



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
CAMPUS PARNAÍBA

# PHP - MYSQL

Athanio Souza

[www.ifpi.edu.br](http://www.ifpi.edu.br)



## Softwares necessários

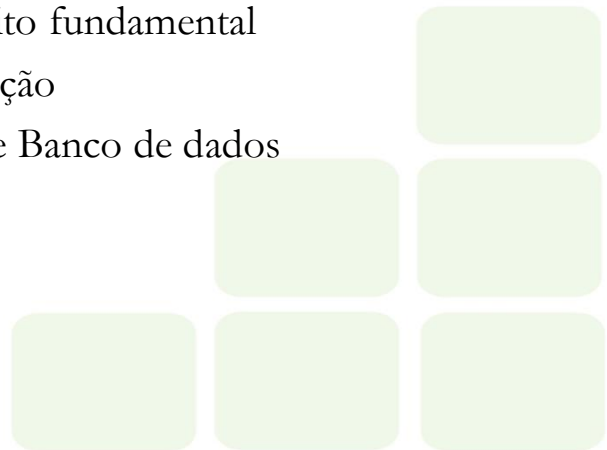
- Servidor WWW - Apache
- PHP
- Banco de dados MySql



## Notas Importantes

---

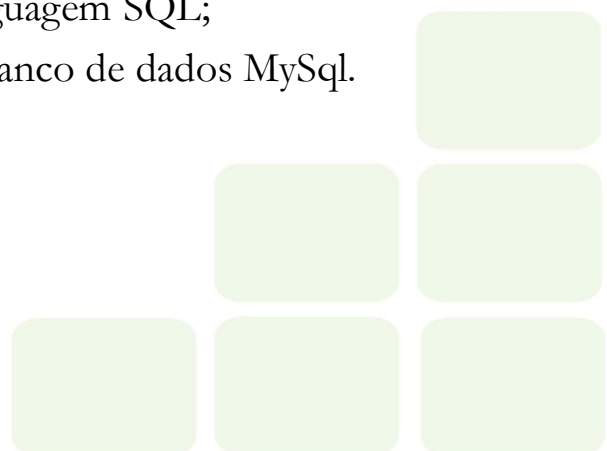
- HTML – Um pré-requisito fundamental
- Linguagem de Programação
- Conhecimento Básico de Banco de dados



## Programação

---

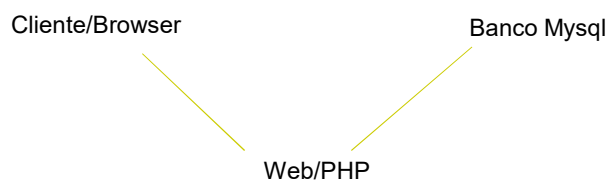
- Conceitos básicos de programação PHP;
- Conceitos básicos da linguagem SQL;
- Integrar o PHP com o Banco de dados MySql.





## Visão geral

### Interação dos softwares



## Fundamentos do PHP

- O que vem a ser PHP ?
- O que pode ser feito com o PHP ?
- Quanto custa o PHP ?
- Como surgiu o PHP ?
- PHP x JavaScript
- PHP x ASP



## O que vem a ser PHP ?

- O PHP (Hiptertext Preprocessor) é uma linguagem de script voltada para o desenvolvimento de páginas dinâmicas para a internet de fácil uso e aprendizagem, permitindo ao programador inserir seus comandos diretamente nos códigos HTML de uma página.
- O PHP oferece suporte de conexão com diversos bancos de dados, tais como: MySql, Oracle, Sybase, Dbase, PostgreSQL etc.
- Outra grande vantagem ao ser comparado com outras linguagens, tais como: ASP e ColdFusion , é a possibilidade de ser executado em diversos sistemas operacionais (Multiplataforma) como Linux, Windows, Unix, OS/2, Macintosh, NT etc., além de permitir mudanças de plataforma com nenhuma ou pouquíssimas alterações em seus códigos fontes.



## Quanto custa o PHP ?

- O PHP é distribuído sobre GPL ( Licença Pública Geral), ou seja você não precisa pagar nada para usá-lo.



## O que pode ser feito com o PHP ?

- Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feita também com PHP, como coletar dados de um formulário, gerar páginas dinamicamente .
- PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP.
- Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP. Ainda é possível abrir *sockets* e interagir com outros protocolos.



## Como surgiu o PHP ?

- Em 1994, Rasmus Lerdorf desenvolveu algumas ferramentas que tinham por objetivo monitorar os visitantes de seu site por intermédio de um contador de acesso, um livro de visitas e outros pequenos aplicativos. Era o início do PHP Personal Home Page tools (ferramentas para home page pessoal).
- Um ano depois, Lerdorf aprimorou sua linguagem e incorporou ao PHP um pacote para interpretação de formulários (FI – Form Interpretation) e algumas ferramentas para acesso a banco de dados, passando o PHP a ser conhecido como PHP-FI
- Democráticamente, Lerdorf abriu o código-fonte de sua linguagem, dando início a uma verdadeira avalanche de novos contribuintes ao PHP-FI.
- Em 1997, o PHP-FI ganhou dois novos colaboradores de peso, Zeev Suraski e Andi Gutmans, que reescreveram o interpretador, criando a versão 3.0. Deste ponto em diante, o PHP3 deixa de ser Personal Home Page Tools para se transformar em Hipertext Preprocessor.
- Em maio de 2000, desembarca no mercado a versão 4 do PHP com muitas novidades, como suporte a sessões, permitindo a identificação de usuários que solicitam informações em um site.



## PHP x JavaScript

- Uma das diferenças marcantes entre o PHP e o JavaScript consiste em que os códigos PHP são processados no servidor, sendo apenas enviado ao cliente o HTML puro, ou seja o usuário não tem acesso aos scripts embutidos no HTML, como ocorre com o JavaScript que processa os dados no computador do cliente. Este procedimento pode ser de grande valia, quando estamos lidando com dados confidenciais, senhas etc.



## PHP x ASP

- O PHP é multiplataforma, podendo ser executado em quase todos os sistemas operacionais, enquanto ASP somente é executado em sistemas operacionais da Microsoft. Também podemos destacar que o acesso à banco de dados por intermédio do PHP é feito de forma mais simples do que o ASP.
- Com relação a velocidade, o PHP é comprovadamente mais rápido que o ASP, além de possuir um gerenciamento de memória superior ao do ASP.



## O que são páginas estáticas

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<b> Olá! Hoje é 06/06/2016 </b>
</BODY>
</HTML>
```



## O que são páginas dinâmicas

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<b> Olá! Hoje é </b>
<?
    echo date("d/m/Y");
?>
</BODY>
</HTML>
```



## PHP – Sintaxe

- Um bloco de script PHP começa com **<?php** ou **<? e termina com ?>**
- As instruções são finalizadas com **;**

```
<html>
  <body>
    <?php
      echo "Hello World";
    ?>
  </body>
</html>
```



## Iniciando o PHP

Sintaxe básica para um programa em PHP

- O código PHP fica embutido no próprio HTML. O interpretador identifica quando um código é PHP pelas seguintes tags:

```
<?php
Comandos;
?>

<script language="php">
Comandos;
</script>

<?
comandos
?>

<%
comandos
%>
```







## Comentário de uma única linha

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<b> Olá! Hoje é </b>
<? // A linha seguinte exibe a data atual do servidor
  echo date("d/m/Y") ;
?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Comentários com mais de uma linha

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<b> Olá! Hoje é </b>
<? /* Esta linha inicia o comentário
    o script seguinte é a data atual */
  echo date("d/m/Y") ;
?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



### Como exibir texto em um navegador

Em PHP usamos a instrução echo seguida pelo texto a ser exibido entre aspas.

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<?
    // Exibindo texto
    echo “IFPI– Ensino, Pesquisa e Extensão” ;
?>
?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



### Como enviar códigos HTML dentro de um script PHP

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
    <? // Inserindo tags HTML
    echo “<b> Oi pessoal </b> Bem-vindo ao curso PHP <br>” ;
    echo “<b> <font color = \”#FF0000\”> Olá , amigo!</b></font> Bem-vindo .”;
    ?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



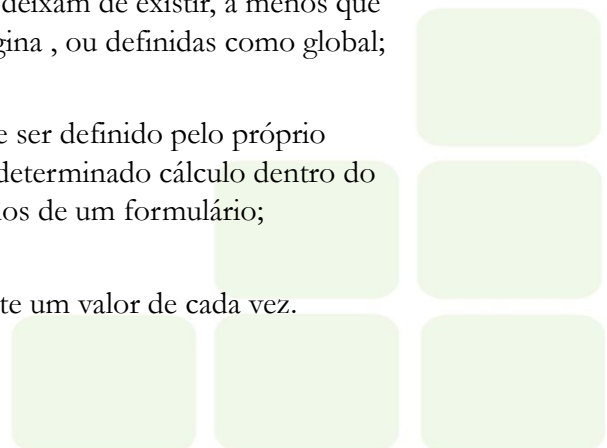
## Exercício

- 1 – Utilizando o PHP monte uma pagina e escreva na tela seu nome, curso e a instituição que estuda um abaixo do outro;
- 2 – Mostre a data atual com tag em HTML;
- 3 – Some 3 dias na data atual.



