**Conceitos básicos do HTML (3/2/23)**

* Client Side são programas que rodam na máquina do cliente (HTML, CSS, JavaScript e XML);
* Server Side são os programas que rodam no servidor (linguagens como PHP, ASPX, JSP, Banco de dados) e através dessas linguagens são gerados arquivos que somente o computador pode ler;
* HTML – linguagem de marcação de hipertexto. Tem como função distribuir informação globalmente;
* Ferramentas gratuitas para desenvolvimento web: Microsoft Expression Web 4, Brackets, Bloco de notas (nesse caso salvar o arquivo com a extensão HTML ou HTM), NetBeans ou Notpad++ (Visual Studio Code também é gratuito e muito melhor);
* O W3C é o órgão responsável por desenvolver padrões para a Web;
* Tag = marcas;
* Um elemento é formado por tag, atributos, valores e filhos;
* Os marcadores de HTML não são case sensitive, ou seja, escrevendo com maiúsculas ou minúsculas, ele vai entender e aplicar o solicitado;
* <meta> - define propriedades da página, como codificação dos caracteres, descrição da página, autor etc. O atributo name e content são para dar informações pertinentes a página. Essas informações são utilizadas pelos mecanismos de buscas para retornar o resultado da pesquisa;
* <script> - local onde é definido a programação de funções com scripts (JavaScript por exemplo);
* Motores de renderização é um mecanismo utilizado pelos navegadores para ler as informações de estilo da Web. Os principais são: WebKit (mais compatível com HTML5. Safari e Chorme tem essa tecnologia), Gecko (Firefox, sigla Moz) e Trident (Internet Explorer, sigla MS);
* Opera, sigla O nas CSS;

Texto

Descrição gerada automaticamente

* <button id=”btnBotal”> </button>: comando para fazer um botão;
* Atividades e exemplos de HTML: [Exemplos de HTML (w3schools.com)](https://www.w3schools.com/html/html_examples.asp);
* É importante fechar as tags corretamente para facilitar a renderização do arquivo;
* O atributo title define alguma informação extra sobre um elemento;
* Lista de atributos: <https://www.w3schools.com/tags/ref_attributes.asp>
* Lista de tags: <https://www.w3schools.com/tags/default.asp>;

**Introdução HTML5 e Principais Tags de Estrutura (6/2/23)**

* O HTML serve para estruturar textos, mas precisa das CSS para formatação do estilo e do Java Script para dar interatividade;
* Existem tags que não precisam ser fechadas pelo </ >, apenas abertas, como é o caso do <br>;
* Elementos que podem ter conteúdo, necessitam de abertura e fechamento, já as tags (elementos) que não tenham conteúdo (como imagens e linha horizontal) não necessitam de fechamento;
* Para formação dos arquivos, há algumas estruturas a serem seguidas: criar documentos bem formados, tags e atributos sempre com letras minúsculas, usar tags de fechamento, elementos vazios (sem conteúdo) podem ser fechados com “/”, os valores dos atributos devem ser descritos dentro de aspas e por fim todos os atributos precisam ter nome e valor associados;
* No HTML existem alguns caracteres especiais que se digitados o navegador o entende como um comando e não parte do texto. Exemplo deles são: “, ‘, &, <, >. Para colocar esses caracteres no texto é necessário utilizar os códigos abaixo:

Tabela

Descrição gerada automaticamente

* Em HTML, mesmo dando vários espaços em um texto, vai contar apenas como um espaço na demonstração da frase no browser;

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamenteTexto

Descrição gerada automaticamente

* O enter no HTML serve apenas para organizar o documento, pois para a demonstração no browser, não faz diferença;
* Inline – formatação rápida feita para um bloco específico dentro do próprio código;
* Algumas das tags não utilizadas após o uso das CSS: <b>Negrito</b>, <i>Italico</i>, <u>sunlinhado</u> e <hr>quebra temática de linha Linha 1<hr/>Linha2;

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* É possível criar listas ordenadas (ordem numérica, alfabética etc) ou listas não ordenadas (marcadores diversos, como imagens, bolinhas etc). As ordenadas são representadas por <ol></ol> e as não ordenadas por <ul></ul>. Os itens dentro dessas listas serão colocados dentro da tag <li></li>;

Diagrama

Descrição gerada automaticamente com confiança baixa

* Para criar tabelas, utilizamos as seguintes tags: <table></table>, <tr></tr> e <td></td>;

