

# session\_start

(PHP 4, PHP 5, PHP 7)

session\_start — Start new or resume existing session

## Description

```
bool session_start ([ array $options = [] ] )
```

session\_start() creates a session or resumes the current one based on a session identifier passed via a GET or POST request, or passed via a cookie.

When session\_start() is called or when a session auto starts, PHP will call the open and read session save handlers. These will either be a built-in save handler provided by default or by PHP extensions (such as SQLite or Memcached); or can be custom handler as defined by [session\\_set\\_save\\_handler\(\)](#). The read callback will retrieve any existing session data (stored in a special serialized format) and will be unserialized and used to automatically populate the \$\_SESSION superglobal when the read callback returns the saved session data back to PHP session handling.

To use a named session, call [session\\_name\(\)](#) before calling session\_start().

When [session.use\\_trans\\_sid](#) is enabled, the session\_start() function will register an internal output handler for URL rewriting.

If a user uses [ob\\_gzhandler](#) or similar with [ob\\_start\(\)](#), the function order is important for proper output. For example, [ob\\_gzhandler](#) must be registered before starting the session.

## Parameters

### options

If provided, this is an associative array of options that will override the currently set [session configuration directives](#). The keys should not include the *session.* prefix.

In addition to the normal set of configuration directives, *read\_and\_close* option may also be provided. If set to **TRUE**, this will result in the session being closed immediately after being read, thereby avoiding unnecessary locking if the session data won't be changed.

## Return Values

This function returns **TRUE** if a session was successfully started, otherwise **FALSE**.

## Changelog

Version	Description
7.0.0	The <a href="#">options</a> parameter was added.
5.3.0	If a session fails to start, then <b>FALSE</b> is returned. Previously <b>TRUE</b> was returned.
4.3.3	As of PHP 4.3.3, calling <b>session_start()</b> after the session was previously started will result in an error of level <b>E_NOTICE</b> . Also, the second session start will simply be ignored.

## Examples

### A basic session example

Example #1 *page1.php*

```
<?php
// page1.php

session_start();

echo 'Welcome to page #1';

$_SESSION['favcolor'] = 'green';
$_SESSION['animal']   = 'cat';
$_SESSION['time']     = time();

// Works if session cookie was accepted
echo '<br /><a href="page2.php">page 2</a>';

// Or maybe pass along the session id, if needed
echo '<br /><a href="page2.php?" . SID . "'>page 2</a>';
?>
```

After viewing `page1.php`, the second page `page2.php` will magically contain the session data. Read the [session reference](#) for information on [propagating session ids](#) as it, for example, explains what the constant `SID` is all about.

## Example #2 *page2.php*

```
<?php
// page2.php

session_start();

echo 'Welcome to page #2<br />';

echo $_SESSION['favcolor']; // green
echo $_SESSION['animal'];   // cat
echo date('Y m d H:i:s', $_SESSION['time']);

// You may want to use SID here, like we did in page1.php
echo '<br /><a href="page1.php">page 1</a>';
?>
```

## Providing options to **session\_start()**

### Example #3 Overriding the cookie lifetime

```
<?php
// This sends a persistent cookie that lasts a day.
session_start([
    'cookie_lifetime' => 86400,
]);
?>
```

### Example #4 Reading the session and closing it

```
<?php
// If we know we don't need to change anything in the
// session, we can just read and close rightaway to avoid
// locking the session file and blocking other pages
session_start([
    'cookie_lifetime' => 86400,
    'read_and_close'  => true,
]);
```

## Notes

**Note:**

To use cookie-based sessions, `session_start()` must be called before outputting anything to the browser.

**Note:**

Use of [zlib.output\\_compression](#) is recommended instead of [ob\\_gzhandler\(\)](#)

**Note:**

This function sends out several HTTP headers depending on the configuration.  
See [session\\_cache\\_limiter\(\)](#) to customize these headers.