**PHP**

**PHP:**The most popular server-side language on the web, [PHP](https://www.upwork.com/hiring/development/php-frameworks-hiring-a-php-developer/) is designed to pull and edit information in the database. It’s most commonly bundled with databases written in the SQL language. PHP was designed strictly for the web and remains one of the most widely used languages around. It’s easy to install and deploy, is staying competitive with lots of modern frameworks, and is the foundation for a number of content-management systems. *PHP-powered sites: WordPress, Wikipedia, Facebook*

Several predefined variables in PHP are "superglobals", which means they are available in all scopes throughout a script. There is no need to do global $variable; to access them within functions or methods.

These superglobal variables are:

* [$GLOBALS](http://php.net/manual/en/reserved.variables.globals.php)- An associative [array](http://php.net/manual/en/language.types.array.php) containing references to all variables which are currently defined in the global scope of the script. The variable names are the keys of the array.
* [$\_SERVER](http://php.net/manual/en/reserved.variables.server.php)- $\_SERVER is an array containing information such as headers, paths, and script locations. The entries in this array are created by the web server. There is no guarantee that every web server will provide any of these; servers may omit some, or provide others not listed here. That said, a large number of these variables are accounted for in the [» CGI/1.1 specification](http://www.faqs.org/rfcs/rfc3875), so you should be able to expect those.
* [$\_GET](http://php.net/manual/en/reserved.variables.get.php)- $HTTP\_GET\_VARS contains the same initial information, but is not a [superglobal](http://php.net/manual/en/language.variables.superglobals.php). (Note that $HTTP\_GET\_VARS and $\_GET are different variables and that PHP handles them as such)
* [$\_POST](http://php.net/manual/en/reserved.variables.post.php)- An associative array of variables passed to the current script via the HTTP POST method when using application/x-www-form-urlencodedor multipart/form-data as the HTTP Content-Type in the request. $HTTP\_POST\_VARS contains the same initial information, but is not a [superglobal](http://php.net/manual/en/language.variables.superglobals.php). (Note that $HTTP\_POST\_VARS and $\_POST are different variables and that PHP handles them as such)
* [$\_FILES](http://php.net/manual/en/reserved.variables.files.php)- An associative [array](http://php.net/manual/en/language.types.array.php) of items uploaded to the current script via the HTTP POST method. The structure of this array is outlined in the[POST method uploads](http://php.net/manual/en/features.file-upload.post-method.php) section. $HTTP\_POST\_FILES contains the same initial information, but is not a [superglobal](http://php.net/manual/en/language.variables.superglobals.php). (Note that $HTTP\_POST\_FILES and $\_FILES are different variables and that PHP handles them as such)
* [$\_COOKIE](http://php.net/manual/en/reserved.variables.cookies.php)- An associative [array](http://php.net/manual/en/language.types.array.php) of variables passed to the current script via HTTP Cookies. $HTTP\_COOKIE\_VARS contains the same initial information, but is not a [superglobal](http://php.net/manual/en/language.variables.superglobals.php). (Note that $HTTP\_COOKIE\_VARS and $\_COOKIE are different variables and that PHP handles them as such)
* [$\_SESSION](http://php.net/manual/en/reserved.variables.session.php)- An associative array containing session variables available to the current script. See the [Session functions](http://php.net/manual/en/ref.session.php) documentation for more information on how this is used. $HTTP\_SESSION\_VARS contains the same initial information, but is not a [superglobal](http://php.net/manual/en/language.variables.superglobals.php). (Note that $HTTP\_SESSION\_VARS and $\_SESSIONare different variables and that PHP handles them as such)
* [$\_REQUEST](http://php.net/manual/en/reserved.variables.request.php)- An associative [array](http://php.net/manual/en/language.types.array.php) that by default contains the contents of [$\_GET](http://php.net/manual/en/reserved.variables.get.php),[$\_POST](http://php.net/manual/en/reserved.variables.post.php) and [$\_COOKIE](http://php.net/manual/en/reserved.variables.cookies.php).
* [$\_ENV](http://php.net/manual/en/reserved.variables.environment.php)- These variables are imported into PHP's global namespace from the environment under which the PHP parser is running. Many are provided by the shell under which PHP is running and different systems are likely running different kinds of shells, a definitive list is impossible.

**PHP SESSION FUNCTIONS**

* [session\_abort](http://php.net/manual/en/function.session-abort.php) — Discard session array changes and finish session
* [session\_cache\_expire](http://php.net/manual/en/function.session-cache-expire.php) — Return current cache expire
* [session\_cache\_limiter](http://php.net/manual/en/function.session-cache-limiter.php) — Get and/or set the current cache limiter
* [session\_commit](http://php.net/manual/en/function.session-commit.php) — Alias of session\_write\_close
* [session\_create\_id](http://php.net/manual/en/function.session-create-id.php) — Create new session id
* [session\_decode](http://php.net/manual/en/function.session-decode.php) — Decodes session data from a session encoded string
* [session\_destroy](http://php.net/manual/en/function.session-destroy.php) — Destroys all data registered to a session
* [session\_encode](http://php.net/manual/en/function.session-encode.php) — Encodes the current session data as a session encoded string
* [session\_gc](http://php.net/manual/en/function.session-gc.php) — Perform session data garbage collection
* [session\_get\_cookie\_params](http://php.net/manual/en/function.session-get-cookie-params.php) — Get the session cookie parameters
* [session\_id](http://php.net/manual/en/function.session-id.php) — Get and/or set the current session id
* [session\_is\_registered](http://php.net/manual/en/function.session-is-registered.php) — Find out whether a global variable is registered in a session
* [session\_module\_name](http://php.net/manual/en/function.session-module-name.php) — Get and/or set the current session module
* [session\_name](http://php.net/manual/en/function.session-name.php) — Get and/or set the current session name
* [session\_regenerate\_id](http://php.net/manual/en/function.session-regenerate-id.php) — Update the current session id with a newly generated one
* [session\_register\_shutdown](http://php.net/manual/en/function.session-register-shutdown.php) — Session shutdown function
* [session\_register](http://php.net/manual/en/function.session-register.php) — Register one or more global variables with the current session
* [session\_reset](http://php.net/manual/en/function.session-reset.php) — Re-initialize session array with original values
* [session\_save\_path](http://php.net/manual/en/function.session-save-path.php) — Get and/or set the current session save path
* [session\_set\_cookie\_params](http://php.net/manual/en/function.session-set-cookie-params.php) — Set the session cookie parameters
* [session\_set\_save\_handler](http://php.net/manual/en/function.session-set-save-handler.php) — Sets user-level session storage functions
* [session\_start](http://php.net/manual/en/function.session-start.php) — Start new or resume existing session
* [session\_status](http://php.net/manual/en/function.session-status.php) — Returns the current session status
* [session\_unregister](http://php.net/manual/en/function.session-unregister.php) — Unregister a global variable from the current session
* [session\_unset](http://php.net/manual/en/function.session-unset.php) — Free all session variables
* [session\_write\_close](http://php.net/manual/en/function.session-write-close.php) — Write session data and end session