

Linguagem de Programação II

Prof.^a Ma. Jessica Oliveira

Aula 07 – 16/09/2024

Formulários HTML e Introdução ao CSS

HTML: Estrutura e Componentes Principais

Conceito de HTML

- O HTML (*HyperText Markup Language*) é uma **linguagem de marcação** utilizada para estruturar e organizar o conteúdo de páginas na web.
- Por meio de elementos que se definem através de *tags*, o HTML estabelece a organização de textos, imagens, links e demais componentes, permitindo a correta exibição pelo navegador.

Conceito de HTML

- A linguagem é **interpretada** pelo navegador para construir a interface de um site, de maneira que o usuário final possa interagir com o conteúdo conforme a estrutura estabelecida.
- Cabe ao desenvolvedor organizar os diversos elementos de forma lógica, garantindo a integridade semântica da página.

Estrutura fundamental de um documento HTML

- Um documento HTML segue uma estrutura padronizada que é composta por três seções principais:
 - **<!DOCTYPE html>**: declara o tipo de documento e indica que o código segue as normas do HTML5.
 - **<html>**: é a raiz do documento, englobando todo o conteúdo da página.
 - **<head>**: seção destinada aos metadados e links importantes, como arquivos CSS e scripts, além do título da página, que aparece na aba do navegador.
 - **<body>**: contém todo o conteúdo visível da página, como textos, imagens, listas, tabelas e outros elementos interativos.

Principais elementos de uma página HTML

- **Cabeçalhos:** variam de **<h1>** a **<h6>** e são responsáveis por hierarquizar os títulos da página, do mais importante (**<h1>**) ao menos relevante (**<h6>**). Eles facilitam a leitura do conteúdo e têm impacto positivo na indexação por mecanismos de busca.
- **Parágrafos:** o elemento **<p>** é utilizado para agrupar blocos de texto, delimitando parágrafos.
- **Listas:** podem ser ordenadas (****) ou não ordenadas (****). Cada item da lista é definido com ****.

Principais elementos de uma página HTML

- **Links:** a tag `<a>` cria *hiperlinks* para outras páginas ou documentos.
- **Imagens:** a tag `` exibe imagens na página, sendo necessário definir o caminho da imagem no atributo “src”.

Formulários HTML

Definição e finalidade

- Formulários são componentes fundamentais em páginas web, utilizados para coletar informações dos usuários.
- Por meio deles, é possível capturar dados como nomes, endereços de e-mail, opiniões, preferências, entre outros.
- Os dados enviados pelos formulários são processados em um servidor, geralmente por linguagens *server-side*, como o PHP.

Estrutura de um formulário

- A tag **<form>** delimita o início e o fim de um formulário, englobando todos os campos de entrada e botões necessários para a coleta de dados.
- Dois atributos são essenciais para o funcionamento do formulário:
 - **action:** define o **destino** para onde os dados serão enviados.
 - **method:** especifica o **método de envio** dos dados (GET ou POST).

| Característica | GET | POST |
|------------------------|---|--|
| Visibilidade | Os dados são anexados à URL. | Os dados são enviados no corpo da requisição. |
| Limite de dados | Limitado (tamanho máximo da URL). | Sem limite específico para o tamanho dos dados. |
| Segurança | Não seguro para dados sensíveis. | Mais seguro que GET, mas exige HTTPS para segurança total. |
| Uso | Requisições de leitura (pesquisas, filtros). | Envio de dados confidenciais, cadastros, uploads. |
| Armazenamento no Cache | Os dados ficam armazenados no histórico do navegador. | Os dados não são armazenados no histórico. |

Tipos de campos de entrada

- Cada campo de entrada é especificado com a tag **<input>**, que define diversos tipos de dados a serem inseridos, conforme o atributo *“type”*.
 - **<input type="text">**: aceita textos curtos.
 - **<input type="password">**: esconde o texto digitado, ideal para senhas.
 - **<input type="email">**: exige que o dado seja um endereço de e-mail válido.
 - **<input type="number">**: permite a inserção de números.

Outros componentes do formulário

- **Área de texto:** a *tag* `<textarea>` cria uma área de texto maior, útil para coletar mensagens ou descrições mais extensas.
- **Campos de seleção:** utilizando do `<select>`, oferecem listas suspensas (*dropdown*) para escolha de opções (*option*).
- **Botões:** são fundamentais para o envio dos dados ou para a redefinição dos campos preenchidos. Os dois tipos principais são o *submit* (envia os dados para o servidor) e o *reset* (limpa todos os campos preenchidos).

