PHP – PARTE 1

DIOGO CEZAR TEIXEIRA BATISTA
http://inf.cp.utfpr.edu.br/diogo
diogo@diogocezar.com

INTRODUÇÃO

- PHP: *PHP HyperText Preprocessor;*
- linguagem de código fonte aberto;
- muito utilizada na web;
- criada para desenvolvimento de aplicações WEB.

INTRODUÇÃO

- diferencial:
  - ao invés de escrever um programa com um monte de comandos para imprimir HTML, você escreve um arquivo "HTML" com algum código inserido para fazer alguma coisa;
- o código PHP é delimitado por tags iniciais e finais que lhe permitem pular pra dentro e pra fora do "modo PHP";
  - dentro do modo PHP, têm-se todos os recursos oferecidos pela linguagem, como: acesso a base de dados, manipulação de imagens, entre outros.

INTRODUÇÃO

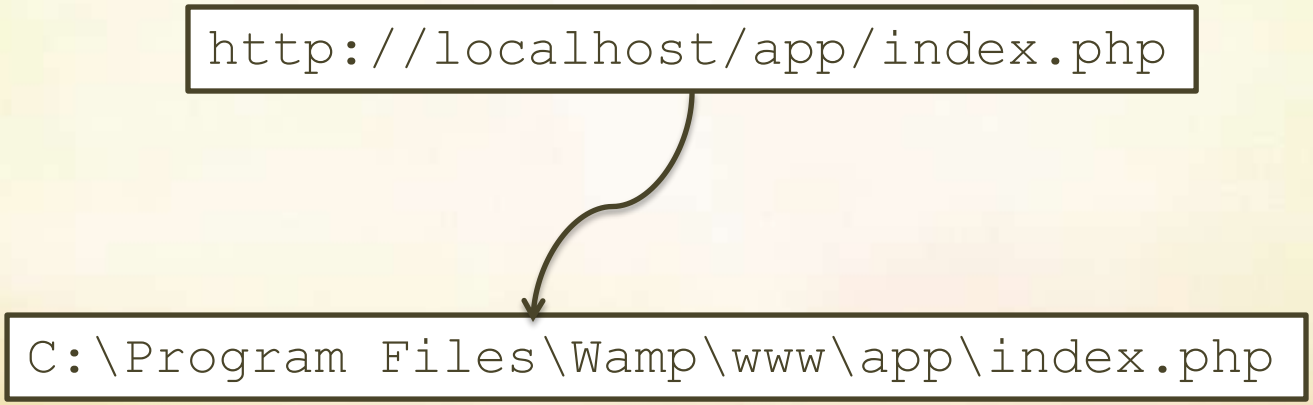
- a melhor coisa em usar PHP está no fato de ele ser extremamente **simples** para um iniciante, mas oferece muitos recursos para o programador profissional.

INTRODUÇÃO

- para testar o **PHP**, devemos possuir um servidor (Apache) devidamente configurado, bem como os arquivos necessários para a interpretação do código;
- a configuração do Apache + PHP já foi estudada nas aulas anteriores.

INTRODUÇÃO

- qualquer editor de textos pode ser usado para escrever os scripts PHP;
- as páginas PHP devem ser salvas no diretório raiz do servidor, e serão executadas somente quando acessadas pela url correspondente em seu navegador, exemplo:



PRIMEIRO SCRIPT

- para criar o primeiro exemplo, digite o seguinte código-fonte no seu editor e salve com o nome de *teste.php* dentro do diretório raiz do servidor:

