

# Programação Web com PHP

Prof. Wyllyams Barbosa Santos

wyllyamss@gmail.com

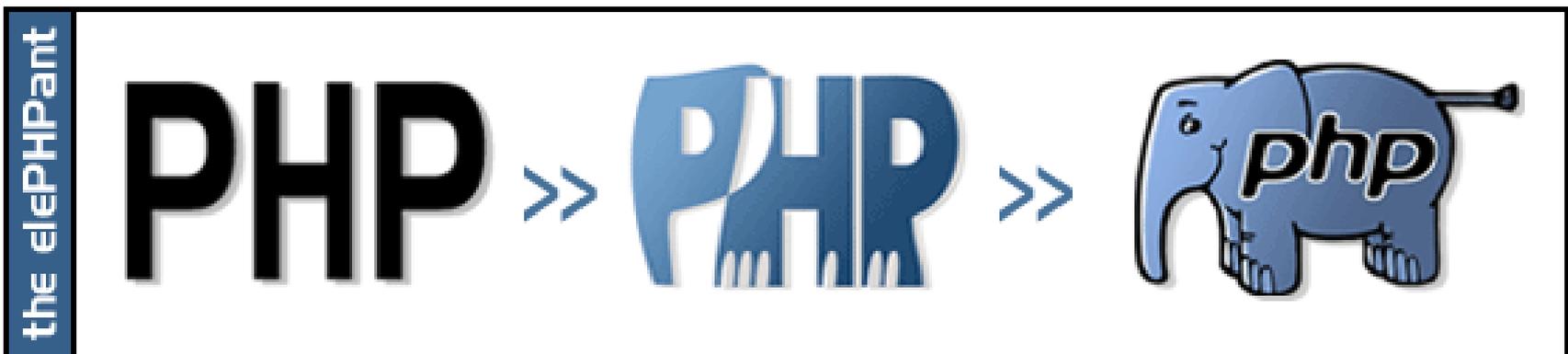
Optativa IV – Projetos de Sistemas Web

# Agenda

- O que é PHP?
- O que da pra fazer com PHP?
- Como a web funciona?
- Onde, quando e por que usar PHP.
- PHP na prática:
  - Sintaxe, leitura de dados de formulário e acesso a banco de dados MySQL;

# O que é PHP?

- PHP é uma linguagem de programação multiparadigma, com tipagem dinâmica, voltada para o desenvolvimento de websites dinâmicos;



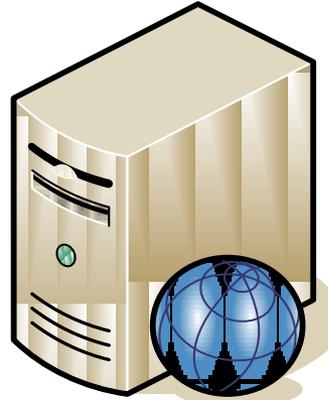
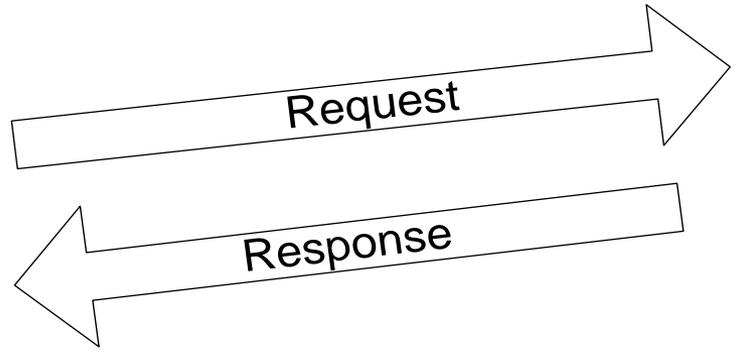
# O que da pra fazer com PHP?

