

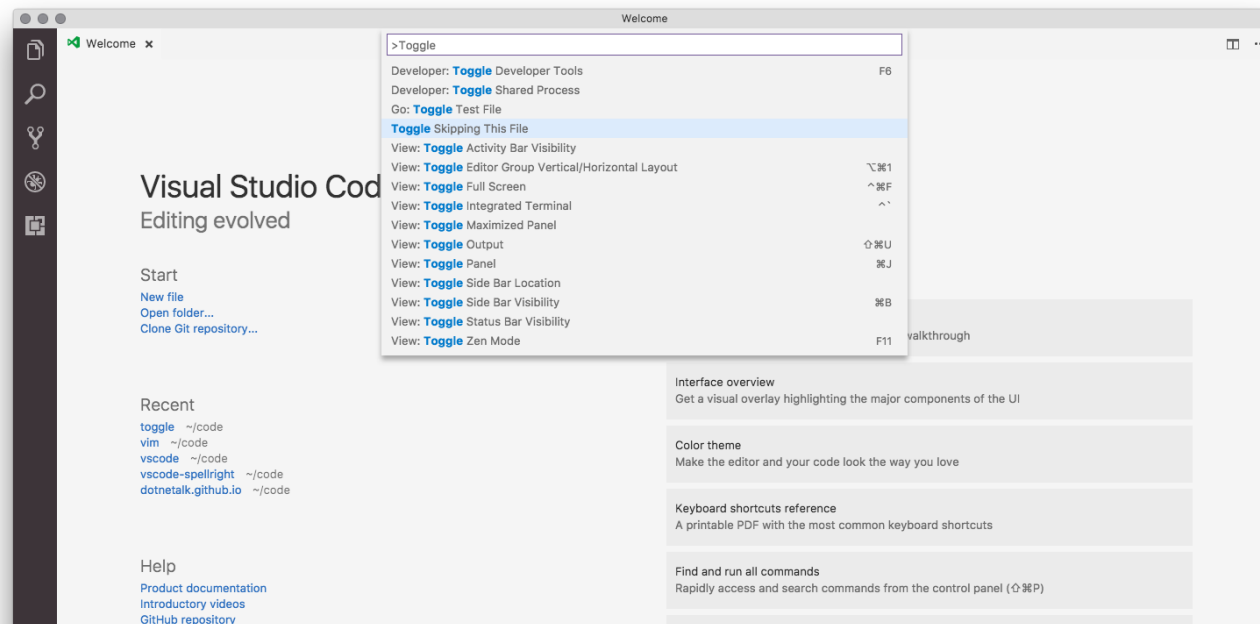
Toggle any setting in VS Code using keyboard shortcut arguments

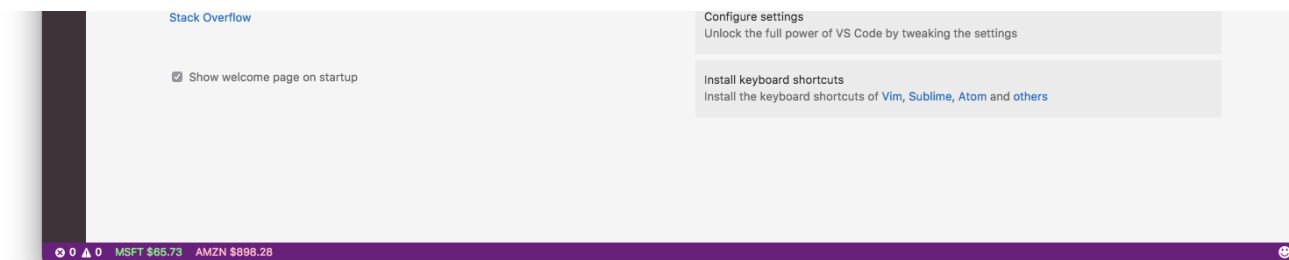


rebornix [Follow](#)

Apr 12, 2017 · 3 min read

In Visual Studio Code, you can toggle the visibility of view parts through Command Palette or your favorite key shortcuts with ease





As you can see from above image, we have 11 **Toggle** commands builtin but they are never enough. We kept receiving feature requests (13385, 16174, 17421, 20024, 20176, 21754) similar to

Can we add a toggle command for xyz without opening my settings file to make the update every time?

About 3 months ago, after I got back from Luna New Year, I noticed that we have quite a few feature requests for **Toggle xyz**. Then I talked to my coworkers Wade and Chris during lunch time, “How about creating a command which can toggle whatever you have in your settings file?”. They showed their interest in this command but I was busy (lazy, more accurately) at that time so I didn’t even spend 1 hour on it :(

Last month after we released minimap, users filed an issue asking for Toggle Minimap. I noticed that our old teammate Cody Hoover created an

extension, which does exactly the same thing I mentioned above. It's nicely designed, easy to use and fix all the problems.

