



Olá, Professor!
Peço que realize a **avaliação do conteúdo** com o intuito de manter o nosso material sempre atualizado.

Avalie este
conteúdo! 🚀 ✨



<https://bit.ly/451BDqS>



Desenvolvimento WEB em HTML5, CSS, Javascript e PHP

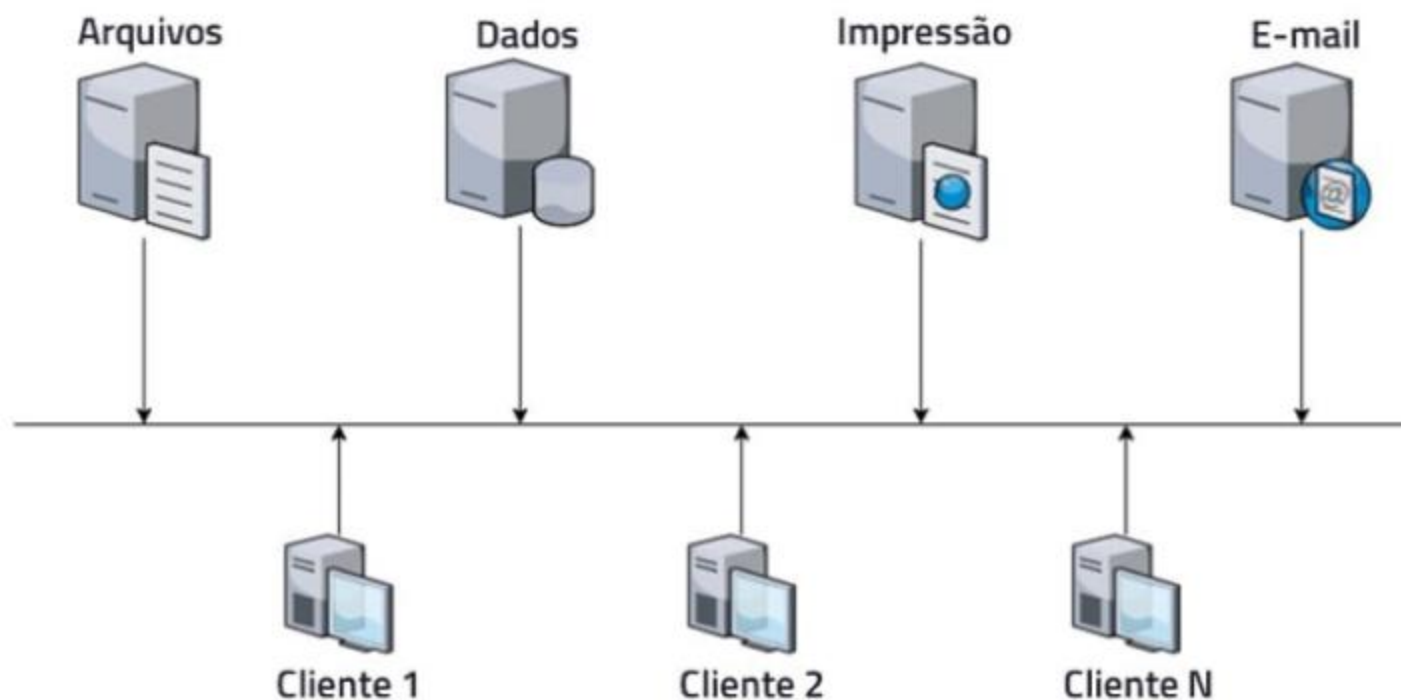


O ambiente internet: cliente X servidor e as tecnologias



— Modelo cliente X servidor

Ambiente cliente X servidor



Arquitetura cliente X servidor em uma rede interna.

— Modelo cliente X servidor

Aplicações no modelo cliente X servidor

- Modelo possibilitou desenvolvimento de aplicações com arquitetura distribuída.
- Aplicações baseadas no conceito de desenvolvimento em camadas.
- Surgiram modelos de duas, três e quatro (ou N) camadas.

— Modelo cliente X servidor

Modelo de duas camadas



Modelo de duas camadas.

— Modelo cliente X servidor

Modelo de três camadas

Camada de apresentação

Representada pela aplicação instalada na máquina cliente. Era responsável pela interface com o usuário e passou a acessar o servidor de aplicação, perdendo o acesso direto ao servidor de dados.

Camada de aplicação

Representada por um servidor responsável pela lógica e pelas regras de negócio, assim como pelo controle de acesso ao servidor de dados.

Camada de dados

Representada por um servidor responsável pelo armazenamento dos dados.

— Modelo cliente X servidor

Modelo de quatro camadas



Cliente

Passou a precisar apenas de um navegador para ter acesso à aplicação.



Servidor

Composto por três servidores — o de aplicações, o de dados e o web

— Ambiente web

Cliente

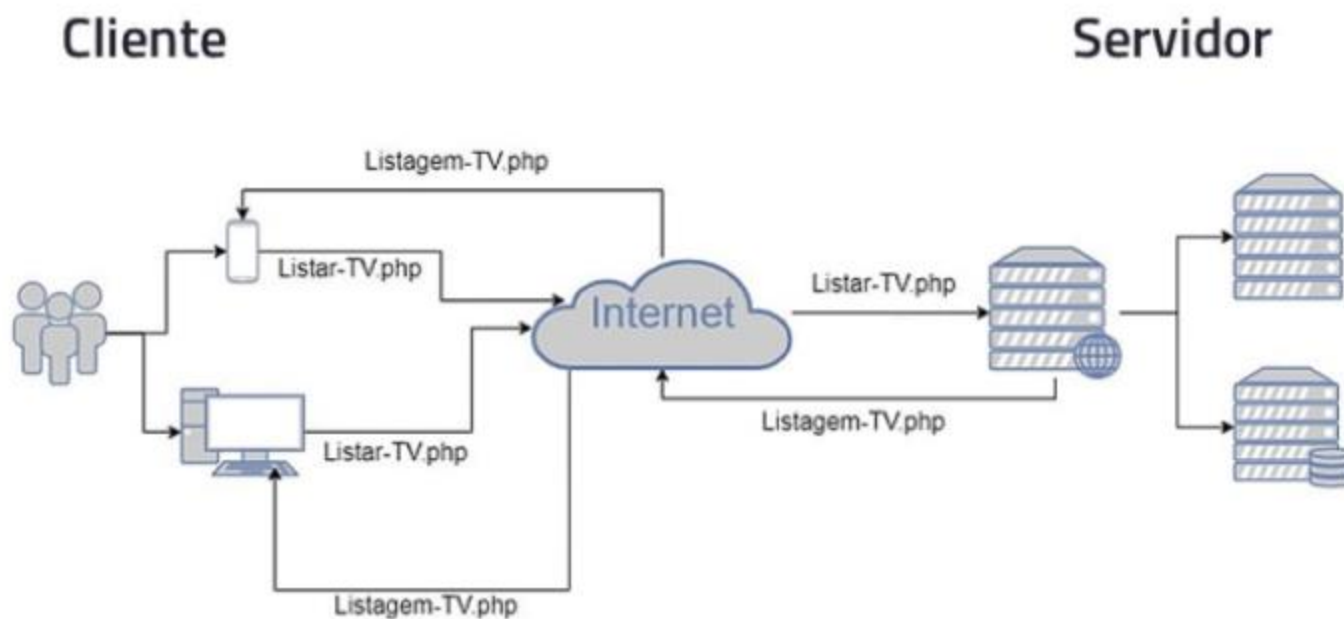
Utiliza um navegador ou aplicativo e consome serviços hospedados em um servidor web.

