



Desenvolvimento de Aplicações para Internet

Aula 5

Celso Olivete Júnior

`olivete@fct.unesp.br`

`www.fct.unesp.br/docentes/dmec/olivete`



Na aula passada

- ❑ Linguagem PHP:
 - Introdução
 - tipos de dados
 - operadores
 - estruturas de controles
 - funções.

Na aula de hoje

- ❑ Reutilização de códigos
- ❑ Interação → formulários x processamento servidor (PHP)
- ❑ Manipulação de banco de dados





Reutilização de códigos

- ❑ Situação 1: Imagine que um portal possua 50 páginas (no formato htm, html, php,...) e todas elas apresentam a mesma estrutura (cabeçalho, rodapé, área de apresentação, menu, ...) conforme exemplo



Reutilização de códigos

The screenshot shows a website header with the unesp logo and the text "UNIVERSIDADE ESTADUAL PAULISTA 'JÚLIO DE MESQUITA FILHO' Campus de Presidente Prudente". Below the header is a navigation menu with links for "Home", "Compiladores 2012-1", "Programa do Curso", "Material Didático", "Trabalhos Práticos", "Projeto", and "Notas". The footer contains the copyright notice "© Celso Olivete Júnior", the contact email "Contato: olivete@fct.unesp.br", and the address "DMaC - Departamento de Matemática e Computação, Rua Roberto Simonsen, 305, CEP: 19060-900, Presidente Prudente - SP".

- ❑ Como pode ser observado as áreas de cabeçalho e rodapé são as mesmas para todas as páginas
 - ❑ Imagine se fosse necessário fazer uma alteração no rodapé – seria necessário abrir todos os arquivos e alterar um a um



Reutilização de códigos

- ❑ Uma forma de viabilizar esse processo, neste caso, é criar um arquivo (html, php) de cabeçalho, rodapé e até mesmo um que representa o menu para utilizá-los em todas as páginas. Dessa forma, você deverá:
 - ❑ Identificar no código HTML onde está a **<div>** de cabeçalho
 - ❑ Retirar este código HTML da página, criar um novo arquivo e adicionar o respectivo código.



Reutilização de códigos

- ❑ Situação 2: Imagine que seu site possua 50 páginas, e existe uma ou mais **funções** que é utilizada em todas elas. Você não precisa repetir o código dessa função em cada uma das páginas.
 - ❑ Utilizar uma forma de reutilização de código



