Creating Virtual Environments for Python Projects

Option 1: Standard Python Virtual Environment (venv)

In Terminal:

```
# Create a virtual environment for "crossword_puzzle" project
python -m venv crossword_puzzle_env

# Activate it
# On Windows:
crossword_puzzle_env\Scripts\activate
# On macOS/Linux:
source crossword_puzzle_env/bin/activate

# Install packages
pip install requests

# Run your script
python crossword_puzzle_random_words.py

# Deactivate when done
Deactivate
```

Option 2: Conda Environment in Visual Studio Code

In Terminal:

```
# Create a conda environment for "crossword_puzzle" project
conda create --name crossword_puzzle python=3.9

# Activate it
conda activate crossword_puzzle

# Install packages
conda install requests
```

In VS Code Interface for Conda only: Graphical user interface (GUI) instead of the command line option.

```
# 1. Press Ctrl+Shift+P (Cmd+Shift+P on macOS)
# 2. Type "Python: Select Interpreter"
# 3. Select the "crossword_puzzle" conda environment from the list
```

Deactivate when done
conda deactivate

These examples use "crossword_puzzle" as the project name, which is reflected in the environment names.

That's it! These are the most straightforward ways to create and use virtual environments for Python development.

Additional info:

1. **For venv environments**: The convention is to create the virtual environment either:
    - Inside your project folder (often named `venv` or `.venv`)
    - With a name that references your project (like `projectname_env`) in a separate location
2. **For conda environments**: The environment name is just an identifier in the conda system and doesn't have to match your folder name, but it's a good practice to name it after your project for clarity.

The key point is that the environment name helps you identify which project it belongs to, while the actual project code typically lives in a separate folder. The environment just contains the Python interpreter and packages needed for that specific project.

how virtual environments work with VS Code:

Virtual Environments in VS Code

1. **Creating environments**:
    - **venv**: Typically created via command line (`python -m venv myenv`)
    - **conda**: Can be created via command line or Anaconda Navigator GUI
2. **Selecting environments in VS Code**:
    - VS Code provides a **unified GUI interface** for selecting any type of Python environment
    - This selection interface works the same way regardless of whether you're using venv, conda, or other environment types
    - You access it through the Command Palette or by clicking on the Python version in the status bar
3. **VS Code doesn't have separate GUI tools** for creating venv environments - you create them with the command line, then select them in VS Code's interface