**Lektion 4: Arbeiten mit Zeit und Daten**

In dieser Lektion, werden wir versuchen, die vielen verschiedenen Optionen für die Arbeit mit Zeiten und Daten in PHP zu beleuchten. Wir haben in der vergangenen Lektion ein paar sehr einfache Bespiele bearbeitet, einfach um zu zeigen, was PHP eigentlich ist. In dieser Lektion betrachten wir die PHP-eigene [[Dokumentation](http://php.net/manual/en/function.date.php)date](http://php.net/manual/en/function.date.php)-Function.

**Zeit- und Datumsfunktionen**

PHP bringt eine breite Auswahl an [Functions bzgl. Zeit und Datum](http://php.net/manual/en/ref.datetime.php) mit. In dieser Lektion schauen wir uns die wichtigste dieser Funktionen an: Dokumentationdate.

Mit verschiedenen Parametern kann die Dokumentationdate-Function das aktuelle Datum/Zeit in vielen verschiedenen Formaten/Formatierungen wiedergeben. Einige der nützlichsten Parameter sind:

date("y")

Gibt das aktuelle Jahr eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **13**

date("m")

Gibt den aktuellen Monat eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **01**

date("F")

Gibt den Namen des aktuellen Monats eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **January**

date("d")

Gibt den aktuellen Tag eines Monats eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **12**

date("l")

Gibt den Namen des aktuellen Wochentages eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **Saturday**

date("w")

Gibt den aktuellen Tag der Woche eines gegebenen Datums aus - mit dem heutigen Datum ist dies: **6**

date("H")

Gibt die aktuelle Stunde einer gegebenen Zeit aus - mit der aktuellen Zeit ist dies: **13**

date("i")

Gibt die aktuelle Minute einer gegebenen Zeit aus - mit der aktuellen Zeit ist dies: **53**

date("s")

Gibt die aktuelle Sekunde einer gegebenen Zeit aus - mit der aktuellen Zeit ist dies: **15**

Dieses Beispiel zeigt den Gebrauch der Dokumentationdate-Function:

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body>

<?php

echo "<p>Heute ist " . date("l") . "</p>";

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex1.php)

**Die Zeit ist 1357995195**

Und jetzt kurz innehalten ..., weil es ein wenig "nerdy" wird! Die Function [[Dokumentation](http://www.php.net/manual/en/function.time.php)time()](http://www.php.net/manual/en/function.time.php) gibt die aktuelle Zeit als Anzahl der Sekunden, die seit dem 01. Januar 1970, 12:00 Uhr mittags GMT (= Greenwich Zeit) vergangen sind, aus.

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body>

<?php

echo "<p>Es sind genau " . time() . " Sekunden seit dem 01. Januar 1970, 12:00 Uhr GMT vergangen</ p> ";

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex2.php)

Die Zeit, die in vergangenen Sekunden seit dem 01. Januar 1970, 12:00 Uhr GMT ausgedrückt wird, ist ein sog. "timestamp" (auch UNIX timestamp) und ist ziemlich nützlich, wenn man mit Daten/Zeiten in der Zukunft oder der Vergangenheit arbeitet.

Standardmäßig nutzt die [[Dokumentation](http://www.php.net/manual/en/function.date.php)date](http://www.php.net/manual/en/function.date.php)-Function den aktuellen timestamp (z.B. den aktuellen Wert von [[Dokumentation](http://www.php.net/manual/en/function.time.php)time()](http://www.php.net/manual/en/function.time.php)). Aber mit einem extra Parameter kann man einen anderen timestamp festlegen und somit mit der Zukunft oder der Vergangenheit arbeiten. Im folgenden Beispiel setzen wir den timestamp auf null (0) Sekunden seit dem 01. Januar 1970, 12:00 Uhr mittags GMT. So können wir herausfinden, welcher Wochentag der 01. Januar 1970 gewesen ist.

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body>

<?php

echo "<p>Der 01. Januar 1970 war ein " . date("l",0) . "</p>";

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex3.php)

Wenn Sie nicht gerade ein Mathematik-Genie sind, wird es schnell schwierig, die Anzahl der Sekunden seit dem 01. Januar 1970, zu einem bestimmten Datum in der Zukunft oder der Vergangenheit zu zählen. Aber hier gibt es Hilfe von einer anderen raffinierten Function: [[Dokumentation](http://www.php.net/manual/en/function.mktime.php)mktime](http://www.php.net/manual/en/function.mktime.php), diese führt diese Berechnung für Sie aus.

Die Syntax (Reihenfolge der Argumente) für [[Dokumentation](http://www.php.net/manual/en/function.mktime.php)mktime](http://www.php.net/manual/en/function.mktime.php) ist *(Stunde, Minute, Sekunde, Monat, Tag, Jahr)*. Das folgende Beispiel konvertiert den Zeitpunkt des ersten Schrittes auf dem Mond (21. Juli 1969, 02:56 Uhr):

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body>

<?php

echo mktime (2,56,0,7,21,1969);

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex4.php)

Beachten Sie, dass eine negative Zahl ausgegeben wird, weil das Datum vor dem 01. Januar 1970 liegt

Wir können dies jetzt zusammen mit der [[Dokumentation](http://www.php.net/manual/en/function.date.php)date](http://www.php.net/manual/en/function.date.php)-Function verwenden und so herausfinden, an welchem Wochentag dieses historische Ereignis stattfand.

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body>

<?php

echo date("l", mktime(2,56,0,7,21,1969));

?>

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex5.php)

**Wozu kann man das verwenden?**

All das mutet jetzt wahrscheinlich noch ein wenig theoretisch an. Wozu kann man eine Function wie [[Dokumentation](http://www.php.net/manual/en/function.time.php)time()](http://www.php.net/manual/en/function.time.php) benutzen? Oder noch wichtiger, wann werden Sie etwas lernen, was Sie auch auf Ihren Seiten nutzen können?

Die Antwort ist, dass Sie hier Bausteine (kennen)lernen - die einzigen Beschränkungen dafür, was Sie daraus bauen, sind Ihre Kreativität und Vorstellungskraft! Ich würde wagen zu sagen, dass Sie bereits mehr gelernt haben, als Sie denken. Können Sie sich z.B. vorstellen, dass Sie eine Webseite erstellen können, die an jedem Tag der Woche ein anderes Hintergrundbild hat und das in allen Browsern funktioniert?

Natürlich können Sie! Schauen Sie sich dieses Beispiel an:

<html>

<head>

<title>Zeit und Datum</title>

</head>

<body background="background\_<?php echo date("w"); ?>.png">

</body>

</html>

 - [Beispiel anzeigen](http://de.html.net/tutorials/php/lesson4_ex6.php)

Das obige Beispiel mit dem dynamischen Hintergrundbild erfordert nur, dass Sie sieben Bilder erstellen und diese wie folgt benennen: background\_1.png, background\_2.png, background\_3.png usw.

Wenn ein Benutzer Ihre Seite an einem Dienstag besucht, wird die Seite das Hintergrundbild background\_2.png haben, am nächsten Tag background\_3.png usw. Einfach oder?

In der nächsten Lektion lernen Sie einen neuen Baustein kennen, der Schleifen und Wiederholungen in Ihrem Skript bewerkstelligen kann.

PHP macht Spaß, oder?