

Programação PHP

Para começarmos a programar em PHP primeiro abra o seu editor de texto ou IDE preferido, analise entre as melhores IDE para PHP e tabela comparativo entre as melhores IDE para PHP ambos em inglês. Uma boa saída é o Aptana e o plugin para desenvolvimento em PHP mais você pode utilizar o seu preferido.

Delimitadores De Código PHP

Assim como o HTML temos as tags no PHP temos os delimitadores de código, que são `<?php` e `?>` respectivamente a tag de abertura e a tag de fechamento, em que o código deve ser inserido. Veja o exemplo a seguir:

1	<code><?php</code>
2	
3	<code>// Código PHP aqui!!!</code>
4	
5	<code>?></code>

Além dos delimitadores de código `<?php` e `?>` todas as suas páginas devem possuir a extensão `.php` por exemplo: `meu-primeiro-programa.php`

Ou ainda como mencionado no artigo anterior sobre o que é o PHP podemos **misturar o PHP com o HTML** como pode ser observado no exemplo a seguir.

01	<code><?php</code>
02	
03	<code>// Sim também podemos ter código PHP antes do DocType.</code>
04	
05	<code>?></code>
06	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"></code>
07	<code><html xmlns="http://www.w3.org/1999/xhtml"></code>
08	<code><head></code>
09	<code><meta http-equiv="Content-Type" content="text/html; charset=utf-8" /></code>
10	<code><title>Titulo da minha página</title></code>
11	<code><?php</code>
12	
13	<code>// Código PHP aqui dentro do head.</code>
14	
15	<code>?></code>
16	<code></head></code>

17		
18		<body>
19		<?php
20		
21	// Código PHP aqui dentro do body!!!	
22		
23		?>
24		</body>
25		</html>
26		<?php
27		
28	// Código PHP aqui até depois de fecharmos a tag html!	
29		
30		?>

Como você pode perceber podemos escrever código PHP misturado com o HTML em qualquer parte do código. Além do exemplo que eu mostrei imagine que você queira gerar o título da sua página pelo PHP dinamicamente o que você faria? Isso mesmo colocaria as tags do PHP dentro da tag title e realizaria uma rotina para gerar o título dinamicamente.

Caso você já tenha programado algo em PHP ou baixado algum script em PHP talvez já tenha se deparado com as tags respectivamente de abertura e fechamento assim:

1. <? ?>
2. <% %>
3. <script language="PHP"> </script>

O mais utilizado é o primeiro caso, no entanto **não o utilize**. Pois com o avanço do PHP este tipo de escrita pode ser desabilitado e ainda pode ocorrer a possibilidade de seu servidor web não aceitar este tipo de tag.

Por isso prefira sempre as tags <?php e ?>.

Conceito De Variável

Variáveis como você já deve estar imaginando apenas pelo nome é tudo aquilo que é sujeito a **variações**, que é **incerto**, **instável** ou **inconstante**, ou seja, pode **mudar durante sua existência**. Vamos a uma analogia para entendermos o conceito de variáveis. Imagine a seguinte situação: Joãozinho mora na casa de seus pais desde que nasceu e passa a maior parte do dia em seu quarto.

Com o passar dos anos o quarto de Joãozinho foi mudando, seus brinquedos foram trocados por um computador, seu berço foi trocado por uma cama, seu guarda-roupa de criança foi trocado por um de adulto, as paredes foram pintadas de cores diferentes e Joãozinho ganhou uma televisão em seu quarto. Como você já deve ter imaginado a nossa variável aqui foi o quarto de Joãozinho, que com o passar do tempo foi variando seu conteúdo.

Trazendo o conceito de variável ao PHP e a nossas páginas de internet. Voltemos ao exemplo anterior em que misturamos PHP e HTML. O conteúdo que ficaria dentro da tag body gerado pelo PHP seria uma variável, que dependo da página que estaríamos acessando este valor, ou seja, ele seria variável. A página inicial teria um valor naquela variável e página de contato já teria um valor diferente em sua variável.

