

# PHP

## Aula 1

# O que é?

- Linguagem que permite criar sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e links.
- Código PHP é executado no servidor, sendo enviado para o cliente apenas html puro.

# Características

- É uma linguagem de fácil aprendizado;
- Tem suporte a um grande número de bancos de dados como: MySQL, Oracle, Sybase, PostgreSQL e vários outros.
- Tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP;
- É multiplataforma, tendo suporte aos sistemas Operacionais mais utilizados no mercado;
- Seu código é livre, não é preciso pagar por sua utilização e pode ser alterado pelo usuário na medida da necessidade de cada usuário.
- Não precisa ser compilado.

# Instalação

- Para que o php funcione é necessária a instalação e configuração do servidor Apache e do PHP, além de um servidor de banco de dados.
- Existem aplicativos que instalam “em um único pacote” o Apache, o PHP e o Mysql. Dentre esses podemos destacar: EasyPHP, Wamp, Xampp.

# Testando a instalação do PHP

```
<html>
<head><title>Aprendendo PHP</title></head>
<body>
<?php
    echo "Primeiro Exemplo";
?>
</body>
</html>
```

Salve este arquivo como exemplo.php no diretório configurado no servidor.

# Testando a instalação do PHP

- No browser acesse:

`http://localhost/exemplo.php`

Se for exibido o texto “Primeiro Exemplo” na sua página, significa que a instalação está correta.

# Sintaxe

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

```
<?php  
  comandos;  
?>
```

```
<?  
  comandos;  
?>
```

```
<script language="php">  
  comandos;  
</script>
```

# Sintaxe

- Utiliza o ponto e vírgula (;) ao final de cada linha de comando.
- Comentários:
  - Uma linha: delimitador //

```
<?php  
echo "teste"; // este teste é similar ao anterior  
// sql "teste";  
?>
```



# Sintaxe

- Comentários:
  - Mais de uma linha: caracteres “/\*” para o início do bloco e “\*/” para o final do comentário.

```
<?php
/*
    echo “Este é o nosso primeiro exemplo”;
    echo “Este texto não será exibido na página”;
*/
?>
```

# Exibindo na tela

- Comandos:
  - Echo
  - Print

```
<?php
    $a = 10;
    $b = 20;
    $soma = $a + $b;
    //Imprimindo a soma na tela com o comando print.
    print ("A soma é: ". $soma."<br/>");
    //Imprimindo a soma na tela com o comando echo.
    echo ("A soma é: ". $soma." <br/>");
    echo $a." ".$b;
?>
```

# Variáveis

- Toda variável em PHP tem seu nome composto pelo caracter \$(dólar) e uma string, que deve iniciar por uma letra ou o caracter “\_”.
- O PHP é case sensitive, ou seja, as variáveis \$teste e \$TESTE são diferentes.

# Tipos de variáveis

- Inteiro
- Ponto flutuante
- String
- Array

# Tipos de variáveis

- O PHP utiliza checagem de tipos dinâmica, ou seja, uma variável pode conter valores de diferentes tipos em diferentes momentos da execução do script. Por este motivo não é necessário declarar o tipo de uma variável para usá-la. O interpretador PHP decidirá qual o tipo daquela variável, verificando o conteúdo em tempo de execução.
- Ainda assim, é permitido converter os valores de um tipo para outro desejado, utilizando o typecasting ou a função `settype`.

# Tipo inteiro (integer)

- `$php = 1234;` # inteiro positivo na base decimal
- `$php = -234;` # inteiro negativo na base decimal
- `$php = 0234;` # inteiro na base octal-simbolizado pelo 0  
# equivale a 156 decimal
- `$php = 0x34;` # inteiro na base hexadecimal(simbolizado  
# pelo 0x) - equivale a 52 decimal.

# Tipo Ponto Flutuante (double ou float)

- `$php = 1.234;`
- `$php = 23e4;` # equivale a 230.000
- A diferença entre float e double está no número de bytes utilizados para armazenar a variável.

# Tipo String

- Strings podem ser atribuídas de duas maneiras:
  - a) utilizando aspas simples ( ' ) - Desta maneira, o valor da variável será exatamente o texto contido entre as aspas (com exceção de \\ e \') ;
  - b) utilizando aspas duplas ( " ) - Desta maneira, qualquer variável ou caracter de escape será expandido antes de ser atribuído



# Tipo String

- Exemplo:

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

A saída desse script será "---\$teste--\n".

# Tipo String

- Exemplo

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

A saída desse script será "---Brasil--" (com uma quebra de linha no final).

# Caracteres de Escape

Sintaxe	Significado
<code>\n</code>	Nova linha
<code>\\</code>	A própria barra ( <code>\</code> )
<code>\\$</code>	O símbolo \$
<code>\'</code>	Aspa simples
<code>\"</code>	Aspa dupla

# Tipo Array

- Tipo array é um dicionário onde os índices são as chaves de acesso.
- Vale ressaltar que os índices podem ser valores de qualquer tipo e não somente inteiros. Inclusive, se os índices forem todos inteiros, estes não precisam formar um intervalo contínuo.