Reutilização de códigos

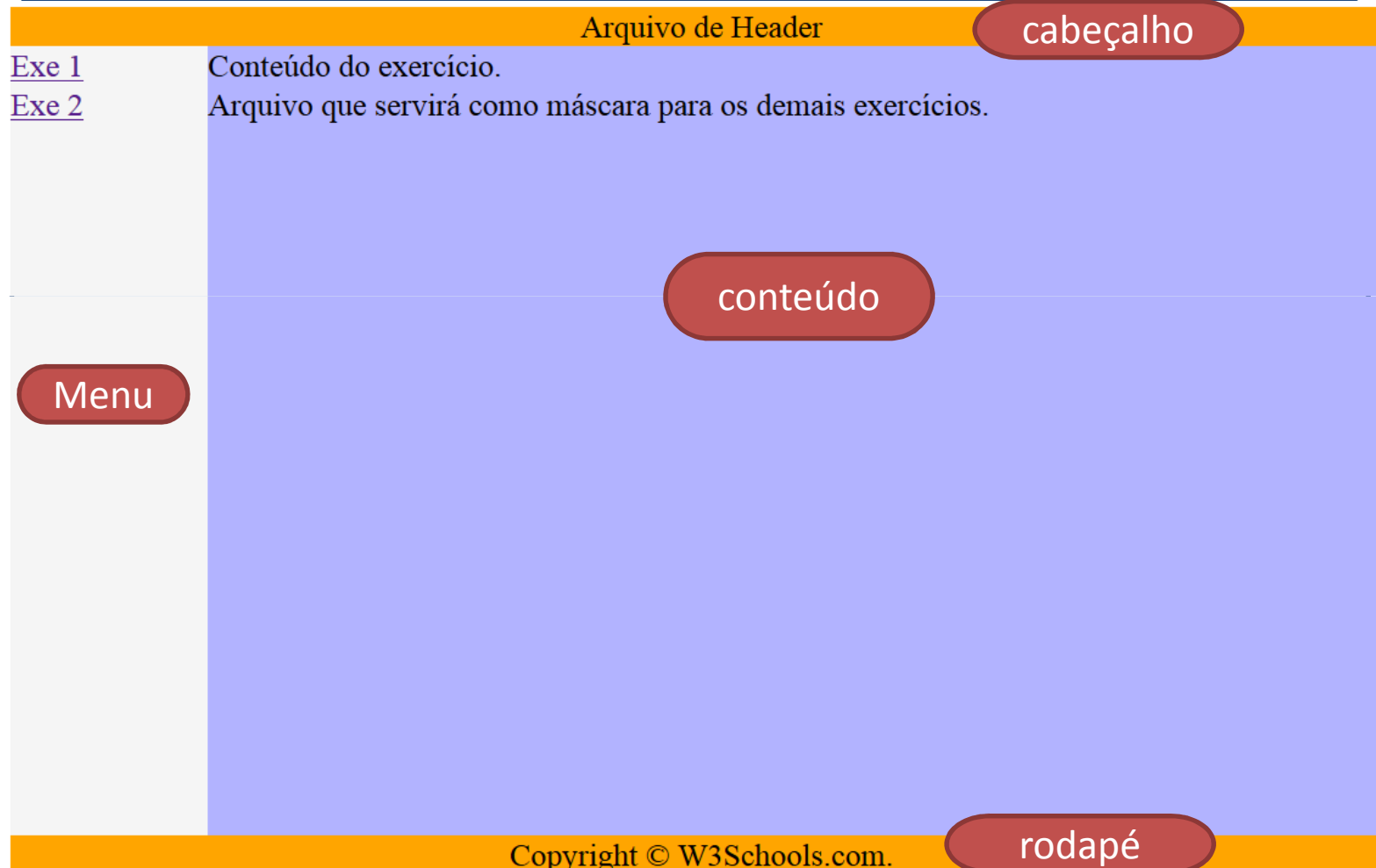
❑ Formas de reutilização:

- **include():** Tenta incluir uma página. Caso dê algum erro, o *script* retorna um *warning* (aviso) e prossegue com a execução do script. Aceita a passagem de variáveis (GET) na string. Pode ser utilizado em uma estrutura de condicionais que terá seu efeito propagado todas as vezes que for chamado.
- **require():** Tenta incluir uma página. Caso dê algum erro, o *script* retorna um *fatal_error* e aborta a execução do *script*. Não aceita a passagem de variáveis (GET) na string.
- **include_once() e require_once():** Idênticas às suas funções simples, porém se o arquivo referenciado já foi incluído na página anteriormente, a função retorna 'false' e o arquivo não é incluído.



Exemplo

Estrutura principal





Definindo a estrutura CSS

```
.menu{  
background-color: #F5F5F5;  
height: 400px;  
width: 100px;  
float: left;  
}
```

Menu

```
.content{  
background-color: #B2B3FF;  
height: 400px;  
width: 600px;  
float: left;  
}
```

conteúdo

```
.footer{  
background-color:#FFA500;  
clear:both;  
text-align:center;  
width:700px;  
}
```

rodapé

```
.header{  
background-color:#FFA500;  
width:700px;  
text-align:center;  
}
```

cabeçalho



Definindo os arquivos

menu.php

Menu

```
<?php  
echo '<div class="menu" >  
    <a href="exe1.php" title="Exe 1"> Exe 1</a> <br>  
    <a href="exe2.php" title="Exe 2"> Exe 2</a> <br>  
</div>';  
?>
```



Definindo os arquivos footer.php

rodapé

```
<?php  
echo '  
  <div class="footer">  
    Copyright © W3Schools.com.  
  </div>';  
?>
```



Definindo os arquivos

header.php

cabeçalho

```
<?php  
echo '  
  <div class="header">  
    Arquivo de Header  
  </div>';  
?>
```

```
<?php
echo '
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="css/estilos.css">
</head>
<body>';
    include('includes/header.php');
    include('includes/menu.php');
echo '
<div class="content">
    Conteúdo do exercício. <br> Arquivo que servirá como
    máscara para os demais exercícios.
    //incluir o conteudo de cada script aqui..
</div> ';
    include('includes/footer.php');
echo '
</body>
</html>
';
?>
```

Todo conteúdo do html
deve vir dentro do
comando *echo*

Arquivo
padrão

Arquivo de
configuração

Carrega os
arquivos php
header e menu

Todo conteúdo da
página deve ser
incluído aqui...