- PHP é uma linguagem voltada para o desenvolvimento de websites dinâmicos;
- Com PHP, podemos criar websites capazes de:
  - Enviar e-mail utilizando serviços de e-mail;
  - Fazer diversas operações em banco de dados;
  - Trabalhar com dados enviados por formulário;
  - Trabalhar com cookies e sessões HTTP;

# Como a web funciona?



Cliente (Navegador)



Servidor Web

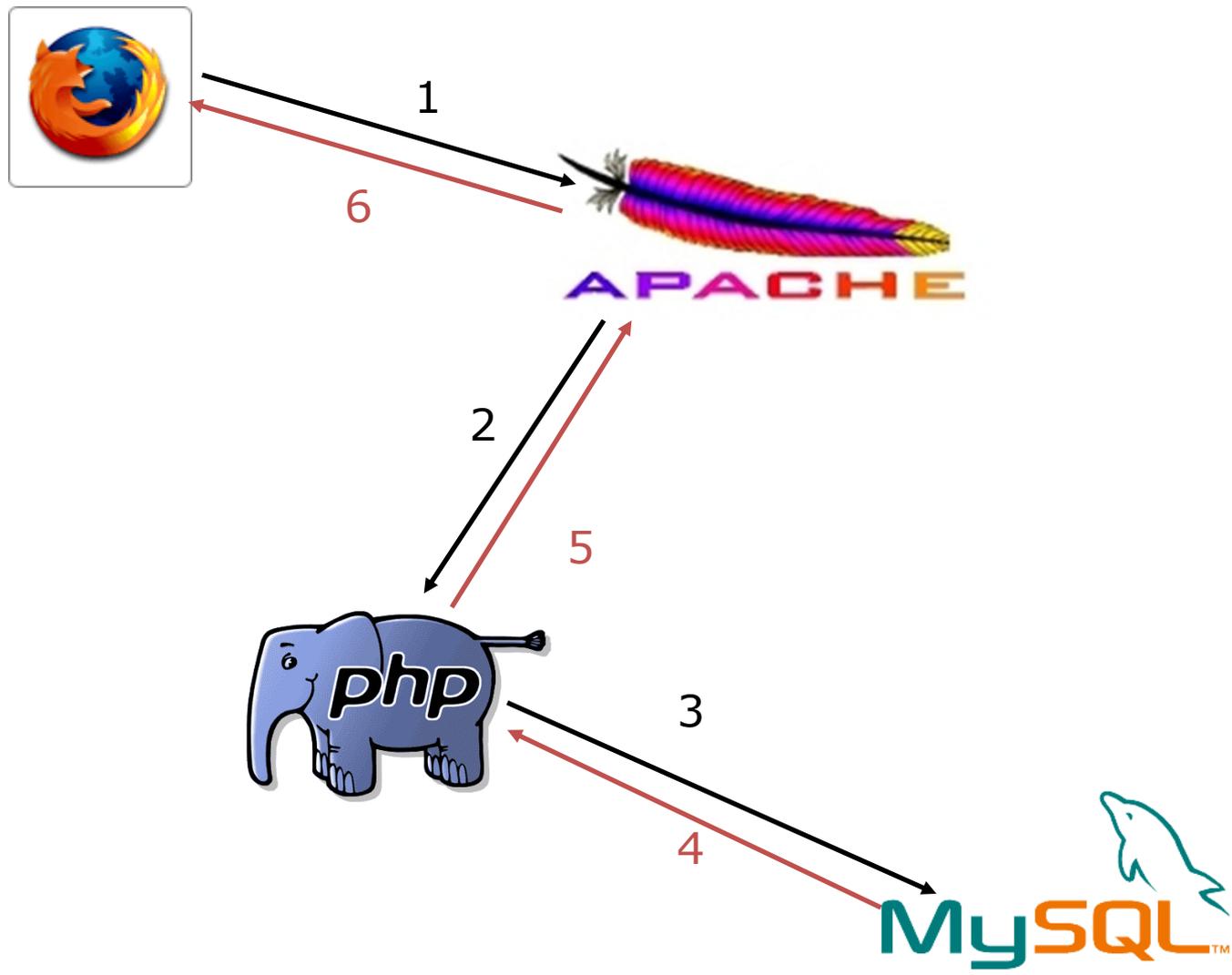
# Cliente x Servidor

- Na web, o cliente é o navegador. O navegador funciona como um interpretador
- O navegador é capaz de interpretar **apenas** HTML, Javascript e CSS!
- PHP é interpretado no servidor e não no cliente.

# Cliente x Servidor

- O servidor web interpreta a resposta, transforma a mesma em HTML e envia ao cliente;
- O cliente (navegador) interpreta o HTML e exibe na tela o resultado da interpretação.