As Variáveis No PHP

Agora que você já possui um conceito de variável iremos ver como criar nossas variáveis no PHP. As variáveis servem para armazenar dados que podem ser utilizado em qualquer parte do programa. Para criar uma variável em PHP você deve seguir algumas regras, são todas bem fáceis de se acostumar, veja a seguir:

1. As variáveis são representadas por um cifrão (\$) seguido pelo nome da variável
2. O PHP é case sensitive, ou seja, ele diferencia minúsculas de maiúsculas. Sendo então \$nome diferente de \$Nome e \$NoMe
3. Nomes de variáveis devem ser iniciados apenas com uma letra ou _ (sublinhado) e podem ser seguido de letras ou algarismos, ou seja, números podem aparecer em outras posições exceto na primeira
4. Nunca utilize caracteres especiais em nome de uma variável como acentos (é í ó) cedilha (ç)

Exemplo De Variáveis Validas No PHP

Veja a seguir exemplos de nomes válidos para variáveis, baseadas nas regras que estudamos anteriormente.

1	<?php
2	
3	\$nome = 'Mauro George';
4	\$nascimento = '06/09/1989';
5	\$sobre_nome = 'Oliveira Tavares';
6	
7	?>

Exemplo de variáveis invalidas no PHP

Agora alguns nomes de variáveis invalidas no PHP que **não** devem ser utilizadas.

1	<?php
2	
3	\$20_anos_e_nome = 'Mauro George';
4	\$ nascimento = '06/09/1989';
5	\$*este_é_meu_sobrenome = 'Oliveira Tavares';
6	
7	?>

Como você já deve ter imaginado o sinal de = (igual) é utilizado para atribuir um valor a variável, aprenderemos mais sobre eles quando chegarmos em operadores de atribuição.

Separação De Instruções

Você também deve ter reparado no exemplo anterior que depois definirmos um valor a variável terminamos utilizando o sinal de ; (ponto e vírgula). Este é o separador de instruções do PHP, ou seja, sempre que definirmos uma variável ou imprimirmos algo na tela devemos utilizar o ;.

Enviando resultados ao browser, o uso do comando echo

Você deve estar imaginando acabei de criar varias variáveis mais como que eu faço para exibir o conteúdo que guardei em cada uma delas? E neste ponto que aprenderemos exibir o valor de nossas variáveis na tela. Mais primeiro exibiremos o mais clichê de todas as linguagens de programação o famoso "Hello World".

Aconselho a partir deste ponto criar uma pasta com o nome de **estudos** em seu diretório HTDOCS e vá testando os exemplos que serão mostrados a seguir.

Não se esquecendo de salvar os arquivos com a extensão .php

1	<?php
2	
3	echo 'Hello World';
4	
5	?>

Repare que ainda não exibimos os dados de uma variável apenas exibimos um texto comum.

Exibindo Dados Das Variáveis No PHP

Agora iremos exibir os valores de nossas variáveis no browser utilizando o PHP, para isso utilizaremos as variáveis que criamos anteriormente.

01	<?php
02	
03	\$nome = 'Mauro George';
04	\$nascimento = '06/09/1989';
05	\$sobre_nome = 'Oliveira Tavares';
06	
07	
08	echo \$nome;
09	echo \$sobre_nome;
10	echo \$nascimento;
11	

12	?>
----	----

Você deve ter percebido que como exibimos varias variáveis quando terminou o dado de uma ele “co-lou” com o dado da próxima variável mais isto será concertado quando chegarmos a concatenação de strings, mais neste ponto o interessante é vermos que conseguimos exibir o valor da variável.

Posso Exibir Dados Das Variáveis No PHP Misturado Com O HTML?

SIM! Como o PHP se mistura ao HTML podemos exibir os dados em conjuntos vejam um exemplo pratico.

