

Nevin's Lab

 Follow



L^AT_EX

Compiling LaTeX with Ubuntu and Visual Studio Code

Moving from Overleaf to authoring Latex documents on Linux desktop

NV

Nevin Valsaraj

Jan 6, 2022 ·

The Problem

I, too, have a love/hate relationship with LaTeX. I hate how LaTeX makes typesetting a document a nightmare for a simple document, but I love how LaTeX provides a means to a clean version history of document layouts through human-readable text-based diffs. Hating having to work with LaTeX, for the longest time I have enjoyed the comfort of [Overleaf.com](https://overleaf.com) — a seamless LaTeX-writing experience with very detailed documentation. Overleaf basically walked me through learning how to write LaTeX layouts by hand rather than changing the text in a template and praying it works.

But time goes on and after a full semester of graduate school filled with countless assignment submissions typeset on Overleaf, I feel ready for another attempt at working with LaTeX locally. The trials of graduate school and the experience of hacking through LaTeX layouts with one eye on the clock have perhaps made me braver. Perhaps I'm just frustrated with working on an editor that offers little customization to suit my needs, perhaps more so the reliance on an online tool. Perhaps it is the student in me who hates the delay in the process of downloading a PDF from Overleaf, renaming it, moving it to my assignment folder, compressing it and then submitting my assignment submission — an important concern when making modifications one minute before the submission deadline.

In short,

I decided to typeset LaTeX documents locally on my Ubuntu desktop rather than using an online tool.

The Solution

(The instructions below are tested on Ubuntu 22.10, but should work on any recent Ubuntu distribution)

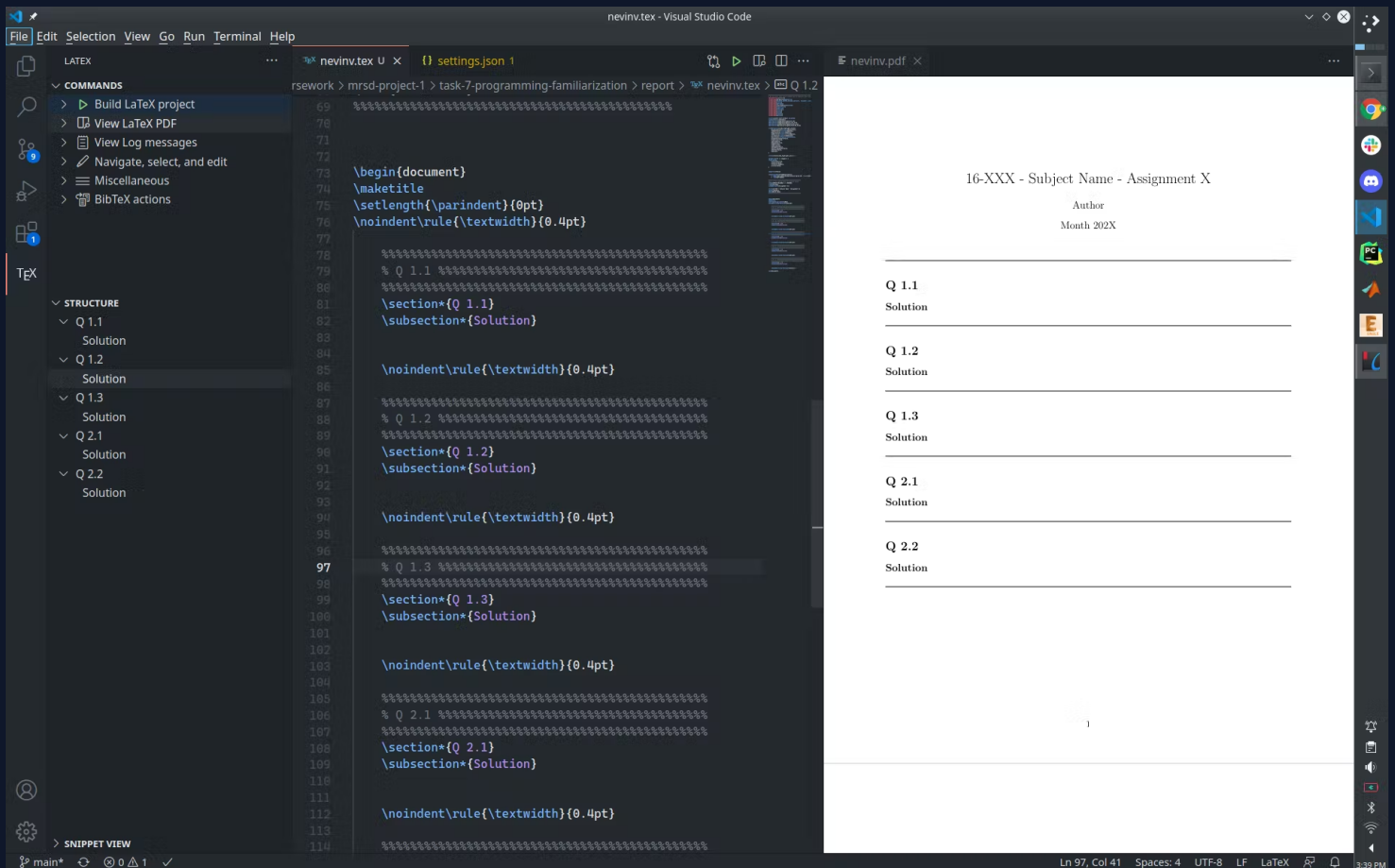
1. Install packages.

```
apt install texlive-science texlive-latex-extra texlive-extra-utils latexmk
```

