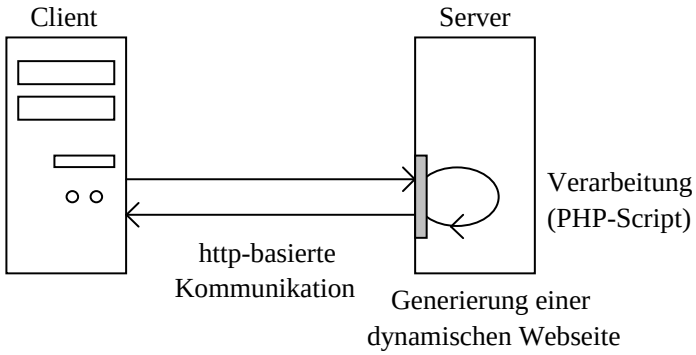
PHP – Eine Einführung

Das Server – Client – Prinzip

Der Client erteilt einen Arbeitsauftrag, der *serverseitig* abgearbeitet wird.
Das Ergebnis wird als HTML-Dokument zurückgesendet.



Als *Web-Server* nutzen wir einen APACHE-Server, der

- Dateien in einem Web-Verzeichnis bereithalten kann (z.B.: **`http://gauss/Lehrer203/test2.html`**) und
- Programme in der Script-Sprache PHP abarbeiten kann (z.B.: **`http://gauss/Lehrer203/test2.php`**).

Alle erforderlichen Dateien werden zunächst lokal erstellt und anschließend mit einem *FTP-Client* im Web-Verzeichnis abgelegt („hochgeladen“).

Das erste PHP-Script

Erstellen Sie in einem Editor das folgende Script, speichern Sie es unter dem Namen „`test1.php`“ ab und laden Sie anschließend die Datei in Ihr Verzeichnis auf dem Webserver hoch! Betrachten Sie danach die Datei in Ihrem Browser!




```
<html>
 <head>
  <title>PHP-Testprogramm</title>
 </head>
 <body>
  <h2>Mein erstes PHP-Programm</h2>

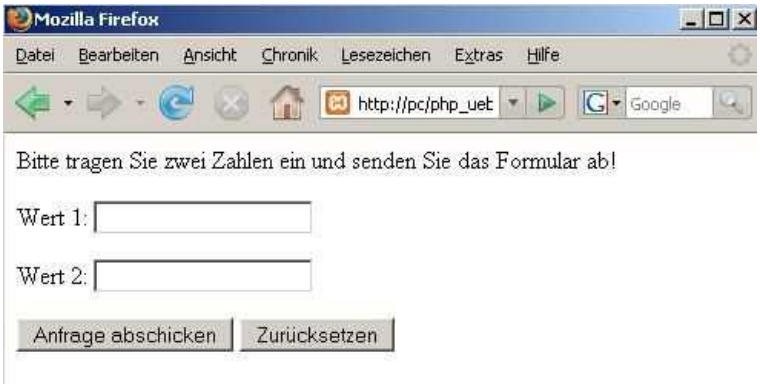
  <?php // Beginn des PHP-Scripts
   $Fach = "Profilinformatik"; // Definition einer Variable mit Wert
   echo "Das Unterrichtsfach ".$Fach." macht richtig Spa&szlig;!";
   echo "<br><br>"; // HTML-Tags im PHP-Script
   $Summand1 = 12;
   $Summand2 = 33;
   $Summe = $Summand1 + $Summand2;
   echo "Und nun rechnen wir: ".$Summand1." + ".$Summand2." = ".$Summe;
   echo "<br><br>";
   echo "<p>Der PHP-Einstieg ist ja ganz einfach!</p>";
  ?> // Ende des PHP-Scripts

 </body>
</html>
```

Verkettungsoperator

Wie kann der Client mit dem Server *variabel* kommunizieren?

Variablenübergabe mittels POST



Im Allgemeinen wird man die Server-Kommunikation *formularbasiert* organisieren.
Dazu wird das Formular als „gewöhnliches“ HTML-Dokument erstellt:

Übung 1

```
<html>
<body>
<p>Bitte tragen Sie zwei Zahlen ein und senden Sie das Formular ab!</p>
<form action="test2.php" method="post">
  <p>Wert 1: <input name = "w1"></p>
  <p>Wert 2: <input name = "w2"></p>
  <input type = "submit">
  <input type = "reset">
</form>
</body>
</html>
```

Die Attribute `action="test2.php" method="post"` sorgen für die korrekte Parameterübergabe. Die entsprechenden Daten müssen nun vom PHP-Script aufgenommen und verarbeitet werden:

```
html>
<body>
<?php
  $erg = $_POST["w1"] + $_POST["w2"]; // Auswertung der vom Formular übergebenen
                                      Variablen
  echo "Die Summe von " . $_POST["w1"] . " und " . $_POST["w2"] . " ist $erg.";
?>
</body>
</html>
```

Die vom Server zurückgelieferte Antwort sollte dann wie in dem nebenstehenden Bild aussehen.



Übung 2

Es soll eine Webseite zu Berechnungen am Kreiszylinder erstellt werden. Nach Eingabe von Radius und Höhe des Kreiszylinders in einem Formular soll das zugehörige PHP-Script den Oberflächeninhalt und das Volumen des Kreiszylinders berechnen und auf einer neu erstellten Webseite an den Browser zurückgeben (siehe Abbildungen). Erstellen Sie dafür zunächst das entsprechende Formular!

```
<form action="kreiszylinder.php" method="post">
 <table border="0" cellpadding="2">
  <tr>
   <td align="right">Radius in cm:   r =</td>
   <td align="left"><input name="r" value="3.45" size="10" type="text"></td>
  </tr>
  <tr>
  <tr>
   <td align="right">Höhe in cm:   h =</td>
   <td align="left"><input name="h" value="12.25" size="10" type="text"></td>
  </tr>
  <tr>
   <td> </td>
  </tr>
  <tr>
   <td align="right"><input value="Berechnen" type="submit"></td>
   <td align="left"><input value="Abbrechen" type="reset"></td>
  </tr>
 </table>
</form>
```

Und nun das PHP-Script zur Verarbeitung der Eingabedaten. Ergänzen Sie die fehlenden Codezeilen! Orientieren Sie sich dabei an den obigen Beispielen!

```
<?php
 $r = $_POST["r"];
 $h = $_POST["h"];
 $volumen = pi()*$r*$r*$h;
 $oberflaeche = 2*pi()*$r*($r+$h);
 ...
?>
```