Carrega o arquivo
de rodapé

Exercício 1

```
<?php
  echo '
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="css/estilos.css">
</head>
<body>;
  include('includes/header.php');
  include('includes/menu.php');
echo '
<div class="content">
  Exercicio 1 - Usando comando FOR';
  for ($i=0;$i<10;$i++)
    echo '<br> Valor do i = ' . $i;
echo '
</div> ';
  include('includes/footer.php');
echo '
</body>
</html>
';
?>
```



Reutilização de códigos

❑ Exemplo de utilização

- ❑ Carregando um arquivo de cabeçalho

```
<?php
```

```
....
```

```
include ("cabecalho.html");
```

```
....
```

```
?>
```


❑ Exemplo de utilização

❑ Carregando um arquivo de funções

```
<?php
    include("funcoes.php");
    $x=10;
    $y=20;
    $res = soma($x,$y);
    echo "resultado da soma:... ".$res;
    $res = dif($x,$y);
    echo "resultado da subtração:...
    ".$res;
?>
```

```
funcoes.php
<?php
function soma($a,$b)
{
    return $a+$b;
}

function dif($a,$b)
{
    return $a-$b;
}

?>
```



Reutilização de códigos

- Pode-se referenciar qualquer arquivo, utilizando caminhos relativos ou absolutos. Pode-se incluir arquivos do tipo txt, htm, html, inc, php, css, jsp...



Formulários

- ❑ Um formulário consiste em uma série de campos que são preenchidos pelo usuário e recebidos pelo servidor web para serem processados.
- ❑ Toda seção de uma página que contenha um formulário deve sempre estar entre as tags `<form>...</form>`
- ❑ A tag `<form>` contém uma série de atributos, sendo que os mais importantes são:



Formulários

- ❑ **Action:** Informa ao servidor a página que deve ser carregada assim que o usuário clicar sobre o botão “enviar” contido no formulário.
- ❑ **Method:** Controla como as informações contidas no formulário devem ser enviadas ao servidor. Este procedimento pode ser executado de duas maneiras diferentes: uma usando o método **GET** e a outra usando o método **POST**
 - ✓ O método **GET** os dados enviados fazem parte do URL (endereço) associado à consulta enviada para o servidor. Suporta até 128 caracteres. Ex: `www.xxx.com.br?nome=teste`
 - O método **POST** transmite essas informações de modo invisível, ou seja, não as insere na URL, mas sim no corpo da requisição do HTML, além de permitir uma quantidade maior de informações a ser enviada de uma página para outra.

EX: `<FORM METHOD="POST" action = "pagina.html">`



Formulários

- ❑ Principais controles do HTML que são usados na coleta de dados em um formulário:
 - Caixa de texto de uma linha (Text)
 - Caixa de texto de rolagem (TextArea)
 - Caixa de seleção (CheckBox)
 - Botão de opção (Radio)
 - Menu suspenso (Select e Option)
 - Botão de envio (Submit)



Formulários

- Pode conter qualquer formatação - parágrafos, listas, tabelas, imagens - exceto outros formulários
- Geralmente utiliza-se `<INPUT>`, `<SELECT>` e `<TEXTAREA>`.
- **OBS:** todos os campos de entrada devem ter um atributo `NAME` → que será utilizado posteriormente pelo *script*



Formulários

☐ Exemplo:

TIPO TEXTO (1)

Nome: `<INPUT TYPE=TEXT NAME="Nome">`

Produzirá...

