

PHP

1. Einleitung

PHP ist die Abkürzung für Hypertext Preprocessor (ursprünglich „personal Homepage Tools“ was auch die Abkürzung an sich erklärt). PHP ist ein OSS-Produkt (Open-source-software) und wird unter den Bedingungen der eigenen „The PHP License, version 3.0“ (http://www.php.net/license/3_0.txt) welche sogar noch mehr Freiheiten gewährt als die GPL. PHP selber ist im engeren Sinne keine Programmiersprache sondern gehört zur Klasse der Interpretersprachen, genauso wie z.B. Perl (obwohl für beide Sprachen bereits Compiler tools existieren).

Interpreter bedeutet, der Programmcode wird erst zur Laufzeit des Programms in Maschinensprache übersetzt, im Gegensatz zu Compilerprogrammen die schon vor dem Einsatz übersetzt wurden (wie z.B. C++, C, Java uvm.) Dieses Vorgehen ermöglicht es dem PHP-Skripter Änderungen am Code sehr schnell vorzunehmen.

2. Prinzip der Verwendung

PHP ist bekannt geworden als Webprogrammiersprache und findet auch fast ausschließlich (andere Versuche wie etwa PHP-GTK scheinen nicht sehr erfolgsversprechend zu sein) auf Webservern Verwendung. Durch den offenen Code existieren Portierung für alle gängigen Serverbetriebssysteme.

Der Besucher einer Homepage bekommt dabei nur indirekt mit, dass er eine Webseite besucht, die über PHP gesteuert wird. Der Ablauf ist wie folgt:

- User gibt Adresse in den Browser, dieser sendet einen Get-Request an den Server.
- Der Server beantwortet die Anfrage (HTTP) und sendet das Dokument zurück. Dies ist der Augenblick in dem PHP einsetzt. Der Webserver erkennt, ob das angeforderte Dokument in PHP geschrieben ist, öffnet es, liest den Quellcode ein, sendet ihn an den Interpreter und gibt die Informationen zurück an den Webserver. Dieser kann nun diese Ausgabe an den Benutzer zurücksenden. Daher erscheint es für den Benutzer genauso, als ob er ein normales HTML-Dokument aufgerufen hätte.
- Der Browser des Benutzers interpretiert nun ebenfalls die ihm zugesendeten Informationen (anhand der Informationen im Header) und schickt gegebenenfalls weitere Get-Anfragen an den Webserver zurück, um evtl. Noch Bilder die zur Darstellung notwendig sind vom Server zu laden (*ein Bild wäre hierzu nicht schlecht*)

3. Grundlagen

Ein PHP-Dokument kann sowohl PHP-Code als HTML-Code enthalten. Der PHP-Code steht dabei stets umklammert von `<? ?>` bzw. `<?php ?>`

Alles zwischen diesen Klammern wird vom Interpreter geparkt, alles außerhalb der Klammern wird unverändert (genau wie in einem HTML-Dokument) an den Benutzer zurückgeschickt.

Kommentare werden zwischen `/*` und `*/` bzw. innerhalb einer Zeile nach `//` oder `#` geschrieben. Die Syntax orientiert sich stark am ANSI-C++-Standard.

Auffallend ist, wie leicht es einem die Sprache macht mit ihr zu arbeiten. Variablen müssen beispielsweise nicht extra deklariert werden, der Debugger ist äußerst fehlertolerant. Eine Beispieldatei könnte etwa so aussehen:

```

<?
// Das hier ist ein gar lustig Script!
$mein_Satz = „Hallo Welt“;
echo $mein_Satz;
?>

```

Das \$-Zeichen symbolisiert eine Variable (Schreibweise case-sensitive!)
 Eine Befehlszeile wird mit einem ; beendet.
 Der Befehl echo gibt die variable aus.