# Cliente x Servidor



# Onde, quando e por que usar PHP?

- Websites dinâmicos;
- Facilidade de uso e de aprendizagem;
- Dupla formada pelo PHP e MySQL;
- Frameworks MVC para PHP

# PHP in Action!

# Tags

Quanto utilizamos HTML, tudo é tag!

```
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    Texto da página.
  </body>
</html>
```

# Tags

- Temos tag para o documento HTML, tag para o título da página...;
- Temos tag para colocar nosso código PHP!
- Todo o nosso código ficara entre as tags do PHP, que são `<?php ?>`.

```
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    <?php
      // Código PHP aqui!
    ?>
  </body>
</html>
```



# Sintaxe

```
<?php  
  
class Faladora {  
    public function dizerOi($nome) {  
        echo "Olá $nome! <br />";  
    }  
}  
  
$falador = new Faladora();  
$falador->dizerOi("Francisco");  
?>
```

# Sintaxe

Comando *echo*, originado da linguagem *Perl*.

Comando *print*, semelhante ao `print` do *Python*.

**ATENÇÃO!**

```
<?php  
    echo "Olá mundo!";  
    print "Olá mundo!";  
    printf("Olá mundo!");  
?>
```

Função *printf*, originada da linguagem *C*.

# Variáveis

- Variáveis são definidas utilizando um cifrão (\$);
- Em PHP, variáveis possuem tipagem dinâmica;
- Tipagem dinâmica significa que uma variável assume um tipo de acordo com o valor que lhe é atribuído;



# Variáveis

A variável n1 é do tipo *inteiro*.

```
$n1 = 10;  
$n2 = "5";
```

A variável n2 é do tipo *string*.

# Controle de fluxo

- Controle de fluxo em PHP é idêntico ao controle de fluxo em C, Java, C++, C# e qualquer linguagem com sintaxe baseada em C;
- Para controle de decisão temos o *if... else if... else* e o *switch... case*;
- Para controle de repetição, temos o *for*, o *foreach*, o *while* e o *do... while*;



# if

```
if($numero < 20){  
    echo "$numero é menor que 20";  
} else {  
    echo "$numero é maior que 20";  
}
```



```
if("condição"){  
    <faz alguma coisa quando "condição" é verdadeira>  
} else {  
    <faz alguma coisa quando "condição" é falsa>  
}
```

# Switch

Você pode usar strings!

ATENÇÃO!

```
switch ($nome) {  
  case "José":  
    echo "Olá senhor José!";  
    break;  
  case "Maria":  
    echo "Olá senhora Maria!";  
    break;  
  default:  
    echo "Olá $nome!";  
}
```



**UFRPE**

Universidade  
Federal Rural  
de Pernambuco

# While



```
while($numero < 200) {  
    $numero++;  
}
```



```
while("condição"){  
    <faz alguma coisa enquanto "condição" for verdadeira>  
}
```



# do... while

```
do{  
    $numero++;  
} while($numero < 200);
```

do {  
 <faz "alguma coisa" e verifica se "condição" é verdadeira.  
 Em seguida, repete alguma coisa enquanto  
 "condição" for verdadeira>  
} while('condição');



# for

```
for($i = 1; $i <= 80; $i++){  
    echo "$i<br />";  
}
```

for(posição inicial, "condição", passo){  
 <faz alguma coisa enquanto "condição" for verdadeira.  
}

# Estruturas de Dados em PHP

- PHP suporta apenas uma estrutura de dados: o *array/vetor*;
- O *array* do PHP consegue ser a implementação de praticamente todas as estruturas de dados: vetor, matriz, lista, pilha, fila, tabela hash, etc.;

# Array

- *Arrays* podem possuir índices numéricos (inteiros), alfanuméricos e alfabéticos (strings) ou mesmo objetos personalizados – *Arrays associativos*;
- *Arrays* podem ser acessado apenas utilizando funções, simulando listas, pilhas ou filas;



# Array

Declarando um *array*.

```
$vetor = array(1, 2, 3);  
  
foreach($vetor as $numero) {  
    echo "$numero<br />";  
}
```

```
foreach(array as variavel) {  
    <acessa cada elemento de array unicamente,  
    armazenando seu conteúdo temporariamente  
    em variavel>  
}
```



# Array

```
$vetor = array(1, 2, 3);  
  
$tamanho = sizeof($vetor);  
for($i = 0; $i < $tamanho; $i++) {  
    echo "$vetor[$i]";  
}
```



**UFRPE**  
Universidade  
Federal Rural  
de Pernambuco



# Funções

# Funções

- Definir funções em PHP é simples;
- A palavra chave que define funções é bem sugestiva: *function*;
- A sintaxe é bem declarativa.



