PHP

PHP ist eine Skriptsprache ähnlich wie Python oder so, und auch genauso einfach. So richtig witzig ist es aber in Verbindung mit HTML, weil man da PHP direkt reinschreiben kann.

Der Aufruf schaut in HTML so aus :

```
<html>
<body>
das ist ein test, diese zeile ist normales html
<?php
print "das hier ist php-code";
?>
</body>
</html>
```

So ein File muß dann .phtml oder .php (je nach Server-konfiguration) heißen, damit der Webserver das kapiert...

Hinter jede Zeile ; setzen

Variable : $name = "Alf";
$zahl = 176;
$klasse[1] = "Anton";
$klasse[2] = "Berta";

Eingabe : braucht man nicht, Variablen kommen immer vom HTML-Client

Ausgabe : echo $name;
print $name; (identische Funktion..)
print "Name ist :".$name."
";
(Der Punkt fügt den Ausdruck zusammen)

Html-Tags werden als String ausgegeben.

Schleifen, Bedingungen :

```
if ($a>$b)
{
    print "a ist größer";
}
```

```
if ($a>$b)
{
    print "a größer";
}
else
{
    print "b größer";
}
```

```
$i=1;
while ($i<10)
{
    print  $i;
    $i = $i+1;
}
```

```
for ($i=1; $i<10; $i++)
{
    print $i;
}
```

....usw, also das übliche !

Beispiel für server-sided Datenverarbeitung :

1) HTML-File mit enthaltenem HTML-Form. Das wird zunächst mit dem Browser geholt. Damit wir uns besser verstehen und das übersichtlicher wird, mache ich die Struktur ab jetzt immer so.
Diese HTML-Seite mit dem Form ruft das PHP auf, ich nenne sie deswegen die „aufrufende Seite“:

```
<html>
<body>
hallo, ich bin html <br><br>
<form action = "./aktion.php" method = "post">
        <input type=”text” name="alf"><br><br>
        <input type="submit">
</form>
</body>
</html>
```

Am Browser erscheint „hallo, ich bin html“ und das Eingabefenster für die form-Daten. Ein Textfeld holt einen String, der in die Variable „alf“ geschrieben wird Mit dem „submit“-Knopf wird der Wert an den Server geschickt und das HTML-File „aktion.php“ aufgerufen, welches den PHP-Code zur Verarbeitung enthält .

2) Die vom Form aufgerufene HTML-Seite aktion.php mit ausführbarem Code drin nenne ich die „aufgerufene Seite“.

```
<html>
<body>
hallo, hier HTML !
<?php
   print „hallo, hier PHP “;
   print "Die Eingabe war : $_POST[alf]“;
?>
</body>
</html>
```

Wird vom Apache also aus dem DocumentRoot geholt, und dann nach dem <?php untersucht. Alles zwischen diesem Zeichen und dem ?> wird an den PHP-Interpreter übergeben. Alle Ausgaben (print..) gehen über den cgi-Mechanismus zurück an den Browser und werden angezeigt.

Ein kleines Problem hierbei ist die Syntax, eigentlich dürfte :
print "Die Eingabe war : $_POST[alf]“

gar nicht funktionieren, weil der Variablenzugriff ja innerhalb eines Strings geschieht, und nicht ausgewertet werden sollte. Geht aber bei PHP doch, wenn es die richtige PHP-Version ist.

Alternative Syntax wäre :
print "Die Eingabe war :“. $_POST[alf];

Das schaut syntaktisch wesentlich sympathischer aus, hier wird der String „Die Eingabe war :“ mit dem „Verbindungspunkt“ an die Ausgabe der Variable angeheftet.

Probieren Sie aus, welche Variante bei ihrem PHP funktioniert, man kann es auch bei Goolge nachblättern .....

lächerliche PHP-Übung

1. Schreiben Sie eine HTML-Seite und dazu ein PHP-Skript, das am Webclient über ein Form Ihren Namen einliest, und dann über PHP wieder ausgibt : "hallo Herr ……."

2. Probieren Sie den Aufruf aus einem Links aus, und übergeben Sie einen Parameter, der dann am Client angezeigt werden soll

3. Schreiben Sie eine HTML-Seite, die das Eingeben eines Vor- und eines Familiennames erlaubt. Diese Werte schicken Sie dann mit POST an eine PHP-Seite, die untersuchen soll, ob der Vorname oder der Familienname der Ihre ist. Ist das der Fall, soll „Hallo Meister" ausgegeben werden, wenn nicht : „Hallo Fremder"