01	<?php
02	
03	\$titulo = 'Aqui vai o titulo da minha página';
04	\$css = '<link rel="stylesheet" type="text/css" href="css/estilos.css" />';
05	\$conteudo = 'Aqui é o conteudo mais como não tenho nenhum... vai apenas um Hello World';
06	
07	?>
08	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
09	<html xmlns="http://www.w3.org/1999/xhtml">
10	<head>
11	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
12	<title><?php echo \$titulo; ?></title>
13	
14	<?php
15	echo \$css;
16	?>
17	</head>
18	
19	<body>
20	<?php
21	
22	echo \$conteudo;
23	
24	?>

25	
26	<p>Eu posso repetir o valor da variavel sempre que eu quiser sabia? Veja aqui o nosso titulo denovo "<?php echo \$titulo; ?>"</p>
27	<p>E não necessariamente deveria imprimir \$titulo apenas porque eu a declarei primeiro poderia imprimir \$css antes e depois \$titulo ou qualquer uma variavel isso vai de acordo com sua necessidade</p>
28	<p>Agora irei exibir \$conteudo novamente olhe ela ai</p>
29	<p><?php echo \$conteudo; ?></p>
30	
31	</body>
32	</html>

Repare que além do PHP esta misturado ao HTML também a tags de HTML no valor da minha variável em PHP.

As variáveis podem ser exibidas sempre que necessário na tela, veja que repeti algumas variáveis mais de uma vez.

E ainda independente da ordem de declaração das variáveis elas podem ser chamadas a sua escolha. Mesmo eu tendo declarado \$titulo, \$css e \$conteudo eu poderia exibir na tela primeiro \$conteudo e \$css e por ultimo exibir \$titulo a ordem de exibição sou eu que faço.

No entanto o seguinte exemplo geraria um erro:

01	<?php
02	
03	echo \$nome;
04	echo \$sobre_nome;
05	echo \$nascimento;
06	
07	\$nome = 'Mauro George';
08	\$nascimento = '06/09/1989';
09	\$sobre_nome = 'Oliveira Tavares';
10	
11	?>

Você sabe me dizer o por quê? Se você respondeu que devido a tentarmos exibir uma variável sem antes declará-la você acertou. Ou seja, antes de exibirmos qualquer dado antes devemos o ter declarado previamente.

Comentários no PHP

O PHP nos fornece um suporte a comentários que vem a ser muito útil quando estamos desenvolvendo sistemas. Temos os seguintes tipos de comentários no PHP:

1. // comentário de uma linha apenas
2. # também comentário de uma linha apenas
3. /* */ comentário de múltiplas linhas

Veja todos eles em funcionamento abaixo:

01	<?php
02	
03	// Comentario de uma linha apenas
04	# Outro modo de escrever um comentario de uma linha
05	
06	/* Comentario de varias linhas com
07	este podemos quebra linha e escrever
08	quantas linhas quisermos.
09	*/
10	
11	// Apenas um nome
12	\$nome = 'Mauro George';
13	# A data de nascimento de uma pessoa
14	\$nascimento = '06/09/1989';
15	// O Sobre nome de uma pessoa
16	\$sobre_nome = 'Oliveira Tavares';
17	
18	/* Aqui exibimos o nome
19	o sobre nome e a data de nascimento de
20	uma pessoa
21	*/
22	echo \$nome;
23	echo \$sobre_nome;
24	echo \$nascimento;
25	

26	?>
----	----

Constantes no PHP

Como você já deve ter imaginado as constantes no PHP guardam valores que nunca serão alterados. Diferente das variáveis que possuem valores que podem ser alterados, sendo assim após definida uma constante ela não pode ser alterada ou removida.

Para definirmos uma constante utilizamos o comando `define()`; que tem sua sintaxe a seguir:

`define('NOME_DA_CONSTANTE', 'VALOR DA CONSTANTE');`

O nome de uma constante tem a mesma regra de qualquer identificador PHP, ou seja, as mesmas regras de nomes de variáveis exceto pelo fato de constantes **não** iniciarem o nome com cifrão (\$). Veja um exemplo a seguir em que utilizamos uma constante.