## VARIÁVEIS

- As variáveis contidas em um programa PHP são voláteis e somente existem enquanto um script estiver sendo executado, ou seja, ao se mudar de uma página para outra elas deixam de existir, a menos que sejam transferidas para a próxima página, ou definidas como global;
- O valor atribuído a uma variável pode ser definido pelo próprio programador, pelo resultado de um determinado cálculo dentro do programa ou por uma entrada de dados de um formulário;
- Uma variável pode armazenar somente um valor de cada vez.





## Variáveis

- No PHP as variáveis começam com o símbolo **\$**
- Para criar uma variável basta que façamos **\$variavel = valor;**
- O PHP é uma linguagem sem tipo para as variáveis, ou seja, dispensa explicitar com que dados iremos tratar.

### Exemplo

```
<?php
$txt = "Hello World!";
$num = 16;
?>
```

- Para cada tipo de dado (String, Inteiro, Float, Bool) existem inúmeras funções para se trabalhar.

```
= "Iabc";
2 = 1;
```



## Como dar nome a variável

- Por ser esta linguagem case-sensitive (sensível a letras maiúsculas e minúsculas), sendo “A” (maiúsculo) diferente de “a” (minúsculo), assim uma variável de nome Contador é diferente de contador.
- Em PHP todas as variáveis são obrigatoriamente iniciadas com o símbolo \$ (cifrão), seguido pelo seu nome.
- Exemplos válidos : \$contador, \$\_autor, \$VALOR
- Exemplos não válidos : \$5Valor, \$titulo livro, \$segundo.nome



## Atribuição de valor

- Para atribuir um valor a uma variável, você deve usar o sinal de =, seguido pelo valor a ser armazenado.
- Os valores alfanuméricos (strings) a serem armazenados devem estar sempre entre aspas (“”)
- Valores numéricos são armazenados sem o uso das aspas.
- Em PHP você não precisa definir uma variável para usá-la, como é comum em outras linguagens, basta atribuir um valor diretamente a ela, e a partir deste instante ela será automaticamente criada.
- Exemplos:

\$titulo = “Programando em PHP”;

\$precolibro = 129.30;



## PHP com uso de variáveis

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
```

```
<? // variáveis
$A = 5;
$B = “Estou trabalhando na universidade há 18 anos”;
echo “O valor da variável A é : $A” ;
echo “<br>”;
echo “ O valor da variável B é : $B”;
?>
```

```
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Tipo de dados

### ■ Numéricos

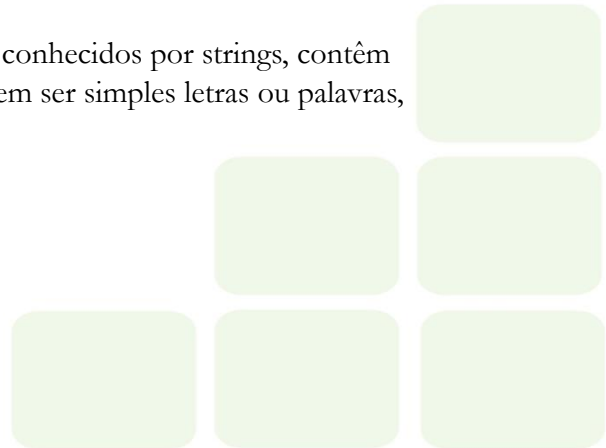
- Numéricos inteiros ou absolutos e números fracionários, também conhecido como ponto flutuante.

### ■ Alfanuméricos

- Os dados alfanuméricos, também conhecidos por strings, contêm uma cadeia de caracteres que podem ser simples letras ou palavras, até frases inteiras

#### Exemplos:

- \$A = 25;
- \$B = -1234;
- \$C = 78.56;
- \$D = -9.3245;
- \$E = “ Olá, como vai? “;
- \$F = “ 78.999”;



## Exercícios

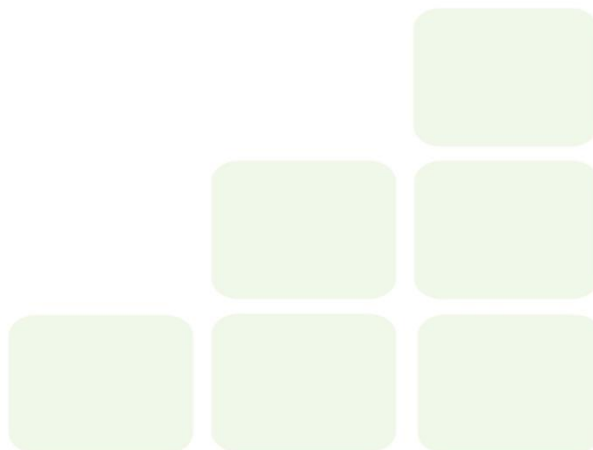
- 2 – Utilizando o PHP monte uma pagina que receba um valor numa variável e diga se é par ou impar.
- Dica: Utilize comando IF
- E o sinal %





## Operadores Aritméticos

- +** Adição
- Subtração
- \*** Multiplicação
- /** Divisão
- %** Resto da divisão



## Operadores

- Matemática:  $x = 2;$

Operador	Descrição	Exemplo	Resultado
+	Adição	$x+2;$	4
-	Subtração	$5-x;$	3
*	Multiplicação	$x*5;$	10
/	Divisão	$5/x;$	2.5
%	Mod (resto da divisão)	$5\%x$	1
++	Incremento	$x++$	3
--	Decremento	$x--$	1





## Como usar os operadores

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
```

```
<? // usando operadores aritméticos
$A = 30;
$B = 2;
$C = $A + $B;
$D = $A / $B;
$A = $A * 3;
echo "O valor da variável C é : $C <br>" ;
echo " O valor da variável D é : $D <br>";
echo " O valor da variável A é : $A <br>";
?>
```

```
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Prioridade dos operadores

A prioridade adotada nos cálculos em PHP segue de acordo com as mesmas regras matemáticas, ou seja, os cálculos são executados de acordo com a seguinte ordem:

*Parênteses => divisão => Multiplicação => Adição => Subtração*





## PHP – Operadores

### ■ Atribuição

Operador	Exemplo	O mesmo que:
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

33



## PHP – Operadores

### ■ Comparação

Operador	Descrição	Exemplo
==	É igual a	5==8 retorna false
!=	Não é igual a	5!=8 retorna true
>	É maior que	5>8 retorna false
<	É menor que	5<8 retorna true
>=	É maior ou igual a	5>=8 retorna false
<=	É menor ou igual a	5<=8 retorna true

34



## PHP – Operadores

### ■ Lógicos

Operador	Descrição	Exemplo
&&	E (and)	x=6; y=3; (x < 10 && y > 1) retorna true
	OU (or)	x=6; y=3; (x==5    y==5) retorna false
!	NÃO (not)	x=6; y=3; !(x==y) retorna true

35

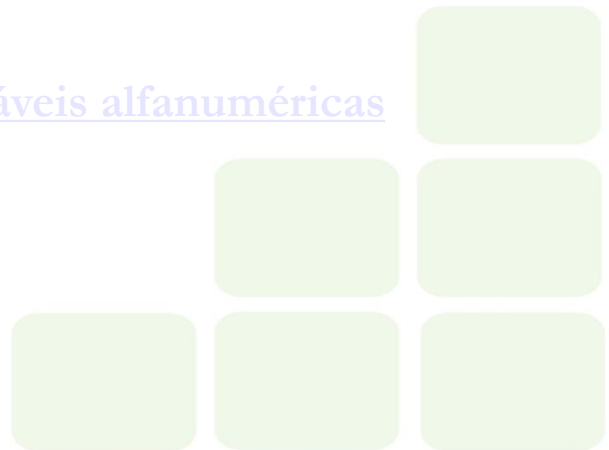


## Exercícios

- 3 – Utilizando os operadores matemáticos escreva uma sequência de operações.
- Onde x=5 e y=2;
- Multiplicação, divisão e resto.
- Mostre a data atual e some mais 2 dias.