# Funções

```
function escreverNaTela($texto) {  
    echo $texto;  
}
```

```
function maiorValor($n1, $n2) {  
    if($n1 > $n2) {  
        return $n1;  
    } else {  
        return $n2;  
    }  
}
```

# Valor padrão para argumentos

- É possível definir uma valor padrão para um ou mais argumentos de uma função;

```
function contar($limite = 100) {  
    for($i = 0; $i < $limite; $i++) {  
        echo "$i<br />";  
    }  
}
```

# Valor ou Referência?

- O PHP possui passagem de parâmetros por referências;
- Utiliza-se o & na declaração da função;
- O & também pode ser utilizado no momento em que você chama a função;

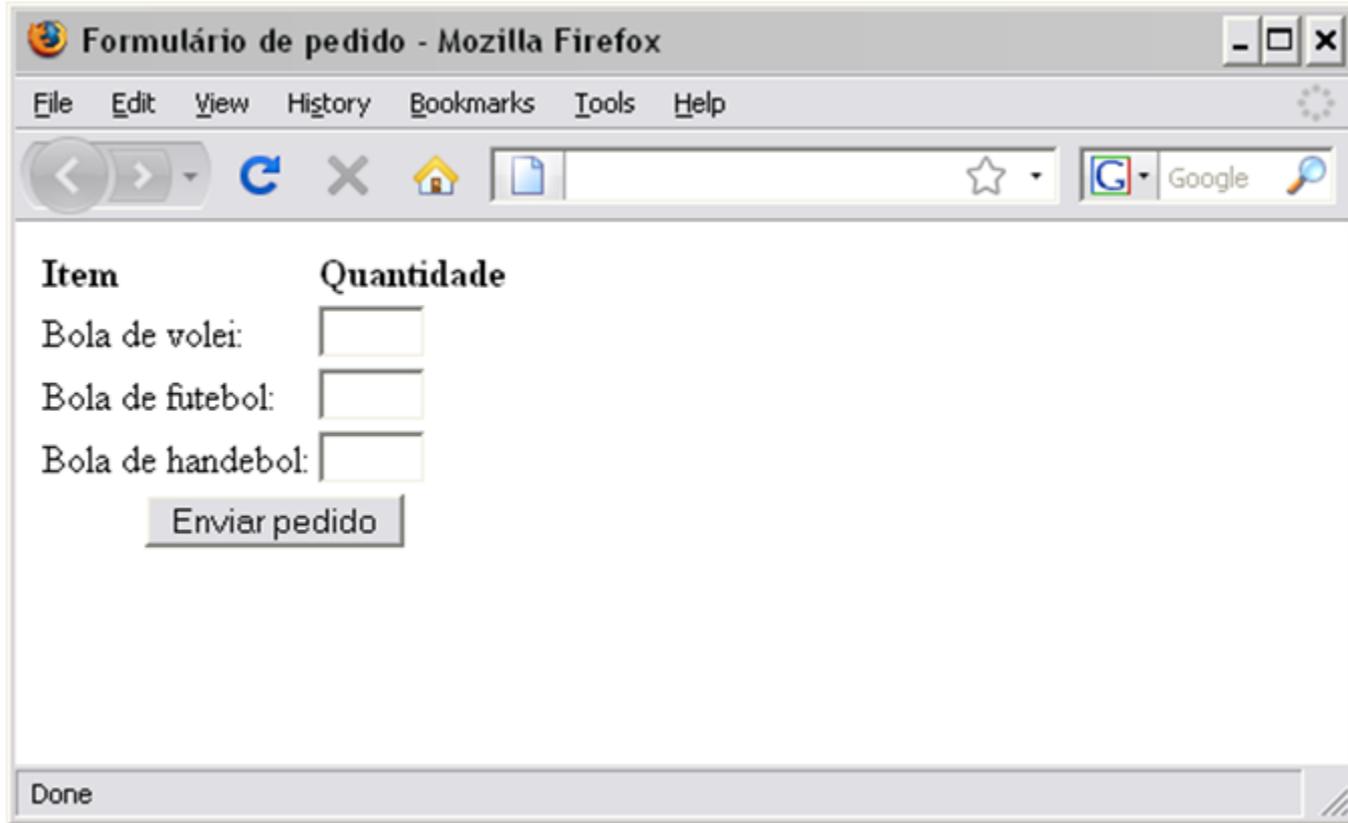
# Valor ou Referência?

```
function setarComDez (&$val) {  
    $val = 10;  
}  
  
function setarComVinte ($val) {  
    $val = 20;  
}  
  
setarComDez ($x);  
setarComVinte (&$y);
```