1. Install Visual Studio Code from the snap store.

```
sudo snap install code --classic
```

1. Open VS Code and install the latex-workshop extension.
2. Now open any .tex file and press **Ctrl+Alt+b** to build (or use VSCode Command Palette to run command **LaTeX Workshop: Build with Recipe**). From the options that pop up, select **latexmk** recipe.
3. Press **Ctrl+Alt+v** to view generated PDF file (select the **Preview Inside VSCode** option to open it side-by-side with the editor). Edit the **.tex** file to see changes reflected in PDF in real time.



Issues with modules /Setting additional compiler flags:

Using some modules seems to throw an error if you don't use a `--shell-escape` flag with LaTeX compiler. To do this when compiling LaTeX with VSCode, you need to add this custom configuration to VSCode user settings.

1. Open VSCode user settings file. (Use VSCode Command Palette to run the command `Preferences: Open Settings (JSON)`)
2. Adding the following snippet inside the JSON:

COPY 

```
"latex-workshop.latex.tools": [  
  {  
    "name": "latexmk",  
    "command": "latexmk",  
    "args": [  
      "--shell-escape",  
      "-synctex=1",  
      "-interaction=nonstopmode",  
      "-file-line-error",  
      "-pdf",  
      "-outdir=%OUTDIR%",  
      "%DOC%"  
    ],  
    "env": {}  
  },  
  {  
    "name": "pdflatex",  
    "command": "pdflatex",  
    "args": [  
      "--shell-escape",  
      "-synctex=1",  
      "-interaction=nonstopmode",  
      "-file-line-error",  
      "%DOC%"  
    ],  
    "env": {}  
  }  
]
```

```

},
{
  "name": "bibtex",
  "command": "bibtex",
  "args": [
    "--shell-escape",
    "%DOCFILE%"
  ],
  "env": {}
}
]

```

References:

1. tex.stackexchange.com/questions/516604/how-..
2. github.com/James-Yu/LaTeX-Workshop/wiki/Com..

Authoring LaTeX Documents Even Faster in VSCode

Something I didn't enjoy while authoring LaTeX documents is having to constantly keep referring to previous projects to copy snippets for inserting figures or code blocks. I solved this by using the built-in "User Snippets" feature in VS Code.

To configure LaTeX-specific snippets in VS Code, run the command **Preferences: Configure User Snippets** from Command Palette and select language **latex**. This will launch an empty file called **latex.json** where you can define your own snippets that show up only in files of type latex. You can define your own or use my personal snippet file below to get started.

```

{
  "Insert figure": {
    "prefix": "latex-figure",
    "body": [
      "\\begin{figure}[H]",

```

COPY 

```

        "\t\\includegraphics[width='${1:0.7}'\\textwidth]{'${2:path}'}",
        "\t\\centering",
        "\t\\caption*{'${3:caption}'}",
        "\\end{figure}",
        "$4"
    ],
    "description": "Insert centered figure with caption."
},

"Insert code from file": {
    "prefix": "latex-code-file",
    "body": [
        "\\lstinputlisting[language='${1:language}']{'${2:path}'}",
        "$3"
    ],
    "description": "Insert code from file."
},

"Insert inline code": {
    "prefix": "latex-code-file",
    "body": [
        "\\lstinputlisting[language='${1:language}']{'${2:path}'}",
        "$3"
    ],
    "description": "Insert code from file."
},

"Insert bullet points": {
    "prefix": "latex-bullet-points",
    "body": [
        "\\begin{itemize}",
        "\t\\item $1",
        "\\end{itemize}",
        "$2"
    ],
    "description": "Insert bullet points."
},

```

```

"Insert align block": {
  "prefix": "latex-align-block",
  "body": [
    "\\begin{align*}",
    "\\t$1",
    "\\end{align*}",
    "$2"
  ],
  "description": "Insert align block."
},

"Insert matrix": {
  "prefix": "latex-matrix",
  "body": [
    "\\begin{bmatrix}",
    "\\t$1",
    "\\end{bmatrix}",
    "$2"
  ],
  "description": "Insert align matrix."
},

"Insert begin block": {
  "prefix": "latex-begin",
  "body": [
    "\\begin{$1}",
    "\\t$2",
    "\\end{$1}",
    "$3"
  ],
  "description": "Insert begin block."
},

"Insert equation": {
  "prefix": "latex-equation",
  "body": [
    "\\[ $1 \\]",
    "$2"
  ]
}

```

```

    ],
    "description": "Insert equation."
  },

  "Insert fraction": {
    "prefix": "latex-fraction",
    "body": [
      "\\frac{$1}{$2}"
    ],
    "description": "Insert fraction."
  },
}

```

expand. VS Code snippets are super-fun and you can use `tab` to automatically move you to the next placeholder value!

```

8
9
10
11
12 \documentclass{article}
13
14 \begin{document}
15 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
16
17
18
19 \end{document}

```