

# Linguagem de Programação I PHP Básico



Prof. Esp. Norton Guimarães  
[norton.guimaraes@ifgoiano.edu.br](mailto:norton.guimaraes@ifgoiano.edu.br)

# Agenda

- Parte I
  - Histórico
  - Familiarizando com o PHP
  - Tipos de Dados, Operadores, Estruturas de Controle, Funções, Escopo de Variáveis, Include, Arrays
  - Conceitos de Orientação a Objetos
- Parte II
  - Trabalhando com Arquivos
  - Trabalhando com OO
  - Formulários HTML
  - Cookies e Sessões
  - Outros componentes e funções PHP



# PARTE I



# Histórico

- O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem interpretada.
- O PHP é sucessor do chamado PHP/FI. PHP/FI foi criado por Rasmus Lerdorf em 1995.
- Atualmente a versão do PHP é o 5.3. O PHP 5 foi lançado em julho de 2004.



# O mascote

- O mascote do PHP é o Elefante.

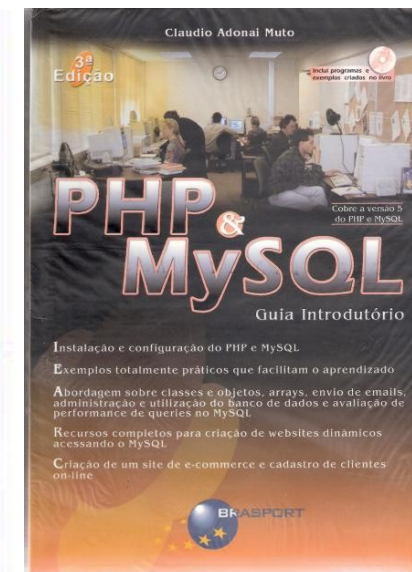
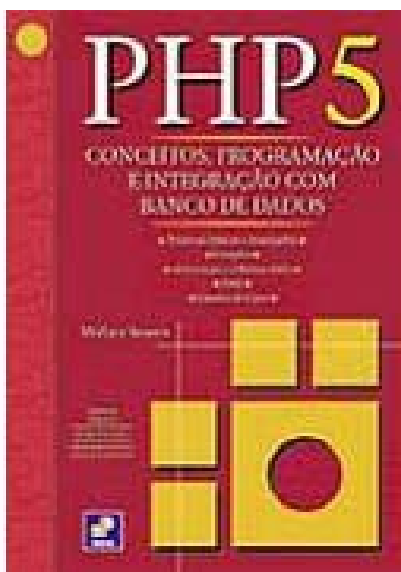
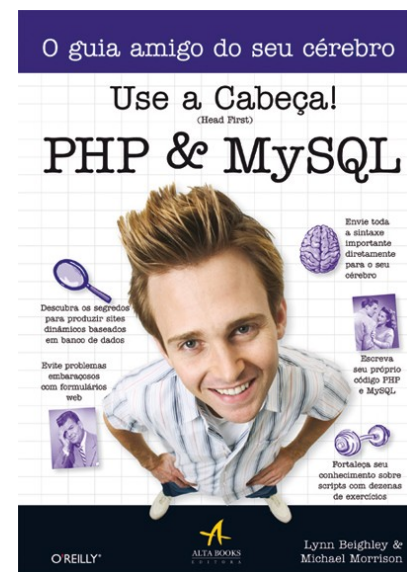
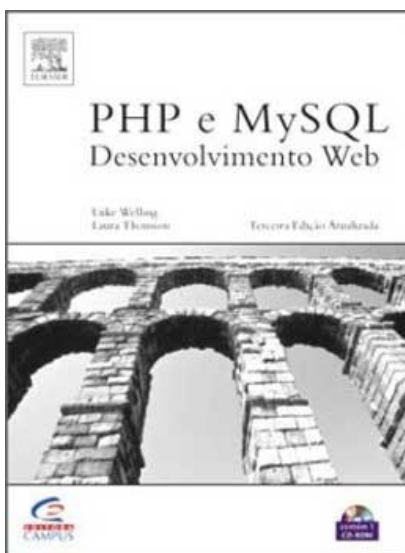
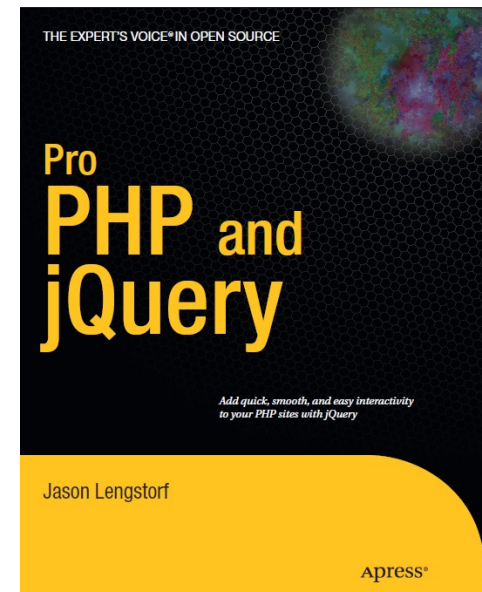
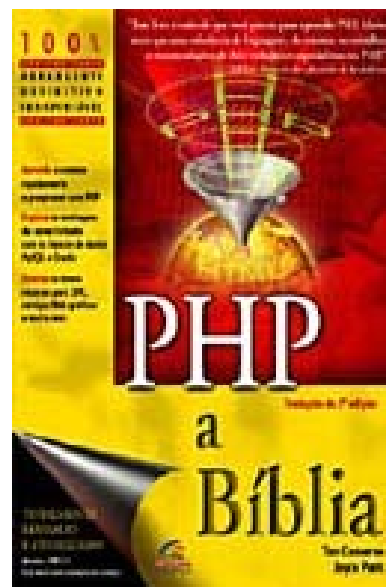
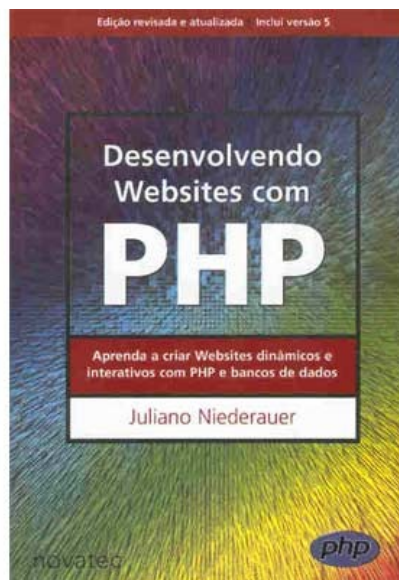




