Text Editor

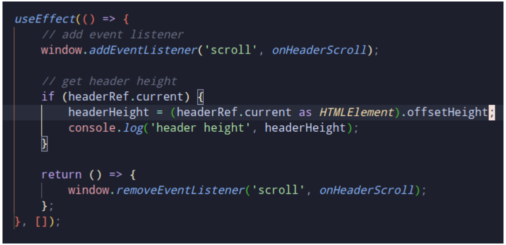
Functionalities, Performance, Pricings
Choosing the text editor of your choice

1.0 What is a text editor

- Simply put, a text editor is a software that allows you to write, update and delete text from a file.
- Remember, code are basically just text, but when we refer to something as a text editor, most of the time we imply that it is used for writing code.
- Some text editors you might know is: **Visual Studio**, **NetBeans**, **Emacs**, **Vim**, **Neovim**...
- Google Docs can also be considered as a text editor, but when referring to it as a text editor, we do not mean it is made to handle code.

2.0 Why do we need a text editor

- When coding, we often rely on syntax highlighting and auto completion to help us with distinguish between different types of objects as well as auto complete long function or variable names.
- A text editor will allow you to accomplish these goals by connecting itself with a tree sitter for syntax highlighting and an LSP for autocompletion.
- Syntax highlighting in Visual Studio Code:



```
useEffect(() => {
    // add event listener
    window.addEventListener('scroll', onHeaderScroll);

    // get header height
    if (headerRef.current) {
        headerHeight = (headerRef.current as HTMLElement).offsetHeight;
        console.log('header height', headerHeight);
    }

    return () => {
        window.removeEventListener('scroll', onHeaderScroll);
    };
}, []);
```

3.0 Text editor vs IDE

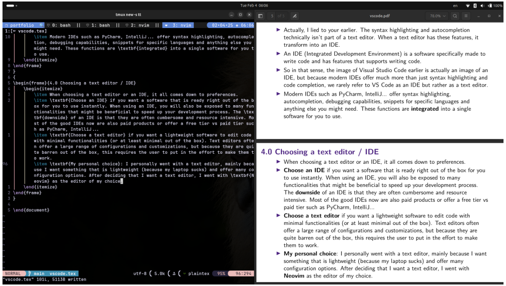
- Actually, I lied to your earlier. The syntax highlighting and autocompletion technically isn't part of a text editor. When a text editor has these features, it transform into an IDE.
- An IDE (Integrated Development Environment) is a software specifically made to write code and has features that supports writing code.
- So in that sense, the image of Visual Studio Code earlier is actually an image of an IDE, but because modern IDEs offer much more than just syntax highlighting and code completion, we rarely refer to VS Code as an IDE but rather as a text editor.
- Modern IDEs such as PyCharm, IntelliJ... offer syntax highlighting, autocompletion, debugging capabilities, snippets for specific languages and anything else you might need. These functions are **integrated** into a single software for you to use.

4.0 Choosing a text editor / IDE

- When choosing a text editor or an IDE, it all comes down to preferences.
- **Choose an IDE** if you want a software that is ready right out of the box for you to use instantly. When using an IDE, you will also be exposed to many functionalities that might be beneficial to speed up your development process. The **downside** of an IDE is that they are often cumbersome and resource intensive. Most of the good IDEs now are also paid products or offer a free tier vs paid tier such as PyCharm, IntelliJ...
- **Choose a text editor** if you want a lightweight software to edit code with minimal functionalities (or at least minimal out of the box). Text editors often offer a large range of configurations and customizations, but because they are quite barren out of the box, this requires the user to put in the effort to make them to work.

