



# Linting WordPress PHP code

One thing I've always tried to do is lint my PHP code as I go when working with WordPress so it won't bite me later if I try to submit the theme or plugin to the WordPress repositories.

The best, and as far as I know only, way to efficiently lint PHP code is with [PHP Code Sniffer](#) and the [WordPress rules](#) to accompany it.

The process includes installing PHP Code sniffer, optionally creating a configuration file, and running the code, either from command line or from an existing build file.

## Installation Methods

There are many ways to install PHP Code Sniffer. I will cover the two I consider the easiest ones.

### Install with Homebrew on macOS

For macOS users the easiest way is to install PHP Code Sniffer via [Homebrew](#) with this command

```
brew install php-code-sniffer
```

### Install with Composer

The next easiest way is to install PHP Code Sniffer is to use [PHP Composer](phpcs --config-set tab\_width 4) dependency manager

Follow the instructions in [Installation - Linux / Unix / macOS](#) to install Composer.

Once it's set up, update it.

```
composer self-update && \  
composer global update
```

And do a global install of the required packages

```
composer g require --dev wp-coding-standards/wpcs \  
dealerdirect/phpcodesniffer-composer-installer -W
```

## Running the sniffer

I normally run the sniffer from the command line.

I redirect the test results to a text file (**-report-file=**) so I can read it later, tell it that I want to use the WordPress standard set of sniffers (**-standard=WordPress**) and to lint all the PHP files in the current and all children directory (**\*\*/\*.php**).

```
phpcs --report-file=PHPCS-report.txt \  
--standard=WordPress **/*.php
```

We can also incorporate this command into the scripts portion of a package.json file

```
scripts: {  
  "lint:php": "phpcs --report-file=PH"phpcs --report-file=PHPCS-report.txt
```

We can also use the same command to lint individual files; for example, if we're working on a file and want to lint to make sure we fixed problems in previous runs.

The following command line will lint functions.php and write the report to a text file.

```
phpcs --report-file=PHPCS-report-functionsphp.txt --standard=WordPress fun
```

# Further work

Although the WordPress PHPCS settings are meant to exclude Javascript files so they can be linted with other tools, PHPCS will still lint them.

You could also customize the WordPress Coding standard ruleset to make it work for you.

Customization is not always a good idea because it may prevent your theme or plugin from being accepted in the WordPress [theme](#) or [plugin](#) directories, but if you're working on your own content or have additional sniffs to run then creating a custom rule that includes all the rules we want to lint against.

See:

- PHP Coding Standard [annotated ruleset](#)
- [WordPress coding standards](#)
- [PHP compatibility](#)