Servidor web

Sua estrutura pode comportar tanto as camadas de apresentação, aplicação e dados numa única máquina quanto em diversas máquinas, sendo essa distribuição indistinguível para o cliente

— Ambiente web

Comunicação no ambiente web



Comunicação no ambiente web.

— Ambiente web

Solicitação e resposta

- Processo de comunicação na web: solicitação (request) e resposta (response).
- Solicitação geralmente iniciada pelo cliente.
- Servidor também pode iniciar solicitação, como em serviços PUSH.
- PUSH: disparo de notificações/mensagens para clientes que optaram por recebê-las.

— Ambiente web

Client side X Server side

- Expressões comuns: "lado cliente" e "lado servidor".
- Relacionadas a tecnologias e códigos na web.
- "Lado cliente": dispositivo do usuário para fazer requisições.
- "Lado servidor": tecnologias e códigos no servidor.

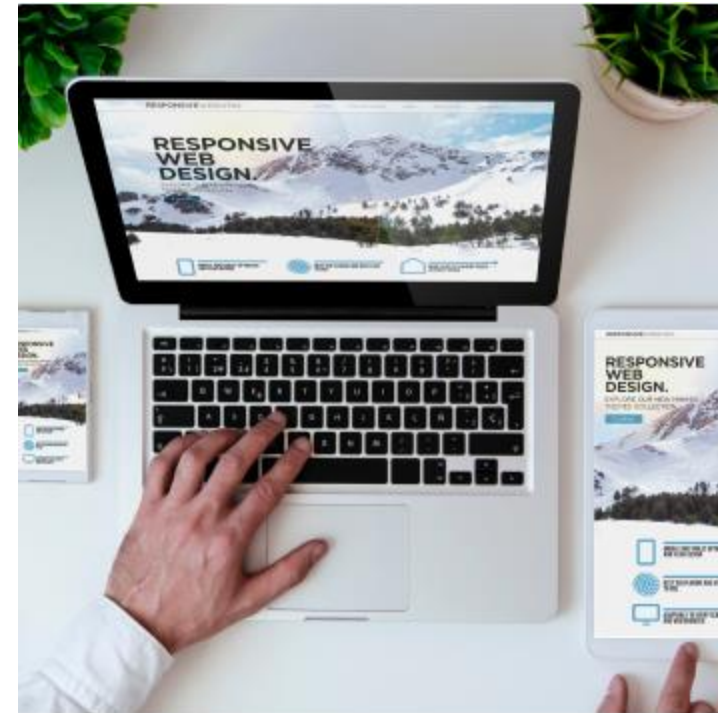
— O conceito de interface

- Interação Humano-Computador (IHC).
- IHC: estudo da interação entre pessoas e computadores.
- Interface: facilita interação humano-computador.
- Evolução da interação: linha de comando e interfaces gráficas.
- No início, interação era primária, foco em hardware e software.
- Avanço tecnológico: acesso a computadores e dispositivos.
- Necessidade crescente de melhorar a interação.

— O conceito de interface

A interface do lado cliente

- Evolução tecnológica.
- Dispositivos variados.
- Garantia de usabilidade.
- Flexibilidade.
- Design responsivo.
- Usabilidade alinhada ao design responsivo.



— O conceito do design responsivo

Design responsivo

- Design responsivo: Responde ao comportamento e ambiente do usuário.
- Baseado em tamanho de tela, plataforma e orientação do dispositivo.
- Implicações práticas: Adaptação automática às preferências do usuário.
- Evita construir diferentes versões para dispositivos variados.

— O conceito do design responsivo

A origem do design responsivo

- Projeto Arquitetônico Responsivo.
- Sala ou espaço ajusta-se automaticamente ao número e fluxo de pessoas.
- Utiliza robótica e tecnologia.
- Materiais que dobram, flexionam e expandem são empregados.
- Paralelo no ambiente web: não criar versões diferentes para características dos visitantes.
- Evita custos adicionais.

— O conceito do design responsivo

Design responsivo na prática

Layouts fluidos

Fazem uso de unidades flexíveis — no lugar de definir as dimensões com o uso de quantidades fixas são utilizados valores flexíveis.

Media query

É a utilização de media types (tipos de mídia) a partir de uma ou mais expressões para definir formatações para dispositivos diversos.

Script

Adiciona interação a uma página web, permitindo, por exemplo, a atualização dinâmica de conteúdos, o controle de multimídia, a animação de imagens e muito mais.

— O conceito do design responsivo

Design responsivo X Design adaptativo



Design responsivo

Medias queries são utilizadas, em conjunto com scripts, para criar um layout fluido que se adapte



Design adaptativo

Um site é planejado e construído com a definição de seis layouts predefinidos, em que são previstos pontos de quebra para que a página se adapte às seis diferentes dimensões utilizadas.

— As tecnologias HTML

HTML

Estruturais

Juntamente com o elemento de definição do DocType, como pode ser visto na imagem seguinte, compõem a estrutura obrigatória de uma página web.

De conteúdo

Como nome sugere, têm o papel de marcar o conteúdo pelo seu tipo.

Semânticas

Relacionadas ao tipo de conteúdo e à criação de seções para agrupá-lo de acordo com sua função no documento. .

— As tecnologias HTML

HTML5

- Novos atributos e elementos, com foco sobretudo na semântica.
- Melhorias de conectividade.
- Possibilidade de armazenamento de dados no lado cliente.
- Otimização nas operações off-line.
- Suporte estendido a multimídia – áudio e vídeo.

— Outras tecnologias: CSS e JavaScript

CSS

O seletor

Um elemento HTML (body, div, p etc.) ou o seu identificador (atributo ID) ou classe (atributo class).

A propriedade

Característica do elemento (cor, fonte, posição etc.).

O valor

Novo parâmetro a ser aplicado à característica do elemento.

— Outras tecnologias: CSS e JavaScript

Como inserir o CSS na página web

Inline

Os estilos são aplicados com a utilização do atributo “style” seguido de uma ou mais propriedades/valores.

Interno

Os estilos são definidos com a utilização da tag <style>, dentro da tag <head> no documento.

Externo

É utilizado um arquivo externo, com extensão “.css”, contendo apenas estilos.

Escopo

Um estilo pode ser definido em nível de escopo, ou seja, declarado em seções específicas do documento.

— Outras tecnologias: CSS e JavaScript

Boas práticas relacionadas à CSS

- Utilizar a forma externa para incluir CSS em uma página web.
- Compactar o arquivo — normalmente chamamos este processo de minificação.

— Outras tecnologias: CSS e JavaScript

Outras considerações sobre a CSS

- Nova funcionalidade ganhando espaço na CSS: pré-processadores.
- Exemplos: Sass, Less, Stylus, entre outros.
- Pré-processador gera CSS a partir de uma sintaxe própria.
- Sintaxe inclui facilidades não nativas do CSS.
- Fluxo: escrita do código com regras usando a sintaxe, compilação para gerar código CSS.

— Outras tecnologias: CSS e JavaScript

JavaScript



Eventos de manipulação DOM

Essa linguagem oferece amplo suporte à manipulação de eventos relacionados a elementos HTML.



Mensagem e entrada de dados

O JavaScript possui suporte a funções nativas para a exibição de caixas de diálogo para entrada de dados ou exibição de mensagens.

