**Lektion 3: Ihre erste PHP-Seite**

Aus den Lektionen 1 und 2 wissen Sie bereits ein wenig, was PHP ist und Sie haben Zugang zu einem Webserver, bzw. haben diesen auf Ihrem Rechner installiert. Nun sind wir bereit, unsere erste PHP-Seite zu erstellen. Wir halten diese so einfach wie möglich, aber nachdem Sie diese Lektion durchgearbeitet haben, werden Sie noch wesentlich mehr Verständnis dafür haben, was PHP ist und was man damit machen kann.

Im Grunde genommen ist eine PHP-Datei eine Textdatei mit der Dateiendung **.php**, die aus den folgenden Teilen besteht/bestehen kann:

* Text
* HTML-Tags
* PHP-Skripte

Sie wissen schon, was Text und HTML-Tags sind. Also schauen wir jetzt ein wenig genauer auf die PHP-Skripte.

**PHP-Code**

Die PHP Documentation Group hat eine detaillierte [[Dokumentation](http://php.net/manual/en/index.php)Dokumentation für PHP](http://php.net/manual/en/index.php) herausgegeben. Während des Tutorials wird es viele Links zu dieser Dokumentation geben. Das Ziel ist, dass Sie sich daran gewöhnen, selbst nachzuschauen und Antworten auf Ihre Fragen zu finden. PHP ist so umfangreich, dass man nicht alle Facetten in diesem Tutorial beleuchten kann. Aber PHP ist nicht schwierig! Im Gegenteil, PHP ist oft ziemlich ähnlich zur Englischen Sprache.

Fangen wir nun mit Ihrer ersten PHP-Seite an.

**Beispiel: Hallo Welt!**

Starten Sie, indem Sie ein normales HTML-Dokument erstellen, aber nennen Sie die Datei *page.php* und speichern Sie es im Hauptverzeichnis (root) Ihrer Seite. Wenn Sie XAMPP nutzen (siehe [Lektion 2](http://de.html.net/tutorials/php/lesson2.php)), ist der Pfad "c:\xampp\htdocs\page.php" auf Ihrem Computer (der nun (auch) ein Server ist).

Der HTML-Code sollte so aussehen:

<html>

<head>

<title>Meine erste PHP-Seite</title>

</head>

<body>

</body>

</html>

Wie Sie sich vielleicht aus der Lektion 1 erinnern, ist PHP nichts anderes, als **einem Server Anweisungen zu geben**. Also geben wir dem Server eine Anweisung.

Erst müssen wir dem Server mitteilen, wann bzw. wo das PHP-Skript **beginnt** und wann es **endet**. Für PHP müssen wir die Tags **<?php** und **?>** benutzen, um Anfang und Ende des PHP-Codes, die der Server ausführen soll zu markieren. Auf vielen Servern sollte es auch genügen, nur **<?** als Starttag zu verwenden, aber **<?php** ist am 'korrektesten'. Bitte gewöhnen Sie sich diese Schreibweise an.

Jetzt fügen Sie bitte das folgende Code-Schnipsel (engl. snippet) in Ihren HTML-Code ein:

<html>

<head>

<title>Meine erste PHP-Seite</title>

</head>

<body>

<?php

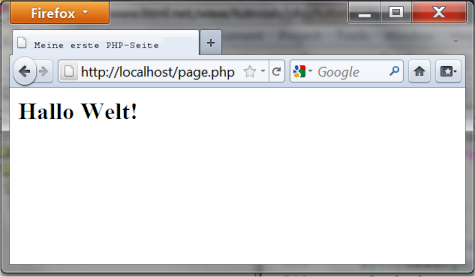
echo "<h1>Hallo Welt!</h1>";

?>

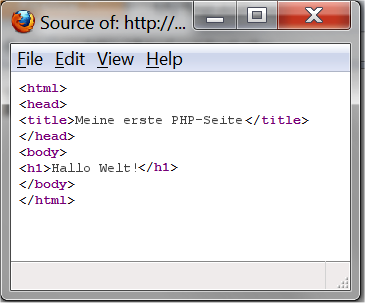
</body>

</html>

Wenn wir jetzt das PHP-Dokument in einem Browser aufrufen, sollte es so aussehen:



Aber es wird interessant, wenn Sie sich den HTML-Code im Browser anschauen (über "Seitenquelltext anzeigen"):



Der PHP-Code ist weg! Wie Sie sich vielleicht aus Lektion 1 erinnern, es ist nur der Server, der diesen Code sieht - **der Client (also der Browser) sieht nur das Ergebnis!**

Lassen Sie uns nachschauen, was passiert ist. Wir haben den Server aufgefordert <h1> Hello Welt!</h1> zu schreiben. In einer mehr technischen Sprache könnte man sagen, dass wir die String-Function [[Dokumentation](http://php.net/manual/en/function.echo.php)echo](http://php.net/manual/en/function.echo.php) benutzt haben, um dem Client eine festgelegte Zeichenfolge auszugeben, wobei das Semikolon die Anweisung beendet. Aber keine Angst! In diesem Tutorial werden wir versuchen, die technische Sprache auf ein Minimum zu beschränken.