**Warning: Call-time pass-by-reference has been deprecated;**

# Trabalhando com Formulários

# Formulários



Formulário de pedido - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Navigation icons: Back, Forward, Refresh, Stop, Home, Search, Star, Google

Item	Quantidade
Bola de volei:	<input type="text"/>
Bola de futebol:	<input type="text"/>
Bola de handebol:	<input type="text"/>

Done

- ❑ Os dados enviados em um formulário são armazenados em um *array*!
- ❑ Você pode acessar facilmente estes dados através deste array;
- ❑ Existem três *arrays* especiais capazes de tratar estes dados enviados por formulários:
  - ❑ *\$\_GET* (para requisições do tipo GET);
  - ❑ *\$\_POST* (para requisições do tipo POST);
  - ❑ *\$\_REQUEST* (para ambos os tipos de requisição);
- ❑ Formulário enviará os dados via requisição *POST* para uma página PHP;

# Formulários

- Primeiramente, escreveremos em nossa página uma mensagem informando que o pedido foi processado;
- Depois pegaremos as quantidades informadas em nosso array  $\$_POST$  (lembre-se que fizemos uma requisição do tipo POST);
- Declararemos uma constante que representará o custo de cada uma das bolas;
- Multiplicaremos o valor da quantidade de cada bola pelo seu respectivo preço;
- Mostraremos na tela o valor total de cada item e o valor total da compra.



**UFRPE**

Universidade  
Federal Rural  
de Pernambuco

# Formulários



Formulário de pedido - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Navigation icons: Back, Forward, Refresh, Stop, Home, Search, Star, Google

Item	Quantidade
Bola de volei:	<input type="text" value="10"/>
Bola de futebol:	<input type="text" value="10"/>
Bola de handebol:	<input type="text" value="10"/>

