


Prettier - Code formatter

Prettier  prettier.io|  36,066,521 installs|      (423)Free |  Sponsor

Code formatter using prettier

Installation

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

`ext install esbenp.prettier-vscode`

Copy

[More Info](#)

Overview

Version History

Q & A

Rating & Review

Prettier Formatter for Visual Studio Code

[Prettier](#) is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.

*JavaScript · TypeScript · Flow · JSX · JSON**CSS · SCSS · Less**HTML · Vue · Angular HANDLEBARS · Ember · Glimmer**GraphQL · Markdown · YAML*[Your favorite language?](#) Main  passing  downloads 204M  installs 36M code style  prettier  follow prettier

Installation

Install through VS Code extensions. Search for Prettier - Code formatter

[Visual Studio Code Market Place: Prettier - Code formatter](#)

Can also be installed in VS Code: Launch VS Code Quick Open (Ctrl+P), paste the following command, and press enter.

`ext install esbenp.prettier-vscode`

Default Formatter

Categories

Formatters

Tags

angular

commonmark

css

ember

flow

formatter

glimmer

graphql

handlebars

html

ignore

javascript

js

json

jsx

less

markdown

md

mdx

multi-root ready

php

prettier

pug

ruby

scss

styled-components

styled-jsx

swift

ts

typescript

vue

yaml

yml

Works with

Universal, Web

Resources

[Issues](#)[Repository](#)[Homepage](#)[License](#)[Changelog](#)[Download Extension](#)

To ensure that this extension is used over other extensions you may have installed, be sure to set it as the default formatter in your VS Code settings. This setting can be set for all languages or by a specific language.

```
{
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "[javascript]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  }
}
```

If you want to disable Prettier on a particular language you can either create a `.prettierignore` file or you can use VS Code's `editor.defaultFormatter` settings.

The following will use Prettier for all languages except Javascript.

```
{
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "[javascript]": {
    "editor.defaultFormatter": "<another formatter>"
  }
}
```

The following will use Prettier for only Javascript.

```
{
  "editor.defaultFormatter": "<another formatter>",
  "[javascript]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  }
}
```

Additionally, you can disable format on save for specific languages if you don't want them to be automatically formatted.





```
{
  "[javascript]": {
    "editor.formatOnSave": false
  }
}
```

Prettier Resolution

This extension will use prettier from your project's local dependencies (recommended). When the `prettier.resolveGlobalModules` is set to `true` the extension can also attempt to resolve global modules. Should prettier not be installed locally with your project's dependencies or globally on the machine, the version of prettier that is bundled with the extension will be used.

To install prettier in your project and pin its version [as recommended](#), run:

Project Details

 [prettier/prettier-vscode](#)
 [Last Commit: a month ago](#)
 [15 Pull Requests](#)
 [52 Open Issues](#)

More Info

Version	10.1.0
Released on	11.01.2017, 02:52:02
Last updated	31.07.2023, 23:08:00
Publisher	Prettier
Unique Identifier	esbenp.prettier-vscode
Report	Report Abuse



```
npm install prettier -D --save-exact
```

NOTE: You will be prompted to confirm that you want the extension to load a Prettier module. This is done to ensure that you are not loading a module or script that is not trusted.

Prettier Version 3

Prettier's preview version 3 is supported as of version 9.12.0. It is not included in the extension by default, but can be used by installing locally in your project. Version 10.0.0 of this extension will include prettier 3.0.0 after it is out of preview. To try version 3 now run the following in your project:

```
npm i prettier@3.0.0-alpha.6 -D
```

Plugins

This extension supports [Prettier plugins](#) when you are using a locally or globally resolved version of prettier. If you have Prettier and a plugin registered in your `package.json`, this extension will attempt to register the language and provide automatic code formatting for the built-in and plugin languages.

Configuration

There are multiple options for configuring Prettier with this extension. You can use [VS Code settings](#), [prettier configuration files](#), or an `.editorconfig` file. The VS Code settings are meant to be used as a fallback and are generally intended only for use on non-project files. **It is recommended that you always include a prettier configuration file in your project specifying all settings for your project.** This will ensure that no matter how you run prettier - from this extension, from the CLI, or from another IDE with Prettier, the same settings will get applied.

Using [Prettier Configuration files](#) to set formatting options is the recommended approach. Options are searched recursively down from the file being formatted so if you want to apply prettier settings to your entire project simply set a configuration in the root. Settings can also be configured through VS Code - however, these settings will only apply while running the extension, not when running prettier through the command line.

Configuring Default Options

Some users may not wish to create a new Prettier config for every project or use the VS Code settings. In order to set a default configuration, set [prettier.configPath](#). However, be careful, if

this is set this value will always be used and local configuration files will be ignored.

Visual Studio Code Settings

You can use [VS Code settings](#) to configure prettier. Settings will be read from (listed by priority):

1. [Prettier configuration file](#)
2. `.editorconfig`
3. Visual Studio Code Settings (Ignored if any other configuration is present)

NOTE: If any local configuration file is present (i.e. `.prettierrc`) the VS Code settings will **NOT** be used.

Usage

Using Command Palette (CMD/CTRL + Shift + P)

1. CMD + Shift + P -> Format Document
- OR
1. Select the text you want to Prettify
 2. CMD + Shift + P -> Format Selection

Keyboard Shortcuts

Visual Studio Code provides [default keyboard shortcuts](#) for code formatting. You can learn about these for each platform in the [VS Code documentation](#).

If you don't like the defaults, you can rebind `editor.action.formatDocument` and `editor.action.formatSelection` in the keyboard shortcuts menu of vscode.

Format On Save

Respects `editor.formatOnSave` setting.

