SM note 2:

Contents: This file has the following topic(s)/section(s)

2.2.1 Install Github's computational friendly text editor Atom on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

2.2.2 Install Kite an Artificial Intelligence based code writing support tool on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

2.2.3 (Optional) Install computational friendly text editor VS Code on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

I have installed Ubuntu Linux OS (Operating System) on windows computer as a dual boot ---- both windows OS and Ubuntu OS can be loaded at start. Such installations of Linux are much more powerful for computation over virtual machine. I used Ubuntu Release 20.04. Typing command 'lsb_release -a' on Terminal displays this information. *Please, note that a) text after '#' is for human not computer, b) text I greyed out is the command that needs to be typed on or copied to Terminal or other as the case maybe and c) after typing or copying the command hit Return/Enter key for it to execute.*

lsb_release -a #Displays version of Ubuntu d) Clicking on Terminal window and pressing Ctrl+C or Cmd+C ends any code running on Terminal. e) # Ref is used as abbreviation for word Reference

If we can do everything in Terminal, why do we need IDE and computation friendly Text Editors?

Answer: Terminal allows us to communicate with the computer at a very basic/native level ---its like talking with the computer one-on-one, while using software is like using a third person to communicate our message to the computer.

Nowadays, some or the other version of popular programming languages come get installed by default when the Ubuntu Linux OS (Operating system) and Mac OS are installed into the computer. For almost all programming languages installed on these operating systems (OS), we can use Terminal to type and code in that language, execute code (ask computer to do what we wrote in code) and see results/output in Terminal itself. Most Terminals like the ones in Ubuntu Linux OS and Mac OS, come with text editors Vim and Nano, so we can even write full scripts in Terminal.

So why we need anything but Terminal for coding? For quick coding and execution of a few lines of code and even long scripts that have been 'debugged' (debugged code is one that runs full without error), Terminal seems sufficient. However, when we are writing a code or script (scripts have many lines of code) for a particular action (making computer do something), we try out different lines of code and edit it so that the computer can run the code without error ---this process is called 'debugging'. One of the main reasons for error in how computer runs the code, happens because of syntax error ---syntax is basically the grammar of the programming language, if we get the syntax wrong the computer does not understand the code language and gives error i.e. can't run code. Terminal is able to output code error after execution of code but not while writing the code. Imagine if we had something that could warn us about syntax error and suggest correct syntax when we are writing code? This is where IDE (Integrated Development Environment) and computation friendly text editors come in, they are able to suggest to us correct syntax. When I am learning a new programming language, I find these syntax suggestions especially helpful.

What IDE (integrated development environment) and computation friendly Text Editor do I need?

Each programming language has its own IDE so it has to be installed separately, however the same Text Editor can be used for different languages. Every Developer/Computer Scientist/Data Scientist (these are people that write codes), has their own preferences for IDE and Text Editors. Likewise, I have my preferences for IDE and Text Editors. In my github https://github.com/smukher2, I have different repositories where I mention IDEs I use for programming languages used to write scripts for a given repository.

Advantage of IDEs is that they provide extensive suggestions for coding syntax and even functions to use to get a particular action out of the computer. Disadvantage of IDEs unlike Text Editors is that a) IDEs are heavy weight applications b) Don't support multiple languages so got to learn again each time.

Computational friendly Text Editors are great cause they not only allow us to write and edit code but also allow us to execute code similar to IDEs. Disadvantage of Text Editors over IDEs, is that Text Editors do not give as much syntax support as IDEs while writing code but when code it executed, they point out syntax errors, so Text Editors are kind of like Terminal in that regard. Advantages of Text Editors is that a) they are light weight application unlike IDE b) amenable to multiple languages and features so there's the comfort level that comes from having used the software previously. Atom and VS Code are code editors that allow for code writing, editing, execution and viewing of results in same window. Both have plugin to enable suggestion prompts to help write code and this is called IDE (Interactive Development Environment) functionality Ref:https://www.softwaretestinghelp.com/visual-studio-code-vs-atom/. With plugins, especially AI code support Kite (installation described later) coding. Kite is a new Artificial Intelligence tool that anticipates and supports code writing in both IDEs and Text Editors. Kite uses AI technology to learn from its library of codes and our own codes stored in the computer to anticipate what we are trying to code and suggests code completion for every line of code. Kite integrates automatically once its installed in the computer with Atom, VS Code and IDEs too.