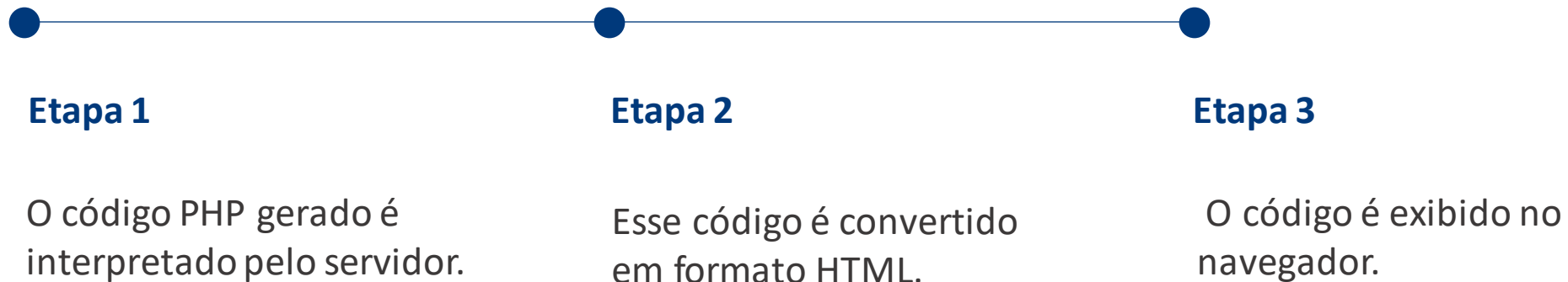
— PHP: uma linguagem de programação server side

PHP

- Linguagem de programação baseada em script e open source.
- Destinada principalmente ao desenvolvimento web.
- Adotou o paradigma de orientação a objetos 10 anos após a criação.
- Foco principal em scripts do lado servidor.
- Suporte a funções como coleta e processamento de dados de formulários HTML.
- Geração de conteúdo dinâmico com acesso a bancos de dados.
- Possibilidade de uso em scripts em linha de comando e criação de aplicações desktop com PHP-GTK.

— PHP: uma linguagem de programação server side

Como o PHP funciona



— PHP: uma linguagem de programação server side

Anatomia de um script PHP

- Script PHP delimitado por tags `<?php` e `?>`.
- A tag de fechamento `?>` não é obrigatória.
- Um mesmo script PHP pode conter código estruturado e orientado a objetos.
- Pode incluir código de marcação HTML.
- Código PHP entre as tags de abertura e fechamento no caso de mesclar com HTML.
- Extensões possíveis para scripts PHP incluem "php", "php3", "php4" e "phtml" (atualmente em desuso).

— PHP: uma linguagem de programação server side

Sintaxe

- Variáveis.
- Tipos de dados.
- Operadores condicionais.
- Laços de repetição.
- Funções e métodos.

— PHP: uma linguagem de programação server side

Inclusão de scripts dentro de scripts

- PHP permite a inclusão de um script dentro de outro.
- Útil para o paradigma de orientação a objetos com diversas classes em diferentes scripts.
- Facilita o uso de classes, métodos ou atributos em um script.
- Funções para inclusão: Include, Require, Include_once, Require_once.

— Páginas dinâmicas e acesso a dados

Páginas dinâmicas

	Páginas estáticas	Páginas dinâmicas
Inclusão/Alteração/Exclusão de conteúdo	Manualmente, direto no código HTML	Automaticamente através de scripts no lado servidor, como PHP
Armazenamento do conteúdo	Na própria página HTML	Em um banco de dados

Comparativo entre páginas estáticas e dinâmicas.

— Páginas dinâmicas e acesso a dados

Acesso a dados

- Ambiente web: cliente e servidor.
- Lado servidor: banco de dados.
- Banco de dados: repositório de informações.
- SGBD: sistemas gerenciadores de bancos de dados.
- SGBD monta estrutura do banco de dados.
- Necessidade de linguagem server side (ex.: PHP) para acessar e exibir dados.

— Páginas dinâmicas e acesso a dados

Formas de acesso a dados

A partir do HTML

Uma das maneiras mais comuns de enviar e recuperar dados a partir do HTML é fazendo uso de formulários.

A partir do Java Script

Possui, essencialmente, duas formas para se comunicar com linguagens do lado servidor: por meio das APIs (*Application Programming Interface*) XMLHttpRequest e Fetch API.

Linguagem de marcação de hipertexto - HTML

```
ol-md-6 col-xs-12 wow fadeInLeft animated">  
Lorem ipsum dolor sit amet, consectetur </h4>  
Lorem ipsum dolor sit amet, consectetur </h2>  
src="img/img.png" alt="">  
orem ipsum dolor sit amet, consectetur adipiscing  
orem ipsum dolor sit amet, consectetur adipiscing  
orem ipsum dolor sit amet, co </p>  
ef="#contact"><button class="download-btn b  
ol-md-6 col-xs-12 horizontal-phone wow fadeIn  
class="img-responsive" src="img/stock-photo-
```

— Estrutura de uma página Web

O princípio da HTML

- HTML: junção de SGML e HyTime.
- SGML: Standard Generalized Markup Language (padrão ISO).
- HyTime: padrão ISO para hiperdocumentos hipermídia.
- HyTime descreve estrutura e conteúdo de hiperdocumentos.
- Inicialmente mantida pelo IETF (Internet Engineering Task Force).
- A partir de 1996 mantida pelo W3C (World Wide Web Consortium).
- Ganhou status de norma internacional (ISO/IEC) em 2000.

— Estrutura de uma página Web

A evolução da HTML: versões

Ao longo dos anos, a HTML tem passado por uma série de aperfeiçoamentos. Tal processo é liderado pelo W3C. Desde a sua criação, a HTML teve diferentes versões: HTML, HTML 2.0, HTML 3.2, HTML 4.01, XHTML, HTML 5, HTML 5.1 e HTML 5.2.

— Definição de tipos de documento - DTD

- HTML: inicialmente tags para marcações simples.
- Primeiras tags para elementos como títulos e parágrafos.
- Expansão rápida da Web e lançamento do navegador Mosaic.
- Necessidade de novas tags para marcações mais complexas.
- Diferentes navegadores implementavam padrões próprios.
- W3C criou DTD (Definição de Tipos de Documentos) para padronização.

— Definição de tipos de documento - DTD

Utilizando Doctypes – Document types

Strict

Determinava uma série de restrições e exigências.

Transitional

Mais maleável em relação aos atributos de apresentação, além de possibilitar a utilização de tags obsoletas.

Frameset

Tag obsoleta da HTML4 usada para combinar múltiplos frames e exibi-los como numa única página.

— Estrutura de uma página Web

Elementos obrigatórios



Raiz

A tag inicial, após a declaração do Doctype, é a `<html>`. Tal elemento é a **raiz do documento**.



Cabeçalho

A tag `<head>` é chamada de **cabeçalho do documento**.



Corpo

Logo abaixo do cabeçalho deve ser inserida a tag `<body>`. Esta é responsável pela **estruturação do documento**.

— Estrutura de uma página Web

Elementos básicos

Estrutura composta por seções:

Cabeçalho

**Barra de
navegação**

Conteúdo

Barra Lateral

Rodapé

— Funcionamento da HTML

- Salvar documento com extensão "html" ou "htm".
- Pronto para ser lido por um navegador Web.
- Conteúdo visível no navegador, não o código ou tags.
- Processo de interpretação e renderização ao exibir a página Web.