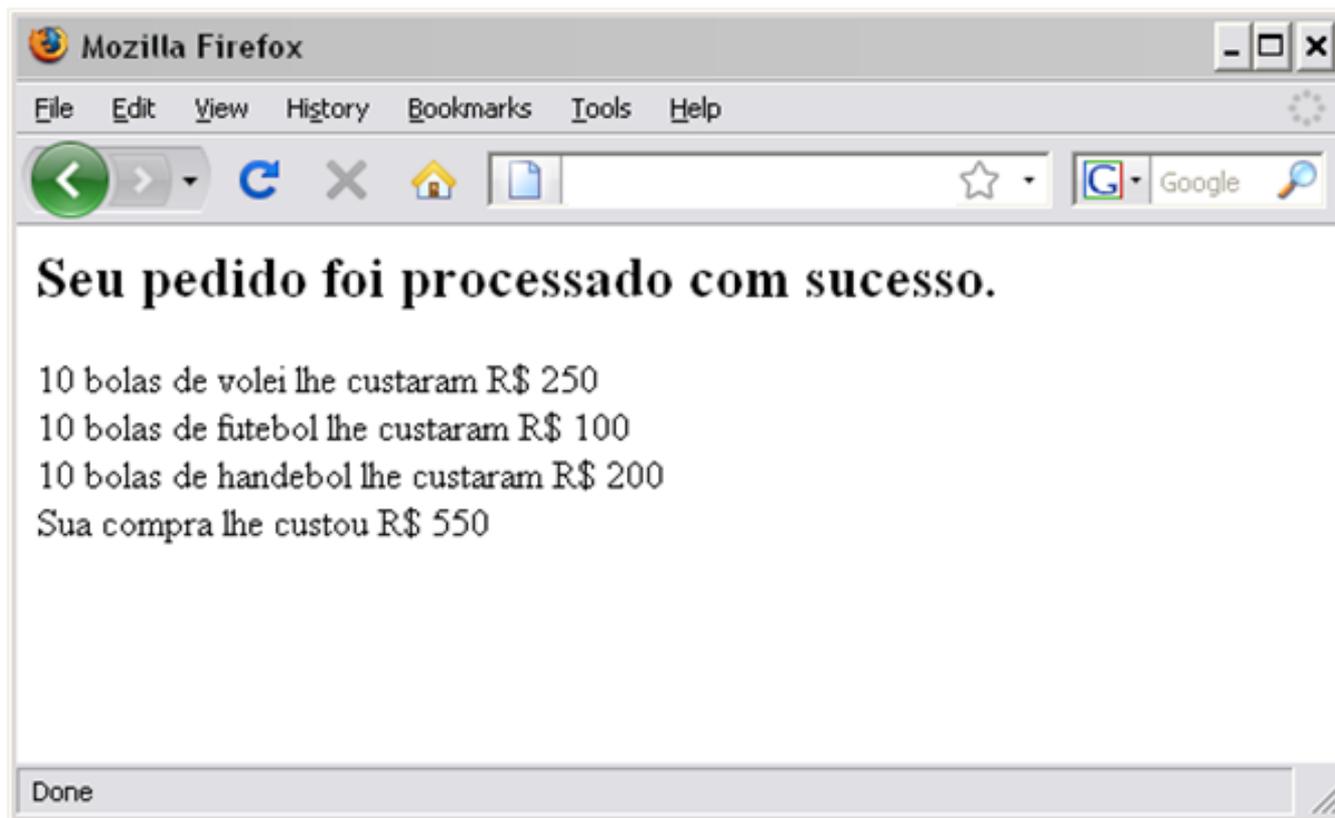
Done



**UFRPE**

Universidade  
Federal Rural  
de Pernambuco

# Formulários



```
<?php
    echo "<h2>Seu pedido foi processado com sucesso.</h2>";
    $BvQtd = $_POST['BvQtd'];
    $BfQtd = $_POST['BfQtd'];
    $BhQtd = $_POST['BhQtd'];

    define('PRECOBV',25);
    define('PRECOBF',10);
    define('PRECOBH',20);

    $valorBV = $BvQtd * PRECOBV;
    $valorBF = $BfQtd * PRECOBF;
    $valorBH = $BhQtd * PRECOBH;
    $valorTotal = $valorBV + $valorBF + $valorBH;

    echo "$BvQtd bolas de volei lhe custaram R$ $valorBV<br />";
    echo "$BfQtd bolas de futebol lhe custaram R$ $valorBF<br />";
    echo "$BhQtd bolas de handebol lhe custaram R$ $valorBH<br />";
    echo "Sua compra lhe custou R$ $valorTotal<br />";
?>
```

# Acessando Banco de Dados MySQL

# Conectando-se ao banco de dados

- Para conectar-se ao banco de dados:

```
$conexao = mysql_connect  
("localhost", "user", "senha", 3306);
```

Porta onde se  
conectará no servidor.

Servidor ao qual estamos nos conectando.

Senha do usuário.

Usuário utilizado para obter conexão.

- A função *mysql\_connect* retorna um identificador ou *false*;

```
if($conexao){  
    echo "Conectado!";  
} else {  
    echo "Não conectado!";  
}
```

- Para recuperarmos dados de um banco de dados utilizando SQL, executamos a função *mysql\_query()*;

```
$rsClientes = mysql_query("SELECT * FROM clientes");  
$rsProdutos = mysql_query("SELECT * FROM produtos");
```

```
$clientes = array();  
$i = 0;  
while($cliente = mysql_fetch_assoc($rsClientes)) {  
    $clientes[$i] = $cliente;  
    $i++;  
}  
  
$produtos = array();  
$i = 0;  
while($produto = mysql_fetch_assoc($rsProdutos)) {  
    $produtos[$i] = $produto;  
    $i++;  
}
```

# Referências

- THOMSON, Wellin. PHP e MySQL. Desenvolvimento Web. 3. ed. Rio de Janeiro: Campus, 2005. 712 p.;
- I Jornada Acadêmica de Ciência, tecnologia e Cultura;
- [www.php.net](http://www.php.net);



**UFRPE**  
Universidade  
Federal Rural  
de Pernambuco

