de uma ação.

Ex. quando se clica em algum lugar e aparece uma barrinha de carregamento, esse carregando é um progresso, ou seja, começo meio e fim.

Obs. pode-se também mudar o comportamento do progress, colocando atributos:

Ex. <progress max=”100” min=”0” value=”50” > </progress>

Obs. E com JavaScript, posso passar uma lógica ao qual a cada intervalo de tempo a barrinha de carregamento seja completada.

Portanto, a finalidade do progress é definir um progresso, alguma coisa está carregando e terminando.

A tag **metter** é um medidor, então se quiser saber quanto de espaço tem no disco, ele vai me dar um valor. Ou seja, não há um progresso e sim um valor.

Então o intuito do metter são valores que podem aumentar ou diminuir.

Obs. também possui mesmos atributos “max. Min, value”.

Ex. metter pode ser utilizado para mostrar a temperatura.

**Tag <mark>**

O JS manipula o HTML, mas quando o HTML já está no browser e renderiza ele passa a se chamar de DOM, então quando for manipular o DOM, significa que estará manipulando dinamicamente o HTML que já foi renderizado pelo browser.

Nas páginas web, quando for buscar por alguma palavra, automaticamente coloca-se um marca texto nas letras digitadas. Esse evento se faz automaticamente, e o programa com JS.

**Tag <canvas>**

É uma tag onde se permite manipular gráficos dentro dela. Então se delimita um espaço do navegador e dentro desse espaço tem-se a liberdade de criar através do JavaScript, gráficos.

Assim, é uma tag especifica para criar animações.

Ref.

https://www.w3schools.com/html/html5\_canvas.asp

**Novos Atributos do HMTL 5**

Link

https://www.w3.org/TR/html5-diff/

Existe um atributo que se chama “**autofocus**” que é quando carregar a página ele dará um foco naquela tag. Por exemplo, se for um formulário, poderá deixar com foco em um campo.

O “**placeholder**” onde se coloca uma descrição dentro da caixa do formulário. Atributo da tag input. Mais para colocar uma explicação sobre o que irá dentro daquele determinado campo.

É possível fazer com que determinados campos sejam obrigatórios no próprio HTML. Com atributo “**required**”. Somente colocar ele dentro da tag input. E na página não irá mudar, mas quando clicar em enviar formulário, ele irá aparecer uma mensagem (formatada de acordo com o browser) pedir para que se preencha o campo do formulário que estiver sido posto o atributo required.

No campo tag fieldset é onde se delimita as partes do formulário, e ao colocar atributo “**disable**” na tag fieldset, é possível desabilitar o preenchimento do campo daquele fieldset.

Obs. também pode-se colocar o atributo disable em um campo específico, por exemplo dentro de uma tag input.

Obs. a ideia do disable na tag fieldset é para que se evite colocar vários atributos disable dentro dos inputs, simplificando o código colocando dentro de uma tag só o fieldset.

Na tag textarea, limitar número de caracteres para serem inseridos através do atributo **“maxlenght=”200”**”. Onde quando for atingido o número de caractere especificado irá para de digitar na caixa. Como também determinar o mínimo de caracteres digitados pelo atributo **“minlenght=”10”**”, assim, ao clicar em enviar aparecerá mensagem informando que o texto precisa ter no mínimo x caracteres.

Obs. O maxlenght, é exclusivo da tag textarea.

Obs. O minlenght pode-se ser utilizados em todos os inputs também além do textarea.

Tudo que estiver dentro da tag **form**, será enviado na requisição, ou seja, enviado para o servidor. No HTML 5 existe a possiblidade de não ter mais esse processo. Para isso, pode-se colocar dentro dos inputs o atributo **form=”form”** somente os inputs que queira ser enviado, não esquecer também de colocar esse atributo dentro dos botões (enviar, limpar, por exemplo). Para isso a tag form precisa ter um “**id**”.

Obs. verificar essas funcionalidades nos diferentes navegadores para ver se funciona.

Com o atributo “**contenteditable**” é possível usuário alterar diferentes lugares do HTML diretamente na página do usuário. Assim, pode-se criar navegadores de páginas dinâmicos.