— Tags HTML básicas

Aprofundando o conceito de tags

Tags são palavras, escritas entre os caracteres de menor “<” e maior “>” e que servem para informar ao navegador a estrutura e o significado do conteúdo inserido em uma página Web.

— Declaração de tags

Como declarar tags

- Tags são padronizadas pelo W3C.
- Não é possível criar as próprias tags HTML.
- As tags são escritas entre sinais de menor e maior.
- Toda tag deve ser obrigatoriamente fechada.
- O fechamento é feito repetindo a declaração da tag com uma barra "/" antes do nome.

— Tipos e composição das tags

Estruturais

Textuais

Semânticas

— Tipos e composição das tags

Tags de formatação e vs e <i>

- A última versão HTML suporta algumas tags para formatação visual de conteúdo.
- Quatro tags merecem atenção especial: , , <i>, .
- aplica negrito, aplica negrito e marca como importante.
- <i> aplica itálico, aplica itálico e dá ênfase.
- Diferença semântica entre e é perceptível em dispositivos de leitura de tela.
- Tags obsoletas devem ser evitadas devido a possíveis desconfigurações e falta de suporte futuro.

— Listas na HTML

Listas

Ordenadas

Usadas quando desejamos listar dados com a necessidade de representar a sua ordenação de forma numérica ou alfabética.

Não ordenadas

Usadas quando não há necessidade de listar ordenadamente.

De definição

Usadas quando precisamos listar itens e atribuímos uma descrição a eles.

— Tabelas na HTML

- Tabelas são usadas para dados tabulares em páginas web.
- Estrutura: `<table>` é container principal, `<tr>` para linhas, `<td>` para colunas.
- `<th>` é usado para títulos com função semântica, em `<tr>`.
- Tags `<thead>` e `<tfoot>` para cabeçalho e rodapé com função semântica.
- Organização padrão: colunas de mesma largura, linhas de mesma altura.
- Atributos `rowspan` e `colspan` expandem linhas ou colunas.
- Uso indevido de tabelas fora da função semântica prejudica acessibilidade e peso da página.

— Mídias na HTML

Mídias: vídeo e áudio

- Suporte à multimídia na HTML melhorou com o tempo.
- HTML5 permite incorporar vídeo e áudio de forma simples.
- Tags utilizadas: <video> para vídeo, <audio> para áudio.
- Atributos essenciais, como "src", indicam endereço do conteúdo.
- Atributos específicos como "controls", "autoplay" e "loop" ajudam a controlar e definir comportamentos.
- Existem outros atributos disponíveis na especificação do HTML5.
- "Media Events" são eventos que permitem controle de mídia com Javascript.

— Formulários em páginas Web

Formulários: composição e validação

- Formulários são essenciais para interação entre usuários e páginas Web.
- Exemplos de uso incluem solicitar informações, serviços, cadastros, reclamações, entre outros.

— Estrutura básica do formulário

Estrutura inicial do formulário

<form>

Container principal do formulário.

<input>

Campo do formulário.

<textarea>

Campo de texto de múltiplas linhas.

<select> e <option>

Campos de seleção.

button

Campo de botão.
Permite que uma ação seja executada no formulário

Grande para idade gestacional

Usado para definir um título, uma legenda.

— Atributos do formulário

O atributo “type”

Password

Hidden

Checkbox

Radio

Submit

Reset

Button

— Atributos do formulário

Novos atributos e tipos

Atributos:

- placeholder
- required
- autofocus
- pattern

Tipos:

- tel
- datetime
- date
- number

— Validação de dados em formulários

Validação de formulários

- Validade dos dados é tão importante quanto tags e atributos em formulários.
- Estrutura, tags e atributos visam melhorar a experiência do usuário ao preencher o formulário.

— Validação de dados em formulários

Como funciona a validação?

- Validação ocorre no lado cliente e servidor.
- No lado cliente, a validação ocorre ao submeter o formulário antes de ser enviado ao servidor.
- Na HTML5, é possível realizar a validação de forma nativa, sem o uso de Javascript.

— Validação de dados em formulários

Tipos de validação

Dois tipos de validação possíveis:

- Que verifica se o dado inserido em um campo é consistente com o seu tipo e/ou padrão (pattern).
- Que verifica se um campo obrigatório foi preenchido.

Linguagem de marcação e estilos - CSS



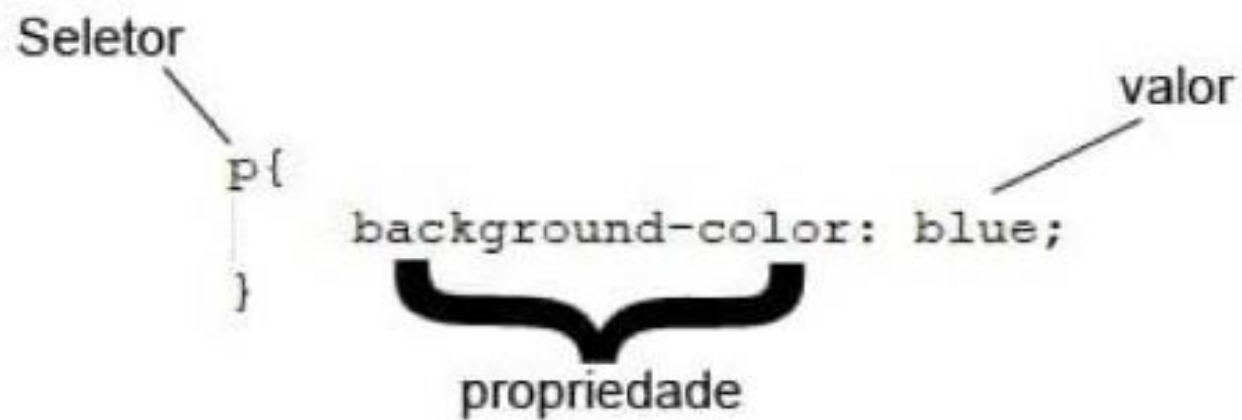
— Fundamentos da CSS

Como a CSS funciona?

- CSS é uma linguagem de estilo que controla a apresentação de um documento HTML.
- Permite alterar forma, posicionamento, cores, fontes, entre outros elementos.
- Permite aplicação seletiva de estilos a elementos na página.
- Pode ser aplicada em um documento inteiro ou em parte dele.
- Mesmo tipo de elemento pode ter estilos diferentes ao longo do documento.

— Fundamentos da CSS

Sintaxe da CSS



Sintaxe da CSS.

— Seletores

- Seletores class e id.
- Restrições e boas práticas na utilização do identificador.
- Seletores de atributo.
- Seletores baseados em relacionamento.

— Propriedades CSS

Background

Border

**Top, Bottom,
Right e Left**

Color

**Font-family,
Font-size, Font-
weight etc.**

Height

**List-style, List-
style-image etc.**

Margin

Padding

Position

Text-...

Width

Z-index

— CSS e HTML

Integrando a CSS à HTML

CSS inline

CSS interna

CSS externa

CSS em escopo

Efeito cascata:

- Herança
- Especificidade

— CSS e HTML

Dicas sobre as regras de precedência

Regra de precedência quanto às formas de inclusão da CSS:

- Os estilos internos e de escopo têm precedência sobre estilos em arquivos externos.
- Os estilos inline têm precedência sobre estilos internos, de escopo e externos.

