

PHP Web Features



This project is funded by
the European Union



Maharah
Code Your Future

LibyanSpider



Web Features

PHP is a language created for the web. Its original purpose was to make it easier to make web pages and it remains heavily focused on server-side scripting. This chapter looks at some of the language features that make it one of the world's most popular server-side web programming languages.



Date and Time



Date and Time

- **Traditional**

Timestamp use function like **date()** and **time()**.

- **Object-oriented**

Use **DateTime** class

Timestamp

Timestamp uses an integer to represent date and time, this integer stores the number of seconds since midnight on 1/1/1970, this point in time is known as the Unix Epoch

Functions

- **time(): int**
return the current time measured in the number of seconds since the Unix Epoch
- **date(string \$format, ?int \$timestamp = null): string**
returns a string formatted according to the provided format string, utilizing the current timestamp if the second's parameter is not provided.
- **strtotime(string \$datetime, ?int \$baseTimestamp = null): int|false**
Parse about any English textual datetime description into a Unix timestamp

Formatting Dates

- Date and time formatting codes are case-sensitive. For example, “y” is a two-digit year and “Y” is a four-digit year.
- Characters in the formatting string that are not recognized as formatting characters will be placed into the output unchanged. So, the string “Y-m-d” would include the hyphens between the year, month, and day when output—like this “2015-12-25”.
- You can find a list of the PHP date and time formatting codes on the manual page,¹⁰ but here are the ones that are in the previous table:

Formatting Dates (2)

| Code | Replaced with | Example |
|------|---|----------------|
| Y | A full four-digit year | 1999 |
| M | Two-digit month, with leading zeroes | 06 |
| d | Day of the month, two digits with leading zeros | 14 |
| D | A three letter textual day | Mon, Tue, Wed |
| H | 24-hour format hour with leading zero | 00, 09, 12, 23 |
| i | Two-digit minute, with leading zeroes | 05,15,25,45 |
| s | Two-digit seconds, with leading zeroes | 05,15,25,45 |
| T | Time zone abbreviation | EST, CET |

<https://php.net/manual/en/function.date.php>

Examples

Here are examples of formatting dates in traditional ways using the date() function in PHP:

```
<?php
$timestamp = time();

$formattedDate = date("Y-m-d H:i:s", $timestamp);
echo $formattedDate; // 2022-06-15 14:30:00

$formattedDate = date("F j, Y, g:i a");
echo $formattedDate; // June 15, 2022, 2:30 pm

$formattedDate = date("m/d/y");
echo $formattedDate; // 06/15/22

$formattedDate = date("l");
echo $formattedDate; // Thursday

$formattedDate = date("jS F Y");
echo $formattedDate; // 15th June 2022
```


Examples (2)

Here are examples and explanatory content for using **strtotime**

```
<?php
$startDate = strtotime("2022-06-15");
$endDate = strtotime("+2 weeks", $startDate);
echo date("Y-m-d", $endDate); // 2022-06-18

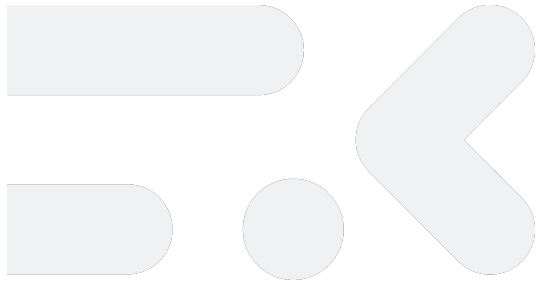
$customDateTimeString = "2022-12-31 18:30:00";
$timestamp = strtotime($customDateTimeString);
echo date("l, F j, Y g:i A", $timestamp); // Thursday, December 31, 2022 6:30 PM

$complexDateTimeString = "first day of next month 10:00";
$timestamp = strtotime($complexDateTimeString);
echo date("l, F j, Y g:i A", $timestamp); // Friday, January 1, 2023 10:00 AM
```

DateTime class

The DateTime class in PHP provides an object-oriented way to work with dates and times. Here are examples

```
<?php
$dateTime = new DateTime();
echo $dateTime->format('Y-m-d H:i:s'); // 2010-11-12 13:14:15
```