01	<?php
02	// Defino o titulo da minha página
03	\$titulo = 'Exemplo utilizando Constantes';
04	// Apenas um nome
05	\$nome = 'Mauro George';
06	// A data de nascimento de uma pessoa
07	\$nascimento = '06/09/1989';
08	// O Sobre nome de uma pessoa
09	\$sobre_nome = 'Oliveira Tavares';
10	// Defino o ESTADO da pessoa que é uma constante
11	define('ESTADO', 'Rio de Janeiro');
12	
13	?>
14	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15	<html xmlns="http://www.w3.org/1999/xhtml">
16	<head>
17	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
18	<title><?php echo \$titulo; ?></title>
19	</head>
20	
21	<body>
22	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>

23	</body>
24	</html>

Agora que você entendeu o funcionamento de uma constante deve estar se perguntando: qual a diferença real entre variáveis e constantes?

Acompanhe o exemplo a seguir e observe que variáveis podem ter seus valores modificados durante a execução de uma página, já as constantes seus valores nunca podem ter seus valores alterados.

01	<?php
02	// Defino o titulo da minha página
03	\$titulo = 'Exemplo utilizando Constantes';
04	// Apenas um nome
05	\$nome = 'Mauro George';
06	// A data de nascimento de uma pessoa
07	\$nascimento = '06/09/1989';
08	// O Sobre nome de uma pessoa
09	\$sobre_nome = 'Oliveira Tavares';
10	// Defino o ESTADO da pessoa que é uma constante
11	define('ESTADO', 'Rio de Janeiro');
12	
13	?>
14	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15	<html xmlns="http://www.w3.org/1999/xhtml">
16	<head>
17	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
18	<title><?php echo \$titulo; ?></title>
19	</head>
20	
21	<body>
22	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>
23	<?php
24	/**
25	* Redefino os valores das seguintes variaveis

26	*
27	*/
28	\$nome = 'Jéssica';
29	\$nascimento = '12/07/1990';
30	\$sobre_nome = 'Monteiro da Silva';
31	define('ESTADO', 'São Paulo');
32	
33	?>
34	<p><?php echo \$nome; ?> <?php echo \$sobre_nome; ?>, nascido em <?php echo \$nascimento; ?> nasceu no <?php echo ESTADO; ?></p>
35	</body>
36	</html>

Por padrão sempre escreva o nome de constantes em maiúsculo e separado por _ (sublinhado). Exemplo:

MINHA_CONSTANTE, UMA_CONSTANTE, CONSTANTE

E não se esqueça que constantes não são iniciadas com o cifrão (\$).

PHP: Como Dar Os Primeiros Passos

Se você sempre teve vontade de programar em PHP, mas nunca soube por onde começar, saiba que não está sozinho. Embora seja possível encontrar praticamente qualquer tipo de conteúdo na internet, muitas vezes os aspirantes a programadores ficam um pouco perdidos com tanta informação e não conseguem encontrar um ponto de partida para a sua empreitada.

Às vezes, ter alguém para apontar o caminho certo é tudo o que uma pessoa precisa para se encontrar no mundo da programação. Por isso, o Tecmundo resolveu criar este guia, mostrando onde encontrar as informações necessárias para começar a programar em PHP e também o que é importante saber e fazer antes de avançar para os níveis mais complexos. Confira!

Por Onde Começar?

Antes de começar a codificar uma página, é importante configurar o seu computador com os aplicativos necessários. Normalmente, quando um programador vai realizar alguma alteração em um site, o trabalho é feito primeiro em uma cópia local dos endereços para só então ser enviado para o servidor. Dessa forma, não há o risco de o portal ficar fora do ar devido a alguma falha durante a manutenção. Para ter essa cópia local funcionando corretamente no seu computador, é preciso instalar e configurar ferramentas como o Apache, PHP e MySQL. Assim, você consegue simular um ambiente de servidor na máquina e abrir a página no navegador mesmo sem estar conectado à internet.

A parte de configuração das ferramentas citadas é de extrema importância, pois garante a exibição correta do conteúdo programado. Para quem nunca realizou a tarefa de instalação dos aplicativos, ter um guia mostrando todos os passos por perto é uma ótima ideia.