Regra de precedência Quanto aos seletores:

- Seletores de elemento são os de menor precedência.
- Seletores de classe têm mais precedência que os de elemento.
- Seletores de identificação têm mais precedência que os de classe.

— CSS e HTML

CSS3

- Melhorias de seletores, com novas possibilidades de seleção.
- Efeito gradiente e de sombra em textos e elementos.
- Borda arredondadas.
- Manipulação de opacidade.
- Controle de rotação e perspectiva.
- Animações.

— Recursos de cores

Cores

- Formas de escrita de cores em CSS incluem palavras-chave, notação hexadecimal, `rgb()`, `rgba()`, `hsl()`, e `hsla()`.
- Propriedades de cor incluem, `background-color`, `border-color` e `outline-color`.
- As propriedades podem ser aplicadas a diferentes elementos HTML.

— Recursos de textos e fontes

Texto

- Alinhamento de texto.
- Espaçamento entre linhas.
- Espaçamento entre letras e palavras.

Fontes

- Font-Family, Font-size, Font-style, Fonte-weight.
- Web fontes.

— Conceitos de box model

- Elementos da página web são representados por caixas retangulares.
- Box Model em CSS está relacionado ao design e ao layout.
- Boxes têm quatro componentes principais:
 - margem (margin)
 - borda (border)
 - preenchimento (padding)
 - conteúdo (content).
- A composição do Box Model é prática e envolve a definição de estilos.

— Conceitos de pseudoclasses e pseudoelementos

Pseudoclasses

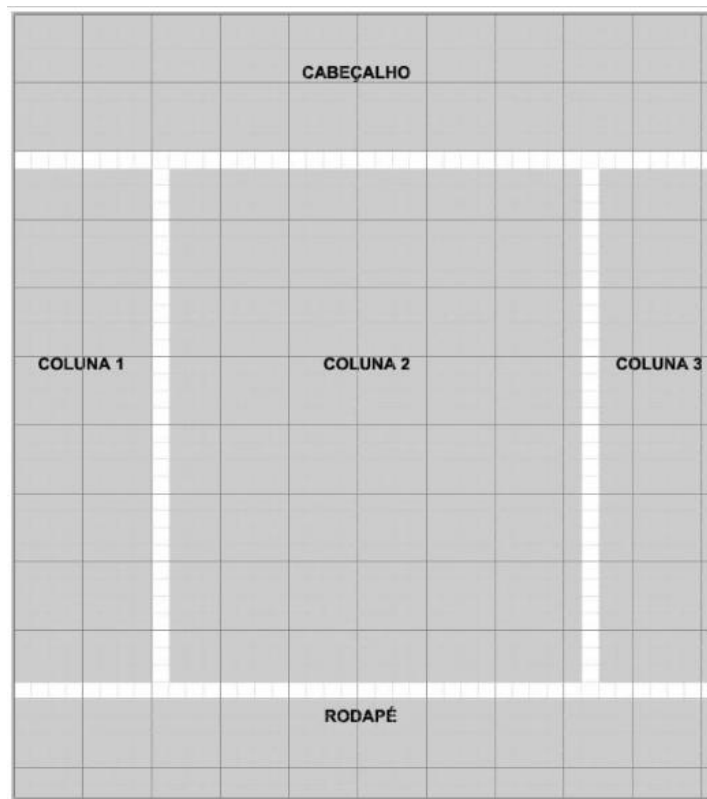
São utilizadas para definir um estado especial de um elemento.

Não ordenadas

São palavras-chave que, adicionadas/relacionadas a um seletor, permitem que uma parte específica dele seja estilizada.

— Conceitos de layout

Layout em colunas e Grid Layout



Layout CSS em colunas.

— Conceitos de layout

A propriedade position

Position static

Essa é a posição padrão dos elementos.

Position relative

Elemento posicionado de modo relativo à sua posição normal.

Position fixed

Define uma posição fixa para um elemento na página.

Position absolute

Elemento posicionado em relação à localização do seu elemento ancestral mais próximo.

Position sticky

Elemento posicionado com base na posição de rolagem da página.

— Frameworks CSS

Frameworks e CSS

- CSS é uma ferramenta poderosa, flexível e complexa, com várias propriedades e valores.
- Marcação HTML tem comportamento natural, mas pode variar entre navegadores.
- Sites frequentemente compartilham layouts semelhantes para melhorar a experiência do usuário.
- Frameworks CSS são conjuntos de componentes reutilizáveis que otimizam o processo de programação.
- Escolha de um framework depende de fatores objetivos e subjetivos.
- Principais frameworks apresentados: Bootstrap, Foundation e Semantic UI.

— Frameworks

- Bootstrap.
- Foundation.
- Semantic UI
- Outros frameworks:
 - Pure
 - Materialize
 - Bulma
 - Skeleton

Linguagem Javascript



```
if self.use_tpu:  
    assert (self.batch_size // self.num_gpus) % 2 == 0  
  
    @property  
    def single_iteration_train_steps(self):  
        if self.single_iteration_train_steps is None:  
            return self.single_iteration_train_steps  
  
    def add_epochs_to_steps(  
        num_epochs=self.single_iteration_train_steps  
    ):  
        return self.single_iteration_train_steps * num_epochs  
  
    @property  
    def single_iteration_eval_steps(self):  
        if self.single_iteration_eval_steps is None:  
            return self.single_iteration_train_steps  
  
    def add_epochs_to_steps(num_epochs=1,  
                             num_eval_steps=1):  
        return self.single_iteration_train_steps * num_epochs +  
            self.single_iteration_eval_steps * num_eval_steps  
  
    @property  
    def single_iteration_train_steps(self):  
        if self.single_iteration_train_steps is None:  
            return self.single_iteration_train_steps  
  
    def add_epochs_to_steps(num_epochs=1):  
        return self.single_iteration_train_steps * num_epochs  
  
    @property  
    def single_iteration_eval_steps(self):  
        if self.single_iteration_eval_steps is None:  
            return self.single_iteration_train_steps  
  
    def add_epochs_to_steps(num_epochs=1):  
        return self.single_iteration_train_steps * num_epochs +  
            self.single_iteration_eval_steps * num_epochs
```