Unser erstes Beispiel ist, zugegebenermaßen, nicht besonders aufregend. Aber immer langsam! Von jetzt an wird es immer interessanter. Schauen wir uns ein anderes Beispiel an.

**Beispiel: Jetzt!**

Lassen Sie uns dem Server eine andere Aufgabe geben. Wir könnten ihn z.B. das aktuelle Datum und die aktuelle Zeit ausgeben lassen:

<html>

<head>

<title>Meine erste PHP-Seite</title>

</head>

<body>

<?php

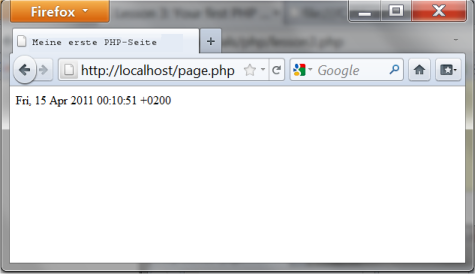
echo date("r");

?>

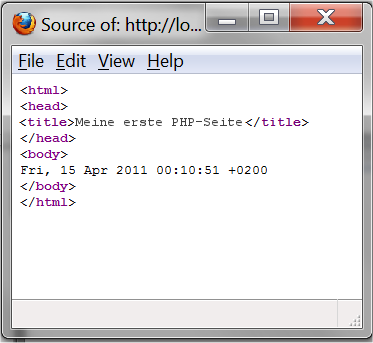
</body>

</html>

Das sieht im Browser so aus:



Und der zugehörige HMTL-Code:



Jetzt wird's interessant - Oder?

Wir lassen den Server das Datum und die Zeit des Anzeigezeitpunktes unserer Seite ausgeben. Sie werden merken, dass beim Neuladen der Seite im Browser immer eine andere Zeit ausgegeben wird. Der Server schreibt immer wieder das aktuelle Datum und die aktuelle Zeit, jedes Mal, wenn die Seite zu einem Client gesendet wird.

Es ist auch wichtig anzumerken, dass der Quellcode nur Datum und Zeit enthält, nicht aber den PHP-Code. Darum hat auch der verwendete Browser keinen Einfluß auf dieses Beispiel. Tatsächlich funktioniert alles, was mit *serverseitigen* Technologien bewerkstelligt wird, immer **in jedem Browser!**

Haben Sie auch wieder das Semikolon am Ende der Codezeile bemerkt? Dieser Separator ist sehr wichtig - ohne ihn würde Ihr Skript nicht funktionieren.

In diesem Beispiel haben wir [[Dokumentation](http://php.net/manual/en/function.date.php)date](http://php.net/manual/en/function.date.php) benutzt. Diese Function liefert das aktuelle Datum und die aktuelle Zeit auf dem Server.

Probieren wir das Beispiel zu erweitern, indem wir beides, den *String* und eine *Function*, verwenden - getrennt von "**.**" (einem Punkt) - und so wird's gemacht:

<html>

<head>

<title>Meine erste PHP-Datei</title>

</head>

<body>

<?php

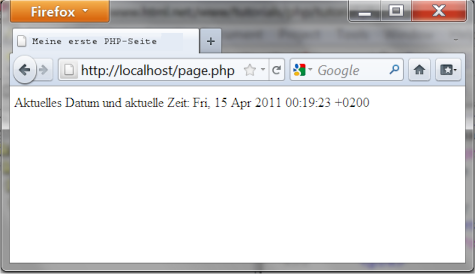
echo "<p>Aktuelles Datum und aktuelle Zeit: " . date("r") . "</p>";

?>

</body>

</html>

Im Browser wird es so aussehen:



Und der entsprechende HTML-Quelltext:



In der nächsten Lektion werden wir uns näher mit der Function [[Dokumentation](http://php.net/manual/en/function.date.php)date](http://php.net/manual/en/function.date.php) und den verschiedenen Formaten und Formatierungen für Datum und Zeit beschäftigen.