Nome:



Formulários – interação com o servidor

```
<form method="POST" action="obter_dados.php">  
Seu nome <input type="text" name="campo1">  
<input type="Submit" value="Enviar" name="b_enviar">  
</form>
```

Aula 4

Seu nome teste

Enviar

❑ Obtendo valores dos campos do formulário...

```
<?php  
    $x=$_POST["campo1"];  
    echo "Nome enviado:... ".$x;  
?>
```




Formulários – interação com o servidor

```
<form method="POST" action="obter_dados.php">  
Seu nome <input type="text" name="campo1">  
<input type="Submit" value="Enviar" name="b_enviar">  
</form>
```

Aula 4

Seu nome teste

Enviar

❑ Obtendo valores dos campos do formulário...

```
<?php  
  $x=$_POST["campo1"];  
  echo "Nome enviado:... ".$x;  
?>
```

Formulários

❑ `<input>` → **text**

Aula 4

Seu nome

- ❑ Os valores são acessados a partir do tipo do atributo **method** do `<form>` juntamente com o nome do campo. No exemplo anterior:

```
<form method="POST" action="obter_dados.php">  
Seu nome <input type="text" name="campo1">  
<input type="Submit" value="Enviar" name="b_enviar">  
</form>
```

- ❑ Neste caso, o conteúdo do input (text) é acessado da seguinte maneira:

```
<?php  
$x=$_POST["campo1"];  
echo "Nome enviado:... ".$x;
```

D ?>

net

Formulários

Aula 4

Seu nome

❑ `<input>` → **text**

- ❑ antes de recuperar o conteúdo de um campo, é interessante verificar se o mesmo foi preenchido.

Para isso, utiliza-se a função **empty**

```
<?php
    if (!empty($_POST["campo1"]))
        $x=$_POST["campo1"];
    else $x="vazio";
    echo "Nome enviado:... ".$x;
?>
```

❑ `<input>` → **checkbox**

Esporte preferido	<input type="checkbox"/> Futebol
	<input checked="" type="checkbox"/> Basquete

- ❑ Campo de escolhas, onde é possível escolher mais de uma opção
- ❑ O atributo **name** deve ser um vetor e, valor retornado se a opção for selecionada é o mesmo do atributo

```
<INPUT TYPE="CHECKBOX" NAME="chk_esportes[]" VALUE="futebol">Futebol <br>  
<INPUT TYPE="CHECKBOX" NAME="chk_esportes[]" VALUE="basquete">Basquete</td>
```

Formulários

```
<INPUT TYPE="CHECKBOX" NAME="chk_esportes[]" VALUE="futebol">Futebol <br>  
<INPUT TYPE="CHECKBOX" NAME="chk_esportes[]" VALUE="basquete">Basquete</td>
```

❑ Acessando valores selecionados...

❑ `<input>` → **checkbox**

Esporte preferido	<input type="checkbox"/> Futebol
	<input checked="" type="checkbox"/> Basquete

```
if(!empty($_POST["chk_esportes"]))  
{  
    $j=count($_POST["chk_esportes"]);  
    $dados = $_POST["chk_esportes"];  
    for ($i=0;$i<$j;$i++)  
        echo $dados[$i];  
}
```

Verifica se algum
item foi selecionado

Conta quantos
elementos tem no
vetor

Imprime os
elementos
selecionados



Formulários

- ❑ `<input>` → **radio**: insere um botão de escolha de valores para uma **única** opção. **NAME** deve ser o mesmo para todos os itens

```
<INPUT TYPE=RADIO NAME="sexo" VALUE="masc" CHECKED>Masculino <br>
```

```
<INPUT TYPE=RADIO NAME="sexo" VALUE="fem">Feminino
```

- ❑ O item selecionado é recuperado de forma semelhante à um campo **text**. O valor exibido será o conteúdo do atributo **value**



Formulários

- ❑ `<TEXTAREA>` abre uma área para entrada de texto de acordo com atributos para número de colunas/linhas e, se for o caso, um valor inicial.
- ❑ Recuperação do conteúdo idêntica ao tipo **text**



Formulários

- ❑ `<TEXTAREA>` abre uma área para entrada de texto de acordo com atributos para número de colunas/linhas e, se for o caso, um valor inicial.
- ❑ Recuperação do conteúdo idêntica ao tipo **text**