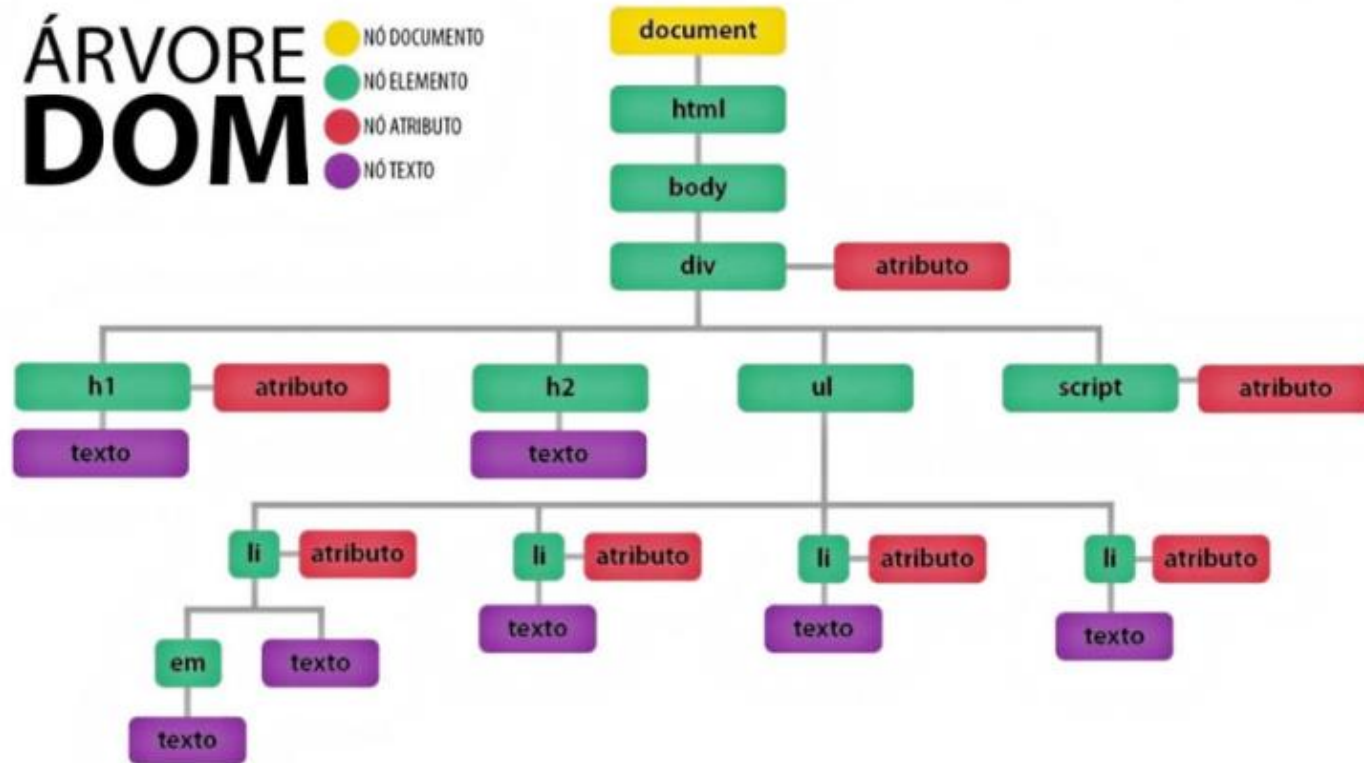

— Conceitos e sintaxe do Javascript

JavaScript

- JavaScript é parte da tríade de tecnologias da web, junto com HTML e CSS.
- Responsável pelo comportamento e interatividade das páginas web.
- Linguagem de programação interpretada e multiparadigma.
- Suporte à programação estruturada, orientada a objetos e funcional.
- Criada inicialmente para o lado cliente da web, mas agora também utilizada no lado servidor e no desenvolvimento de aplicativos móveis.
- Não tem relação com a linguagem de programação Java.

— Árvore DOM

A interface DOM



Árvore DOM (BARBOSA, 2017).

— JavaScript e HTML

Incorporando o JavaScript à HTML



Códigos no corpo da página

Incluindo os códigos diretamente no corpo da página – dentro da seção `<head>` e da tag `<script>`.



Arquivos externos

Mediante arquivos externos, com extensão `js`, linkados ao documento também dentro da seção `<head>`.

— Sintaxe e conceitos

Sintaxe JavaScript

Variáveis

São declaradas utilizando-se a palavra reservada “var”, sucedida pelo seu nome. Não devem ser utilizados caracteres especiais como nomes de variáveis.

Atribuição

Embora declarados da mesma maneira, os tipos de dados são atribuídos de formas distintas.

Ponto e vírgula

Em JavaScript não é obrigatória a utilização de caracteres para indicar o final de uma linha de código, mas adote uma convenção.

Outros elementos

- Alert
- document.getElementByld
- innerHTML
- +=
- Function
- return

— Estruturas condicionais

Estruturas de decisão

if

else

else if

switch

— Laços: estruturas de repetição

Estruturas de repetição

for

while

do/while

for/in

— Vetores

Vetores em JavaScript

Um vetor é um array unidimensional.



A matriz é um array multidimensional (um vetor de vetores).

— Composição e criação de vetores

Composição dos vetores

- Vetor é uma coleção de valores.
- Cada valor é chamado de elemento.
- Cada elemento possui uma posição numérica, conhecida como índice.

— Composição e criação de vetores

Criação de vetores em JavaScript

- Em JavaScript, os vetores não possuem tipo, permitindo a inclusão de números, strings, objetos e outros vetores.
- A capacidade máxima de elementos em um vetor é de 4.294.967.295.
- Vetores em JavaScript têm tamanho dinâmico.

— Manipulação de vetores

Acesso e exibição de elementos do vetor

Em termos de acesso aos elementos de um array, a forma mais simples é utilizando o seu índice.

Javascript



```
1 var alunos = ['Alex', 'Anna', 'João']; // array de strings
2 alert(alunos[0]); // exibirá "Alex" na caixa de diálogo
```


— Manipulação de vetores

push

- O método push é útil para adicionar elementos a um vetor em tempo de execução.
- Sintaxe: `nome_do_array.push(valor)`.
- Possibilidade de inserir múltiplos valores utilizando o método push.

— Manipulação de vetores

Outras formas de adicionar elementos a um vetor

```
JavaScript   
1 alunos[alunos.length] = 'Maria';
```

— Manipulação de vetores

splice

- O método splice em JavaScript é versátil.
- Sintaxe: `Array.splice(posição, 0, novo_elemento, novo_elemento, ...)`.
- `posição` é o índice onde o novo elemento será incluído.
- `0` indica que nenhum elemento do array será excluído.
- `novo_elemento` é o elemento a ser adicionado ao array.

— Manipulação de vetores

Outros métodos para remover elementos do vetor

A linguagem JavaScript possui, além de "delete", outros 3 métodos para remoção de elementos:.

pop

shift

slice

— Ajax e JSON

Requisições síncronas e assíncronas

Requisições síncronas

Quando realizadas, bloqueiam o remetente. Ou seja, o cliente faz a requisição e fica impedido de realizar qualquer nova solicitação até que a anterior seja respondida pelo servidor. Com isso, só é possível realizar uma requisição de cada vez.

Requisições assíncronas

Quando realizadas, não são dotadas de sincronismo. Logo, várias requisições podem ser realizadas em simultâneo, independentemente de ter havido resposta do servidor às solicitações anteriores.

— Requisições assíncronas em JavaScript

AJAX - Requisições assíncronas em JavaScript

- AJAX combina tecnologias como HTML, CSS, JavaScript, DOM, XML, e o objeto XMLHttpRequest.
- Criado em 2005 para requisições assíncronas.
- Permite carregamento rápido e incremental de novos conteúdos.
- O "X" originalmente referia-se a XML, mas outros formatos, como HTML e JSON, são comuns.
- Recursos em JavaScript para requisições: objeto XMLHttpRequest e Fetch API.

— Requisições assíncronas em JavaScript

Outros métodos e propriedades

- API Fetch
- JSON



Programação de páginas dinâmicas com PHP



— A linguagem PHP

- PHP é uma linguagem open source para desenvolvimento Web.
- Integrado ao HTML, proporciona dinamismo às páginas.
- Executado no servidor, ao contrário do JavaScript no lado cliente.
- Surgiu em 1994 como PHP/FI, uma CGI escrita em C por Rasmus Lerdof.
- Código-fonte liberado em 1995, permitindo correções e melhorias.
- Inicialmente chamado de FI, evoluiu para PHP e integrou funcionalidades.
- Principal função inicial: interpretar formulários HTML.
- Possibilita consultas a bancos de dados, processamento e consumo de recursos externos.
- Sintaxe básica inclui variáveis, operadores e integração com HTML.

