

Universidade da Região da Campanha
Centro de Ciências da Economia e Informática
Curso de Informática
Disciplina: Tópicos Especiais em Sistemas de Informação

Apostila de PHP

Prof. Cristiano Cachapuz e Lima

Sumário

1	Introdução.....	3
2	Exemplo de Script	3
3	Configuração (php.ini)	5
4	Sintaxe Básica	6
5	Variáveis.....	8
6	Tipos de dados.....	9
7	Constantes.....	12
8	Expressões	13
9	Operadores.....	14
10	Estruturas de Controle	18
11	Funções.....	22
12	Classes e Objetos.....	23
13	Referências	24
14	Matrizes	26
15	Inclusão de Arquivos.....	27
16	Cookies	28
17	Parâmetros	30
18	Formulários.....	31
19	Uploads.....	33
20	Envio de e-mails	34
21	Introdução ao MySQL.....	35
22	Exibição.....	35
23	Consulta e Ordenação.....	37
24	Inclusão e Atualização.....	38
25	Exclusão	40
	Referências	40

1 Introdução

PHP é uma sigla recursiva que significa *PHP HyperText Preprocessor*. O PHP é uma linguagem de código-fonte aberto, muito utilizada na Internet e especialmente criada para o desenvolvimento de aplicativos *Web*.

Note como isso é diferente de scripts CGI escritos em outras linguagens como Perl ou C --- 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 (nesse caso, imprimir um pouco de texto). O código PHP é delimitado por tags iniciais e finais que lhe permitem pular pra dentro e pra fora do “modo PHP”.

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.

Para testar scripts PHP é necessário um servidor com suporte a esta tecnologia. Normalmente, o mais utilizado é o Apache. O banco de dados mais utilizado com os scripts PHP é o MySQL. Um exemplo de pacote pronto para execução de um ambiente Apache + PHP + MySQL é o EasyPHP (<http://www.easyphp.org>). Qualquer editor de textos pode ser usado para escrever os scripts PHP (ex. bloco de notas ou a ferramenta *free* MPS PHP Designer em <http://www.mpssoftware.dk>).

As páginas PHP devem ser salvas no diretório raiz do servidor. Para testes locais com o EasyPHP, essa pasta é `c:\Arquivos de programas\EasyPHP\www`. Para acessar a página, deve-se abrir o *browser* Internet Explorer e digitar-se o nome do domínio (`http://127.0.0.1`) e o nome da página com extensão `.php`. Quando o EasyPHP está sendo executado, aparece um ícone com uma letra “e” ao lado do relógio do Windows.

2 Exemplo de 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>
```

Figura 1 – Primeiro script

No *browser*, digite o endereço `http://127.0.0.1/teste.php` e veja o resultado. Veja também o código fonte da página (Exibir → Código fonte). É interessante notar que os comandos PHP não aparecem porque o servidor interpreta todos os scripts antes de enviar a página para o *browser*.

O que PHP pode fazer ?

Qualquer coisa. O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como: coletar

dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

Esses são os maiores campos onde os scripts PHP podem se utilizar:

- Script no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para seu trabalho. O interpretador do PHP (como CGI ou módulo), um servidor *web* e um browser. Basta rodar o servidor web conectado a um PHP instalado. Você pode acessar os resultados de seu programa PHP com um browser, visualizando a página PHP através do servidor web.
- Script de linha de comando. Você pode fazer um script PHP funcionar sem um servidor *web* ou *browser*. A única coisa necessária é o interpretador. Esse tipo de uso é ideal para script executados usando o `cron` ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto.
- 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 PHP-GTK é uma extensão do PHP, não disponível na distribuição oficial. Se você está interessado no PHP-GTK, visite o site <http://gtk.php.net>.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS, e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atuais, incluindo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet Servers, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd, e muitos outros. O PHP pode ser configurado como módulo para a maioria dos servidores, e para os outros como um CGI comum.

Com o PHP, portanto, você tem a liberdade para escolher o sistema operacional e o servidor web. Do mesmo modo, você pode escolher entre utilizar programação estrutural ou programação orientada a objeto, ou ainda uma mistura deles. Mesmo não desenvolvendo nenhum recurso padrão de OOP (Object Oriented Programming, Programação Orientada a Objetos) na versão atual do PHP, muitas bibliotecas de código e grandes aplicações (incluindo a biblioteca PEAR) foram escritas somente utilizando OOP.

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) criados dinamicamente. Você pode facilmente criar qualquer padrão texto, como XHTML e outros arquivos XML. O PHP pode gerar esses padrões e os salvar no sistema de arquivos, em vez de imprimi-los, formando um cache dinâmico de suas informações no lado do servidor.

Talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página que consulte um banco de dados é incrivelmente simples. Os seguintes bancos de dados são atualmente suportados:

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

Adicionalmente, o PHP suporta ODBC (Open Database Connection, ou Padrão Aberto de Conexão com Bancos de Dados), permitindo que você utilize qualquer outro banco de dados que suporte esse padrão mundial.

O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros. Você pode abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o intercâmbio de dados complexos WDDX, utilizado em virtualmente todas as linguagens de programação para *web*. Falando de comunicação, o PHP implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP. Você ainda pode usar sua extensão CORBA para acessar objetos remotos.

O PHP é extremamente útil em recursos de processamento de texto, do POSIX Estendido ou expressões regulares Perl até como interpretador para documentos XML. Para acessar e processar documentos XML, são suportados os padrões SAX e DOM. Você ainda pode usar nossa extensão XSLT para transformar documentos XML.

Utilizando o PHP no campo do e-commerce, você poderá usar as funções específicas para Cybescash, CyberMUT, Verysign Payflow Pro e C CVS, práticos sistemas de pagamento online.

Por último mas longe de terminar, temos também outras extensões interessantes: funções para o search engine mnoGoSearch, funções para Gateway IRC, vários utilitários de compressão (gzip, bz2), calendário e conversões de datas, tradução, etc.

3 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`.

Exemplo:

```
[PHP]

;;;;;;;;;;
; WARNING ;
;;;;;;;;;;
; This is the default settings file for new PHP installations.
; By default, PHP installs itself with a configuration suitable for
; development purposes, and *NOT* for production purposes.
; For several security-oriented considerations that should be taken
; before going online with your site, please consult php.ini-
recommended
```

```
; and http://php.net/manual/en/security.php.

