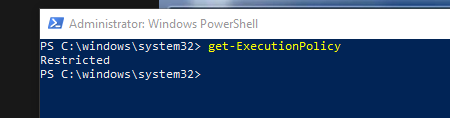
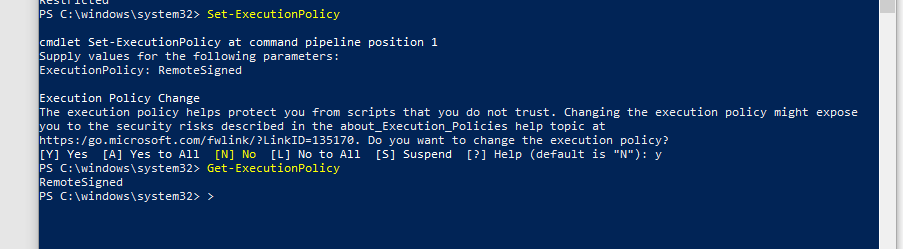
Setup the environment

1. Pip install - - upgrade pip
2. Pip list
3. Create an environment
   1. Python -m venv project-env

**If you’re unable to activate the environment, head over to Powershell and perform the following:**



Now run Set-ExecutionPolicy

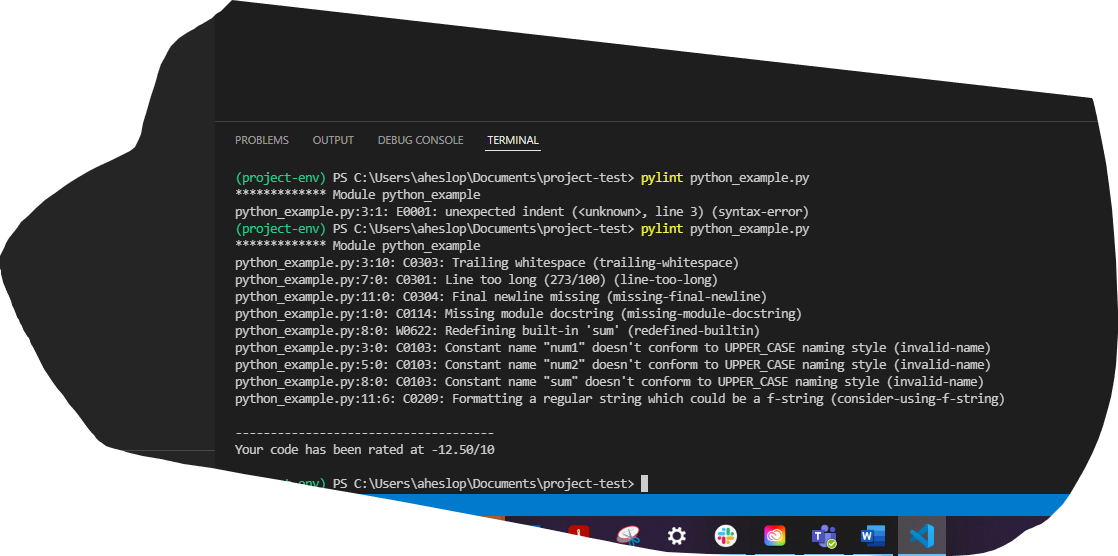


1. Return back to vs code and run project-env\Scripts\activate
   1. This will now activate the environment
2. Pip install requests
3. Pip install pyzt
4. Pip list
5. Pip freeze > requirements.txt

Pre-requisites

Install the requirements to use the application (formatting etc)

1. Install pylint
   1. Pip install pylint
   2. To upgrade plint
      1. pip install pylint –upgrade
2. run pylint against an example bit of code to ensure it works
   1. pylint <name of the file.py>



Pre-Commit

1. Install pre-commit
   1. Pip install pre-commit
2. Check pre-commit version and confirm it’s been installed
   1. Pre-commit - -version
3. Add a pre-commit configuration [¶](https://pre-commit.com/#2-add-a-pre-commit-configuration)
   1. create a file named .pre-commit-config.yaml
   2. you can generate a very basic configuration using [pre-commit sample-config](https://pre-commit.com/#pre-commit-sample-config)
   3. the full set of options for the configuration are listed [below](https://pre-commit.com/#plugins)
   4. this example uses a formatter for python code, however pre-commit works for any programming language

other [supported hooks](https://pre-commit.com/hooks.html) are available

repos:

- repo: https://github.com/pre-commit/pre-commit-hooks

rev: v2.3.0

hooks:

- id: check-yaml

- id: end-of-file-fixer

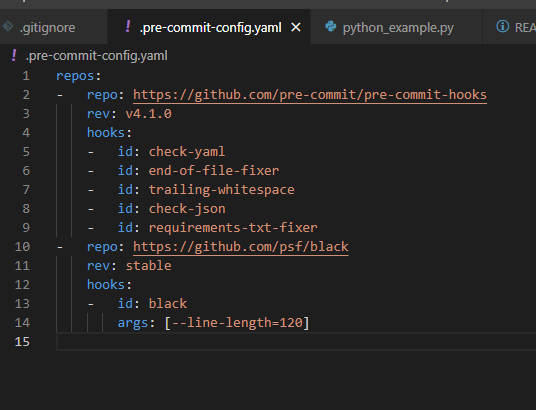
- id: trailing-whitespace

- repo: https://github.com/psf/black

rev: 21.12b0

hooks:

- id: black



Git add .

Pre-commit run

Git commit -m “message” (this now picks up the pre-commit also)