## Trabalhando com variáveis alfanuméricas



## Armazenando Strings

**\$A = 'Olá, Pessoal ! ';**

**\$B = " Rua Javali 3456 ";**





## Diferença entre aspas simples e aspas duplas

- <?
  - \$teste = "Brasil";
  - \$php = '---\$teste--\n';
  - echo "\$php";
  - ?>
- <br>
- <?
  - \$teste = "Brasil";
  - \$php = "---\$teste---\n";
  - echo "\$php";
  - ?>



## Sequências Escape

- \n, \r Nova linha
- \t Tabulação horizontal
- \\ A própria barra ( \ )
- \\$ O símbolo \$
- \' Aspa simples
- \“ Aspa dupla





## Concatenar Strings

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
```

```
<?      // Concatenar strings
$A = “ Olá, pessoal ! “ ;
$B = “ sejam bem-vindos ao PHP ”;
$C = $A . $B;
echo “ $C ” ;
?>
```

```
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Como descobrir o comprimento de uma string

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
```

```
<?      // Contando o número de caracteres
$A = “ Olá, pessoal ! “ ;
$B = “ sejam bem-vindos ao PHP ”;
$C = strlen($A);
$D = strlen($B);
$E = strlen($A . $B);
echo “ A variável A possui $C caracteres <br>” ;
echo “ A variável B possui $D caracteres ” ;
echo “ As duas possuem $E caracteres”;
?>
```

```
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Como eliminar espaços em branco finais e iniciais de uma string

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<?      // Eliminando espaços em branco
$A = "  Olá, pessoal !  ";
$B = trim($A);
$C = strlen($A);
$D = strlen($B);
echo " A variável A possui $C caracteres <br>" ;
echo " A variável B possui $D caracteres " ;
?>

</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Como converter em letras maiúscula ou minúscula

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<?      // Convertendo para maiúsculo ou minúsculo
$A = "  Olá, pessoal !  ";
$B = strtoupper($A); // converte todos os caracteres contidos na string em letras maiúscula
$C = strtolower($A); // converte todos os caracteres contidos na string em letras minúscula
echo " $B <br>" ;
echo " $C " ;
?>

</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Como Trabalhar com Substrings

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<? // Substrings
$país = "Brasil" ;
$B = substr($país,0,2); // Retorna Br
$C = substr($país,2,2); // Retorna as
$D = substr($país,3,1); // Retorna s
echo " $país <br>" ;
echo " $B <br>" ;
echo " $C <br>" ;
echo " $D ";
?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Trabalhando com arrays

```
<HTML>
<HEAD>
<TITLE> Curso PHP – MySql </TITLE>
</HEAD>
<BODY>
<? // Vetor unidimensional
$hoje = date("w"); // w retorna um número entre 0 e 6 correspondente ao dia da semana entre domingo e Sábado
$diaSemana[0] = "Domingo";
$diaSemana[1] = "Segunda-feira";
$diaSemana[2] = "Terça-feira";
$diaSemana[3] = "Quarta-feira";
$diaSemana[4] = "Quinta-feira";
$diaSemana[5] = "Sexta-feira";
$diaSemana[6] = "Sábado";
echo " Hoje é : <b> $diaSemana[$hoje] </b>";
?>
</BODY>
</HTML>
```

Obs: Salvar com a extensão .php



## Exercícios – Prato do dia

Elemento	Dia da semana	Prato do dia	Preço
0	Domingo	Lasanha a quatro queijos	R\$ 12,60
1	Segunda-feira	Frango ao molho madeira	R\$ 10,00
2	Terça-feira	Arroz à grega	R\$ 09,40
3	Quarta-feira	Feijoada à moda da casa	R\$ 11,20
4	Quinta-feira	Nhoque paulista	R\$ 08,50
5	Sexta-feira	Bacalhau ao forno	R\$ 15,20
6	Sábado	Feijão branco	R\$ 10,00

O resultado a ser mostrado na tela deve estar na seguinte ordem :

Hoje é **Domingo**

Nossa sugestão para hoje é : **Lasanha a quatro queijos**

Preço : **12,20**



## Listas

- São usadas para atribuir valores a diversas variáveis simultaneamente.

Ex.:

```
list($dom, $seg, $ter, $qua, $qui, $sex, $sab) = $diaSemana;
```

```
echo "$qua <br>";
```

```
echo "$sex";
```

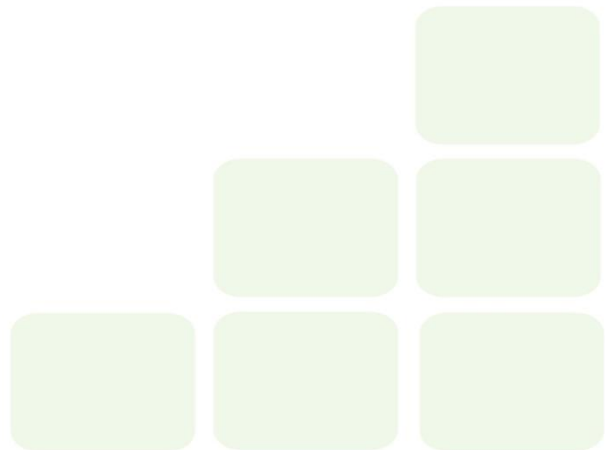




## Objetos

```
class teste {  
    function nada() {  
        echo "nada";  
    }  
}
```

```
$php = new teste;  
$php -> nada();
```



## Booleanos

- 0 → falso (false)
- Outro valor  $\neq$  0 → verdadeiro (true)

■ Ex.:

- 0 == falso
- 1 == verdadeiro
- -2 == verdadeiro





## Transformação de Tipos

### Coerções - Conversão automática

- `$php = "1";` // `$php` é a string "1"
- `$php = $php + 1;` // `$php` é o integer 2
- `$php = $php + 3.7;` // `$php` é o double 5.7
- `$php = 1 + 1.5;` // `$php` é o double 2.5
- `$php = 1 + "10.5";` // `$php == 11.5`
- `$php = 1 + "-1.3e3";` // `$php == -1299`
- `$php = 1 + "teste10.5";` // `$php == 1`
- `$php = 1 + "10testes";` // `$php == 11`
- `$php = 1 + " 10testes";` // `$php == 11`
- `$php = 1 + "+ 10testes";` // `$php == 1`



## Ex. de conversão automática

- `$php = 1 + "10.5";` // `$php == 11.5`
- `$php = 1 + "-1.3e3";` // `$php == -1299`
- `$php = 1 + "teste10.5";` // `$php == 1`
- `$php = 1 + "10testes";` // `$php == 11`
- `$php = 1 + " 10testes";` // `$php == 11`
- `$php = 1 + "+ 10testes";` // `$php == 1`



## Constantes

Ex.:

```
<?
define ("pi", 3.1415926536);
$raio = 5;
$circunf = 2 * pi * $raio;
echo "A área da circunferência é $circunf";
?>
```



## Operadores Aritméticos

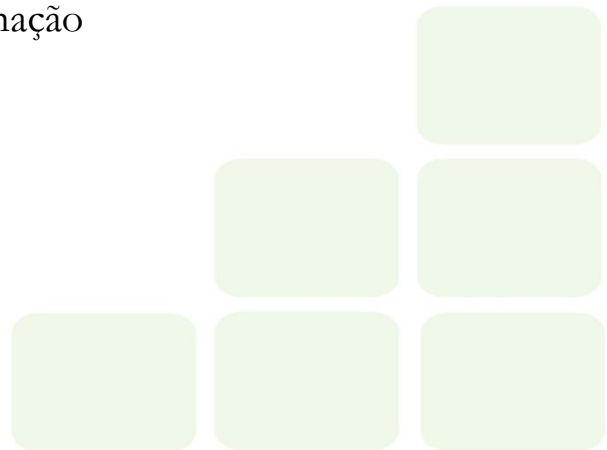
- + adição
- - subtração
- \* multiplicação
- / divisão
- % módulo
- ++ incremento
- -- decremento

```
<?
$x = 5;
echo "$x <br>";
$x = $x + 2 - 4;
echo "$x <br>";
$x++;
echo "$x <br>";
$x = $x * 4 / 2;
echo "$x <br>";
?>
```



## Operador Alfanumérico

- . Operador de concatenação



## Operadores de atribuição

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





## Usando Operadores de Atribuição

<?

\$x = 5; echo "\$x <br>";

\$x += 3; echo "\$x <br>";

\$x -= 2; echo "\$x <br>";

\$x \*= 2; echo "\$x <br>";

\$x = " Olá!"; echo "\$x <br>";

\$x .= \$x; echo "\$x <br>";

?>



## Formulários

- 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:
- ACTION => Informa ao servidor a página que deve ser carregada, assim que o usuário clicar sobre o botão "enviar" contido no formulário.
- Method => Este atributo controla como as informações contidas no formulário devem ser enviadas ao servidor.