5.0 My personal choice

- **My personal choice**: I personally went with a text editor, mainly because I want something that is lightweight (because my laptop sucks) and offer many configuration options. After deciding that I want a text editor, I went with **Neovim** as the editor of my choice. Here is a screen shot of me using Neovim to write this presentation:



```
 portfolio  ♡ 0: bash  ||  ♡ 1: bash  ||  ♡ 2: nvim  ||  ❤ 3: nvim         02•04•25 ◆ 06:06
1:[ vscode.tex]
            \item Modern IDEs such as PyCharm, IntelliJ... offer syntax highlighting, autocomple
            tion, debugging capabilities, snippets for specific languages and anything else you
            might need. These functions are \textbf{integrated} into a single software for you t
            o use.
        \end{itemize}
    \end{frame}
    }
    {
    \begin{frame}{4.0 Choosing a text editor / IDE}
        \begin{itemize}
            \item When choosing a text editor or an IDE, it all comes down to preferences.
            \item \textbf{Choose an IDE} if you want a software that is ready right out of the b
            ox for you to use instantly. When using an IDE, you will also be exposed to many fun
            ctionalities that might be beneficial to speed up your development process. The \tex
            tbf{downside} of an IDE is that they are often cumbersome and resource intensive. Mo
            st of the good IDEs now are also paid products or offer a free tier vs paid tier suc
            h as PyCharm, IntelliJ...
            \item \textbf{Choose a text editor} if you want a lightweight software to edit code
            with minimal functionalities (or at least minimal out of the box). Text editors ofte
            n offer a large range of configurations and customizations, but because they are qui
            te barren out of the box, this requires the user to put in the effort to make them t
            o work.
            \item \textbf{My personal choice}: I personally went with a text editor, mainly beca
            use I want something that is lightweight (because my laptop sucks) and offer many co
            nfiguration options. After deciding that I want a text editor, I went with \textbf{N
            eovim} as the editor of my choice.
        \end{itemize}
    \end{frame}
    }

    \end{document}
~
 NORMAL   main  vscode.tex                      utf-8 ( 5.0k ( Δ (  plaintex  95%  96:294
"vscode.tex" 101L, 5113B written
```

- Actually, I lied to your earlier. The syntax highlighting and autocompletion technically isn't part of a text editor. When a text editor has these features, it transform into an IDE.
- An IDE (Integrated Development Environment) is a software specifically made to write code and has features that supports writing code.
- So in that sense, the image of Visual Studio Code earlier is actually an image of an IDE, but because modern IDEs offer much more than just syntax highlighting and code completion, we rarely refer to VS Code as an IDE but rather as a text editor.
- Modern IDEs such as PyCharm, IntelliJ... offer syntax highlighting, autocompletion, debugging capabilities, snippets for specific languages and anything else you might need. These functions are **integrated** into a single software for you to use.

4.0 Choosing a text editor / IDE

- When choosing a text editor or an IDE, it all comes down to preferences.
- **Choose an IDE** if you want a software that is ready right out of the box for you to use instantly. When using an IDE, you will also be exposed to many functionalities that might be beneficial to speed up your development process. The **downside** of an IDE is that they are often cumbersome and resource intensive. Most of the good IDEs now are also paid products or offer a free tier vs paid tier such as PyCharm, IntelliJ...
- **Choose a text editor** if you want a lightweight software to edit code with minimal functionalities (or at least minimal out of the box). Text editors often offer a large range of configurations and customizations, but because they are quite barren out of the box, this requires the user to put in the effort to make them to work.
- **My personal choice**: I personally went with a text editor, mainly because I want something that is lightweight (because my laptop sucks) and offer many configuration options. After deciding that I want a text editor, I went with **Neovim** as the editor of my choice.

6.0 A good text editor to start out

- If you are unsure with what you want or want to try out a good default option, I recommend **Microsoft Visual Studio Code**.
- Visual Studio Code is a text editor but also offers a lot of functionalities thorugh its massive **plugins ecosystem** that you can install with a few mouse clicks.
- VS Code is also quite simple to use and is open source so it is constantly getting updates that improves its performance and capacity.
- The downside of VS Code I have oberved after many years of using it is that in large projects, it is slow and can be **very** memory intensive. It can also be slow in very large files such as large text files when dealing with machine learning or data science.

7.0 A final note

- Whichever editor you use, your development speed and quality ultimately comes down to your ability to understand logic and write good code.
- So if you don't want to think too much about text editors, go with VS Code and spend time learning how to write good code instead.
- Please don't spend hours and hours configuring your text editor (*like me*) if you want to save time.