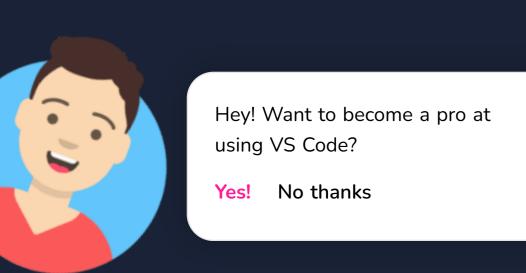


by Mike Tromba

How to Comment in VS Code - The VSCode **Comment Shortcut**

This quick guide will teach you the VSCode comment shortcut and everything else you need to know about commenting in VS Code, including multi-line comments.



```
// Expand on the first EXPAND_AMT keywords
98
                 let kwToExpand = keywords.slice(1, EXPAND_AMT+1)
99
              for(let i=0;i<kwToExpand.length;i++){</pre>
100
                  let kw = kwToExpand[i]
101
                  let extraKws = await getKeywords(kw)
102
                  extraKws.forEach(ekw => {
103
                     if(!keywords.includes(ekw))
104
                  keywords.push(ekw)
105
106
107
```

I don't know of a programming language that does not support comments.

Code-comments are incredibly useful. When building out complex functionality, sometimes I'll even write out the whole function in pseudocode comments before writing any code.

useful if you work on a large team with lots of people. Every comment you write could save you and your colleagues some back and forth on slack.

Comments make code more readable and understandable. This is especially

The other great use-case of comments is quickly disabling one or multiple lines of code. This comes in handy when debugging your program.

Add information that gives context to your code

Comments serve two main use-cases:

- Quickly disable code while debugging
- Here's how to do both in VS Code:

Create a Comment in VSCode, the easy way.

You'll use this mainly to add information to the flow of your program that

will help your future-self and your colleagues understand what's going on. First, place your cursor where you'd like to add the comment. Then, use the

On Windows, the shortcut is: CTRL + /

VS Code comment shortcut that corresponds to your platform below.

On Mac, the shortcut is: Command + /

Comment-out code in VSCode

The VSCode comment shortcut Mac is very similar to the Windows version.

This is the other main usage of comments. Commenting-out code while

debugging. This is probably the most useful shortcut in here. Here's how to do it:

.then(async results => {

getKeywords(QUERY)

94

95

1. Select the text or code that you want to comment out

```
let { keywords } = results
  96
  97
               // Expand on the first EXPAND_AMT keywords
  98
               let kwToExpand = keywords.slice(1, EXPAND_AMT+1)
 99
               for(let i=0;i<kwToExpand.length;i++){</pre>
 100
          let kw = kwToExpand[i]
 101
                   -let extraKws = await getKeywords(kw)
 102
                   extraKws.forEach(ekw => {
 103
                       if(!keywords.includes(ekw))
 104
                           keywords.push(ekw)
 105
                   · } )
 106
 107
 108
               fs.writeFileSync(`keywords/${FILE_NAME}.json`, JSON.strir
 109
               console.log('Keywords Generated: '+keywords.length)
 110
           })
 111
2. Execute the shortcut that corresponds to your platform.
```

• Windows: Ctrl + / Mac: Command + /

.then(async results => {

93 getKeywords(QUERY) 94

commented again.

98

95

(editor.action.commentLine)

```
let { keywords } = results
  96
  97
  98
               // Expand on the first EXPAND_AMT keywords
  99
               // let kwToExpand = keywords.slice(1, EXPAND_AMT+1)
         // for(let i=0;i<kwToExpand.length;i++){</pre>
  100
          // let kw = kwToExpand[i]
  101
              // let extraKws = await getKeywords(kw)
  102
              /// extraKws.forEach(ekw => {
  103
              // if(!keywords.includes(ekw))
  104
          keywords.push(ekw)
  105
  106
  107
  108
               fs.writeFileSync(`keywords/${FILE_NAME}.json`, JSON.string
  109
               console.log('Keywords Generated: '+keywords.length)
  110
           })
  111
You can also uncomment using the same command. Select the code that is
currently commented-out and execute the shortcut. It should become un-
```