Existem os métodos **GET** e **POST**

- **GET** => acrescenta à URL um ponto de interrogação (?) seguido por todos os nomes dos campos ex. <http://www.phb.fap.com.br/formulario.php?txtnome=tim>
- **POST** => Transmite as informações de um modo invisível, ou seja não as insere na URL mas sim no corpo da requisição HTML, além de permitir uma quantidade maior de informações a ser passada de uma página para outra.  
Ex. <http://www.phb.fap.com.br/formulario.php>, ele não apresenta o conteúdo.



## 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 radio (RADIO);
- Menu suspenso (SELECT e OPTION);
- Botão de envio (SUBMIT).



## Caixa de texto de uma linha

### Código HTML

```
<HTML>
<HEAD>
<TITLE> Formulário com caixa de texto </TITLE>
</HEAD>
<BODY>
```

```
<form Method = "POST" Action = "formulario1.php">

  Digite o seu nome : <input type="Text" size="30" name="txtnome" >

  <input type=submit value="Enviar">

</form>
```

```
</BODY>
</HTML>
```

**Obs:** Salvar com uma extensão .htm



## Caixa de texto de uma linha

### Código PHP

```
<HTML>
```

```
<BODY>
```

```
<?>
```

```
echo " Seu nome é : $txtnome ";
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

**Obs: Salvar com o nome formulario1.php**



## Botão de opção (RADIO)

### Código HTML

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Formulário com botão de radio </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Qual é o seu estado civil ?

```
<form Method = "POST" Action = "civil.php">
<input type="radio" name="opcao" value="Casado" >Casado<br>
<input type="radio" name="opcao" value="Solteiro" >Solteiro<br>
<input type="radio" name="opcao" value="Viúvo" >Viúvo<br>
<input type="radio" name="opcao" value="Separado" >Separado<br>
<input type=submit value="Enviar">
</form>
```

```
</BODY>
```

```
</HTML>
```

**Obs: Salvar com a extensão .htm**



## Botão de opção (RADIO)

### Código PHP

```
<HTML>  
<BODY>
```

```
<?  
echo "Seu estado civil é : $opcao";  
?>
```

```
</BODY>  
</HTML>
```

**Obs:** Salvar como civil.php



## Exercícios

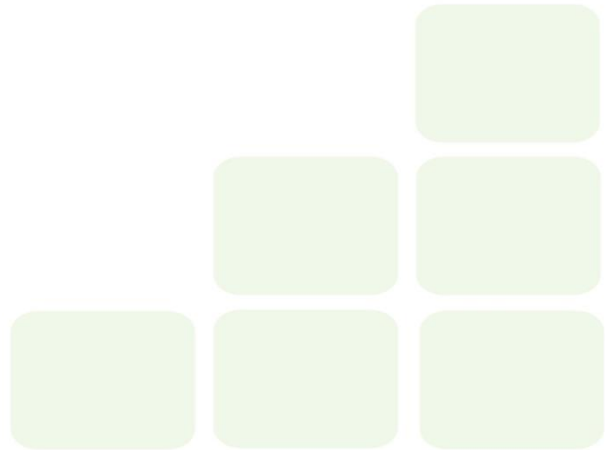
- 1. Crie uma calculadora simples em php, com operações matemáticas.





## Estruturas de Controle

- Blocos:  
Para agrupar vários comandos utiliza-se os caracteres: { e }
- Comandos de Seleção  
if  
switch
- Comandos de Repetição  
while  
do ... while  
for



## A instrução condicional IF

```
<?
$valor = 5;
If ($valor == 5) {
    echo "A variável valor é : $valor";
}
?>
```

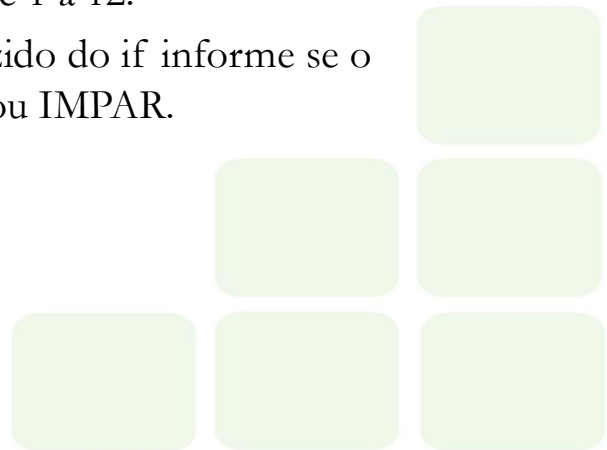
Obs: Salvar com a extensão .php





## Exercícios

- 4 – Utilize o if para retornar o nome de um mês referente a um número de 1 a 12.
- 5 – com o formato reduzido do if informe se o numero digitado é PAR ou IMPAR.



## Instrução condicional switch

```
<?
$valor = 2;
switch ( $valor ) {
    case 1 :
        echo " Primeiro valor";
        break;
    case 2 :
        echo " Segundo Valor ";
        break;
    default :
        echo " Valor maior que 2 ";
        break;
}
?>
```





## Operadores Relacionais

<code>op1 == op2</code>	Verdadeiro se op1 for igual a op2
<code>op1 &gt; op2</code>	Verdadeiro se op1 maior que op2
<code>op1 &lt; op2</code>	Verdadeiro se op1 menor que op2
<code>op1 &gt;= op2</code>	Verdadeiro se op1 maior ou igual a op2
<code>op1 &lt;= op2</code>	Verdadeiro se op1 menor ou igual a op2
<code>op1 != op2</code>	Verdadeiro se op1 diferente de op2



## Operadores lógicos

<code>&amp;&amp;</code>	Verdadeiro se uma expressão A e uma expressão B forem verdadeira
<code>and</code>	Verdadeiro se uma expressão A (AND) e uma expressão B forem verdadeira
<code>  </code>	Verdadeiro se uma expressão A ou uma expressão B forem verdadeira
<code>or</code>	Verdadeiro se uma expressão A ou (OR) uma expressão B forem verdadeira
<code>lop1</code>	Será considerado verdadeiro se o valor de op1 for falso



## Exercício

- Escreva uma página com os seguintes campos de formulário:
  - Nome
  - Endereço
  - Idade
  - Sexo
- Escreva um programa em php para receber os valores do formulário acima e apresentar na tela o nome, endereço e verificar se a idade for maior que 18 anos, mostrar a mensagem : “Minha idade é : XX”, caso contrário a mensagem : “Menor de idade” .



## Exercício: Senha

- Faça um formulário com entradas para Nome de Usuário e Senha e um botão Entrar.
- Esse formulário será interpretado pela página pagina\_protegida.php que deverá exibir uma mensagem de boas vindas para o usuário citando o seu cargo, ou uma mensagem de erro caso a senha ou o nome de usuário estejam incorretos.



## Trabalhando com Loop

### Instrução while e do-while

```
while ( condição )
{
    // instruções ;
}

<?
$i=1;
While($i <= 10)
{
    echo " valor de i = $i;
    i++;
}

$j=1;
Do {
    echo " valor de j = $j;
} while ($j++ < 10)
?>
```



## Trabalhando com Loop

```
for ( inicialização ; condição ; incremento ou decremento )
{
    Trecho a ser repetido ...
}
```

Exemplo:

```
for( $x = 1 ; $x <= 10 ; $x++)
{
    echo " Valor de x é = $x <br> ";
}
```



## Trabalhando com Loop

### Instrução for