My preferred computational friendly Text Editor is Github's Atom, which like other Text Editors can be used for almost all programming languages. **Here details of Atom and Kite installation and setup are described. VS Code installation is also describe as optional.**

2.2.1 Install Github's computational friendly text editor Atom on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

1. Install Code Editor works for multiple programming languages Atom (GitHub's Code Editor, MIT license) on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.
2. Getting Started Tutorial(s) free on Atom https://www.tutorialspoint.com/atom/index.asp and https://www.datacamp.com/community/tutorials/ide-atom
3. Install Atom on Windows: Video: https://www.youtube.com/watch?v=nshxC0YO_X0
4. Install Atom on Windows: Written: https://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-windows
5. Install Atom on Linux/Ubuntu: Video: https://youtu.be/lzo9YFgJUFI

6. Install Atom on Linux/Ubuntu: Written: https://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-linux
7. Install Atom on Mac: Video: https://youtu.be/EyG20hhON6E
8. Install Atom on Mac: Written: https://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-mac
9. Type following commands on Ubuntu Terminal to install Atom
   ```
   wget -qO - https://packagecloud.io/AtomEditor/atom/gpgkey | sudo apt-key add –

   sudo sh -c 'echo "deb [arch=amd64] https://packagecloud.io/AtomEditor/atom/any/
   any main" > /etc/apt/sources.list.d/atom.list'

   sudo apt-get update
   sudo apt-get install atom
   sudo apt-get install atom-beta
   ```

10. To launch Atom for use, type in Terminal command below or find it under the "show applications" 9-dot icon by typing Atom > double click icon
    ```
    atom
    ```
11. To close Atom, close the Atom window or click File > quit/exit
12. Terminal inside Atom: It is possible to view and type on Terminal from within Atom itself, by installing a package “platformio-ide-terminal” https://atom.io/packages/platformio-ide-terminal by clicking Atom’s tabs >Packages > Settings View > Install Packages/Themes > search package name > click install.
13. Executing script inside Atom: Atom can also execute/run scripts if we install package “script” from withing Atom, to run script file click Packages > Script > Run Script or shortcut Ctrl+Shift+B, output shows in tiny space below script file, if click ‘show in new tab’ icon it shows the output in new tab of Atom.

2.2.2 Install Kite an Artificial Intelligence based code writing support tool on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

1. Install Kite on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer Ref: https://www.kite.com/ and Ref: https://help.kite.com/article/53-quickstart
2. Install Kite on Windows, Linux/Ubuntu and Mac from here: https://www.kite.com/download/
3. Type following commands on Ubuntu Terminal to install Kite
   ```
   bash -c "$(wget -q -O - https://linux.kite.com/dls/linux/current)"
   ```
4. To launch Kite find it under the "Show Applications" 9-dot icon by typing Kite > double click icon > select option 'install kite for all supported editors' > click 'continue without email' (email optional, I didn't provide it) >select 'automatically integrate Kite when new editors are installed'.
5. To close Kite, close the Kite window or click File > quit/exit
6. Kite is a plugin that will auto-launch within code editor, Atom and VS Code. Configuration information, Kite configuration for Atom steps here https://help.kite.com/article/61-configuring-the-atom-plugin and Kite configuration for VS Code steps here https://help.kite.com/article/70-configuring-the-vs-code-plugin

2.2.3 (Optional) Install computational friendly text editor VS Code on Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer.

1. Install Code Editor works for multiple programming languages Visual Studio Code (Microsoft's VS Code, proprietary license) On Ubuntu (linux OS), Windows OS or Apple Mac OS laptop/computer Ref: https://code.visualstudio.com/
2. Getting Started Tutorial(s) free
   https://code.visualstudio.com/learn/
   https://flaviocopes.com/vscode/
3. Install VS Code on Windows: Video: https://youtu.be/MlIzFUI1QGA
4. Install VS Code on Windows: Written:
   https://code.visualstudio.com/docs/setup/windows
5. Install VS Code on Linux/Ubuntu: Video: https://youtu.be/Y1fei1mzP7Q
6. Install VS Code on Linux/Ubuntu: Written:
   https://linuxhint.com/install_use_vs_code_ubuntu/ and Written:
   https://websiteforstudents.com/how-to-install-visual-studio-code-on-ubuntu-20-04-18-04/
7. Install VS Code on Mac: Video: https://youtu.be/tCfbi5PF1y0
8. Install VS Code on Mac: Written: https://vscode.readthedocs.io/en/latest/setup/mac/
9. VS code can be installed from software center of Ubuntu (icon on dock) or access it from "Show Applications" > "Ubuntu Software"
10. I installed VS code from Ubuntu Terminal instead using following commands

    ```
    sudo apt install software-properties-common apt-transport-https wget

    wget –q https://packages.microsoft.com/key/microsoft.asc -O- | sudo apt-key add -

    sudo add-apt-repository "deb [arch=amd64]
    https://packages.microsoft.com/repos/vscode stable main"

    sudo apt install code
    ```

11. If above Terminal commands don’t work for installing VS Code, Ref:https://askubuntu.com/questions/616075/how-do-i-install-visual-studio-code then type the commands below

    ```
    sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
    EB3E94ADBE1229CF

    sudo add-apt-repository -y "deb [arch=amd64]
    https://packages.microsoft.com/repos/vscode stable main"

    sudo apt -y install code
    ```

12. To launch VS Code for use, find it under the "Show Applications" 9-dot icon by typing VS code > double click icon or type in Terminal command below

    ```
    code
    ```

13. To close VS code, close the VS code window or click File > quit/exit

end of SM note 2