❑ Formulário **<SELECT>**

- **<SELECT>** apresenta uma lista de valores (menu suspenso) através de campos **OPTION**.

```
<SELECT NAME="sabor">
```

```
<OPTION VALUE = "abacaxi">Abacaxi </OPTION>
```

```
<OPTION VALUE="creme" SELECTED >Creme </OPTION>
```

```
<OPTION VALUE = "morango">Morango </OPTION>
```

```
<OPTION VALUE = "chocolate">Chocolate </OPTION>
```

```
</SELECT>
```

- ❑ Recuperação do conteúdo idêntica ao tipo **text**, ou seja, a partir do atributo **name**



Exercício

1. Modifique seu portal fazendo o uso de reutilização de código (por exemplo: no cabeçalho, rodapé e funções)
2. Crie um formulário com os itens vistos na aula de hoje.
3. A exibição dos dados deve ser em um formulário idêntico ao de entrada de dados, ou seja, após o usuário escolher/digitar os dados e clicar no botão *submit* você deverá redirecionar para um formulário e alterar o atributo **value** no caso de um campo do tipo **text**. Exemplo:

```
echo '<input type="text" name="campo1" value=".' . $_POST["campo1"] . ' readonly> ';
```



Na próxima aula

- Banco de dados MySQL
 - Conectando com o servidor MySQL
 - Selecionando um banco de dados
 - Manipulando informações no BD (inserir, excluir, selecionar, exibir e apagar um registro).
 - Fechar conexão com o BD



Conectando com o servidor MySQL

☐ Função

➤ `mysql_connect(servidor, usuario, senha) or die`
("A conexão com o servidor falhou!");

✓ Servidor:

✓ se for local: localhost

✓ se estiver publicado na Internet, localhost será substituído por um endereço IP

Crie um arquivo php com funções para manipular o BD

Servidor local

usuário

Sem senha

```
$conexao = mysql_connect("localhost", "root", "")  
or die ("A conexão com o servidor falhou!");
```



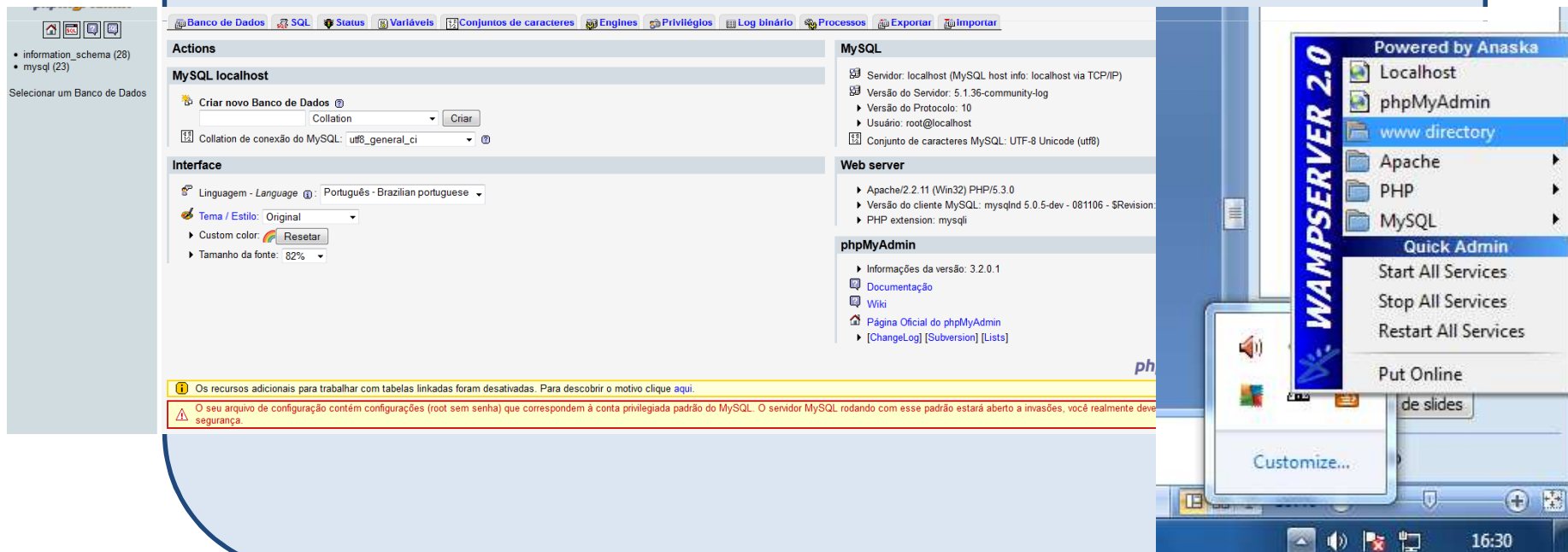
Conectando com o servidor MySQL

```
//no arquivo de funções.... funcoesBD.php
function conecta()
{
    $conexao = mysql_connect("localhost", "root", "")
                or die ("A conexão com o servidor falhou!");
    return true;
}

//no arquivo de chamada
...
include("funcoesBD.php");
if (conecta())
    echo "conectado ao servidor mysql";
...
```