# Livros sobre PHP



<clique para folhear>



# Softwares Necessários

- Requisitos básicos:
  - Servidor Apache (apache.org);
  - Linguagem PHP (php.net);
  - Servidor MySQL (mysql.com);
- Tudo isso é instalado facilmente com o Lamp (Linux) ou Wamp (Windows).
- Existe a junção dos dois anteriores Xampp. Este que utilizaremos no curso.
  - Download:  
[http://www.apachefriends.org/pt\\_br/xampp-windows.html](http://www.apachefriends.org/pt_br/xampp-windows.html)



# Conhecendo o Xampp

- O diretório base é o C:/xampp/htdocs;
- Para acessar o Xampp: <http://localhost>
- Mais informações:
  - [Conhecendo e configurando o xampp](#)



# Familiarizando com o PHP

- O ciclo das páginas

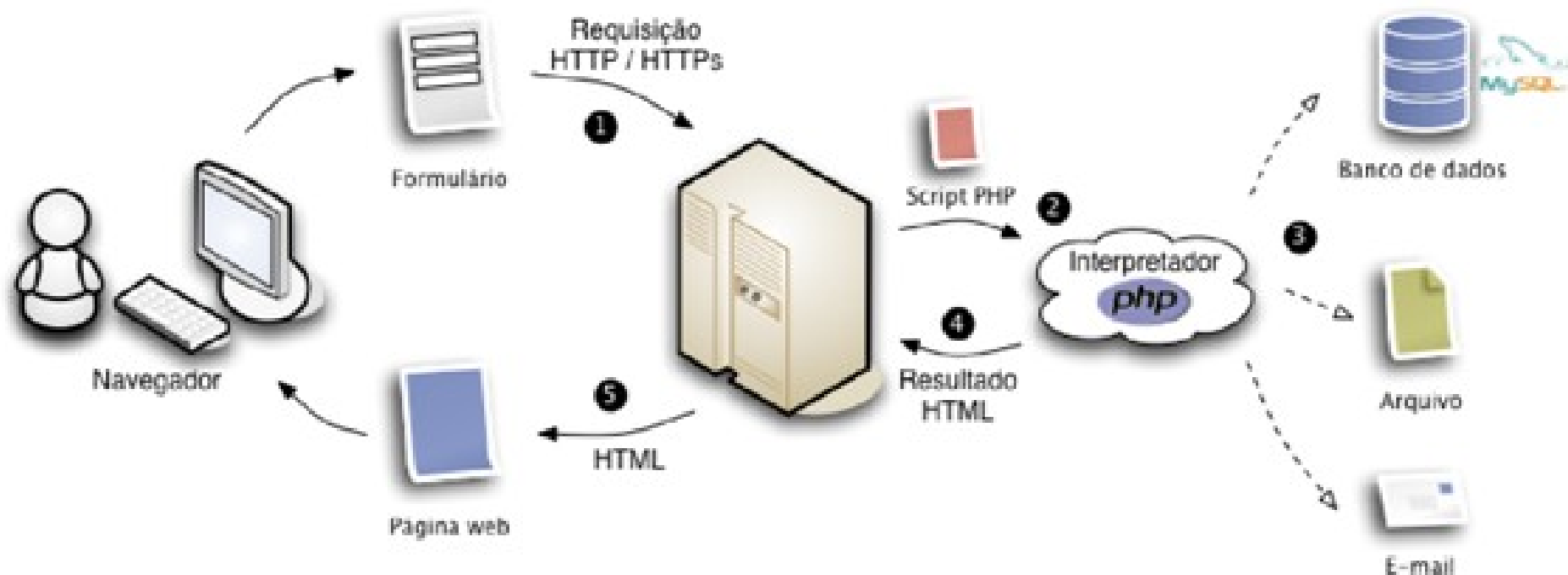


Figura – Como funciona o PHP

Fonte: <http://locobit.blogspot.com/2011/01/apresentando-php.html>

# Primeiro Exemplo

- Olá Mundo!

```
1 <html>
2     <head>
3         <title></title>
4     </head>
5     <body>
6         <?php
7             echo "Olá Mundo!";
8         ?>
9     </body>
10 </html>
```

# Comentários

- Simples
  - Uso do // para comentar uma linha
- Várias Linhas
  - Uso de /\* e \*/ para comentar várias linhas.

```
1 <html>
2   <head>
3       <title></title>
4   </head>
5   <body>
6       <?php
7           //Comenta uma linha
8           echo "Olá Mundo!";
9       /*
10          * Comenta várias linhas
11          */
12       ?>
13   </body>
14 </html>
```



# Variável no PHP

- Toda variável no PHP é precedida de \$.
- A variável no PHP recebe qualquer valor.
- Os nomes de variável no PHP fazem distinção entre maiúsculas e minúsculas.
- Exemplo:

```
<?php
$var = 'Bob';
$Var = 'Joe';
echo "$var, $Var";           // exibe "Bob, Joe"
$4site = 'not yet';          // inválido; começa com um número
$ _4site = 'not yet';        // válido; começa com um sublinhado
$täyte = 'mansikka';         // válido; 'ä' é um caracter ASCII
?>
```

# Uso das Aspas

- Aspas Simples:
  - Utilizada para imprimir exatamente o texto contido entre aspas.
- Aspas Duplas:
  - Utilizado para imprimir com os valores de variáveis e caracteres de escape.

```
<?php  
$var = 'Bob';  
echo 'Meu nome é $var';  
//A saída será Meu nome é $var  
?>
```

```
<?php  
$var = 'Bob';  
echo "Meu nome é $var";  
//A saída será Meu nome é Bob  
?>
```

# Caracteres de Escape

- `\n` : Nova linha.
- `\r` : Retorno de Carro.
- `\t` : Tabulação horizontal.
- `\\` : A própria barra (`\`).
- `\$` : O símbolo \$.
- `\'` : Aspas simples.
- `\"` : Aspas duplas.



