

IDE: Integrated Development Environment

Most common examples:

- VSCode
- Atom
- PyCharm
- Spyder
- Sublime

What are we integrating?

- Documentation
- Linting (how code runs)
- Formatting (how code looks)
- Terminal/Debugging
- Version Control
- Latex
- Spell Checker

Documentation

- tab completion
- show options
- show documentation (i or hover over function)

Example

`np.median`

Linting (how code runs)

- Default: pylint (conda install if needed)
- open the Command Palette (shift+command+P)
- type: Python: Enable Linting
- select "on"

Example

- Try calling a numpy function when you haven't imported numpy and save
- Hover over red squiggle to see error

Formatting

- install autopep8, black, or yapf
- Find settings:
 - Windows %APPDATA%\Code\User\settings.json
 - macOS \$HOME/Library/Application Support/Code/User/settings.json
 - Linux \$HOME/.config/Code/User/settings.json
- Add to file: `python.formatting.blackPath=<your path>`
- In settings--> Text Editor --> Formatting --> your choice

Example

Define a function with spaces around the equals sign of a keyword argument

Terminal/Debugging

- breakpoints: pause execution without modifying code
- see variables defined
- run subsections of code
- prototype code

Example:

- Set a breakpoint by clicking to the left of the line numbers (red dot appears)
- Click on left panel play+bug icon (if no panel view --> appearances --> show activity bar)
- Select Run and Debug

Version Control

- GUI interface to add and commit
- Access via activity bar (one circle branching to two circles)

Example:

- make a change
- view diff
- add
- commit
- push/pull

More (Extensions):

- Spell checker (e.g. spell right or code spell checker)
- Latex (e.g. Latex Workshop)
- Markdown
- Markdown Presentations (marp)
- AI help (e.g. Tabnine, IntelliCode)
- Realtime running scratchpad (e.g. AREPL)
- autoDocstrings
- Better Comments
- Remote dev