Eine einfache Rechenoperation:

```

<?
// Das ist ein Rechenscript
$a = 40;
$i = 0;
while($i < 9) {
    $a++;
    $i++;
}
$ergebnis = sqrt($a);
echo $ergebnis;
?>

```

Das Script initialisiert zwei Variablen, a mit dem Wert 40 und i mit dem Wert 0, daraufhin folgt eine Schleife die insgesamt a um den Wert 9 erhöht.
 Nun wird der Variable ergebnis das Ergebnis der Quadratwurzeloperation zugewiesen und anschließend ausgegeben.

```

<?
// Vergleiche
$x = 5;
$y =4;
if($x <= $y) { echo „X ist kleiner als y“; } else { echo „X ist größer als y!“; }
?>

```

Dieses Script zeigt eine logischen Operator, diese sind wie in fast jeder modernen Programmiersprache mit Bedeutungen versehen.

Es ist aber relativ langweilig, wenn man dem Script keine Daten zur Verarbeitung mitteilen kann außer denen, die bereits im Quellcode definiert sind. Da die PHP-Skripte meistens Serverseitig ablaufen, muss man sich auf die durch das Protokoll vorgegeben Möglichkeiten stützen. Es existieren dabei (im wesentlichen) 3 Übergabearten:

- GET-Methode (das was im Browser in der Adresszeile steht)
- POST-Methode (für Formulare)
- Cookie-Methode (Cookies wie das Wort schon selber sagt)

Darüber hinaus können noch serverbezogene Variablen wie etwa die eigene URL, Document_Root, die IP-Adresse, serverseitige Sessiondaten uvm abgerufen werden. Dies geschieht über global verfügbare Arrays (\$_GET, \$_POST, \$_SERVER usw.)

Ein weiterer Grund für die große Verbreitung von PHP (knapp 10 Millionen Domains weltweit) ist die einfache Möglichkeit Daten auf dem Webserver zu hinterlegen. Während zu Beginn das Speichern der Daten in Textdateien eine gute Lösung zu sein schien, hat man heute die vielen Vorteile von Datenbanken erkannt. PHP wurde auch extra darauf ausgelegt, mit einer Vielzahl von Datenbanken ohne Probleme kommunizieren zu können, besonders populär ist dabei die ebenfalls kostenlose MySQL-Datenbank (www.mysql.com). Die Abfrage geschieht über die Abfragesprache SQL

4. Abschlussbemerkung

Es ist natürlich nicht möglich in so einem kurzen Zeitraum eine gute Einführung in PHP zu präsentieren. Deshalb möchte ich an dieser Stelle gleich vorstellen, wie man zu Hause am besten PHP lernen kann. Zunächst braucht man eine Laufzeitumgebung, dafür bietet sich ein eigener Webserver an. Als sehr erfolgreich hat sich der Apache-Freeware-Webserver erwiesen:

www.apache.org einfacher geht es mit einem bereits vorkonfigurierten Paket, etwa von www.apachefriends.de

Zur Entwicklung kann eigentlich jeder beliebige Texteditor verwendet werden, einfach geht es etwa mit PHPEdit von www.phpedit.org

Gute Tutorials findet man unter anderem auf:

<http://www.php.net/manual/de/>
<http://www.schattenbaum.net/php/>
<http://www.galileocomputing.de/openbook/php4/>
<http://ffm.junetz.de/members/reeg/>

Besser beraten ist man jedoch mit Büchern, etwa PHP4-Grundlagen und Profiwissen von Jörg Krause, Hanserverlag, ISBN: 3446218475

Eine Besonderheit an PHP dürfte wohl auch die einzigartige Community sein, welche begeistert durch den OpenSourcegedanken eine mittlerweile unüberblickbare Anzahl an fertigen Programmen aller Art, Codeschnipsel und Hilfe zur Verfügung stellt. Wer ein bestimmtes Script sucht, wird fast immer auf:

<http://php.resourceindex.com>
<http://www.hotscripts.com>

fündig.

PHP kann also Websites dynamisch machen, helfen diese aktuell zu halten, miteinander zu verknüpfen und portabler zu machen. PHP hat jedoch nichts mit dem Aussehen oder dem Benutzer direkt zu tun.

Abschliessend noch einige bekannte Anwendungen:

- Diskussionsforum (etwa <http://www.thwboard.de/support/>)
- News-Script um nicht immer die eigene Startseite verändern zu müssen
- Gästebuch
- Formmailer uvm.