# Operadores

- = : atribuição simples
- += : atribuição com adição
- -= : atribuição com subtração
- \*= : atribuição com multiplicação
- /= : atribuição com divisão
- %= : atribuição com módulo
- .= : atribuição com concatenação
- ++ : Incremento
- -- : Decremento

# Operadores

- Exemplo:

```
//Operadores com atribuição
```

```
$cont += 1; // é o mesmo que $cont = $cont + 1;
```

```
$cont -= 1; // é o mesmo que $cont = $cont - 1;
```

```
$cont *= 1; // é o mesmo que $cont = $cont * 1;
```

```
$cont /= 1; // é o mesmo que $cont = $cont / 1;
```

```
//Incremento
```

```
$cont++; // é o mesmo que $cont = $cont + 1;
```

```
//decremento
```

```
$cont--; // é o mesmo que $cont = $cont - 1;
```

# Operadores Lógicos

- And : e lógico
- Or : ou lógico
- Xor : ou exclusivo
- ! : não (inversão)



# Comparação

- == : igual a
- != : Diferente de
- < : Menor que
- > : Maior que
- <= : Menor ou igual a
- >= : Maior ou igual a

# Estrutura de Seleção

- Comando IF

```
if ($cor == "azul") {  
    echo "céu";  
} else {  
    echo "Qualquer coisa";  
}
```

```
if ($cor != "azul") {  
    echo "céu";  
} else if($cor == "vermelho") {  
    echo "alerta";  
} else{  
    echo "Qualquer coisa";  
}
```

# Estrutura de Seleção

- Comando SWITCH

```
$num = "";  
switch ($num) {  
    case 1: echo "Um";  
        break;  
    case 2: echo "Dois";  
        break;  
    case 3: echo "Três";  
        break;  
    default: echo "Nove";  
        break;  
}
```



# Lista de Exercícios 01

- 1 - Que informe se um dado ano é ou não bissexto. Obs: um ano é bissexto se ele for divisível por 4 e não por 100.
- 2 - Calcular a média final dadas as notas das 3 provas e produzir uma saída com a média e a situação do aluno de acordo com o seguinte critério:  $\text{média} \geq 7$ , aprovado;  $5 < \text{média} < 7$ , recuperação;  $\text{média} < 5$ , reprovado.

# Estrutura de Repetição

- Comando WHILE

```
//while  
$num = 0;  
while ($num < 3) {  
    echo "Alguma coisa";  
    $num++;  
}
```

# Estrutura de Repetição

- Comando DO..WHILE

```
//Do-While  
$ind = 10;  
do {  
    echo "O número $ind é maior que 5!";  
    echo "<br>";  
    $ind++;  
} while ($ind <= 10);
```

# Estrutura de Repetição

- Comando FOR

```
for($ind=0; $ind<=20; $ind+=2) {  
    echo "Número: $ind";  
    echo "<BR>";  
}
```



## Lista de Exercícios 02 - While

- 1) Exiba 10 linhas de um texto qualquer.
- 2) Exiba a soma de todos os números de 0 a 10.
- 3) Exiba todos os números de 0 a 100;
- 4) Exiba todos os números pares de 0 a 100;
- 5) Sorteie números aleatórios de 0 a 50, imprima todos na tela e só pare quando o número sorteado for 37;

## Lista de Exercícios 03 - For

- 1) Exiba 10 linhas de um texto qualquer.
- 2) Exiba a soma de todos os números de 0 a 10.
- 3) Exiba todos os números de 0 a 100;
- 4) Exiba todos os números pares de 0 a 100;
- 5) Sorteie números aleatórios de 0 a 50, imprima todos na tela e só pare quando o número sorteado for 37;

# Funções

- Realiza uma tarefa específica

```
function alteraValor(&$var1, $var2) {  
    $var1++;  
    echo $var1;  
    $ret = $var1 + $var2;  
    return $ret;  
}
```

nome da função

parâmetros

passagem por referência

Valor a ser retornado

# Include

- A diretiva include é responsável por inserir e interpretar o código PHP, contido em um arquivo externo.

```
include "funcoes.php";
```

```
echo "Código principal";
```

```
echo "<BR>";
```

```
soma(40, 60);
```



# Arrays

- Simples

```
$estado[] = "GO";  
$estado[] = "MG";  
echo $estado[0];  
.
```

- Multi-Dimensionais

```
$lista = array(array('Maria', 'Carlos'), array(50, 60));  
echo $lista[0][0];  
echo '<br />';
```

## Lista de Exercícios 04

- 1) defina um vetor de 365 elementos e inicialize todos seus elementos com o valor numérico 0.
- 2) Defina um vetor de 12 elementos e, inicialize os seis primeiros com o valor lógico VERDADEIRO e os seis restantes com o valor lógico FALSO.
- 3) Defina um vetor para conter os nomes dos dias da semana. Faça as atribuições considerando que o elemento de índice um (1) é o Domingo. Em seguida mostre na tela todo o vetor.