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
<?php echo "<p>Alô Mundo</p>"; ?>
</body>
</html>
```

PRIMEIRO SCRIPT

- no *browser*, digite o endereço:

http://127.0.0.1/teste.php

http://localhost/teste.php

- veja também o código fonte da página (Exibir → Código fonte);
- note que os comando em php não aparecem no código fonte, pois o servidor interpreta todos os scripts antes de enviar a página compilada para o *browser.*

O QUE O PHP PODE FAZER?

- script no lado do servidor (*server-side*):
  - mais tradicional;
  - você precisa de três coisas para seu trabalho:
    - o interpretador do PHP (como CGI ou módulo);
    - um servidor *web;*
    - um browser;
  - basta rodar o servidor web conectado a um PHP instalado, assim você pode acessar os resultados de seu programa PHP com um browser, visualizando a página PHP através do servidor web.

O QUE O PHP PODE FAZER?

- script de linha de comando:
  - você pode fazer um script PHP funcionar sem um servidor *web ou browser;*
  - basta utilizar seu interpretador;
  - usados na maioria das situações para execução de serviços locais.

O QUE O PHP PODE FAZER?

- escrevendo aplicações GUI no lado do cliente (client-side):
  - o PHP não é (provavelmente) a melhor linguagem para produção de aplicações com interfaces em janelas;
  - mas o PHP faz isso muito bem, e se você deseja usar alguns recursos avançados do PHP em aplicações no lado do cliente poderá utilizar o **PHP-GTK** para escrever esses programas;
  - e programas escritos desta forma ainda serão independentes de plataforma.

O QUE O PHP PODE FAZER?

- o PHP pode ser utilizado na maioria dos sistemas operacionais:
  - Linux;
  - várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD);
  - Microsoft Windows;
  - Mac OS X;
  - RISC OS;
  - e provavelmente outros.

O QUE O PHP PODE FAZER?

- o PHP também é suportado pela maioria dos servidores web atuais:
  - Apache;
  - Microsoft Internet Information Server;
  - Personal Web Server;
  - Netscape and iPlanet Servers;
  - Oreilly Website Pro Server;
  - Caudium;
  - Xitami;
  - OmniHTTPd;
  - e muitos outros.

O QUE O PHP PODE FAZER?

- o PHP pode ser configurado como módulo para a maioria dos servidores, e para os outros como um CGI comum.

O QUE O PHP PODE FAZER?

- com o PHP, portanto, você tem a liberdade para escolher o sistema operacional e o servidor web, bem como, você pode escolher entre utilizar programação estrutural ou programação orientada a objeto, ou ainda uma mistura deles.

O QUE O PHP PODE FAZER?

- com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem:
  - geração de imagens;
  - arquivos PDF;
  - e animações Flash (utilizando libswf ou Ming).

O QUE O PHP PODE FAZER?

- talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de banco de dados:

✓Adabas D
✓Ingres
✓Oracle (OCI7 and OCI8)
✓dBase
✓InterBase
✓Ovrimos
✓Empress
✓FrontBase
✓PostgreSQL
✓FilePro (read-only)
✓mSQL
✓Solid
✓Hyperwave
✓Direct MS-SQL
✓Sybase
✓IBM DB2
✓MySQL
✓Velocis
✓Informix
✓ODBC
✓Unix dbm

CONFIGURAÇÃO (PHP.INI)

- as configurações do PHP ficam armazenadas em um arquivo denominado php.ini e que é carregado cada vez que o servidor é iniciado;
- no Windows, ele fica na pasta c:\Windows;
- através de modificações neste arquivo é possível alterar várias opções no comportamento do PHP;
  - todas as linhas iniciadas por ponto-e-vírgula são comentários;
  - ao utilizar a ferramenta WAMP essa configuração está disponível em PHP > php.ini.

SINTAXE BÁSICA

- tags especiais indicam ao PHP onde estão os blocos de código;
- a tag de abertura é formada por um sinal de "menor que" (<), um sinal de interrogação (?) e a sigla php → <?php
- você ainda pode usar uma forma abreviada para a abertura das tags → <?
- a tag de fechamento é formada por um ponto interrogação (?) e sinal de "maior que" (>). → ?>

PRIMEIRO EXEMPLO

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
<?php
$a = 10;
$b = 15;
$c = $a + $b;
echo "$a mais $b é igual a $c";
?>
</body>
</html>
```

PRIMEIRO EXEMPLO

- o sinal de ponto-e-vírgula (;) indica o final de um comando;
- note que, o código php é inserido dentro de um código HTML;
- execute o script e veja qual será seu resultado.