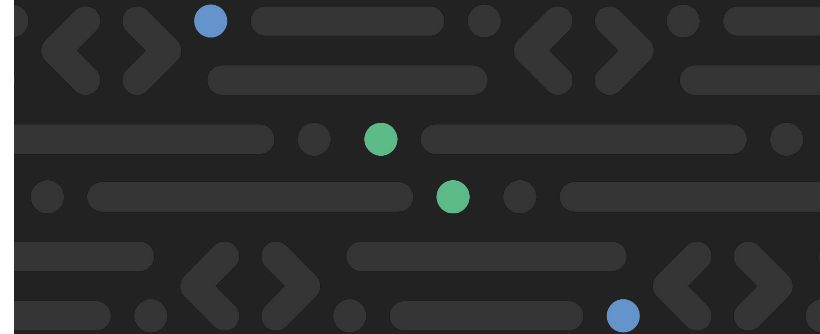
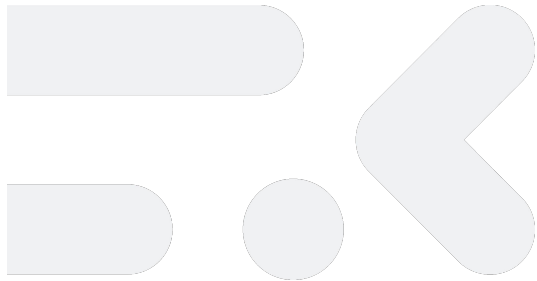




DateTime class

The DateTime class in PHP provides an object-oriented way to work with dates and times. Here are examples

```
<?php
$dateTime = new DateTime();
echo $dateTime->format('Y-m-d H:i:s'); // 2010-11-12 13:14:15
```



Comparing Dates

The **DateTime::diff()** method allows you to compare the difference between two DateTime objects. It returns a DateInterval that contains the period of time between the two dates being represented.

Note that the DateTime class handles time zone and daylight savings time conversions for you.

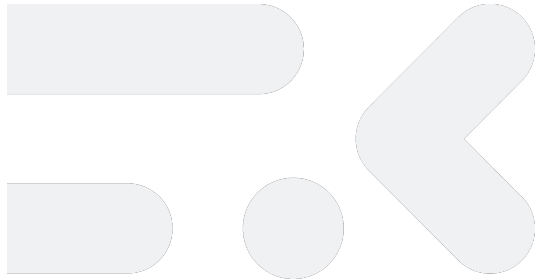
```
<?php
$now = new DateTime();
$maharahDateTime = new DateTime('07 january');
$interval = $maharahDateTime->diff($now);
echo "Maharah bootcamp started from " . $interval->days;
```



Timezone

To set the default timezone for date and time functions in PHP, you can use the `date_default_timezone_set` function. Here's an example:

```
<?php
// Set the default timezone to New York
date_default_timezone_set('Africa/Tripoli');
$dateTime = new DateTime();
echo $dateTime->format('Y-m-d H:i:s');
```





Sessions and Cookies

Sessions

Session in PHP is a way of temporarily storing and making data accessible across all the website pages

Starting a Session

Initiates a new session or resumes the existing one.

```
<?php  
session_destroy();
```

Ending a Session

Destroys all session data and is used to log out

```
<?php  
session_start();
```

Saving and Getting User Data

Saving User Data

\$_SESSION Superglobal: Associative array to store session data.

```
<?php
$_SESSION['username'] = 'john_doe';
$_SESSION['user_id'] = 123;
```

Retrieving Session Data

```
<?php
$username = $_SESSION['username'];
$userID = $_SESSION['user_id'];
```


Best Practices

- **Use HTTPS:** Encrypt data transmission with SSL/TLS.
- **Regenerate Session ID:** using `session_regenerate_id()` function
The `session_regenerate_id` function in PHP is typically used to enhance the security of sessions by regenerating the session ID. This is useful in situations where you want to prevent session fixation attacks.

```
<?php
session_start();
// Perform login logic
// ...
// Regenerate the session ID after successful login
session_regenerate_id(true);
```

Cookies

Cookies in PHP are a way to store small pieces of information on the client's browser. Here are some examples of working with cookies:

Save Cookie

```
<?php
setcookie('username', 'john_doe', time() + 3600, '/');
```

Retrieving Cookie

```
<?php
// Retrieve the value of a cookie
$username = $_COOKIE['username'];
echo "Welcome back, $username!";
```

Cookies (2)

Deleting a Cookie

```
<?php
// Delete a cookie by setting its expiration time to the past
setcookie('username', '', time() - 3600, '/');
```

Cookie with Array Values:

```
<?php
// Set a cookie with an array value
$userInfo = array('username' => 'john_doe', 'email' => 'john@example.com');
setcookie('user_info', json_encode($userInfo), time() + 3600, '/');
```

Any Questions



This project is funded by
the European Union



Maharah
Code Your Future

LibyanSpider