

Das `date()` Objekt liefert zwar zahlreiche Informationen zu einem UNIX-Timestamp aber, auf Englisch. Um die Datumsangaben in Deutsch darzustellen, muss man nur ein Array mit den Wochen- bzw. Monatsnamen anlegen und über den numerischen Wochentag bzw. numerischen Monat auflösen.



Den aktuellen Wochentag auflösen

```
<?php
    $tage = array("Sonntag", "Montag", "Dienstag",
                 "Mittwoch", "Donnerstag", "Freitag", "Samstag,");
    $timestamp = time();
    $wochentagNr = date("w", $timestamp);
    echo $tage[$wochentagNr];
?>
```



Alternativ kann auch `strftime()` verwendet werden!

z. B. `echo strftime("%A", $timestamp);` liefert den Wochentag!
`echo strftime("%B", $timestamp);` liefert das Monat

PHP



`date_default_timezone_set()`

Setzt eine bestimmte Zeitzone. Im Beispiel bleibt der UNIX-Timestamp immer gleich (nämlich der aktuelle). Im Anschluss wird die Uhrzeit nach Greenwich-Mean-Time, nach Chicago und nach Wien-Zeit ausgegeben.

```
<?php
    $zeit = time();
    date_default_timezone_set('GMT');
    echo 'GMT-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
    date_default_timezone_set('America/Chicago');
    echo 'Chicago-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
    date_default_timezone_set('Europe/Vienna');
    echo 'Wien-Zeit: ' . date("H:i:s", $zeit) . '<hr>';
?>
```



Eine Liste der unterstützten Zeitzonen gibt es auf [php.net](https://www.php.net)
<https://www.php.net/manual/de/timezones.php>



Der UNIX-Timestamp wird gerne als Identifikationsnummer (ID) verwendet. Wer es besonders genau braucht, kann mit `microtime(true)` den Zeitstempel mit Mikrosekunden ausgeben. Die Rückgabe ist ein Float-Wert.

```
<?php
    $mikrozeit = microtime(true);
    echo $mikrozeit;
?>
```