;;;;;;;;;;;;;;;;;;;;;;;;;
; About this file ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; This file controls many aspects of PHP's behavior. In order for PHP
to
; read it, it must be named 'php.ini'. PHP looks for it in the
current
; working directory, in the path designated by the environment
variable
; PHPRC, and in the path that was defined in compile time (in that
order).
; Under Windows, the compile-time path is the Windows directory. The
; path in which the php.ini file is looked for can be overridden using
; the -c argument in command line mode.
;
; The syntax of the file is extremely simple. Whitespace and Lines
; beginning with a semicolon are silently ignored (as you probably
guessed).
; Section headers (e.g. [Foo]) are also silently ignored, even though
; they might mean something in the future.
```

Figura 2 – Trecho de exemplo do php.ini

Através de modificações neste arquivo é possível alterar várias opções no comportamento do PHP. Todas as linhas iniciadas por ponto-e-vírgula são comentários.

4 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. A tag de fechamento é formada por um ponto interrogação (?) e sinal de “maior que” (>).

Ex:
<?php
...
?>

Exercício: digite o código da figura 1 e salve no diretório raiz do servidor Apache. Veja o resultado da página através de seu carregamento no *browser*.

O sinal de ponto-e-vírgula (;) indica o final de um comando (ver figura 1). A próxima figura mostra outro 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>
```

Figura 3 – Exemplo de código

Os comentários de mais de uma linha no PHP são obtidos através de /* e */. Os comentários de apenas uma linha são obtidos através de //.

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>

<?php
/*
O código abaixo soma duas variáveis
e exibe o valor encontrado
*/
$a = 10;
$b = 15;
$c = $a + $b;
echo "$a mais $b é igual a $c";
?>

</body>
</html>
```

Figura 4 – Exemplo de código com comentários de mais de uma linha

Os comentários não aparecem no *browser*.

```
<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>
```

Figura 5 - Exemplo de código com comentários de uma linha

Palavras-chave do PHP

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

5 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. Os nomes de variáveis válidos são iniciados por letras ou por um subscrito (_). Existe diferenciação entre nomes de variáveis maiúsculas e minúsculas.

Ex: \$a, \$_A, \$_a

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>

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

</body>
</html>
```

Figura 6 – Exemplo de código com variáveis minúscula e maiúscula

Quando a variável é declarada dentro de uma função, ela só estará disponível para o código desta função. O código a seguir gera um erro devido a essa característica.

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>

<?php

function soma($a)
{
    $b = $a + 5;
}

soma(10);

echo "o valor de 'b' é $b";
```



```
?>
</body>
</html>
```

Figura 7 – Declaração de variável dentro de função

Para evitar este tipo de problema, deve-se definir a variável como global. O código a seguir resolve o problema do código anterior. Compare os resultados dos dois scripts.

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>

<?php

function soma($a)
{
    global $b;
    $b = $a + 5;
}

soma(10);

echo "o valor de 'b' é $b";

?>

</body>
</html>
```

Figura 8 – Declaração de variável global

6 Tipos de dados

O PHP suporta vários tipos de dados:

Inteiro – Números inteiros (isto é, números sem ponto decimal)

Números de dupla precisão – Números reais (isto é, números que contêm um ponto decimal)

String – Texto entre aspas simples (‘ ’) ou duplas (“ ”)

Booleanos – armazenam valores verdadeiros ou falsos, usados em testes de condições

Array – Grupo de elementos do mesmo tipo

Objeto – Grupo de atributos e métodos

Recurso – Uma origem de dados externa

Nulo – Nenhum valor

```
<html>
<head>
<title>Teste PHP</title>
</head>
<body>
```

```

<?php
$a = True;

if ($a)
{
    echo "Verdadeiro";
}
else
{
    echo "Falso";
}

?>

</body>
</html>

```

Figura 9 – Código com dados booleanos

Teste o código anterior alterando o valor da variável para False.

Pode-se armazenar valores inteiros, positivos ou negativos. Pode-se usar também valores hexadecimais.

```

<html>
<head>
<title>Teste PHP</title>
</head>
<body>

<?php

$a = 0x1A; //Coresponde ao decimal 26
$b = -16;
$c = $a + $b;

echo "a + b = $c";

?>

</body>
</html>

```

Figura 10 – Código com variáveis hexadecimal e valor negativo

Os valores de ponto flutuante são representados através de ponto (.).

```

<html>
<head>
<title>Teste PHP</title>
</head>
<body>

<?php

$preco = 11.90;

```

Gracias por visitar este Libro Electrónico

Puedes leer la versión completa de este libro electrónico en diferentes formatos:

- HTML(Gratis / Disponible a todos los usuarios)
- PDF / TXT(Disponible a miembros V.I.P. Los miembros con una membresía básica pueden acceder hasta 5 libros electrónicos en formato PDF/TXT durante el mes.)
- Epub y Mobipocket (Exclusivos para miembros V.I.P.)

Para descargar este libro completo, tan solo seleccione el formato deseado, abajo:

