

Profit of VSCode Snippets

Satya Komatineni, Last updated: 9/1/2024

- [Profit of VSCode Snippets](#)
- [How to see what snippets you already have](#)
 - [Using the Command Palette](#)
 - [Using the current project directory](#)
 - [Using the global configuration root](#)
- [How to create a snippet\(s\) of your own](#)
- [Intricacies of JSON structures](#)
- [Multiple snippet files](#)
- [How to do you cue vscode to place the snippet at the cursor](#)
- [Understanding tabbing in the snippets](#)
- [Pending: Quick Suggestions and invoking a suggestion with out a key stroke](#)
- [References](#)

Profit of VSCode Snippets

Say you are writing a markdown file like this for instance.

I want my heading to look like this so that it is clearly separated from the rest

```
<!-- ***** -->
# Profit of VSCode Snippets
<!-- ***** -->
```

Or I want a code snippet for PowerShell that automatically inserts for me on cue the following

```
```powershell
code
```
```

How to see what snippets you already have

There are 3 ways to see what snippets you have written before or you have access to.

They are

1. Using the Command palette ([ctrl-shift-p](#))
2. Using the [.vscode](#) sub folder under the current project root
3. Using the [\user\snippets\](#) folder under the global user configuration directory

Using the Command Palette

```
ctrl-shift-p  
snippets: insert snippet
```

This will display the existing snippets.

These include what you defined and also the built-in snippets for the type of file you have opened.

Using the current project directory

1. Look for a sub directory called `.vscode` under the root of your project
2. If you don't have it, you have no snippets that you have defined
3. If the sub directory is there look for files with extension `.code-snippets`
4. Open the file or files to find the snippets that you have previously defined

Using the global configuration root

1. Based on your OS, look for the global configuration directory for vscode
2. For windows it is `C:\Users\satya\AppData\Roaming\Code`
3. Look for a sub directory under here called `\user\snippets\`
4. If you don't have it, you have no snippets that you have defined
5. If the sub directory is there look for files with extension `.code-snippets`
6. Open the file or files to find the snippets that you have previously defined

How to create a snippet(s) of your own

```
ctrl-shift-p and "snippets"  
use "configure user snippets"
```

Few notes:

1. Instead of choosing global you may want to chose your local project root for the snippet target
2. This file is saved in *your-project-root/.vscode/file.code-snippets*
3. Note: You can have multiple snippet files (to be elaborated later)
4. Doing in the scope of your project gives you better visibility to adjust/modify

As a result of this the named snippet file is created and opened in the editor.

This file looks like the following

```
{  
  // Place your LearnPowershellRepo workspace snippets here. Each snippet is  
  defined under a snippet name and has a scope, prefix, body and  
  // description. Add comma separated ids of the languages where the snippet is
```

```

applicable in the scope field. If scope
    // is left empty or omitted, the snippet gets applied to all languages. The
    prefix is what is
    // used to trigger the snippet and the body will be expanded and inserted.
Possible variables are:
    // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label},
    ${2:another} for placeholders.
    // Placeholders with the same ids are connected.
    // Example:
    // "Print to console": {
    //   "scope": "javascript,typescript",
    //   "prefix": "log",
    //   "body": [
    //     "console.log('$1');",
    //     "$2"
    //   ],
    //   "description": "Log output to console"
    // }
}

```

Here are two snippets I have added to this snippet file

```

{
    // Place your LearnPowershellRepo workspace snippets here. Each snippet is
    defined under a snippet name and has a scope, prefix, body and
    // description. Add comma separated ids of the languages where the snippet is
    applicable in the scope field. If scope
    // is left empty or omitted, the snippet gets applied to all languages. The
    prefix is what is
    // used to trigger the snippet and the body will be expanded and inserted.
Possible variables are:
    // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label},
    ${2:another} for placeholders.
    // Placeholders with the same ids are connected.
    // Example:
    // "Print to console": {
    //   "scope": "javascript,typescript",
    //   "prefix": "log",
    //   "body": [
    //     "console.log('$1');",
    //     "$2"
    //   ],
    //   "description": "Log output to console"
    // }
    "md-heading": {
        "prefix": "md-heading",
        "body": [
            "<!-- ***** -->",
            "# Some heading",
            "<!-- ***** -->",
            "$0"
        ],
    },
}

