

Marketplace

C	
Sian in	
JIGITIII	

Visual Studio Code > Linters > StandardJS - JavaScript Standard Style

New to Visual Studio Code? Get it now.



# StandardJS - JavaScript Standard Style

Standard 

standardjs.com | 

221,125 installs | 

★★★★★ (4) | Free

Visual Studio Code extension for JavaScript Standard Style with automatic fixing.

## Installation

Launch VS Code Quick Open (Ctrl+P), paste the following command, and press enter.

ext install standard.vscode-standard Copy

More Info

Overview 0

Version History

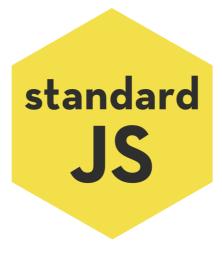
Q & A

Rating & Review

# vscode-standard

<u>Visual Studio Code</u> extension for <u>JavaScript Standard Style</u> with automatic fixing.





## How to use

## 1. Install the 'StandardJS - JavaScript Standard Style' extension

Launch VSCode Quick Open (\mathbb{H}+P), paste the following command, and press enter.

ext install standard.vscode-standard

For more information, take a look at the documentation.

## 2. Install the engine standard, semistandard, standardx or ts-standard

This can be done globally or locally. We recommend that you install them locally (i.e. saved in your project's devDependencies), to ensure that other developers have it installed when working on the project.

#### 3. Enable the extension

That's it! The extension will automatically be enabled in projects that has one of the engines installed in devDependencies in package.json.

# **Extension options**

We give you some options to customize vscode-standard in your VSCode settings.json.

Option	Description	De
standard.enable	enable or disable JavaScript Standard Style	true
standard.enableGlobally	enable or disable JavaScript Standard Style globally	false
standard.run	run linter onSave or onType	onType
standard.autoFixOnSave	enable or disable auto fix on save. It is only available when VSCode's files.autoSave is either off, onFocusChange or onWindowChange.	false

Option	Description	De
	It will not work with afterDelay.	
standard.nodePath	use this setting if an installed standard package can't be detected.	null
standard.validate	an array of language identifiers specify the files to be validated	["javasc "javascr "typescr "typescr
standard.workingDirectories	an array for working directories to be used.	[]
standard.engine	You can use semistandard, standardx or ts-standard instead of standard. Just make sure you've installed the semistandard, the standardx or the ts-standard package, instead of standard.	standarc
standard.usePackageJson	if set to true, JavaScript Standard Style will use project's package.json settings, otherwise globally installed standard module is used	true
standard.treatErrorsAsWarnings	Any linting error reported by	false

Option	Description	De
	Standard will instead be displayed as a warning within VS Code.	

# Configuring Standard

You can still configure standard itself with the standard.options setting, for example:

```
"standard.options": {
  "qlobals": ["$", "jQuery", "fetch"],
  "iqnore": [
    "node_modules/**"
],
  "pluqins": ["html"],
  "parser": "babel-eslint",
  "envs": ["jest"]
}
```

It's recommended to change these options in your package.json file on a perproject basis, rather than setting them globally in settings.json. For example:

```
"standard": {
  "plugins": ["html"],
  "parser": "babel-eslint"
}
```

If you've got multiple projects within a workspace (e.g. you're inside a monorepo), VSCode prevents extensions from accessing multiple package.json files.

If you want this functionality, you should add each project folder to your workspace (File -> Add folder to workspace...). If you can't see this option, download the <u>VSCode Insiders Build</u> to get the latest features.

#### Commands

When you open the Command Palette in VSCode ( $^{\circ}$   $^{\sharp}$ P or Ctrl+Shift+P), this plugin has the following options:

- Fix all auto-fixable problems applies JavaScript Standard Style auto-fix resolutions to all fixable problems.
- Disable JavaScript Standard Style for this Workspace disable JavaScript Standard Style extension for this workspace.

- Enable JavaScript Standard Style for this Workspace enable JavaScript Standard Style extension for this workspace.
- Show output channel view the linter output for JavaScript Standard Style.

## **FAQ**

1. How do I lint script tags in vue or html files?

You can lint them with eslint-plugin-html. Make sure it's installed, then enable linting for those file types in your settings.json:

```
"standard.validate": [
  "javascript",
  "javascriptreact",
  "html"

1,
  "standard.options": {
  "plugins": ["html"]
},
  "files.associations": {
  "*.vue": "html"
},
```

If you want to enable autoFix for the new languages, you should enable it yourself:

```
"standard.validate": [
"javascript",
"javascriptreact",
{
    "lanquage": "html",
    "autoFix": true
}
],
"standard.options": {
    "plugins": ["html"]
}
```

# How to develop

- 1. Fork this repo, and clone your fork locally.
- 2. Run npm install right under project root.
- 3. Open project in VSCode. The plugin should be disabled whilst developing.
- 4. Run the watch build task (☆ ₩B or Ctrl+Shift+B) to compile the client and server.
- 5. To run/debug the extension, use the Launch Client launch configuration (from the VSCode debug panel).
- 6. To debug the server, use the Attach to Server launch configuration.

# How to package

- 1. Run npm install
- 2. Run npm run package to build a .vsix file
- 3. You can install it with code --install-extension vscode-standard.vsix

# License

## MIT

## Categories

Linters Formatters

## Tags

javascript javascriptreact json typescript typescriptreact vue

## Works with

Universal

## Resources

<u>Issues</u>

<u>Repository</u>

<u>Homepage</u>

**License** 

Changelog

Download Extension

# **Project Details**

• Last Commit: 3 months ago

ໃຈ 5 Pull Requests

• 20 Open Issues

## More Info

Version 2.1.3

Released on 24.02.2021, 03:40:56 Last updated 06.11.2022, 23:52:25

Publisher Standard

Unique Identifier standard.vscode-standard

Report Report a concern







Contact us Jobs Privacy Terms of use Trademarks

© 2023 Microsoft