— A sintaxe PHP

- Scripts PHP iniciam com "<?php" e encerram com "?>".
- As instruções PHP devem terminar com ponto e vírgula.
- Comentários de uma linha começam com "//" e de múltiplas linhas usam "/* ... */".
- A tag de fechamento "?>" não é obrigatória em scripts com apenas código PHP.
- Variáveis em PHP são declaradas com "\$" seguido do nome, não requerem tipo definido.
- Variáveis case-sensitive devem começar com letra ou underscore, seguidas por letras, números e underscores.
- Atribuição de valores usa o sinal de igual "=".
- Boa prática é inicializar variáveis, mesmo que não seja obrigatório.

— Entrada de dados: variáveis de requisição HTTP

Métodos de requisição HTTP

GET

HEAD

POST

PUT

DELETE

CONNECT

OPTIONS

TRACE

PATCH

— Operadores PHP

Operadores aritméticos

Usado com números para a realização de cálculos.

Operadores de atribuição

São utilizados na atribuição de valores a variáveis.

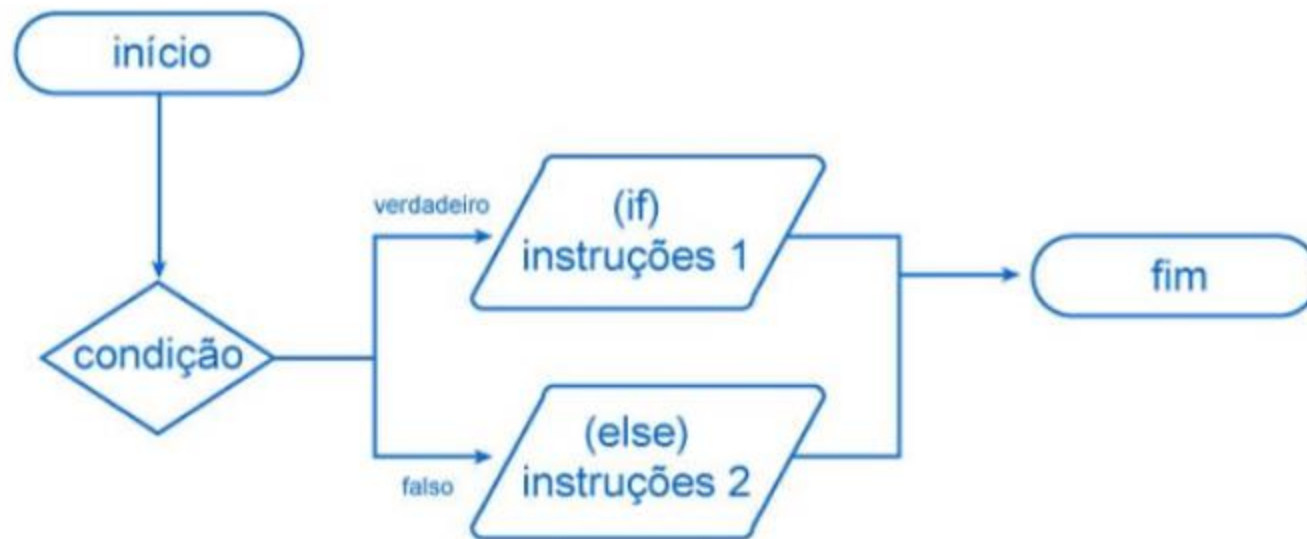
Operadores de comparação

São utilizados para comparar dois valores.

Operadores lógicos

São usados para combinar expressões lógicas.

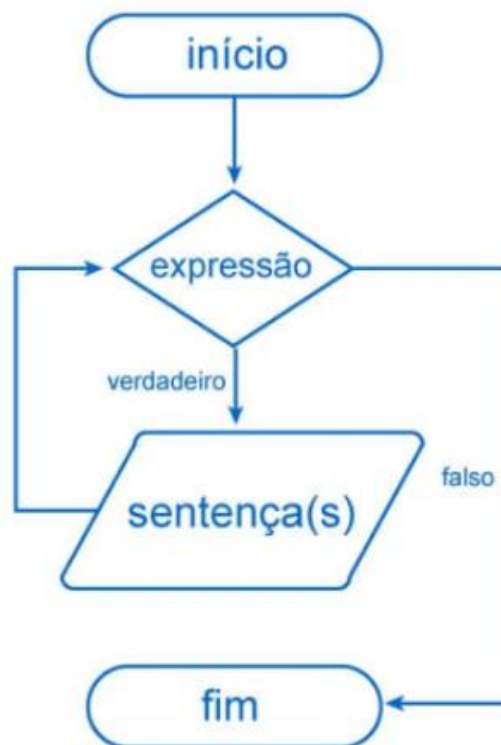
— Condicionais: estruturas de decisão



Fluxo das estruturas de decisão.

— Laços: estruturas de repetição

Em PHP estão disponíveis as seguintes estruturas: **while**, **do-while**, **for**, **foreach**.



Fluxo das estruturas de repetição.

— Condicionais: estruturas de decisão

Formas alternativas

PHP permite que sejam utilizadas formas alternativas das instruções vistas. Em linhas gerais, troca-se a chave de abertura por dois pontos e a de fechamento pela palavra reservada “end” seguida do nome da instrução.

— Arrays: vetores

- array numérico: índice composto por números inteiros.
- array associativo: índice composto por strings.
- array misto: índices numéricos e associativos.

— Arrays: vetores

Remoção de elementos de um array

Há algumas formas de remover elementos de um array.

1

A primeira é definindo o valor do elemento como vazio.

2

Outra forma é fazendo uso de duas funções: `unset` e `array_splice`.

— Funções em PHP

- Funções são blocos de código encapsulados.
- Funções têm um nome, uma definição e são invocadas no código.
- Podem ser nativas (built-in) ou definidas pelo desenvolvedor.
- Funções resolvem problemas específicos e são reutilizáveis.
- A sintaxe de uma função PHP inclui "function", nome, parênteses e chaves para as instruções.
- Parâmetros podem ser declarados dentro dos parênteses.
- Funções podem ou não retornar resultados, utilizando "return".
- Exemplo de funções PHP: "soma" e "imprimir_resultado".

— Funções em PHP

Nomenclaturas de funções e outras boas práticas

- A nomeação de funções em PHP segue as regras de variáveis.
- Pode-se usar underscore ou CamelCase para nomes de funções.
- É recomendado escolher um padrão e segui-lo no código.
- A chave de abertura da função pode seguir duas vertentes: após os parênteses ou na linha seguinte.
- Indentar o código é uma boa prática para melhor compreensão e hierarquia.

— Funções em PHP

Funções nativas

- PHP oferece uma variedade de funções nativas.
- Funções abrangem manipulação de arrays, strings, arquivos, acesso a bancos de dados, entre outras.
- A documentação oficial (Manual do PHP) é uma fonte extensa e essencial para referência.