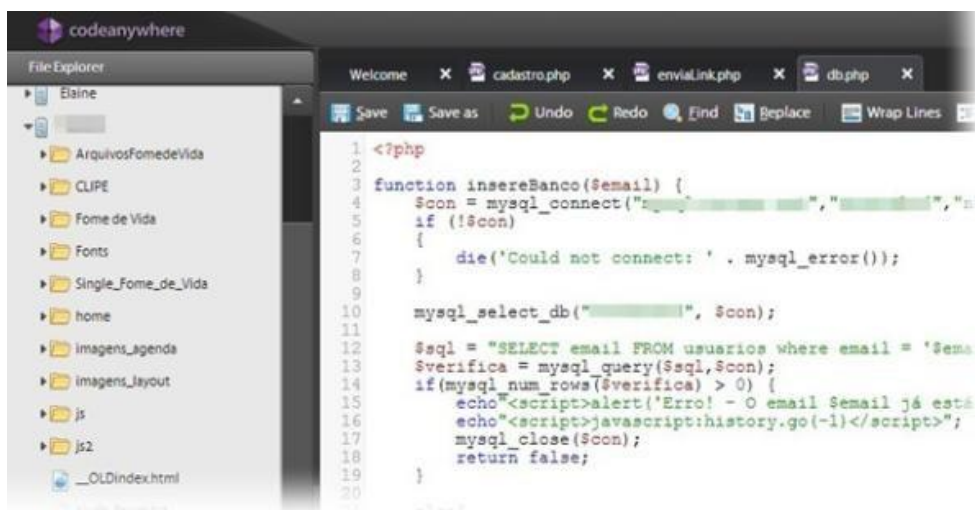


Se você digitar no Google os termos “instalar apache php mysql windows”, certamente vai encontrar diversos fóruns e tutoriais mostrando todo o processo de forma detalhada. Além disso, existem alguns pacotes que facilitam bastante a tarefa de instalação, pois oferecem todos os recursos necessários para rodar sua página localmente em uma só aplicação. Confira abaixo algumas dessas ferramentas.

- XAMPP
- EasyPHP
- PHP Triad
- WampServer

Escolhendo Um Bom Editor

Outra ferramenta indispensável para um programador é o editor ou IDE utilizado na hora de mexer no código. Uma ferramenta não precisa ter dezenas de funcionalidades para ser boa, executando as funções básicas de forma satisfatória pode ser o suficiente.



Cada programador possui a sua ferramenta preferida. Tenha em mente que funcionalidades como autocompletar e destaque de sintaxe não são obrigatórias, mas ajudam muito na hora de programar, principalmente se você está começando e não lembra muito bem os comandos de cabeça. Boas opções de editor para PHP (e outras linguagens também) são:

- PHP Editor
- PHP Anywhere
- Bluefish
- Eclipse PDT
- Notepad++

Agora Sim, Programar!

Se você decidiu criar um site em PHP, mas não tem ideia de como começar a programar, uma boa ideia é entender melhor alguns conceitos de programação, saber bem como funciona a linguagem PHP e procurar alguns tutoriais bem básicos na internet.

Apesar de já instalado e configurado o Apache, PHP e MySQL, entender a utilidade de cada um desses itens pode ser de grande ajuda, pois, caso haja algum problema no futuro, fica mais fácil diagnosticar o que o está causando.

O **Curso de PHP em Vídeo Aulas**, disponível para download no Baixaki, pode ser uma ótima forma de estudar alguns conceitos e aprender na prática os passos básicos para programar em PHP. No site **Apostilando.com** também é possível encontrar diversas apostilas gratuitas que explicam muito bem os conceitos para quem deseja embarcar nesse mundo do desenvolvimento de sites.

Há também o portal **PHP.net**. Além de explicar bem o que é o PHP (não a linguagem), você encontra diversos tutoriais ensinando a instalar e configurar o serviço no seu computador.

