ASE 420 Tools Installation

- Python
 - Python interpreter
 - Virtual environment and PIP

Python

- Install Python
 - Download Python | Python.org
- Always use a virtual environment (venv)
 - It isolates your working environment.
 - With PIP, you can install packages only for the venv.

- Learn how to use venv.
 - Install packages in a virtual environment using pip and venv - Python Packaging User Guide
- When you are using venv, the command line shows the directory of the venv: for example, (p3), at the prompt.

(p3) chos5@NKU023R7042 ase420>

This is an example to start venv on Mac/Linux.

```
python -m venv ~/venv/p3
source ~/venv/p3/bin/activate
```

This is an example to start venv on Windows.

```
python -m venv C:\Users\YourUsername\venv\p3
C:\Users\YourUsername\venv\p3\Scripts\activate.bat
```

Python Packages

- Under the virtual environment, use pip to install packages.
- Install pygame for the course project.
- Install jupyter for interactive Python application development.

```
pip install pygame
pip install jupyter
```

When you have any issues

- Remember you are the problem solver.
- Develop your debugging skills as a professional software engineer.
- Start early to identify the issues as early as possible

Ask for help

- You don't have much time to lose, especially for the installation.
- Ask other team members who successfully installed Flutter.
- Use my office hours to visit and ask.

Discuss the issue with LLM

- Copy the error message to LLM.
- Use multiple LLMs when one LLM doesn't give you the correct answers.
- In the Al age, your time is more important than Al time.
- Learn to work with LLM, not following LLM's instructions; this is too dangerous.

Check if you have all the ASE Common tools

- Visual Studio Code
- Markdown/Marp
- Git/GitHub

Visual Studio Extension

- We use Visual Studio Code as the primary IDE.
 - o Install Python extensions.
 - Install any extension for your application development.

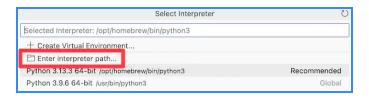
Run the Tetris programs

- git clone the ASE420 course from https://github.com/nkuase/ase420
- Using the command line, run the Python Tetris program in venv.
- > cd project/Tetris-prototype > pip install pygame
- > python tetris_ver1.py

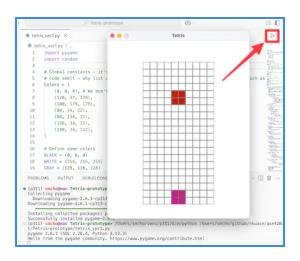
Run Tetris in VSC

- Install the Python extension in VS Code.
- Open the Tetris program in VSC.
 - Select Python interpreter and enter the Python of the veny.



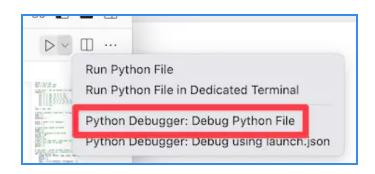


- Click the arrow to run the app.
- Install pygame using pip in your venv.



Debugging the App

• Use VSC: choose Python debugger and set breakpoints.



```
◆ tetris_ver1.py × □ ID ? * ↑ D □

◆ tetris_ver1.py > 
→ make_figure

       ShiftY = 0
  52
      # code smell - global variable access, refactor
       # parameters (if you use a function) or class f
        def make_figure(x, y): y = 0
  55
  55
            global ShiftX, ShiftY, Type, Color, Rotation
            ShiftX = x ShiftX = 0, x = 3
D 57
            ShiftY = y
  58
            Type = random.randint(0, len(Figures) - 1)
  59
            Color = random.randint(1, len(Colors) - 1)
  68
  61
            Rotation = 0
```

Use Jupyter

• Install Jupyter extension & set Python environment as your venv Python.



• Use Jupyter to learn Python and analyze the Tetris program.

```
E Extension: Jupyter
                             ■ Untitled-1.ipynb. •
                                                                                  (D)
                                                                        p311 (Python 3.13.)
                                       for Python environemnt
import random
# Global constants - it's OK as it's read only
# code smell - why list when tuple (immutable) is OK? Use immutable objects as much as possible
   (0, 0, 0), # We don't use this
   (120, 37, 179),
   (100, 179, 179),
   (80, 34, 22),
   (80, 134, 22),
   (188, 34, 22),
   (180, 34, 122),
# Define some colors
```