```
<?
for ($x = 1 ; $x <=10 ; $x++) {
    echo " A variável X neste momento é : $x ";
}
?>
```

Obs: Salvar com a extensão .php



## Quebra de Fluxo

### ■ break

Usado em laços for, while, do...while e switch.  
Interrompe imediatamente a execução do laço.

### ■ continue

Semelhante ao break, mas ao invés de sair do laço,  
volta pro início dele.



## Exercício

- Utilize uma estrutura de repetição para gerar tags <Hx>.

Ex.:

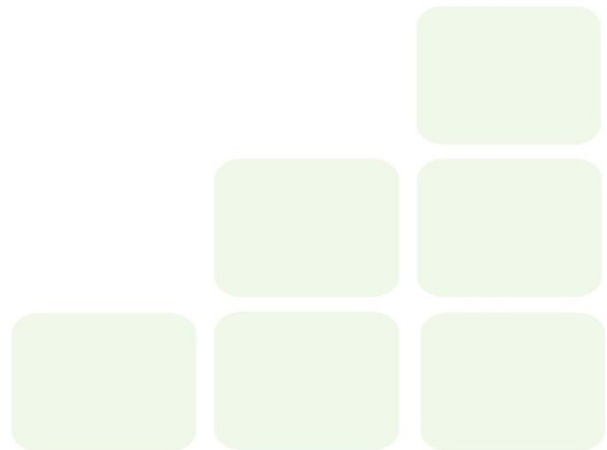
**Letra H1**

**Letra H2**

**Letra H3**

**Letra H4**

**Letra H5**



## Exercícios

- Fazer a tabela Prato do Dia utilizando a estrutura de repetição For.





## Exercício: Formatação via formulário

- Crie um formulário com os seguintes itens:
- Uma caixa de texto
- Três Checkbox para Itálico, Negrito e Sublinhado
- Um Select com opções de alinhamento
- Um select com opções de cor
- Uma caixa de texto para digitar o número de repetições.
- Um select com números correspondentes ao tamanho.



## Exercício: Calculadora

- Fazer uma calculadora simplificada.  
A calculadora possuirá:
- Dois campos onde serão digitados os termos
- Um seletor tipo Radio com as operações básicas
- Um botão Calcular que será usado para realizar o cálculo
- O resultado deve ser apresentado na mesma página do formulário.





## Funções

1. O emprego das funções permite estruturar um programa de modo bem mais eficiente e produtivo, pois garante a criação de pequenas seções de código que podem ser reutilizadas a qualquer momento
1. Podemos definir uma função como sendo um pequeno programa, inserido em seu programa principal, que pode ser chamado a qualquer instante para executar um determinada operação.



## Sintaxe para declaração de funções

- `function nome_da_função($arg1, $arg2, $arg3)`
- `{`
- `Comandos;`
- `...`
- `return <valor de retorno>;`
- `}`



## Retorno

- Não é obrigatório
- Pode-se retornar qualquer tipo de dados, inclusive vetores.
- Sintaxe:  
return valor;



## Argumentos

- Devem ser declarados dentro dos parênteses
- Não precisa especificar o tipo
- Ex.:  
function **imprime**(\$texto)  
{  
    echo \$texto;  
}  
imprime("teste de funções");



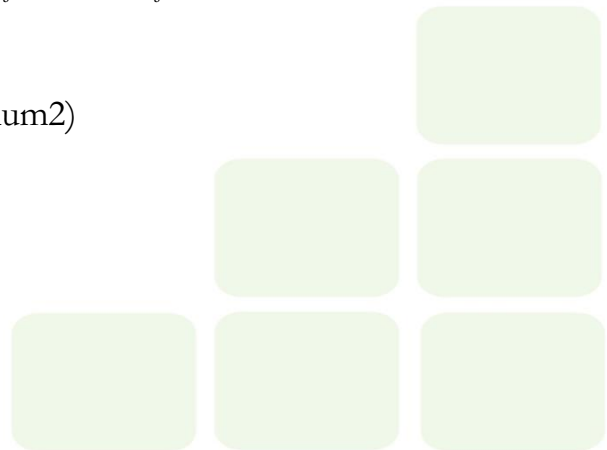


## Passagem de parâmetros por referência

- Utiliza-se o caracter & antes do nome do parâmetro.
- Pode ser utilizado na declaração da função ou na chamada da função.

■ Ex.:

```
function mais5(&$num1, $num2)
{
    $num1 += 5;
    $num2 += 5;
}
mais5($a, &$b);
```



## Argumentos com valores pré-definidos

■ Ex.:

```
function teste($php = "testando")
{
    echo $php;
}
teste(); // imprime "testando"
teste("outro teste"); // imprime "outro teste"
```





## Escopo

```
$php = "Testando";
function Teste()
{
    echo $php;
}
Teste();
```

```
$php = "Testando";
function Teste()
{
    global $php;
    echo $php;
}
Teste();
```



## Funções - Exemplos

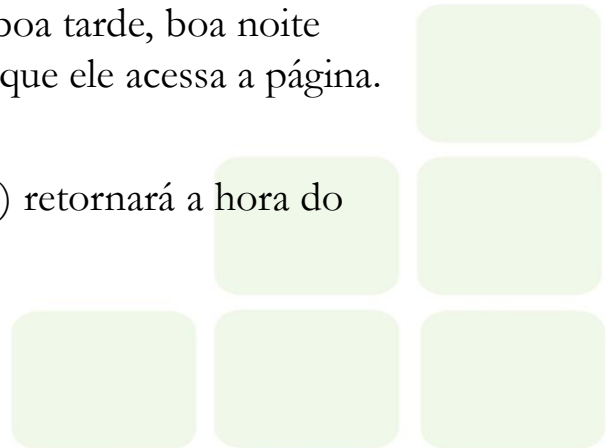
```
<?
function comissao($valor)
{
    $valor = $valor * 0.06;
    return $valor;
}
echo " Ana vendeu R$ 25000, sua comissão é : R$ " . comissao(25000);
echo "<br>";
echo " Carolina vendeu R$ 34680, sua comissão é : R$ " . comissao(34680);
}
?>
```

Obs: Salvar com a extensão .php



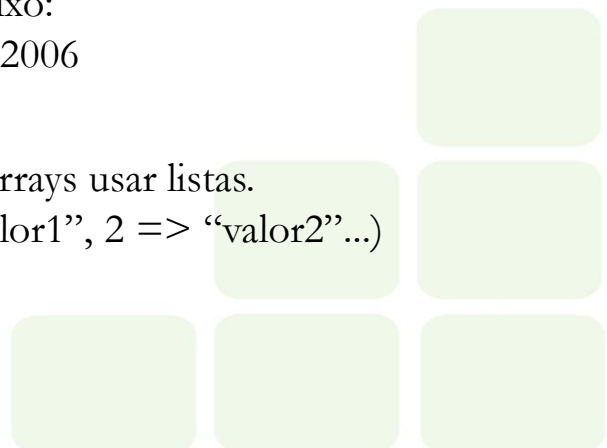
## Exercício: Cumprimento

- Fazer uma função que retorna uma string de cumprimento: Bom dia, boa tarde, boa noite dependendo da hora em que ele acessa a página.
- Dica: A função `date("H")` retornará a hora do dia.



## Exercício: Data por extenso

- Fazer uma função que retorne a data atual no formato do exemplo abaixo:  
Domingo, 29 de abril de 2006
- Dica: Para inicializar os arrays usar listas.  
Ex.: `$x = array(1 => "valor1", 2 => "valor2"...)`





## Variáveis

### ■ Modificador Static

Cria uma variável local, porém seu valor não será perdido ao sair da função.

Ex.:

