

Server variables

`$_SERVER` is an array containing information such as headers, paths, and script locations. The entries in this array are created by the web server. There is no guarantee that every web server will provide any of these; servers may omit some, or provide others not listed here. That said, a large number of these variables are accounted for in the » [CGI/1.1 specification](#), so you should be able to expect those.

<http://php.net/manual/en/reserved.variables.server.php>

Comments

```
//
```

```
Or
```

```
/*
```

```
*/
```

Stop error reporting on or off

```
error_reporting(0);
```

Sample course code

```
// Show difference between print and echo
```

```
/*
```

```
$myVariable = false;
```

```
$myVariable ? print "true<BR>" : print "false<BR>";
```

```
$myVar = print 'hello world';
```

```
*/
```

```
//$myVar = 'some value';
```

```
//error_reporting(0);
```

```
///echo $myVar;
```

```
echo $_SERVER['DOCUMENT_ROOT'];
```

Error Functions

- `debug_backtrace()`
- `debug_print_backtrace()`
- `error_get_last()`
- `error_log()`
- `error_reporting()`
- `restore_error_handler()`
- `restore_exception_handler()`
- `set_error_handler()`
- `set_exception_handler()`
- `trigger_error()`
- `user_error()`

Error Constants

- **E_ERROR1**
Fatal run-time errors. Errors that cannot be recovered from. Execution of the script is halted
- **E_WARNING2**
Non-fatal run-time errors. Execution of the script is not halted
- **E_PARSE4**
Compile-time parse errors. Parse errors should only be generated by the parser
- **E_NOTICE8**
Run-time notices. The script found something that might be an error, but could also happen when running a script normally
- **E_CORE_ERROR16**
Fatal errors at PHP startup. This is like an E_ERROR in the PHP core
- **E_CORE_WARNING32**
Non-fatal errors at PHP startup. This is like an E_WARNING in the PHP core
- **E_COMPILE_ERROR64**
Fatal compile-time errors. This is like an E_ERROR generated by the Zend Scripting Engine
- **E_COMPILE_WARNING128**
Non-fatal compile-time errors. This is like an E_WARNING generated by the Zend Scripting Engine
- **E_USER_ERROR256**
Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function `trigger_error()`
- **E_USER_WARNING512**
Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function `trigger_error()`
- **E_USER_NOTICE1024**
User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function `trigger_error()`
- **E_STRICT2048**
Run-time notices. PHP suggest changes to your code to help interoperability and compatibility of the code
- **E_RECOVERABLE_ERROR4096**
Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also `set_error_handler()`)
- **E_ALL6143**
All errors and warnings, except of level E_STRICT