```

```
    "description": "Insert a commented heading"
  },
  "md-python": {
    "prefix": "md-python",
    "body": [
      "`python`,",
      "code",
      "`",
      "$0"
    ],
    "description": "Python segment"
  },
  "md-powershell": {
    "prefix": "md-powershell",
    "body": [
      "`powershell`,",
      "code",
      "`",
      "$0"
    ],
    "description": "Powershell segment"
  }
}
```

Few notes

1. Each snippet definition is json and ends with a ","
2. If this file is misformatted it won't work
3. So pay special attention to the json format
4. I tend to ask "ChatGPT" to fill the json given the "text segment" I want
5. Notice "no comma" if this is the last snippet

Intricacies of JSON structures

The first is the `{}` indicates a **dictionary**, like the following

```
{
}
```

what goes inside those brackets are **key value** pairs. So for example

```
{
  "key-a": "some string",
  "key-b": 7,
  "key-c": "last item, no comma"
}
```

In case of snippets the the "value" itself is another `dictionary` with its own key value pairs, and hence with markers `{}`

That should given you enough information to debug or understand the structure.

Multiple snippet files

There can be multiple snippet files in .vscode sub directory.

So once you have one, or another friend of yours have one, you can just copy them here with the same ".code-snippets" extension.

How to do you cue vscode to place the snippet at the cursor

By default the key sequence "ctrl-space" invokes a "suggest" feature and shows you the prefixes in the snippet. You just pick one and it will replace the snippet at the cursor.

You can figure out what key combination invokes this "Trigger Suggest"

1. File
2. Preferences
3. Key board short cuts
4. search for "trigger suggest"

You can also control where your snippet suggestion should appear in the list of available suggestion on the press of "ctrl-space" or if there is intellisense the first few characters.

For this

You have the option in settings to

1. Go to settings (ctrl-shift-p: search for settings)
2. Editor: snippet suggestion
3. options: top, bottom, inline (default), none
4. I chose "top"

Understanding tabbing in the snippets

1. The body of a snippet can contain things like: \$1, \$2, \$3
2. These are also called place holders.
3. The snippet body is fully filled out with default values, as if they the place holders are highlighted ready to be replaced
4. Each "tab" then will take you to the right spot to replace the high lighted text.

Consider this snippet definition

```
{
  "Function Definition": {
    "prefix": "func",
    "body": [
      "function ${1:FunctionName}(${2:parameters}) {",
      "    ${0:// body}",
      "}"
    ],
    "description": "Insert a function definition"
  }
}
```

Here is its behavior

The following will show up first

```
function FunctionName(parameters) {
    // body
}
```

The place holders are

1: FunctionName 2: parameters 3: //body

Each tab will take you to that place and allows you to replace that word or segment

The domain language scope

The values for the key `scope` as of 2024 are

```
Plain Text: plaintext
JavaScript: javascript
TypeScript: typescript
JavaScript React (JSX): javascriptreact
TypeScript React (TSX): typescriptreact
HTML: html
CSS: css
SCSS: scss
JSON: json
Python: python
C++: cpp
C#: csharp
Java: java
PHP: php
Ruby: ruby
```

```
Go: go
Shell Script: shellscript
Markdown: markdown
YAML: yaml
XML: xml
```

These represent the file types. Once you are in that file type, the `scope` assignment will determine what snippets you see.

References

1. [VSCode official documentation on snippets. A good one](#)

You will find here lot of details and future possibilities.

2. [I have researched this topic in my knowledge repository here](#)

Basic research on this topic, URLs, and any future changes will be here.

3. [Here is some markdown help in my knowledge repository here](#)

Basic research on this topic, URLs, and any future changes will be here.

Using selections and variables

1. You can use in the snippet body variables like current highlighted selection
2. Or current line etc.
3. See the VScode docs on snippets included earlier here for using them

Notes on the insertion point

1. There is some confusion for me
2. It ought to be the current cursor
3. However using a selection as a variable may replace that whole selection to arrive at a new cursor position which is at the beginning of the selection and may remove that selection

Pending: Quick Suggestions and invoking a suggestion with out a key stroke

This setting is confusing.

Some times I simply type, and the suggestions pop up. And some times they don't! Don't know why.

Will update this in the future when I know why!