## Lista de Exercícios 04 cont..

- 4) Preencha uma matriz de 10 linhas por 15 colunas com números aleatórios de 1000 a 2000. Em seguida exiba todo o seu conteúdo.

## Lista de Exercícios 04 cont..

- 5) Armazene numa matriz  $12 \times 2$  de inteiros o número do mês com a quantidade de dias que o mês possui. Para fevereiro considere 29 dias. Armazene num vetor os nomes dos meses. Em seguida, pergunte ao usuário qual o número do mês que ele quer consultar. Após a sua resposta exiba o nome do mês e a quantidade de dias que ele possui.



# Conceitos de Orientação a Objetos

## ● Objeto

- A instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.
- No PHP a criação de objetos é através da palavra reservada **new**.

```
<?php  
    $teste = new Foo();  
?>
```

---

# Conceitos de Orientação a Objetos

## ● Classe

- Representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos.
- No PHP:
  - atributos de uma classe são definidos através da declaração "var".
  - métodos são definidos através de funções.
  - Para acessar atributos dentro do método, você deve usar o operador "\$this->" precedendo o atributo.

# Exemplo de Classe

```
<?php
class Foo {
    var $x;
    var $multiplicador;

    function Foo() {
        //Esse é o método construtor e será chamado
        //pela primeira vez quando a classe for criada
        $this->multiplicador = 4;
    }
    function set($prop, $value) {
        $this->$prop = $value;
    }
    function get($prop) {
        return $this->$prop;
    }
    function soma() {
        return $this->x * $this->multiplicador;
    }
}
?>
```

# Conceitos de Orientação a Objetos

- Visibilidade

```
class Disciplinas {  
  
    private $cod;  
    public $nome;  
    protected $carga;  
  
    private function consultarTodos() {
```



# Conceitos de Orientação a Objetos

- Herança

```
<?php  
class Homem extends Pessoa{  
  
}  
?>
```

## PARTE II

Características da linguagem: sessões, upload de arquivos, cookies, autenticação HTTP, conexão persistente com banco de dados; Conceitos de Programação (Funcionamento, Boas Práticas de Programação); Sessões e Cookies (Criação e exclusão de Cookies), Serialização de Dados; Cabeçalhos e E-mail (Controle de Cabeçalhos do HTTP, Autenticação com HTTP, Envio de E-mails);

# Session - Sessão

- Suporte a sessões no PHP consiste de uma maneira de preservar dados através de acessos subsequentes.

```
<?php
session_start();

if(isset($_SESSION['views']))
{
    $_SESSION['views']=$_SESSION['views']+1;
}
else
{
    $_SESSION['views']=1;
    echo "Views=". $_SESSION['views'];
}
?>
```

# Cookies

- Cookies são um mecanismo para guardar dados no navegador remoto. Ele é bom para:
  - Verificar se um usuário já logou no site (isto é, validar se o cookie existe no computador)
  - Verificar se um usuário já votou na enquete do site ( || )
  - Carrinho de compras na hora de armazenar os produtos comprados



# Cookies Exemplo

<?php

```
//Cookie
$user = "nortoncg";
//Cria o Cookie
setcookie("usuário", $user, time() + 172800);

$nome_do_cookie = "usuário";
if (isset($_COOKIE[$nome_do_cookie])) {
    echo "O cookie $nome_do_cookie existe! ";
} else {
    echo "O cookie $nome_do_cookie não existe!";
}

//escrever o valor do Cookie
echo $_COOKIE["usuário"];
```

?>

# Excluir Cookies Exemplo

- `setcookie("usuário", "");`

# Formulário em PHP

- Suporta Método Get e Post

```
<form name="frmLogin" method="post"  
      action="<?php echo $_SERVER['PHP_SELF']; ?>">
```

- Todo elemento HTML tem que ter "ID" e "Nome" definidos.

```
<input name="txtSenha" type="password" id="txtSenha" size="20">
```

# Métodos PHP

## ● Método Post

- As informações do Formulário HTML são encapsulados no Cabeçalho do Protocolo HTTP.
- Mais Seguro
- Mais lento
- Utiliza a Tag:

```
$_POST['txtSenha']
```

