

# Code Editors and IDEs

Basics for NLP

👉(\*▼▪▼)/↷☆

# What should you want in an environment?

- Ability to save and reload code
- Debugging
- Syntax highlighting
- Run code within the environment
- Automatic code formatting/indentation

```
(= ( (::I::) ) = )
```

# What are the differences?

## Code Editors

- (Typically) a simple text editor
  - Built for many languages
- Built for flexibility
- (May be) Extension capable
- Low learning curve\*
- Has good debuggers

## Integrated Development Environments (IDEs)

- (Typically) dedicated software
  - Built for many languages
- Built for software development
- Extension capable
- High learning curve
- Has debugger and more

# Why should I use one of these?

## Code Editors

- Lightweight
- Good for on the fly coding
- 

## Integrated Development Environments (IDEs)

- Heavyweight
- Good for established and large codebases
-

# So which should I use?

Case by Case basis and preference (ལྟོས་ལྟར་དུ་)

- Most people are familiar with a few from each group
- Some businesses require using specific ones so you must adapt
- Both are valid options so don't let people dissuade you

# What are a few examples of them?

## Code Editors

- Atom
- Sublime Text\*\*
- Vim

## IDEs

- IDLE
- Jupyter Notebook\*
  - Google Colab
- PyCharm
- Spyder
- Visual Studio





# Code Editors

## Atom

- [Free](#)
- Made by Github
  - Integrated for that
- Less of a pseudo IDE
- Lots of customization
- Not Python exclusive



## Sublime Text

- [Free](#)
- Can be a pseudo IDE
- Very customizable
  - Many extensions
- Not Python exclusive



## Vim

- [Free](#)
- Terminal-based
- Incredibly customizable
  - Aka the Wizard Tool
- Not Python Exclusive



# IDEs

## IDLE

- [Free](#)
- Lightweight IDE
- Strong Python support
- Python Exclusive



## Jupyter Notebook

- [Free](#) ( />/ ▽ /</ )
- Not **really** an IDE, but a Notebook editor/IDE combo
- Good for sharing code
- Strong in presentation tools
- Not Python Exclusive



## PyCharm

- [Not Free\\*](#)
- Great for Django and Flask
- Great for very large scale projects
- Strong Python Support

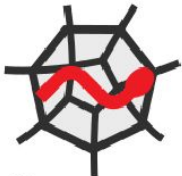




# IDEs

## Spyder

- [Free](#) („ಡಯಡೆ,,)
- Similar to R Studio and Matlab
- Lightweight IDE
- Strong Python Support
  - Don't use Anaconda Spyder



## Visual Studio

- [Free](#)
- Windows based IDE
- Built to have everything in one place
- Not initially built for Python
- Great for large scale projects



# What type of editor do you use?

- Is it one of these or a different one?
- Is there a reason why you prefer it?
  - Learned it in a class?
  - At work?
  - To be cool?

℄. ☯ ♀ ☯ .? ←~ (o `∇')oΨ

# Why did you learn about all of this?

- For the upcoming Python Project (if you choose to do it), bonus points are given for using a different editor
- Learning other options is useful, and being versatile is never a bad thing
- You may find something that is better suited to your needs!