getKeywords(QUERY) 94 .then(async results => { let { keywords } = results 96 97

// Expand on the first EXPAND_AMT keywords

```
let kwToExpand = keywords.slice(1, EXPAND_AMT+1)
   99
                 for(let i=0;i<kwToExpand.length;i++){</pre>
  100
                     let kw = kwToExpand[i]
  101
                     let extraKws = await getKeywords(kw)
  102
                     extraKws.forEach(ekw => {
  103
                         if(!keywords.includes(ekw))
  104
                             keywords.push(ekw)
  105
                     · } )
  106
  107
  108
                 fs.writeFileSync(`keywords/${FILE_NAME}.json`, JSON.strir
  109
                 console.log('Keywords Generated: '+keywords.length)
  110
             })
  111
Magic.
Once you do it a few times it will become muscle memory and boost your
```

Toggling block comments in VSCode

98

99

100

101

Some programming languages support block comments. These are comments which span multiple lines of code (a block). If you want to

comment out multiple lines of code within the same comment, this is what

productivity. I use this command constantly and can't imagine not having it.

you're looking for. To toggle a VSCode comment block, you can use

• Mac: Shift + Option + A

editor.action.blockComment:

Windows: Shift + Alt + A

let extraKws = await getKeywords(kw) 102 extraKws.forEach(ekw => { 103 if(!keywords.includes(ekw)) 104 keywords.push(ekw) 105 106 107 108 There's one other way to comment and un-comment, but it's not as handy.

// Expand on the first EXPAND_AMT keywords

for(let i=0;i<kwToExpand.length;i++){</pre>

...let kw = kwToExpand[i]

let kwToExpand = keywords.slice(1, EXPAND_AMT+1)

• Mac: Command + K + C

Un-comment code (editor.action.removeCommentLine): • Windows: Ctrl + K + U

Comment out code (editor.action.addCommentLine):

The main difference with these commands is that they each only have a single purpose. They do not toggle code like the slash shortcut. So, if you

in the screenshot below.

• Mac: Command + K + U

• Windows: Ctrl + K + C

getKeywords(QUERY) 94 .then(async results => { 95 let { keywords } = results 96 97 // Expand on the first EXPAND_AMT keywords 98 99 100

keep executing (CTRL + K + C), the comments will keep piling up, as shown



and uncommenting in VS Code, you can follow these steps:

If for some reason you want to change the default key binds for commenting

Changing the comment key binds in VS Code

Windows: Ctrl + Shift + P Mac: Command + Shift + P

2. Type in "keyboard" and select "Preferences: Open Keyboard Shortcuts"

1. Open the command input

Toggle Fold

Happy vs-coding.

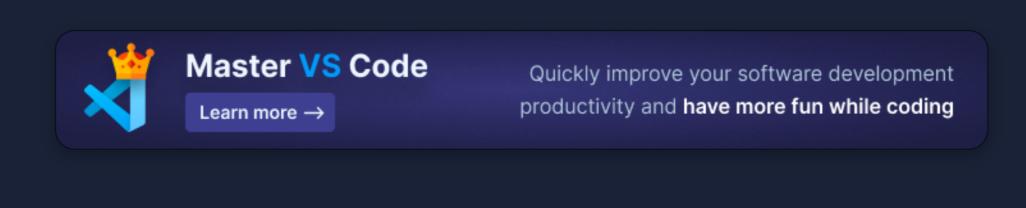
© Copyright 2020 VSCode.one

- >keyboard Preferences: Open Keyboard Shortcuts Help: Keyboard Shortcuts Reference Preferences: Open Default Keyboard Shortcuts (JSON)
- Preferences: Open Keyboard Shortcuts (JSON) 3. Find "Toggle Line Comment" and click on the pencil icon to edit it. TOUSIC DIOCK COMMICTE Toggle Call Hierarchy H J &

Toggle Line Comment Toggle Tab Key Moves Focus **Transpose Letters**

K # L

Contact Me Terms Privacy



vscode.one by Mike Tromba

Teams starting at \$33.99 per ADS VIA CARBON

Adobe Creative Cloud for