PRIMEIRO EXEMPLO

- Você pode utilizar o modo php para customizar suas páginas em XHTML:

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
<?php
    $a = 10;
    if($a > 0){
?>
    <h1>A variável a é maior que 10!</h1>
<?
    }
    else{
?>
    <h1>A variável a é menor que 10!</h1>
<?
    }
?>
</body>
</html>
```

COMENTÁRIOS

- os comentários de mais de uma linha no PHP são obtidos através de /* e */;
- os comentários de apenas uma linha são obtidos através de // ou #;
- os comentários não aparecem no *browser.*

COMENTÁRIOS

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
<?php
$a = 10; //A variável $a recebe o valor 10
$b = 15; //A variável $b recebe o valor 15
# A variável $c recebe o valor da soma
$c = $a + $b;
/* O resultado obtido é exibido */
echo "$a mais $b é igual a $c";
?>
</body>
</html>
```

PALAVRAS-CHAVE DO PHP

- as seguintes palavras são reservadas:

✓and	✓var	✓switch
✓do	✓case	✓xor
✓for	✓elseif	✓continue
✓Include	✓function	✓false
✓require	✓new	✓If
✓true	✓static	✓or
✓break	✓virtual	✓this
✓else	✓class	✓while
✓foreach	✓extends	✓default
✓list	✓global	
✓return	✓not	

VARIÁVEIS

- variáveis armazenam valores. Pode-se referir a variáveis para obter seu valor ou para alterar seu conteúdo;
- no PHP elas são representadas por um cifrão ($) mais o nome da variável;
  - ex: $identificacao;
- os nomes de variáveis válidos são iniciados por letras ou por um subscrito (_);
  - ex: $num, $_idade;
- existe diferenciação entre nomes de variáveis maiúsculas e minúsculas;
  - Ex: $a, $_A, $_a.

EXEMPLO

```php
<?php
$a = 10;
$A = 20;
echo "O valor de '$a' é $a e o de '$A' é $A";
?>
```

Resultado:
O valor de $a é 10 e o valor de $A é 20

VARIÁVEIS

- as variáveis criadas dentro de uma função (function) valerão somente enquanto aquela função existir;
- para trabalhar com uma variável externa à função devemos definir a variável como global.

EXEMPLO

```php
<?php
function soma($a, $b){
        global $msg;
        return $msg.($a + $b);
}
$msg = "A soma é ";
echo soma(10, 2);
?>
```

Resultado:
A soma é 12

INTERPRETAÇÃO DE VARIÁVEIS

- é comum utilizarmos uma mistura de strings e variáveis para imprimir o conteúdo desejado;
- para que isso ocorra podemos utilizar 2 tipos de sintaxe:
  - um simples;
  - um complexo.

INTERPRETAÇÃO DE VARIÁVEIS

- sintaxe simples:
  - se um sinal de cifrão ($) é encontrado, o interpretador tentará obter vários identificadores para formar um nome válido para aquela variável;
  - envolva o nome da variável com chaves {} para especificar explicitamente o fim do nome;

```
$cerveja = 'Skol';
echo "Ele bebeu algumas $cervejas"; // não funciona
echo "Ele bebeu algumas ${cerveja}s"; // funciona
echo "Ele bebeu algumas {$cerveja}s"; // funciona
```

INTERPRETAÇÃO DE VARIÁVEIS

- sintaxe complexa:
  - é chamada de sintaxe complexa porque você pode incluir expressões complexas dessa maneira;
  - basta escrever a expressão da mesma maneira que faria fora da string, e posteriormente envolvê-la por chaves {};

```
echo "IMPRIMINDO ELEMENTO DO ARRAY: {$ARRAY[0][0]}";
echo "IMPRIMINDO ATRIBUTO DO OBJ: {$OBJ->NOME}";
```

INTERPRETAÇÃO DE VARIÁVEIS

- variáveis dentro de aspas duplas "" são interpretados e mostram seu valor;
- variáveis dentro de aspas simples '' não são interpretadas, e mostram o que está escrito entre as aspas;

```
echo "$var"; // imprime conteúdo de $var
echo '$var'; // imprime $var;
```

- para concatenar variáveis utiliza-se o caractere ponto .

```
$var1. " ".$var2;
```

ATIVIDADE

- construa uma aplicação MISTA (XHTML + PHP) que funcionará como uma calculadora;
  - as entradas de dados $n1, $n2 e $op serão declaradas no código;
  - construa uma função calcula que irá capturar as variáveis globais definidas fora da função;
  - trate uma possível divisão por zero, e imprima um erro dentro da tag <h1></h1>
  - o XHTML deverá informar:

```
O RESULTADO DA SOMA DA VARIÁVEL $n1 = 10 e $n2 = 20 é:

30
```