# Métodos PHP

- Método Get
  - As informações do Formulário HTML são exibidas na URL.
  - Menos Seguro
  - Mais rápido
  - Utiliza a Tag:

```
$ _GET [ 'txtSenha' ] ;
```



# Nosso Padrão

```
$erro = "";  
  
//Verifica se foi dado Post na Página  
if (!empty($_POST)) {  
    $objeto = new Usuarios();  
    $objeto->set('login', $_POST['txtLogin']);  
    $objeto->set('senha', $_POST['txtSenha']);  
  
    if ($objeto->verificarUsuario()) {  
        session_start();  
        $_SESSION["usuario"] = $objeto;  
        //Redireciona para outra página  
        header("Location: painel_controle.php");  
    } else {  
        //Emite uma mensagem  
        $erro = "Desculpe, nenhum registro encontrado!";  
    }  
}
```

# Exercícios

- Construa um Formulário (cadPessoa.php) com os seguintes Campos:
  - ID – Inteiro – size: 2;
  - Nome – String – size: 20;
  - Idade – Inteiro – size: 3;
  - Sexo – String (Masculino e Feminino);
  - Endereço – String – size: 200
  - Botão Enviar e Limpar
- Construa uma Classe Pessoa para o formulário
- Utilize o Padrão apresentado<sub>47</sub>

# Padrão de Nomenclatura

- Botão – btnEnviar
- Formulário – frmLogin
- Texto – txtLogin
- Checkbox – chkPermite
- Seleção – cboEstado
- RadioButton – rdbSexo
- Label - lblNome

## PARTE III

Manipulação de Banco de dados (Como Acessar Bancos de dados, Execução de Consultas, Tratamento de Erros, Recuperação de Registros).

# Linguagem SQL

- SELECT
- INSERT
- UPDADE
- DELETE



# Linguagem SQL - SELECT

- SELECT \*
  - FROM Aluno
- SELECT \*
  - FROM Aluno
  - WHERE idAluno = 1
- SELECT idAluno, nomeAluno
  - FROM Aluno
  - GROUP BY nomeAluno
- SELECT count(idAluno)
  - FROM Aluno
  - GROUP BY idAluno

# Linguagem SQL - INSERT

- INSERT INTO
  - Aluno (nomeAluno)
  - VALUES ("Maria Aparecida")

# Linguagem SQL - UPDATE

- UPDATE Aluno
  - SET nomeAluno = "Maria José"
  - WHERE idAluno = 1

# Linguagem SQL - DELETE

- DELETE FROM Aluno
  - WHERE idAluno = 1

# Funções MySql no PHP

- `mysql_connect(<host>,<user>,<pass>)`
  - Abre a conexão no servidor MySQL
- `mysql_select_db(<banco>)`
  - Seleciona o banco de dados
- `mysql_query(<sql>)`
  - Executa o comando SQL
- `mysql_close()`
  - Fecha a conexão



# Tratamento de Erro no PHP

- Utiliza o bloco Try-Catch
- Utiliza a Classe Exception
  - getMessage() - Mensagem de erro.
  - getCode() - Código do erro.
  - getFile() - Retorna o path do arquivo onde a exceção foi gerada.
  - getLine() - Retorna o número da linha onde a exceção foi gerada.

# Tratamento de Erro no PHP

- Exemplo

```
// instância objeto
$cidade = new cidades();

// try(tentar)
try {
    $cidade->pesquisa("Campinas");
}
// catch (pega exceção)
catch (Exception $e) {
    echo ($e->getMessage());
}
```

## PARTE VI

Características da linguagem: upload de arquivos.

# Upload de Arquivo

- Altera a tag FORM do HTML

```
<form name="frm" method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>"  
enctype = "multipart/form-data">
```

```
<input class="texto" name="txtFoto" type="file" value="">
```

# Upload de Arquivo

## ● Exemplo de Upload

```
//se existir a imagem da foto
if (isset($_FILES["txtFoto"])) {

    $arquivo = $_FILES["txtFoto"];

    $pasta_dir = "../imagens/fotos/"; //diretorio dos arquivos
    //se não existir a pasta ele cria uma
    if (!file_exists($pasta_dir)) {
        mkdir($pasta_dir);
    }

    $arquivo_nome = $pasta_dir . $arquivo["name"];

    // Faz o upload da imagem
    move_uploaded_file($arquivo["tmp_name"], $arquivo_nome);
}
```