Texto, Carta

Descrição gerada automaticamente

* Referências absolutas: é aquela que inclui todo o caminho de um arquivo, inclusive o protocolo de comunicação (https). São mais utilizadas no caso de apontar arquivos que não estão dentro da mesma máquina da página construída;



* Referências relativas: é aquela utilizada para quando o arquivo se encontra na mesma máquina da página construída;



* Imagens suportadas nas páginas web: Gif/Gif animado, JPG e PNG;
* Para acessar um arquivo que esteja dentro da mesma página principal do projeto podemos utilizar <a>;



* Nunca criar links com o caminho físico do arquivo (por exemplo C:, sempre pular direto pro caminho do arquivo), pois quando disponibilizamos na Web ou em outros servidores, vai ocorrer o erro “404 – page not found”, que é arquivo não encontrado. Prefira sempre apontar para arquivos nas nuvens;
* Elemento de semântica são aqueles que tem algum significado e existem dois tipos: elementos inline (em linha) e elementos block (em bloco). Apesar de terem significados diferentes, sua apresentação será igual;

**Inline:**

Texto

Descrição gerada automaticamente

**Block:**

Texto

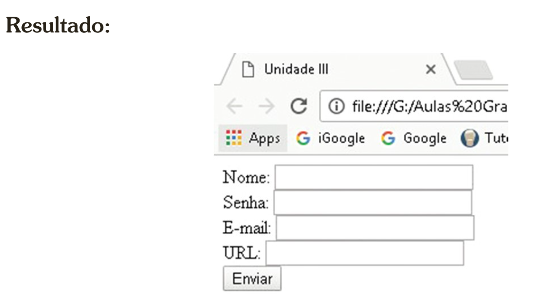
Descrição gerada automaticamente

* A ordenação dos blocos acima pode ser feita da forma desejada;

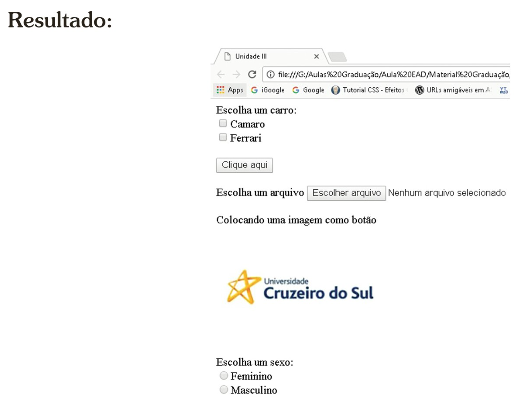
**Formulários e Vídeos**

* Os formulários servem basicamente para coletar dados do usuário para realização de algum procedimento no servidor;
* Podemos criar algumas validações de informações nos formulários com o HTML5, não sendo necessário o uso de JavaScript em todos os casos;
* Para criar um formulário em HTML, utilizamos a tag <form></form>;

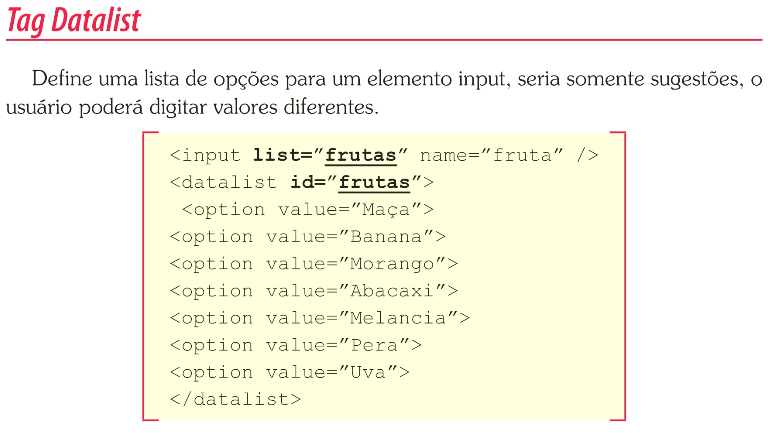




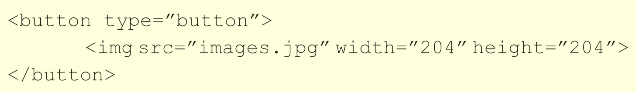




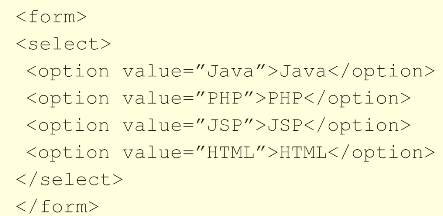
* Sempre que quisermos que o usuário escolha mais de uma opção no checkbox, utilizamos o type=”checkbox”, mas quando quisermos que seja escolhida apenas uma opção, utilizamos type=”radio”;
* Apesar de type=”hidden” não aparecer no browser, ele existe no código fonte para que o desenvolvedor possa incluir informações internamente;
* Tag Datalist – serve para dar sugestões para o usuário, mas ele poderá digitar qualquer coisa no campo, não sendo necessário ser exatamente o sugerido. É como um “preenchimento automático” não obrigatório. Nessa tag é necessário que o id seja igual ao list para vincular as informações;



