### # Install phpcs Windows

<https://learnwithdaniel.com/2019/06/atom-phpcs-ubuntu/#wordpress-vip>

### # Requirements php and composer

**composer global require squizlabs/php\_codesniffer**

**composer global require --dev dealerdirect/phpcodesniffer-composer-installer**

**phpcs -i**

**composer global config bin-dir --absolute**

**composer global require --dev dealerdirect/phpcodesniffer-composer-installer**

### [#](https://learnwithdaniel.com/2019/06/atom-phpcs-ubuntu/#wordpress) [WordPress](https://github.com/WordPress-Coding-Standards/WordPress-Coding-Standards)

**composer global require wp-coding-standards/wpcs:\***

**composer global require --dev phpcompatibility/phpcompatibility-wp:\***

### [#](https://learnwithdaniel.com/2019/06/atom-phpcs-ubuntu/#wordpress-vip) [WordPress VIP](https://github.com/Automattic/VIP-Coding-Standards)

**composer global require --dev automattic/vipwpcs:\***

**composer global require wptrt/wpthemereview**

### # WooCommerce

**Run cmd as administrator**

**composer global require woocommerce/woocommerce-sniffs**

### # Install phpcs Ubuntu

#!/bin/bash

sudo apt update -y

sudo apt install wget curl -y

cd

wget https://az764295.vo.msecnd.net/stable/fd6f3bce6709b121a895d042d343d71f317d74e7/code\_1.54.2-1615424848\_amd64.deb

sudo chmod +x code\_1.54.2-1615424848\_amd64.deb

sudo dpkg -i code\_1.54.2-1615424848\_amd64.deb

sudo apt install -f

sudo apt-get install software-properties-common -y && sudo add-apt-repository ppa:ondrej/php -y && sudo apt-get update -y

sudo apt install php8.2 php8.2-{gd,oauth,curl,intl,bcmath,ctype,dom,iconv,mbstring,simplexml,soap,xsl,zip,mysql} -y

sudo curl -s https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

sudo chmod +x /usr/local/bin/composer

sudo apt update -y

composer global require squizlabs/php\_codesniffer

echo "export PATH=/home/cedcoss/.config/composer/vendor/bin:$PATH">>/home/cedcoss/.profile

source /home/cedcoss/.profile

composer global require --dev dealerdirect/phpcodesniffer-composer-installer

composer global require wp-coding-standards/wpcs:\*

composer global require --dev phpcompatibility/phpcompatibility-wp:\*

composer global require --dev automattic/vipwpcs:\*

composer global require wptrt/wpthemereview

mkdir -p /home/cedcoss/.config/Code/User

cat >> /home/cedcoss/.config/Code/User/settings.json << EOF

{

"phpcs.enable": true,

"phpcs.standard": "WordPress",

"phpcs.executablePath": null

}

EOF

# **PHPCS.XML**

**<?xml version="1.0"?>**

**<ruleset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="PHP\_CodeSniffer" xsi:noNamespaceSchemaLocation="phpcs.xsd">**