```
function Teste()
{
$a = 0;
echo $a;
$a++;
}
```

```
function Teste()
{
static $a = 0;
echo $a;
$a++;
}
```



## Variáveis Variáveis

- \$a = "teste";
- \$\$a = "Mauricio Vivas";

O exemplo acima é equivalente ao seguinte:

- \$a = "teste";
- \$teste = "Mauricio Vivas";



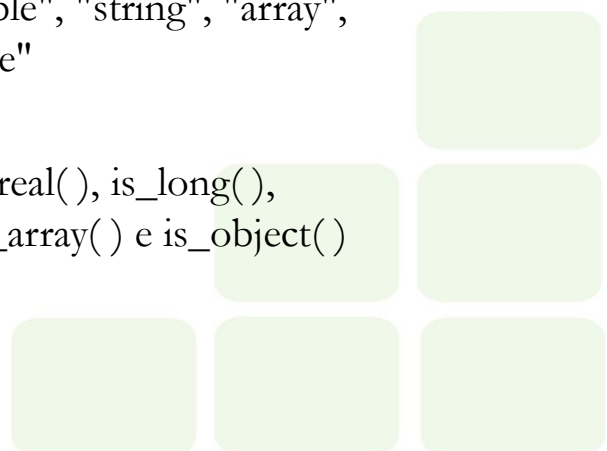
## Variáveis de Ambiente

- Ex.:
  - `$_SERVER["PHP_SELF"]`
  - `$_SERVER["REQUEST_URI"]`
- `phpinfo()`;



## Obter o tipo da variável

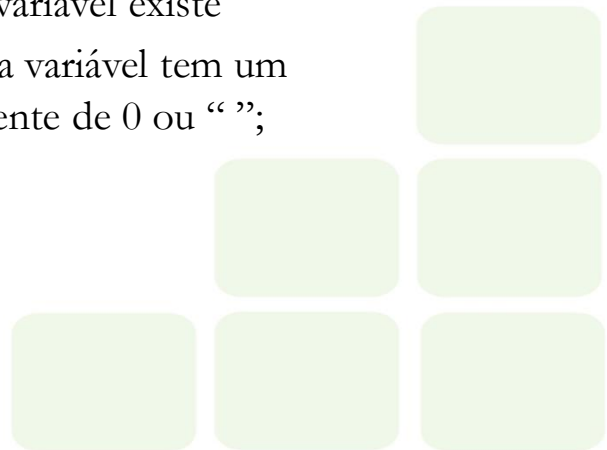
- `gettype($x)`;
- Retorna: "integer", "double", "string", "array", "object" e "unknown type"
- `is_int()`, `is_integer()`, `is_real()`, `is_long()`, `is_float()`, `is_string()`, `is_array()` e `is_object()`





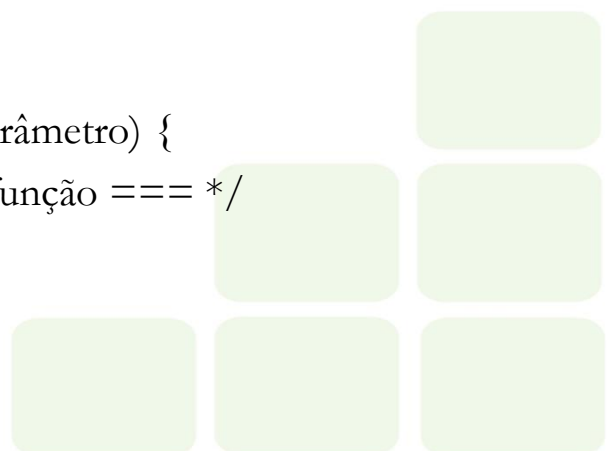
## Outras operações

- `unset($x);` //Destrói a variável
- `isset($x);` //Verifica se a variável existe
- `empty($x);` //Verifica se a variável tem um valor diferente de 0 ou “”;



## Classes e Objetos

- `class Nome_da_classe {`
- `var $variavel1;`
- `var $variavel2;`
- `function funcao1 ($parâmetro) {`
- `/* === corpo da função === */`
- `}`
- `}`

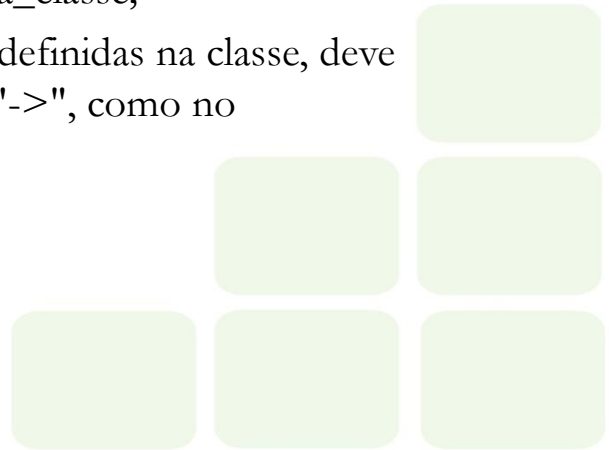






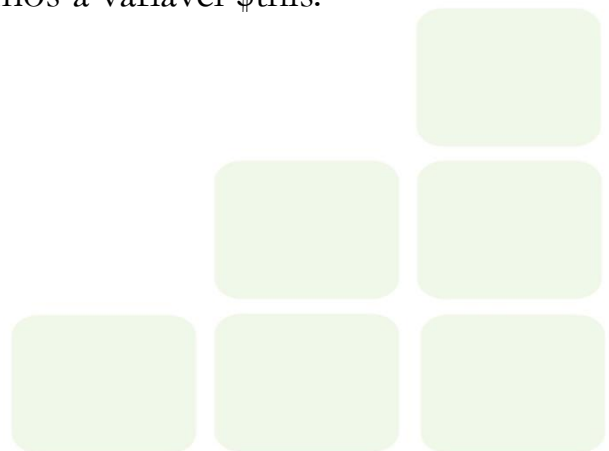
## Objetos

- Sintaxe para criação de objetos:  
`$variável = new nome_da_classe;`
- Para utilizar as funções definidas na classe, deve ser utilizado o operador "->", como no exemplo:
- `$variavel->funcao1();`



## A variável \$this

- Quando for necessário fazer chamadas a funções do próprio objeto utilizamos a variável `$this`.





## Exemplo usando \$this

```

■ class conta
■ {
■     var $saldo;
■     function saldo() {
■         return $this->saldo;
■     }
■     function credito($valor) {
■         $this->saldo += $valor;
■     }
■ }
■ $minhaconta = new conta;
■ echo $minhaconta->saldo() . '<br>';
■ $minhaconta->credito(50);
■ echo $minhaconta->saldo();      // retorna 50

```



## Sub-Classes

```

■ Exemplo de subclasses:
■ class novaconta extends conta
■ {
■     var $numero;
■     function numero() {
■         return $this->numero;
■     }
■ }
■ $outraconta = new novaconta;
■ $outraconta->numero = 123456;
■ echo $outraconta->numero();

```



## Construtores

Um construtor é uma função definida na classe que é automaticamente chamada no momento em que a classe é instanciada



## Exemplo de Construtores

```
■ class conta {  
■     var $saldo;  
■     function conta() {  
■         $this->saldo = 0;  
■     }  
■     function saldo() {  
■         return $this->saldo;  
■     }  
■     function credito($valor) {  
■         $this->saldo += $valor;  
■     }  
■ }
```





## Cookies

- Cookies são mecanismos para armazenar e consultar informações nos browsers dos visitantes da página. O PHP atribui cookies utilizando a função **setcookie**, que deve ser utilizada antes da tag <html> numa página.
- Não use cookies para acessar informações sigilosas. Os dados dos cookies são armazenados no diretório de arquivos temporários do visitante, sendo facilmente visualizado por pessoas mal intencionadas.
- Atenção para a opção "aceitar cookies" que pode ser desativada a qualquer momento pelo visitante.



## setcookie

- Setcookie("nome\_cookie", "seu\_valor", "tempo\_vida")



## Cookies: Exemplo

- <?
- \$valor = 'Texto armazenado no cookie';  
setcookie("TesteCookie", \$valor, time()+3600);
- ?>
  
- <?
- echo \$\_COOKIE["TesteCookie"];
- ?>
- A partir de agora qualquer página do site terá acesso a  
\$\_COOKIE["TesteCookie"];



## Sessões

- Sessões são mecanismos parecidos com os cookies.
- Suas diferenças são que sessões são armazenadas no próprio servidor e não expiram.
  
- Cookie => Computador do usuário
- Session => Servidor



## Sessões: Exemplo 1.1

```

■ <?
  session_start();
?>

<?
echo 'Bem vindo à pagina 1';

$_SESSION['cor'] = 'verde';
$_SESSION['animal'] = 'gato';
$_SESSION['acesso'] = time();

echo '<br><a href="pagina2.php">Página 2</a>';

?>

```



## Sessões: Exemplo 1.2

```

■ <?
  session_start();

?>

<?
echo 'Bem vindo a pagina 2<br>';

echo $_SESSION['animal'] . '<br>'; // gato
echo $_SESSION['cor']; // verde
echo date('d/m/Y H:i:s', $_SESSION['acesso']);

echo '<br /><a href="pagina1.php">Pagina 1</a>';

?>

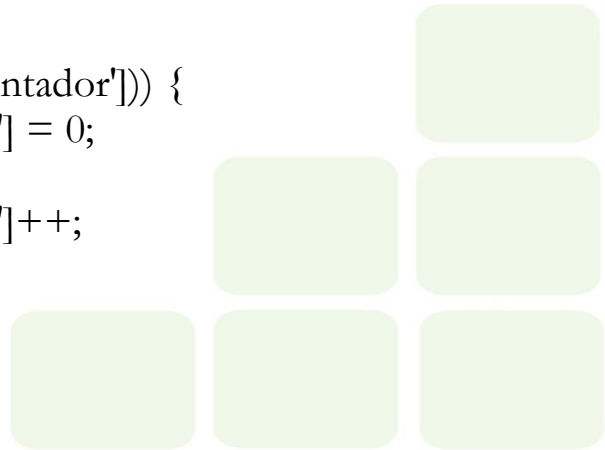
```



## Sessões: Exemplo 2.1

- Contador de visitas

- ```
<?
session_start();
if (!isset($_SESSION['contador'])) {
    $_SESSION['contador'] = 0;
} else {
    $_SESSION['contador']++;
}
?>
```



## Sessões: Exemplo 2.2

- ```
<?php
session_start();
unset($_SESSION['contador']);
?>
```





## Sessões: Exemplo 3.1

```

■ <?
  session_start();