* Tag Button – este comando pode ser utilizado fora dos formulários para transformar um botão em uma imagem (precisa de JavaScript para funcionar);

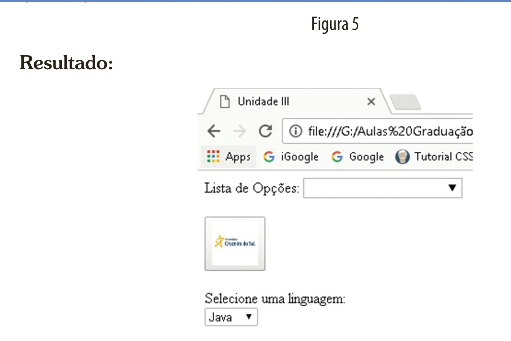


* Tag Select – conhecido como combobox, cria uma lista de opções para a seleção. A ordem de criação é option dentro do select e o select dentro do option;

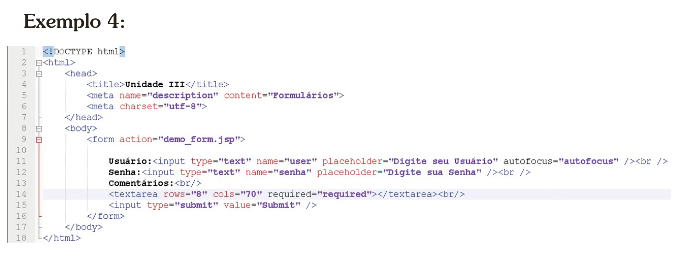


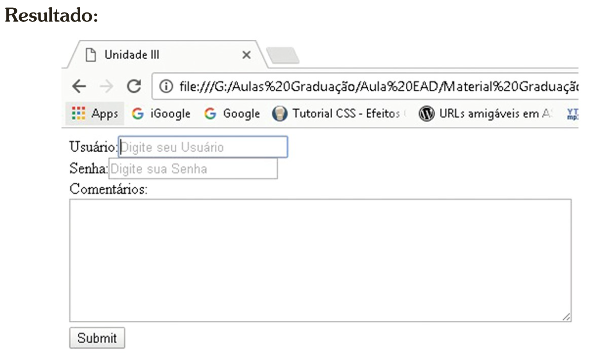
* Exemplo acima estruturados:



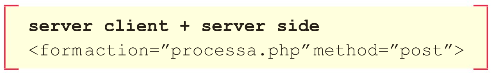


* Atributo Requerid – esse atributa faz com que seja obrigatório a digitação de uma informação;
* Atributo Autofocus – esse atributo dá foco a determinada informação ao carregar a página, como no exemplo abaixo do campo “Usuário”, que ao carregar a p[página já está com o cursor nele;
* Atributo Placeholder – esse atributo é exclusivo de elementos input e textarea que tem como função dar uma dica do que pode ser digitado no campo, sendo que conforme o usuário digita a informação no campo, essa dica desaparece;
* O atributo type=”file” faz com que seja aberto uma caixa de seleção no browser para selecionar o arquivo desejado. Esse atributo é apenas para estrutura, para capturar o arquivo para o servidor, será necessário programar um comando com uso de JavaScript;

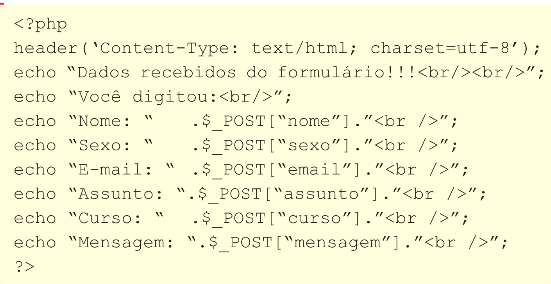


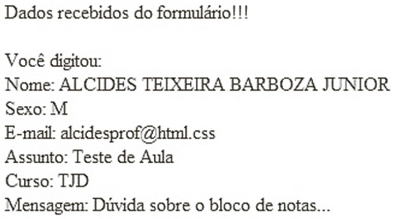


* O arquivo “processa.php” é um aqui que captura as informações digitadas e leva para o servidor.