Integração do PHP com banco de dados

```
E id='$idOrder'; } mysql_close($conn);  
nect.php'; $id = $_POST['id']; $idOrder  
ystapił nieoczekiwany błąd'; } mysql_  
if(mysqli_query($conn,$sql)){echo 'Usuni  
id']; $idOrder=(int)$id; $sql = "DELETE  
d'; } mysqli_close($conn); }<?php incl  
$sql)){echo 'Usunięto użytkownika'; }  
der=(int)$id; $sql = "DELETE FROM user  
mysqli_close($conn); }<?php include 'co  
ho 'Usunięto użytkownika'; } else{echo  
"DELETE FROM user WHERE id='$idOrder'  
php include 'connect.php'; $id = $_POST  
a'; } else{echo 'Wystąpił nieoczekiwany  
E id='$idOrder'; if(mysqli_query($conn,  
nect.php'; $id = $_POST['id']; $idOrder  
ystapił nieoczekiwany błąd'; } mysql_  
if(mysqli_query($conn,$sql)){echo 'Usuni  
id']; $idOrder=(int)$id; $sql = "DELETE  
d'; } mysqli_close($conn); }<?php incl  
$sql)){echo 'Usunięto użytkownika'; }  
der=(int)$id; $sql = "DELETE FROM user  
mysqli_close($conn); }<?php include 'c  
ho 'Usunięto użytkownika'; } else{echo  
"DELETE FROM user WHERE id='$idOrder'  
php include 'connect.php'; $id = $_POST  
a'; } else{echo 'Wystąpił nieoczekiwany  
E id='$idOrder'; if(mysqli_query($conn,  
nect.php'; $id = $_POST['id']; $idOrder  
ystapił nieoczekiwany błąd'; } mysql_  
if(mysqli_query($conn,$sql)){echo 'Usuni  
id']; $idOrder=(int)$id; $sql = "DELETE
```

— Banco de dados e linguagem PHP

- Bancos de dados são essenciais para armazenar e recuperar informações.
- PHP oferece amplo suporte à integração com bancos de dados.
- Sistemas Gerenciadores de Bancos de Dados (SGBD) são comumente associados ao PHP.

— Banco de dados e linguagem PHP

Sistemas Gerenciadores de Banco de Dados



MySQL

Trata-se de um dos SGBDs mais utilizados na web, sendo, normalmente, associado ao PHP, dada a facilidade de conexão entre ambos.

Arquivos externos

O PostgreSQL é um SGBD de código aberto, atualmente disponível sob a licença BSD, criado em 1996. Enquanto o MySQL é um SGBD puramente relacional, o PostgreSQL é considerado um SGBD objeto-relacional.

— Banco de dados e linguagem PHP

Conceitos básicos da linguagem de programação PHP

Constantes

São identificadores ou nomes que contêm um valor único e imutável.

Variáveis

São objetos reservados na memória do computador.

Estruturas de decisão e repetição

Estão disponíveis em PHP: if, else, elseif e switch.

Arrays

São variáveis que guardam uma lista de itens relacionados.

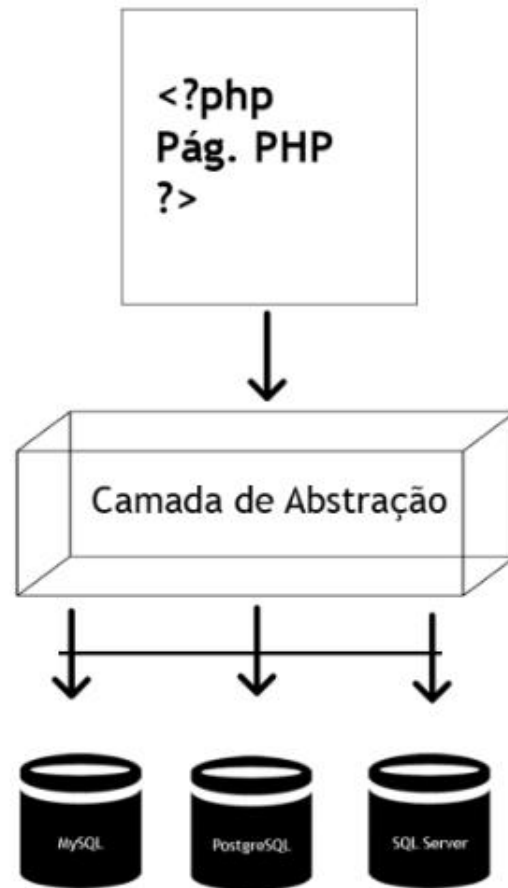
Funções

São pedaços de código que podem ser chamados a partir de qualquer parte do programa

Extensões, bibliotecas e classes

São um importante recurso presente na maioria das linguagens de programação.

— A Classe PDO



Funcionamento da Camada de Abstração.

— Exceções em PHP

Tratamento de exceções

- Boa prática: tratar exceções.
- A instrução "try/catch" é comumente utilizada para tratamento de exceções.
- Um bloco "try" envolve o código que pode lançar exceções, seguido por blocos "catch" ou "finally".
- O objeto lançado como exceção deve ser uma instância ou subclasse da classe Exception.
- O bloco "catch" pode lidar com diferentes tipos de exceções.
- O bloco "finally" é sempre executado após o bloco "try" ou "catch".

— Encerramento de conexões e conexões persistentes

Encerramento de conexões

- Atribuir NULL à variável que armazena a instância da classe PDO.
- Aplicável também às variáveis que guardam resultados de execuções SQL, como Exec e Query.

— Encerramento de conexões e conexões persistentes

Conexões persistentes

- PDO suporta conexões persistentes.
- Conexões persistentes não são encerradas ao final da execução de um script.
- Cache da conexão permite reutilizá-la quando scripts futuros requerem uma conexão semelhante.
- É necessário incluir um parâmetro adicional no momento da criação da instância para usar conexões persistentes.

— Métodos da Classe PDO

Orientação a objetos em PHP

- Conceito vinculado ao paradigma de orientação a objetos.
- Representa uma mudança na forma de construir programas.
- Substitui o conjunto de procedimentos e variáveis por objetos e estruturas (classes).
- Proporciona uma visão mais próxima ao mundo real.
- Introduz conceitos como encapsulamento, herança e polimorfismo.

— Métodos da Classe PDO

Classes e objetos

- Base da programação orientada a objetos.
- Classes definem tipos de dados com atributos e métodos.
- Exemplo de classe: Pessoa com atributos como nome, data de nascimento, CPF.
- Diagrama de Classes exemplifica a estrutura da classe Pessoa.
- Herança permite criar classes que compartilham atributos e métodos de uma classe "pai".

— Os métodos da Classe PDO

A Classe PDO e seus métodos

- Classe PDO dividida em duas:
 - PDO: Conexão com SGBDs, controle de transações, execução de queries.
 - PDOStatement: Métodos relacionados a instruções e retorno.
- Métodos:
 - Método EXEC.
 - Método Query.
 - Outros métodos importantes: Prepare e Execute.

— O Ataque SQL injection

SQL injection

- SQL Injection: Ataque manipula e altera instruções SQL.
- Explora vulnerabilidades em sistemas que aceitam dados sem tratamento.
- Injeção em formulários de login é exemplo prático.
- Código PHP vulnerável a valores manipulados em formulário.
- Comandos maliciosos podem modificar instruções SQL originais.
- Possibilidade de acesso não autorizado ao sistema.

— O Ataque SQL injection

Métodos Prepare e Execute

- Método Prepare:
 - Prepara instrução antes de execução.
 - Utiliza recurso bind para vincular variáveis à instrução SQL.
 - Previne SQL Injection com tratamentos internos automáticos.
- Método Execute:
 - Realiza o bind entre named parameters e variáveis PHP.
 - Named parameters substituem valores provenientes de formulário HTML.