Outro serviço que certamente é muito útil para os programadores web é o **w3schools.com**. Nele, você encontra tutoriais, exemplos e explicações detalhadas de utilização para quase todos os comandos utilizados nas linguagens de desenvolvimento mais comuns para a internet. O conteúdo encontra-se apenas em inglês.

Apesar de ter um nome um tanto peculiar, o **Dummies.com** é um bom lugar para encontrar informações básicas a respeito de programação e desenvolvimento com PHP. Infelizmente, o site não possui tradução para o português, mas não é preciso dominar muito o inglês para compreender os textos disponíveis.

Não há muitos segredos para desenvolver um site em PHP, mas é preciso ter um pouco de paciência no começo e insistir. Se você não entender algum conceito ou termo presente nas apostilas, livros ou portais, procure os fóruns de programadores e não tenha vergonha de perguntar.

O que é o PHP?

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

Ótimo, mas o que isso significa? Por exemplo:

Exemplo #1 Um exemplo introdutório

```
<!DOCTYPE HTML>
<html>
<head>
<title>Exemplo</title>
</head>
<body>

<?php
echo "Olá, eu sou um script PHP!";
?>
</body>
</html>
```

Em vez de muitos comandos para mostrar HTML (como acontece com C ou Perl), as páginas PHP contêm HTML em código mesclado que faz "alguma coisa" (neste caso, mostra "Olá, eu sou um script PHP!"). O código PHP é delimitado pelas instruções de processamento (tags) de início e fim `<?php` e `?>` que permitem que você pule para dentro e para fora do "modo PHP".

O que distingue o PHP de algo como o Javascript no lado do cliente é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse script, mas não sabe qual era o código fonte. Você pode inclusive configurar seu servidor web para processar todos os seus arquivos HTML com o PHP, e então não haverá realmente nenhum modo dos usuários descobrirem se você usa essa linguagem ou não.

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos para um programador profissional. Não tenha medo de ler a longa lista de funções do PHP. Pode entrar com tudo, o mais rápido que puder, e comece a escrever scripts simples em poucas horas.

Apesar do desenvolvimento do PHP ser focado nos scripts do lado do servidor, você pode fazer muito mais com ele. Veja sobre isso na seção *O que o PHP pode fazer?*, ou vá diretamente para tutorial introdutório se você estiver interessado em programação web.

O que o PHP pode fazer?

Qualquer coisa. O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer: 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 ser utilizados:

- Scripts no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar. O interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web. Você precisa rodar o servidor web conectado em uma instalação PHP. Você pode acessar os resultados de seu programa PHP com um navegador web, visualizando a página PHP através do servidor web. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP. Veja a seção das instruções de instalação para mais informações.

- Scripts de linha de comando. Você pode fazer um script PHP para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (Unix, Linux) ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples. Veja a seção *Utilizando o PHP em linha de comando* para mais informações.

- Escrever aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Você também tem a possibilidade de escrever aplicações multi-plataformas desse jeito. O PHP-GTK é uma extensão do PHP, não disponibilizada na distribuição oficial. Caso esteja interessado no PHP-GTK, visite » o site do projeto.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como 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 atualmente. Isso inclui o Apache, o IIS e muitos outros. Também qualquer servidor web que pode utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI.

Com o PHP, portanto, você tem liberdade de escolha de sistema operacional e de servidor web. Do mesmo modo, você pode escolher entre utilizar programação estruturada ou programação orientada a objeto (OOP), ou ainda uma mistura das duas.

Com PHP você não está limitado a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e até animações Flash (utilizando libswf ou Ming) criados dinamicamente, on the fly. 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 mostrá-los em tela, formando um cache no lado do servidor para seu conteúdo dinâmico.

Uma das características mais fortes e mais significativas do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página web consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de um banco de dados (e.g., mysql), ou usando uma camada de abstração como o PDO ou conectar a qualquer banco de dados que suporte o padrão "Open Database Connection" usando a extensão ODBC. Outros bancos de dados podem utilizar cURL ou sockets, como o CouchDB.

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 também pode abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o