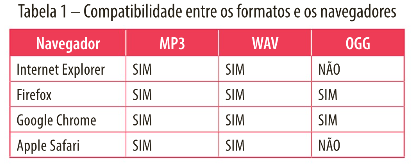


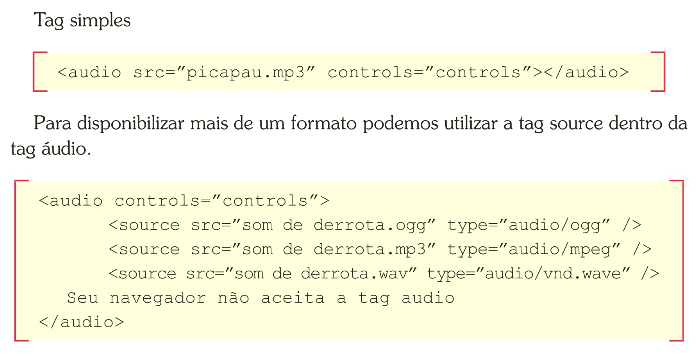
* Exemplo de relatório apresentado para o cliente após preenchimento das informações:



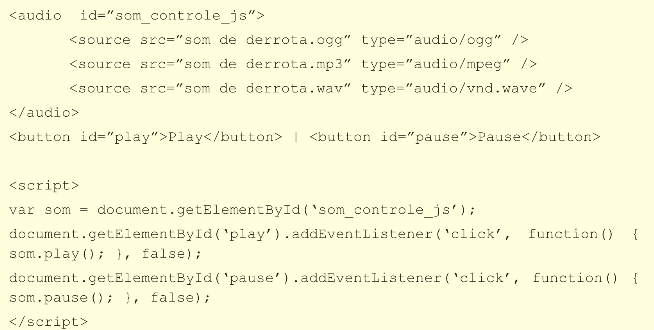


* Para incluir audio em navegadores utilizamos a tag <audio></audio>. Ao utiliza-la é necessário implementar diversos formatos de audio no código, pois ainda não decidiram o Codec que será utilizado para ela. Para isso, é utilizada a tag <source>…</source>;

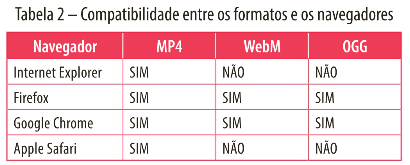




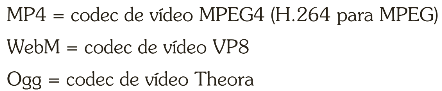
* Exemplo de controle de áudio com JavaScript (apenas demonstração):

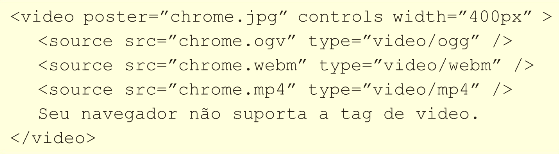


* Antes do HTML5 não havia padrão para reprodução de áudios ou vídeos, para isso eram utilizados plug-ins;
* Para incluir vídeos utilizamos a tag <video></video>. Como também não houve consenso sobre o formato de vídeo adotado (Codec), será necessário identificar o formato para cada navegador utilizando <source></source>;



* Os arquivos de vídeo utilizam compressão e codec, por isso as empresas não concordam em adotar um único formato;

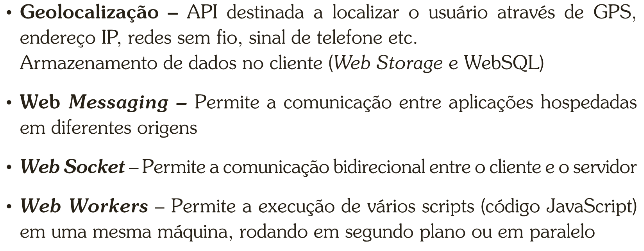




* Exemplo de controle de video com JavaScript (apenas demonstração):



* O HTML5 tem diversos recursos, mas exigem programação em JavaScript. Apesar de nesse capítulo não abordar esse assunto, há algumas explicações válidas abaixo:



**Aplicando Estilo à Estrutura do Documento**

**Layouts**

**Introdução CMS**