**<description>The coding standard for PHP\_CodeSniffer itself.</description>**

**<file>autoload.php</file>**

**<file>bin</file>**

**<file>scripts</file>**

**<file>src</file>**

**<file>tests</file>**

**<exclude-pattern>\*/src/Standards/\*/Tests/\*\.(inc|css|js)$</exclude-pattern>**

**<exclude-pattern>\*/tests/Core/\*/\*\.(inc|css|js)$</exclude-pattern>**

**<exclude-pattern>\*/tests/Core/\*/Fixtures/\*\.php$</exclude-pattern>**

**<arg name="basepath" value="."/>**

**<arg name="colors"/>**

**<arg name="parallel" value="75"/>**

**<arg value="np"/>**

**<!-- Don't hide tokenizer exceptions -->**

**<rule ref="Internal.Tokenizer.Exception">**

**<type>error</type>**

**</rule>**

**<!-- Include the whole PEAR standard -->**

**<rule ref="PEAR">**

**<exclude name="PEAR.NamingConventions.ValidFunctionName"/>**

**<exclude name="PEAR.NamingConventions.ValidVariableName"/>**

**<exclude name="PEAR.Commenting.ClassComment"/>**

**<exclude name="PEAR.Commenting.FileComment.MissingCategoryTag"/>**

**<exclude name="PEAR.Commenting.FileComment.MissingPackageTag"/>**

**<exclude name="PEAR.Commenting.FileComment.MissingLinkTag"/>**

**<exclude name="PEAR.Commenting.FileComment.MissingVersion"/>**

**<exclude name="PEAR.Commenting.InlineComment"/>**

**</rule>**

**<!-- Include some sniffs from other standards that don't conflict with PEAR -->**

**<rule ref="Squiz.Arrays.ArrayBracketSpacing"/>**

**<rule ref="Squiz.Arrays.ArrayDeclaration"/>**

**<rule ref="Squiz.Commenting.ClosingDeclarationComment"/>**

**<rule ref="Squiz.ControlStructures.ControlSignature"/>**

**<rule ref="Squiz.ControlStructures.ElseIfDeclaration"/>**

**<rule ref="Squiz.Commenting.BlockComment"/>**

**<rule ref="Squiz.Commenting.DocCommentAlignment"/>**

**<rule ref="Squiz.Commenting.EmptyCatchComment"/>**

**<rule ref="Squiz.Commenting.InlineComment"/>**

**<rule ref="Squiz.Commenting.LongConditionClosingComment"/>**

**<rule ref="Squiz.Commenting.PostStatementComment"/>**

**<rule ref="Squiz.Commenting.VariableComment"/>**

**<rule ref="Squiz.Formatting.OperatorBracket"/>**

**<rule ref="Squiz.Functions.FunctionDeclarationArgumentSpacing"/>**

**<rule ref="Squiz.Operators.ComparisonOperatorUsage"/>**

**<rule ref="Squiz.PHP.DisallowInlineIf"/>**

**<rule ref="Squiz.Scope.MethodScope"/>**

**<rule ref="Squiz.Strings.ConcatenationSpacing"/>**

**<rule ref="Squiz.WhiteSpace.ControlStructureSpacing"/>**

**<rule ref="Squiz.WhiteSpace.FunctionClosingBraceSpace"/>**

**<rule ref="Squiz.WhiteSpace.FunctionSpacing"/>**

**<rule ref="Squiz.WhiteSpace.MemberVarSpacing"/>**

**<rule ref="Squiz.WhiteSpace.OperatorSpacing"/>**

**<rule ref="Squiz.WhiteSpace.SuperfluousWhitespace"/>**

**<rule ref="Generic.Arrays.DisallowLongArraySyntax"/>**

**<rule ref="Generic.Commenting.Todo"/>**

**<rule ref="Generic.ControlStructures.DisallowYodaConditions"/>**

**<rule ref="Generic.ControlStructures.InlineControlStructure"/>**

**<rule ref="Generic.Formatting.DisallowMultipleStatements"/>**

**<rule ref="Generic.Formatting.SpaceAfterCast"/>**

**<rule ref="Generic.NamingConventions.ConstructorName"/>**

**<rule ref="Generic.PHP.DeprecatedFunctions"/>**

**<rule ref="Generic.PHP.LowerCaseKeyword"/>**

**<rule ref="Generic.Strings.UnnecessaryStringConcat"/>**

**<rule ref="Generic.WhiteSpace.IncrementDecrementSpacing"/>**

**<rule ref="PSR2.Classes.PropertyDeclaration"/>**

**<rule ref="PSR2.Methods.MethodDeclaration"/>**

**<rule ref="PSR2.Files.EndFileNewline"/>**

**<rule ref="PSR12.Files.OpenTag"/>**

**<rule ref="Zend.Files.ClosingTag"/>**

**<!-- PEAR uses warnings for inline control structures, so switch back to errors -->**

**<rule ref="Generic.ControlStructures.InlineControlStructure">**

**<properties>**

**<property name="error" value="true"/>**

**</properties>**

**</rule>**

**<!-- We use custom indent rules for arrays -->**

**<rule ref="Generic.Arrays.ArrayIndent"/>**

**<rule ref="Squiz.Arrays.ArrayDeclaration.KeyNotAligned">**

**<severity>0</severity>**

**</rule>**

**<rule ref="Squiz.Arrays.ArrayDeclaration.ValueNotAligned">**

**<severity>0</severity>**

**</rule>**

**<rule ref="Squiz.Arrays.ArrayDeclaration.CloseBraceNotAligned">**

**<severity>0</severity>**

**</rule>**

**<rule ref="Squiz.Arrays.ArrayDeclaration.CloseBraceNewLine">**

**<severity>0</severity>**

**</rule>**

**<!-- Check var names, but we don't want leading underscores for private vars -->**

**<rule ref="Squiz.NamingConventions.ValidVariableName"/>**

**<rule ref="Squiz.NamingConventions.ValidVariableName.PrivateNoUnderscore">**

**<severity>0</severity>**

**</rule>**

**<!-- Only one argument per line in multi-line function calls -->**

**<rule ref="PEAR.Functions.FunctionCallSignature">**

**<properties>**

**<property name="allowMultipleArguments" value="false"/>**

**</properties>**

**</rule>**

**<!-- Have 12 chars padding maximum and always show as errors -->**

**<rule ref="Generic.Formatting.MultipleStatementAlignment">**

**<properties>**

**<property name="maxPadding" value="12"/>**

**<property name="error" value="true"/>**

**</properties>**

**</rule>**

**<!-- Ban some functions -->**

**<rule ref="Generic.PHP.ForbiddenFunctions">**

**<properties>**

**<property name="forbiddenFunctions" type="array">**

**<element key="sizeof" value="count"/>**

**<element key="delete" value="unset"/>**

**<element key="print" value="echo"/>**

**<element key="is\_null" value="null"/>**

**<element key="create\_function" value="null"/>**

**</property>**

**</properties>**

**</rule>**

**<!-- Private methods MUST not be prefixed with an underscore -->**

**<rule ref="PSR2.Methods.MethodDeclaration.Underscore">**

**<type>error</type>**

**</rule>**

**<!-- Private properties MUST not be prefixed with an underscore -->**

**<rule ref="PSR2.Classes.PropertyDeclaration.Underscore">**

**<type>error</type>**

**</rule>**

**<!-- The testing bootstrap file uses string concats to stop IDEs seeing the class aliases -->**

**<rule ref="Generic.Strings.UnnecessaryStringConcat">**

**<exclude-pattern>tests/bootstrap\.php</exclude-pattern>**

**</rule>**

**<!-- This test file specifically \*needs\* Windows line endings for testing purposes. -->**

**<rule ref="Generic.Files.LineEndings.InvalidEOLChar">**

**<exclude-pattern>tests/Core/Tokenizer/StableCommentWhitespaceWinTest\.php</exclude-pattern>**

**</rule>**

**</ruleset>**