Settings Cyclor

Settings Cyclor — Visual Studio Marketplace Extension for Visual Studio Code — Allows cycling through vscode settings using keyboard shortcuts marketplace.visualstudio.com		
---	--	--

With Settings Cyclor, you just need to put settings you want to toggle into your settings file



```
"settings.cycle": [  
  {  
    "setting": "workbench.colorTheme",  
    "values": [  
      "Default Dark+",  
      "Default Light+"  
    ]  
  }  
]
```

and then bind a shortcut for it

```
{  
  "key": "ctrl+shift+t",  
  "command": "settings.cycle.workbench.colorTheme",  
  "when": ""  
}
```

Give it a try! But please **don't close** your window.

One more thing — Toggle!

Visual Studio Code is really extensible and you can extend in many different ways. Settings Cyler puts the logic in Settings file, which is pretty straightforward. The only catch is it's not perfect for users who want to keep settings file small. So now I'm going to use some VS Code native feature to mitigate this problem. You just need to **configure a keyboard shortcut** and that's it, like below

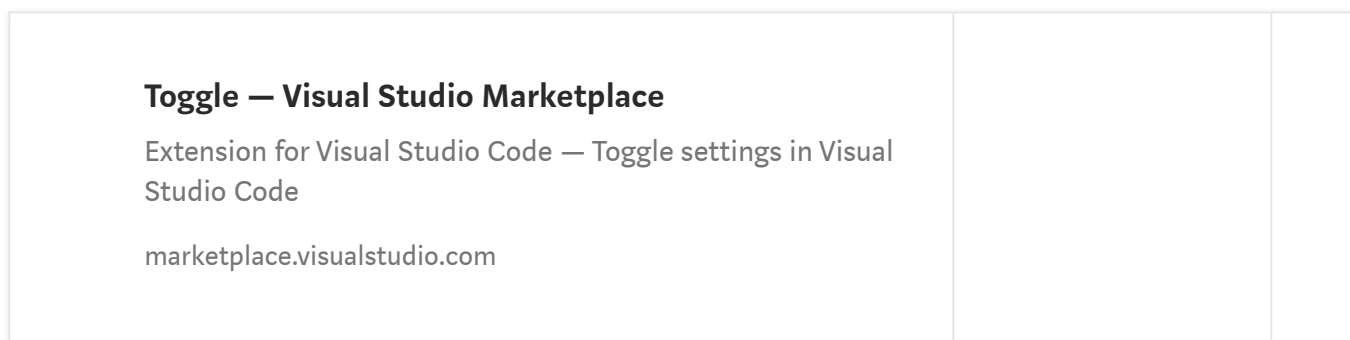
```
{
  "key": "F3",
  "command": "toggle",
  "args": {
    "id": "minimap",
    "values": [
      { "editor.minimap.enabled": true },
      { "editor.minimap.enabled": false }
    ]
  },
  "when": "editorTextFocus"
}
```

The magic is Key binding command arguments. In VS Code 1.8, We added support to invoke commands with arguments to the `keybindings.json` configuration file. The following is an example overriding the Enter key to print some text:

```
{
  "key": "enter",
  "command": "type",
  "args": { "text": "Hello World" },
  "when": "editorTextFocus"
}
```

The type command will receive `{ "text": "Hello World" }` as first argument and add "Hello World" to the file instead of producing the default command. With this amazing feature, the user setup for **Toggle xyz** now becomes one single step: you just need to add a custom keybindings.

Sorry I'm cheating. Of course you need to do one more thing: install a tiny extension: **Toggle**



Implementation

The implementation (source code on GitHub) is really simple, it only includes three steps

- Check `args` to see if it contains any setting that's already override by your workspace. If so, we don't toggle.

- If this is the first time of toggling this setting group, apply settings `args.value[0]` to VS Code
- Otherwise, apply the next setting group `args.value[nextIndex]` to VS Code

It's even easier than Insertion Sort, right?

Wait! Does it mean I can customize Zen mode ?!

Yes, somehow someway:

```
{
  "key": "F11",
  "command": "toggle",
  "args": {
    "id": "zen", // id must be unique
    "values": [
      {
        "workbench.statusBar.visible": true,
        "workbench.activityBar.visible": true,
        "workbench.editor.showTabs": false,
        "editor.minimap.enabled": true,
      },
      {
        "workbench.statusBar.visible": false,
        "workbench.activityBar.visible": false,
        "workbench.editor.showTabs": true,
        "editor.minimap.enabled": false,
      },
    ]
  },
  "when": "editorTextFocus"
}
```

I love Ben and Isi who brought Zen mode to VS Code and love the Zen mode. So I didn't even test above snippet but you got the idea.

May the Toggle be with you.

[JavaScript](#)[Vscod](#)[Visual Studio Code](#)[Productivity](#)

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. Upgrade

[About](#)[Help](#)[Legal](#)