  $autoriza = 0;

  if (isset($_SESSION['usuario'])) {
    $autoriza = 1;
  } else {
    $usuario = $_GET['usuario'];
    $senha = $_GET['senha'];
    if (($usuario == 'fulano') && ($senha == '123'))
      $autoriza = 1;
  }

  if (!$autoriza)
    echo "Você deve fazer login.";
  else
    echo "Você acessou a página com sucesso";
?>

```



## Include e Require

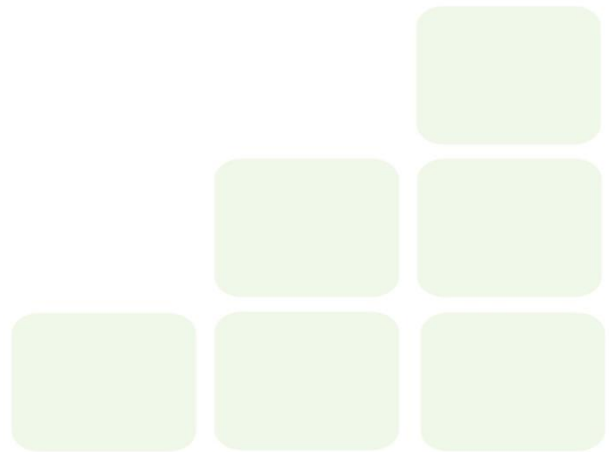
- A função `require()` põe o conteúdo de um outro arquivo no arquivo php atual, antes de ser executado. Quando o interpretador do PHP ler este arquivo, ele encontrará todo o conteúdo dos "require's" adicionado no arquivo corrente.
- A função `include()` é semelhante à função `require()`, com a diferença que o código do arquivo incluído é processado em tempo de execução,





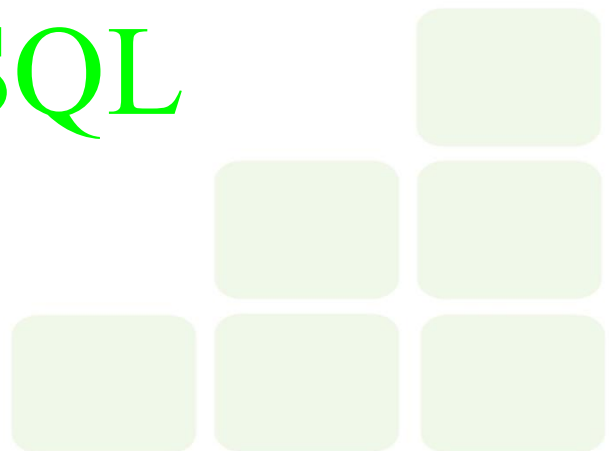
## Exemplos

- `require ("arquivo.php");`
- `include("arquivo.php");`



## BANCO DE DADOS

MySQL





## BANCO DE DADOS - MySQL

- Como você já sabe, o PHP torna as páginas da WEB dinâmicas, mas o que realmente faz com que o PHP seja uma linguagem largamente empregada na internet é a sua capacidade em interagir com banco de dados
- Um Banco de dados deve ser capaz de gerenciar todas as informações contidas em todas as tabelas que o compõem, bem como seus relacionamentos e sua integridade referencial.



## PHP MyAdmin

- Gerenciador de banco de dados MySQL escrito em PHP.
- Executa as principais operações de gerenciamento. (Criação e edição de tabelas, inserção, edição, exclusão de registros...)
- Linha de comandos SQL
- Acessar: [mysql.curso.com.br](http://mysql.curso.com.br)



## BANCO DE DADOS – MySQL

atributo	Tipo	Tamanho
Nome	VarChar	45
Matricula	VarChar	06
Curso	VarChar	45
Endereco	VarChar	30
Cidade	VarChar	30
Estado	VarChar	02
Bairro	VarChar	15
Cep	VarChar	08



## BANCO DE DADOS – MySQL

Host	mysql.curso.com.br
Base de dados	alunoXX
Conta	root
Senha	curso



## A linguagem SQL

- O SQL (Structured Query Language) é uma linguagem criada pela IBM na década de 60. Graças ao seu alto desempenho e grande facilidade de manuseio, o SQL se transformou, em pouco tempo, em uma linguagem de uso universal e praticamente um padrão nos dias de hoje na manipulação de banco de dados.
- Usamos o SQL para realizar várias tarefas com banco de dados, entre elas:
  1. Fazer consultas simples ou complexas em uma base de dados;
  2. Administrar uma base de dados, mantendo suas tabelas atualizadas, por meio de inserções, alterações e cancelamento de registros;



## A linguagem SQL

### Como EXIBIR os dados de uma tabela :

```
SELECT * FROM CIDADE;
```

```
SELECT NOME, ENDERECO, CIDADE
FROM ALUNO
ORDER BY NOME ;
```

```
SELECT NOME, ENDERECO, CIDADE
FROM ALUNO
WHERE CIDADE = 'BURITI DOS LOPES';
```



## A linguagem SQL

### Como INSERIR os dados de uma tabela :

```
INSERT INTO <NOME_DA_TABELA> (CAMPO1,CAMPO2,CAMPO3)
VALUES (VALOR_CAMPO1,VALOR_CAMPO2,VALOR_CAMPO3)
```

#### Exemplo:

```
Insert into aluno(nome,matricula,curso)
Values ('Frederico','456734','Eng. Mecânica');
```



## A linguagem SQL

### Como ATUALIZAR os dados de uma tabela :

```
UPDATE <NOME_DA_TABELA> SET NOME_CAMPOX='PARNAIBA'
WHERE NOME_CAMPO1='CURSO01';
```

#### Exemplo:

```
Update aluno set curso='Engenharia Mecânica'
where curso='Eng. Mecânica'
```



## A linguagem SQL

### Como DELETAR os dados de uma tabela :

**DELETE FROM <NOME\_DA\_TABELA> WHERE NOME\_CAMPO1=VALOR**

#### Exemplo:

Delete from aluno; // Cuidado este comando apaga todos os registros.

Delete from aluno where cidade='TERESINA'

// Este comando apaga todos os registros onde cidade é igual a TERESINA



## PHP - MySQL

### Passo 01 - Efetuar a conexão com o servidor MySQL

- Para que possamos trabalhar com um banco de dados em uma página da WEB, é necessário, antes de mais nada, criar uma variável que contenha uma conexão com o servidor MySQL.

#### Sintaxe:

**\$<nome\_variavel\_conexao> = mysql\_connect("máquina","conta","senha");**

#### Exemplo :

**\$conexao = mysql\_connect("mysql.curso.com.br","root","curso")  
or die (" A conexão não foi executada com sucesso");**



## PHP MySQL

- Selecionar um banco de dados:  
`mysql_select_db("banco_de_dados",$conexao);`
- Exemplo:  
`mysql_select_db("aluno01",$conexao);`



## PHP - MySQL

### Passo 02 - Criar uma instrução SQL

- Uma vez conectado ao banco de dados, devemos criar uma variável que contenha uma instrução SQL para a manipulação de suas tabelas, como , por exemplo visualizar seus registros.

#### Sintaxe:

`$<nome_variavel_SQL> = "<Instrução SQL>";`

#### Exemplo :

`$var_sql = "SELECT * FROM cidade ORDER BY nome";`



## PHP - MySQL

### Passo 03 - Criar uma variável com o resultado obtido pela instrução SQL

- Uma vez executada a instrução SQL, devemos armazenar o seu resultado em uma variável, para que possa por exemplo ser exibido na tela do seu navegador.

#### Sintaxe:

`$<nome_variavel_que contem o resultado da pesquisa SQL> = mysql_query(<nome da variavel que contem a instrução SQL>,<nome da variável de conexão>);`

#### Exemplo :

`$rs = mysql_query($var_sql,$conexão);`



## PHP - MySQL

### Passo 04 - Formatar 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()`.

#### Sintaxe:

`$<nome_variavel_que contem o registro> = mysql_fetch_array(<nome da variavel que contem o resultado da pesquisa SQL>)`

`$<nome da variavel que contem o campo do registro a ser retornado> = <nome da variavel que contem o registro[“nome do campo a ser retornado”] ;`

#### Exemplo :

`$linha = mysql_fetch_array($rs);`

`$nome_cidade = $linha[“nome”];`





## PHP - MySQL

### Passo 06 - Encerrar a conexão

- Ao terminar de usar uma tabela, voce deve fecha-la, bem como sua conexão com o mysql.

#### Sintaxe:

**mysql\_free\_result**(<nome da variavel que contem o resultado da pesquisa SQL>);

**mysql\_close** (<nome da variavel de conexão>);

#### Exemplo :

**mysql\_free\_result**(\$rs);

**mysql\_close**(\$conexão);



## PHP – MySQL – 1º Exemplo