Selecionando um BD

- ❑ Para criar um BD utilize o PhpMyAdmin do próprio Wamp (ou qualquer outro gerenciador...)



The image shows two overlapping screenshots. The background screenshot is the phpMyAdmin interface for MySQL localhost. It features a top navigation bar with tabs like 'Banco de Dados', 'SQL', 'Status', 'Variáveis', 'Conjuntos de caracteres', 'Engines', 'Privilegios', 'Log binário', 'Processos', 'Exportar', and 'Importar'. The main content area is divided into sections: 'Actions' with a 'Criar novo Banco de Dados' button; 'Interface' with language and theme settings; and a status bar at the bottom with warning messages. The foreground screenshot is the WampServer 2.0 control panel, titled 'Powered by Anaska'. It displays a tree view of services: 'Localhost', 'phpMyAdmin', 'www directory', 'Apache', 'PHP', and 'MySQL'. Below this is a 'Quick Admin' section with buttons for 'Start All Services', 'Stop All Services', 'Restart All Services', and 'Put Online'. A 'Customize...' button is also visible at the bottom of the control panel.



Selecionando um BD

- ❑ Função
 - ❑ `$db = mysql_select_db("nome_do_banco")`
`or die ("Não foi possível selecionar o banco de dados!");`
 - ❑ Acrescente essa função no arquivo de funções...



Manipulando um BD

- ❑ **instrução SQL:**
- ❑ Uma vez conectado ao servidor e selecionado o BD, devemos criar uma instrução SQL para a manipulação de suas tabelas, como por exemplo, visualizar seus registros.

```
$sql = "SELECT * FROM livros ORDER BY titulo";
```




Manipulando um BD

- ❑ instrução SQL:
- ❑ Para executar uma instrução SQL utiliza-se a função `mysql_query`. Exemplo:

```
$rs = mysql_query($sql)  
      or die ("Não foi possível efetuar a consulta!");
```



Manipulando um BD

- ❑ Formatando o resultado obtido:
- ❑ Para que os campos de uma tabela possam ser corretamente exibidos, é necessário separarmos seus registros por linha, usando uma função interna do MySQL, denominada `mysql_fetch_array()`. Exemplo:

```
$linha = mysql_fetch_array($rs);  
echo $linha['nome'];  
echo $linha['endereco'];
```



Manipulando um BD

- ❑ No caso de uma instrução SQL do tipo **insert...**

```
$SQL = "INSERT INTO produtos (codigo_produto, nome_produto)  
VALUES (1, 'Camiseta')";
```

```
$insere = mysql_query($SQL)  
or die ("Não foi possível inserir o registro!");
```



Manipulando um BD

- ❑ No caso de uma instrução SQL do tipo **delete**...

```
$SQL = "DELETE from produtos  
      where codigo_produto = '$_POST[cd_enviado].' ";
```

```
$apaga= mysql_query($SQL)  
      or die ("Não foi possível apagar o registro!");
```

- ❑ O mesmo é valido para a instrução **alter table**...



Manipulando um BD

- ❑ Para finalizar é necessário fechar a conexão com o BD mysql
- ❑ Função **mysql_close(nome_variavel_de_conexao);**
- ❑ Exemplo:

```
mysql_close($db);
```



Exercícios BD

- Crie um banco de dados e manipule todas as instruções: inserção, alteração, exclusão, exibir todos os registros e localizar um registro