# Tipo Array

- Exemplo:

```
<?php
    $cor[1] = "vermelho";
    $cor[2] = "verde";
    $cor[3] = "azul";
    $cor["teste"] = 1;
?>
```

- Equivalentemente, pode-se escrever:

```
<?
$cor = array(1 => "vermelho, 2 => "verde, 3 => "azul",
"teste => 1);
?>
```

# Tipo booleano

- Uso das palavras chaves: TRUE e FALSE
- Exemplo:

```
<?php
    $exibir_nome = TRUE;
    if ($exibir_nome){
        echo 'José da Silva';
    }
?>
```

# Transformação de tipos

- Coerções:
  - Quando ocorrem determinadas operações (“+”, por exemplo) entre dois valores de tipos diferentes, o PHP converte o valor de um deles automaticamente (coerção). É interessante notar que se o operando for uma variável, seu valor não será alterado.
  - Exemplo:  
`$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`

# Transformação de tipos

- Coerção

- Exemplos:

- ```
$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
```



# Transformação de tipos

- Transformação explícita:

- Exemplo:

`$php = 15; // $php é integer (15)`

`$php = (double) $php // $php é double (15.0)`

`$php = 3.9 // $php é double (3.9)`

`$php = (int) $php // $php é integer (3) e o valor decimal é truncado`

# Transformação de tipos

- Transformação explícita:
  - Os tipos de cast permitidos são:
    - (int), (integer)  $\Rightarrow$  muda para integer;
    - (real), (double), (float)  $\Rightarrow$  muda para float;
    - (string)  $\Rightarrow$  muda para string;
    - (array)  $\Rightarrow$  muda para array;

# Constantes

- É um valor que não sofre modificações durante a execução do programa.
- Só pode conter valores do tipo boolean, inteiro, ponto flutuante e string.
- Exemplo:

```
<?php  
    define ("MAXIMO_CLIENTES",100);  
    echo MAXIMO_CLIENTES;  
?>
```

# Constantes

- Exemplo:
  - Calcular o comprimento da circunferência

```
<?php  
define ("PI", 3.1415926536);  
$raio = 10.3;  
$circunf = 2*PI*$raio;  
echo $circunf;  
?>
```

# Operadores

- Atribuição

- Exemplo:

```
<?php
```

```
$var = 5;
```

```
$var += 5; // $var = $var + 5;
```

```
$var -= 5; // $var = $var - 5;
```

```
$var *= 5; // $var = $var * 5;
```

```
$var /= 5; // $var = $var / 5;
```

```
?>
```

# Operadores

- Atribuição

Operadores	Descrição
<code>++\$a</code>	Pré-incremento. Incrementa em 1 unidade \$a e retorna \$a.
<code>\$a++</code>	Pós-incremento. Retorna \$a e depois incrementa em 1 unidade \$a
<code>--\$a</code>	Pré-decremento. Decrementa em 1 unidade \$a e retorna \$a.
<code>\$a--</code>	Pós-decremento. Retorna \$a e depois decrementa em 1 unidade \$a

# Operadores

- Aritméticos:

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

# Operadores

- Aritméticos

- Exemplos:

```
<?php
```

```
    $a = 2;
```

```
    $b = 4;
```

```
    echo $a + 3 * 4 + 5 * $b;
```

```
    echo ($a + 3) * 4 + (5 * $b);
```

```
?>
```

```
<?php
```

```
    $a = '10';
```

```
    echo $a + 5;
```

```
?>
```



# Operadores

- Relacionais:
  - Utilizados para a comparação entre valores ou expressões, resultando sempre em um valor boolean (TRUE ou FALSE).

Comparadores	Descrição
==	Igual
!=	Diferente
<	Menor
<=	Menor ou igual
>	Maior
>=	Maior ou igual

# Operadores

- Relacionais

- Exemplo:

```
<?php
```

```
    if ($a==5){
```

```
        echo "A variável tem o valor igual a 5";
```

```
    }
```

```
    if ($a=5){
```

```
        echo "atribui o valor 5 a variável a e retorna verdadeiro.";
```

```
    }
```

```
?>
```

# Operadores

- Relacionais

- Exemplo:

```
<?php
```

```
$e = 0;
```

```
if ($e == FALSE){ //zero é igual a falso
```

```
    echo '$e é falso';
```

```
}
```

```
?>
```

# Operadores

- Lógicos:

Operador	Descrição
&&	“E” lógico
	“Ou” lógico
!	Inversão

# Operadores

- De Strings

- Concatenação: .

- Exemplo:

```
<?php
```

```
    $nome = “João”;
```

```
    $sobrenome = “da Silva”;
```

```
    echo $nome.””.$sobrenome;
```

```
?>
```

# Exercícios

1) Criar um programa que calcule a área de retângulo, considerando a fórmula  $\text{base} * \text{altura}$ .

```
<?php
    $base = 10;
    $altura= 20;
    $area = $base * $altura;
    echo $area;
?>
```

# Exercícios

2) Crie um programa que leia o peso e a altura de uma pessoa.

```
<?php
    define ("ALTURA", 1.59);
    define ("PESO", 60);
    echo "Minha altura é". ALTURA. "<br/>";
    echo "O meu peso é ". PESO." quilos";
?>
```

**OU**

```
<?php
    $altura= 1.59;
    $peso = 60;
    echo "Altura: ". $altura. "<br/>";
    echo "Peso: ". $peso." quilos";
?>
```

# Exercícios

3) Crie um programa em php que calcule o IMC (Índice de Massa Corporal), considerando a fórmula:

$$\text{imc} = \text{peso} / \text{altura} * \text{altura}$$

```
<?php
$peso = 60;
$altura = 1.80;
$imc = $peso/($altura*$altura);
echo $imc;
?>
```