```
<?
$conexao = mysql_connect("mysql.curso.com.br","alunoxx","alunoxx") or die (" A conexão não foi
possivel");
mysql_select_db("alunoXX",$conexao) or die ("Não foi possivel selecionar o banco de dados");
$sql = " select nome , matricula from aluno order by nome";
$rs = mysql_query ($sql, $conexao) or die ("não foi possivel efetuar a consulta");
$linha = mysql_fetch_array($rs);
$codigo_matricula = $linha["matricula"];
$nome_aluno = $linha["nome"];
echo "Matricula : $codigo_matricula ";
echo "<br>";
echo "Nome : $nome_aluno";
mysql_free_result($rs);
mysql_close($conexao);
?>
```



## PHP – MySQL – 2º Exemplo

```
<?
$conexao = mysql_connect("mysql.curso.com.br","alunoxx","alunoxx") or die (" A conexão não foi
possivel");
mysql_select_db("alunoXX",$conexao) or die ("Não foi possível selecionar o banco de dados");
$sql = " select nome , matricula from aluno order by nome";
$rs = mysql_query ($sql, $conexao) or die ("não foi possível efetuar a consulta");
WHILE ($linha = mysql_fetch_array($rs))
{
    $codigo_matricula = $linha["matricula"];
    $nome_aluno = $linha["nome"];
    echo "Matricula : $codigo_matricula ";
    echo "<br>";
    echo "Nome : $nome_aluno";
    echo "<br>";
}
mysql_free_results($rs);
mysql_close($conexao);
?>
```



## PHP – MySQL – 3º Exemplo

Passar valores de um formulário para uma instrução SQL no PHP

```
<form Method = "POST" Action = "procura.php">

Procurar o nome : <input type="Text" size="30" name="txtnome" >

<input type="submit" value="Enviar">

</form>
```

Salvar com a extensão .htm



### PHP – MySQL – 3º Exemplo

#### Usando a variável do formulário na instrução SQL

```
<?
$txtnome = $_POST['txtnome'];
$conexao = mysql_connect("mysql.curso.com.br","root","curso") or die (" A conexão não foi possível");
mysql_select_db("alunoXX",$conexao) or die ("Não foi possível selecionar o banco de dados");
$sql = " select nome , matricula from aluno where nome like '%" . $txtnome . "%' order by nome";
$rs = mysql_query ($sql, $conexao) or die ("não foi possível efetuar a consulta");
while ($linha = mysql_fetch_array($rs))
{
    $codigo_matricula = $linha["matricula"];
    $nome_aluno = $linha["nome"];
    echo "Matricula : $codigo_matricula ";
    echo "<br>";
    echo "Nome : $nome_aluno";
    echo "<br>";
}
mysql_free_results($rs);
mysql_close($conexao);
?>
```



### PHP – MySQL – Exercício

Nome	Tipo	tamanho
id	int (autoincrement)	
descricao	varchar	45
valor	Float	

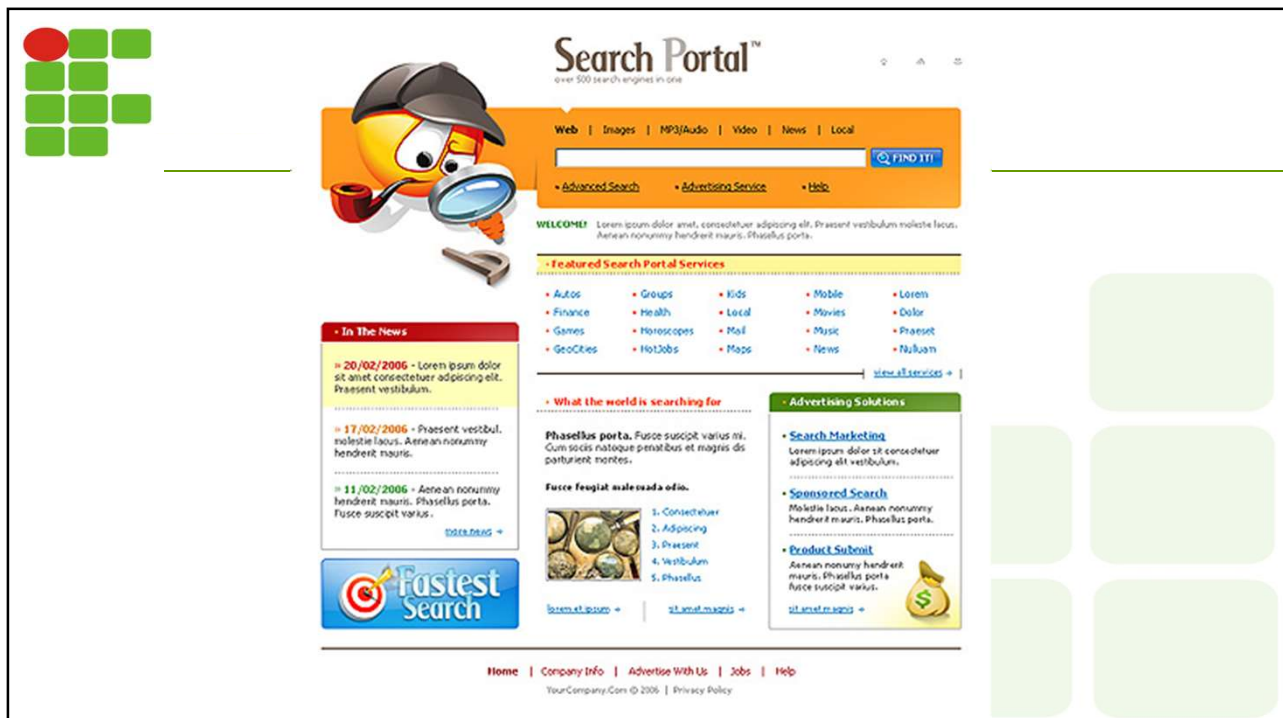
- Você deverá construir um formulário e o seu respectivo programa PHP para cadastrar os produtos e um outro para listar os produtos cadastrados.
- Exemplo de inserção :  

```
insert into produto values ('manteiga',22.98)
```



## Website PHP/MySQL

- Construção de um portal de busca dinâmico em PHP com banco de dados MySQL
- Algumas Características da home page:
  - Pesquisa sites, imagens, músicas, vídeos.
  - Banco de Notícias, Artigos e Publicidade
  - Barra para seleção de tipos de arquivo (Web, Imagens, Mp3, Vídeo...)
  - Links para categorias de sites (Jogos, Carros, Infantil, Horóscopo, Empregos...)
  - Bloco de notícias com notícias mais recentes.
  - Bloco com anúncios dos clientes
  - Bloco com o artigo mais recente.





## Website PHP: Estrutura de Dados

- links
  - id int (autoincrement)
  - titulo varchar 50
  - descricao varchar 200
  - link varchar 150
  - categoria int
  - tipo int



## Website PHP: Estrutura de Dados

- categorias
  - id int (autoincrement)
  - categoria varchar 50





## Website PHP: Estrutura de Dados

- tipos
  - id int (autoincrement)
  - tipo varchar 50



**Search Portal™**  
over 500 search engines in one

Web | Images | MP3/Audio | Video | News | Local

Advanced Search | Advertising Service | Help

**WELCOME!** Lorem ipsum dolor amet, consetetur adipiscing elit. Praesent vestibulum molestie lacus. Aenean nonummy hendrerit mauris. Phasellus porta.

**Featured Search Portal Services**

Autos	Groups	Kids	Mobile	Lorem
Finance	Health	Local	Movies	Dolor
Games	Horoscopes	Mail	Music	Phasellus
GeoCities	HotJobs	Maps	News	Nullam

[view all services](#)

**What the world is searching for**

Phasellus porta. Fusce suscipit varius mi. Cum sociis natoque penatibus et magnis dis parturient montes.

**Forces feugiat malesuada odio.**

1. Consetetur  
2. Adipiscing  
3. Praesent  
4. Vestibulum  
5. Phasellus

[view all items](#)

**Advertising Solutions**

**Search Marketing**  
Lorem ipsum dolor sit consetetur adipiscing elit vestibulum.

**Sponsored Search**  
Molestie lacus. Aenean nonummy hendrerit mauris. Phasellus porta.

**Product Submit**  
Aenean nonummy hendrerit mauris. Phasellus porta. Fusce suscipit varius.

[view all services](#)

**Fastest Search**

Home | Company Info | Advertise With Us | Jobs | Help  
YourCompany.Com © 2006 | Privacy Policy





## Website PHP: Estrutura de Dados

- noticias
  - id int (autoincrement)
  - manchete varchar 50
  - descricao varchar 200
  - data date



## Website PHP: Estrutura de Dados

- anuncios
  - id int (autoincrement)
  - titulo varchar 50
  - descricao varchar 200
  - link varchar 150