Validação de dados

- A validação de dados é essencial para garantir a integridade das informações enviadas pelos usuários.
- O HTML5 fornece uma validação básica por meio de atributos como required (campo obrigatório), pattern (padrão a ser seguido) e maxlength (limita a quantidade de caracteres).

CSS:

Introdução e Estilos

Conceito e função do CSS

- O CSS (*Cascading Style Sheets*) é a linguagem responsável pela definição da **aparência visual** das páginas web.
- Ele atua separando a estrutura do conteúdo (definido em HTML) da apresentação visual, que inclui aspectos como cores, fontes, margens e *layout*.
- Essa separação permite um design mais organizado e reutilizável.

Estrutura de uma regra CSS

- Cada regra CSS é composta por três partes:
 - **Seletor:** identifica o elemento HTML que receberá o estilo.
 - **Propriedade:** define o que será modificado (cor, tamanho, margem).
 - **Valor:** especifica o valor da propriedade.

Seletores

- Os seletores permitem identificar quais elementos HTML serão afetados pelas regras CSS. Os principais tipos de seletores são:
 - **Por *tag*:** aplica estilos a todos os elementos de uma determinada *tag*.
 - **Por *classe*:** aplica estilos a elementos que possuem determinada classe.
 - **Por *ID*:** aplica estilos a um único elemento com ID específico.

Propriedades essenciais do CSS

- Algumas das propriedades mais utilizadas em CSS incluem:
 - **color**: define a cor do texto.
 - **font-size**: altera o tamanho da fonte.
 - **background-color**: configura a cor de fundo de um elemento.
 - **margin**: define o espaço externo entre um elemento e os outros.
 - **padding**: determina o espaçamento interno entre o conteúdo e as bordas do elemento.

Aplicação de estilos em uma página

- **CSS Inline:** é definido diretamente dentro da *tag* do elemento HTML por meio do atributo *style*. Embora seja útil para pequenas modificações, não é recomendado em grandes projetos devido à falta de organização e reutilização de código.
- **Dentro da tag <style> no cabeçalho da página:** o estilo pode ser declarado dentro da tag <style> no cabeçalho (<head>) do documento HTML. Esse método organiza o CSS de maneira mais eficiente do que o *inline*, embora ainda deixe o estilo acoplado ao documento HTML.

Aplicação de estilos em uma página

- **Arquivo CSS Externo:** a maneira mais recomendada de aplicar estilos é utilizando um arquivo CSS separado, que pode ser referenciado no HTML por meio da tag `<link>`. Essa prática promove a separação entre o conteúdo HTML e a estilização, facilitando a manutenção e o reaproveitamento do código CSS em diferentes páginas.

Boas práticas em CSS

- Uma boa prática fundamental é **manter o HTML e o CSS separados**. Todo o estilo de uma página deve ser mantido em um arquivo CSS externo, evitando o uso de estilos *inline*. Isso facilita a manutenção e a reutilização de código.
- Nomeie classes e IDs de maneira clara e descritiva, evitando abreviações desnecessárias ou nomes vagos. As classes devem refletir o propósito ou função de um elemento, não apenas seu estilo visual.
- Não repetir regras similares em várias partes do código.

Vamos para a prática?

Não esqueçam, todos do trio devem realizar as entregas!

Microprojeto 3: Desenvolvimento da Interface de Usuário

Passo 1: Criação de Formulários HTML

- Crie um arquivo `cadastro_eventos.php` onde os usuários poderão cadastrar eventos.
- Utilize HTML para estruturar o formulário.

Passo 2: Estilização com CSS

- Desenvolva um arquivo `style.css` para aplicar estilos ao formulário.
- Adicione estilização básica para melhorar a aparência do formulário.

Passo 3: Validação de Dados

- Adicione validação de dados no lado do cliente usando JavaScript e no servidor usando PHP para garantir que os campos obrigatórios sejam preenchidos corretamente.

Passo 4: Testes

- Teste o formulário em um navegador para garantir que a validação está funcionando corretamente.
- Certifique-se de que os dados estão sendo processados corretamente e que a interface está sendo exibida de maneira adequada.

Na próxima aula...

Aula 08 (23/09/2024) - Revisão para AV1.

Dúvidas?

jessica.oliveira@fbr.edu.br