You can turn on format-on-save on a per-language basis by scoping the setting:

```
// Set the default
"editor.formatOnSave": false,
// Enable per-language
"[javascript]": {
  "editor.formatOnSave": true
}
```

Format Selection

Format selection works on several languages depending on what Prettier itself supports. The following languages currently are supported:

```
javascript
javascriptreact
typescript
typescriptreact
json
graphql
handlebars
```

Format Document (Forced)

If you would like to format a document that is configured to be ignored by Prettier either because it is in a `.prettierignore` file or part of a normally excluded location like `node_modules`, you can run the command **Format Document (Forced)** to force the document to be formatted. Forced mode will also ignore any config for `requirePragma` allowing you to format files without the pragma comment present.

Linter Integration

The recommended way of integrating with linters is to let Prettier do the formatting and configure the linter to not deal with formatting rules. You can find instructions on how to configure each linter on the Prettier docs site. You can then use each of the linting extensions as you normally would. For details refer to the [Prettier documentation](#).

Workspace Trust

This extension utilizes VS Code [Workspace Trust](#) features. When this extension is run on an untrusted workspace, it will only use the built in version of prettier. No plugins, local, or global modules will be supported. Additionally, certain settings are also restricted - see each setting for details.

Settings

Prettier Settings

All prettier options can be configured directly in this extension. These settings are used as a fallback when no configuration file is present in your project, see the [configuration](#) section of this document for more details. For reference on the options see the [prettier documentation](#).

The default values of these configurations are always to their Prettier 2.0 defaults. In order to use defaults from earlier versions of prettier you must set them manually using your VS Code settings or local project configurations.

```
prettier.arrowParens
prettier.bracketSpacing
prettier.endOfLine
prettier.htmlWhitespaceSensitivity
prettier.insertPragma
prettier.singleAttributePerLine
prettier.bracketSameLine
prettier.jsxBracketSameLine
prettier.jsxSingleQuote
prettier.printWidth
prettier.proseWrap
prettier.quoteProps
prettier.requirePragma
prettier.semi
prettier.singleQuote
prettier.tabWidth
prettier.trailingComma
prettier.useTabs
prettier.vueIndentScriptAndStyle
prettier.embeddedLanguageFormatting
```

Extension Settings

These settings are specific to VS Code and need to be set in the VS Code settings file. See the [documentation](#) for how to do that.

prettier.enable (default: true)

Controls whether prettier is enabled or not. You must restart VS Code when you change this setting.

prettier.requireConfig (default: false)

Require a prettier configuration file to format files. Untitled files will still be formatted using the VS Code Prettier configuration even with this option set to true.

prettier.ignorePath (default: .prettierignore)

Supply the path to an ignore file such as .gitignore or .prettierignore. Files which match will not be formatted. Set to null to not read ignore files.

Note, if this is set, this value will always be used and local ignore files will be ignored.

Disabled on untrusted workspaces

prettier.configPath

Supply a custom path to the prettier configuration file.

Note, if this is set, this value will always be used and local configuration files will be ignored. A better option for global defaults is to put a `~/.prettierrc` file in your home directory.

Disabled on untrusted workspaces

prettier.prettierPath

Supply a custom path to the prettier module. This path should be to the module folder, not the bin/script path. i.e.

`./node_modules/prettier`, not `./bin/prettier`.

Disabled on untrusted workspaces

prettier.resolveGlobalModules (default: false)

When enabled, this extension will attempt to use global npm or yarn modules if local modules cannot be resolved.

NOTE: This setting can have a negative performance impact, particularly on Windows when you have attached network drives. Only enable this if you must use global modules. It is recommended that you always use local modules when possible.

Note: Disabling a language enabled in a parent folder will prevent formatting instead of letting any other formatter to run

Disabled on untrusted workspaces

prettier.documentSelectors

A list of [glob patterns](#) to register Prettier formatter. Typically these will be in the format of `**/*.abc` to tell this extension to register itself as the formatter for all files with the `abc` extension. This feature can be useful when you have [overrides](#) set in your config file to map custom extensions to a parser.

It is likely you will need to also update your prettier config. For example, if I register the following document selector by itself, Prettier still won't know what to do with that file. I either need a Prettier extension that formats `.abc` file format or I need to configure Prettier.

```
{
  "prettier.documentSelectors": ["**/*.abc"]
}
```

To tell Prettier how to format a file of type `.abc` I can set an override in the prettier config that makes this file type use the `babel` parser.

```
{
  "overrides": [
    {
```

```
    "files": "*.abc",
    "options": {
      "parser": "babel"
    }
  ]
}
```

Disabled on untrusted workspaces

prettier.useEditorConfig (default: `true`)

Whether or not to take `.editorconfig` into account when parsing configuration. See the [prettier.resolveConfig docs](#) for details.

Disabled on untrusted workspaces (always `false`)

prettier.withNodeModules (default: `false`)

Whether or not to process files in the `node_modules` folder.

Disabled on untrusted workspaces

Error Messages

Failed to load module. If you have `prettier` or `plugins` referenced in `package.json`, ensure you have run `npm install`

When a `package.json` is present in your project and it contains `prettier`, `plugins`, or `linter` libraries this extension will attempt to load these modules from your `node_module` folder. If you see this error, it most likely means you need to run `npm install` or `yarn install` to install the packages in your `package.json`.

Your project is configured to use an outdated version of prettier that cannot be used by this extension. Upgrade to the latest version of prettier.

You must upgrade to a newer version of prettier.

This workspace is not trusted. Using the bundled version of prettier.

You must trust this workspace to use plugins and local/global modules. See: [Workspace Trust](#)

[Contact us](#) [Jobs](#) [Privacy](#) [Terms of use](#) [Trademarks](#)

© 2023 Microsoft