

PROJECT-2

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

INDEX

Sr.no.	Topic name	Pg. no.
1.	Introduction to PyCharm	4
2.	Basic Python development in PyCharm	10
3.	Code Management and Version Control	18
4.	Debugging and Problem Solving	22
5.	Advanced Python Development	23
6.	Python Development Challenge	25

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

Objectives:

The goal of this project is to help users learn and gain practical experience using PyCharm, which is a widely used tool for purpose writing and organizing Python code. The main focus is to make users more productive and efficient in their software development tasks by teaching them how to use the various features and tools available in PyCharm. Learning PyCharm is valuable because it's a tool that can significantly improve the way users write, organize, and manage Python code. PyCharm skill is beneficial for their current tasks and future projects. The main goal of PyCharm is to provide a highly productive and efficient Integrated Development Environment (IDE) for Python developers. PyCharm aims is to make the process of writing, navigating, and managing Python code as smooth and effective as possible.

1. Introduction to PyCharm

“PyCharm” means a charming or attractive tool for Python development. It is a combination of two words. “Py” for python which is popular programming language and “Charm “ it means attractive or appealing quality. PyCharm is a powerful and user-friendly Integrated Development Environment (IDE) specifically designed for Python developers. It's a helpful tool that makes working with Python code easier and faster with intelligent suggestions, auto-completion, and error detection. Instead of just typing code in a regular text editor, PyCharm is like a smart assistant that makes coding easier and faster. An IDE is like a digital workspace that makes writing, organizing, and managing Python code easier and more efficient. Debugging is made easy with built-in tools that help you identify and fix errors in your Python code step by step. PyCharm integrates with version control systems like Git, allowing for efficient collaboration with other developers. It assists in enhancing the quality of your code through the features like code analysis, inspections, and tools for code refactoring. PyCharm provides tailored features and support for popular web frameworks for building a web-applications with python. PyCharm supports the creation and execution of unit tests, ensuring the reliability of your Python code. It suggests words and completes sentences, making it quicker to write code. If there's something wrong in your code, PyCharm shows you where and helps you fix it. Interact with databases directly from the IDE, simplifying tasks related to managing data in your applications. Overview of PyCharm and its importance in Python development.

Why PyCharm?

Efficiency: PyCharm tool is designed to make Python development faster and more efficient.

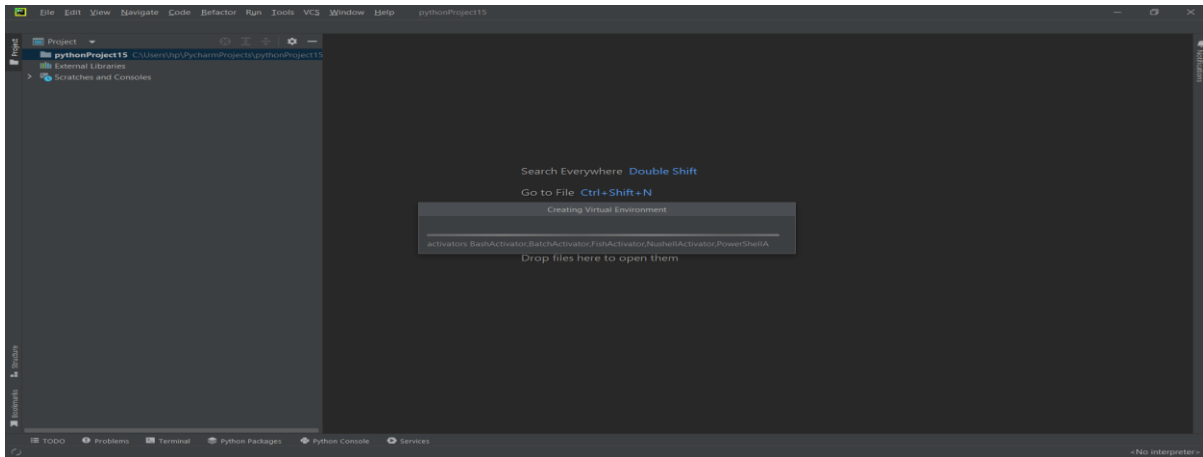
Ease of Use: It is user-friendly interface makes it accessible for both beginners and experienced developers.

Comprehensive Toolset: Start to end i.e. From writing code to debugging and collaboration, PyCharm provides a comprehensive set of tools to users in one place.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

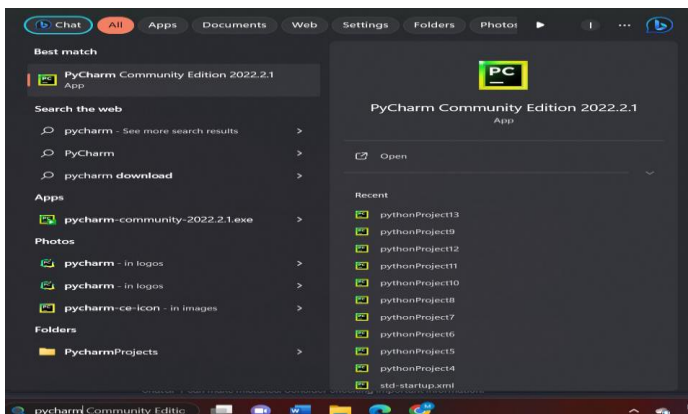
1.1. Setting up and configuring PyCharm for first-time use

Installation steps: Go to chrome browser → search “PyCharm community edition” → Click <https://www.jetbrains.com/pycharm/download/> . → download latest version of python <https://www.python.org/downloads/> .



1.2. Familiarizing with the PyCharm user interface and navigation

This is the first window after open the editor. The main area of the interface is the editor window, where you write and edit Python code. This window is equipped with features like syntax highlighting, code completion, and error checking.

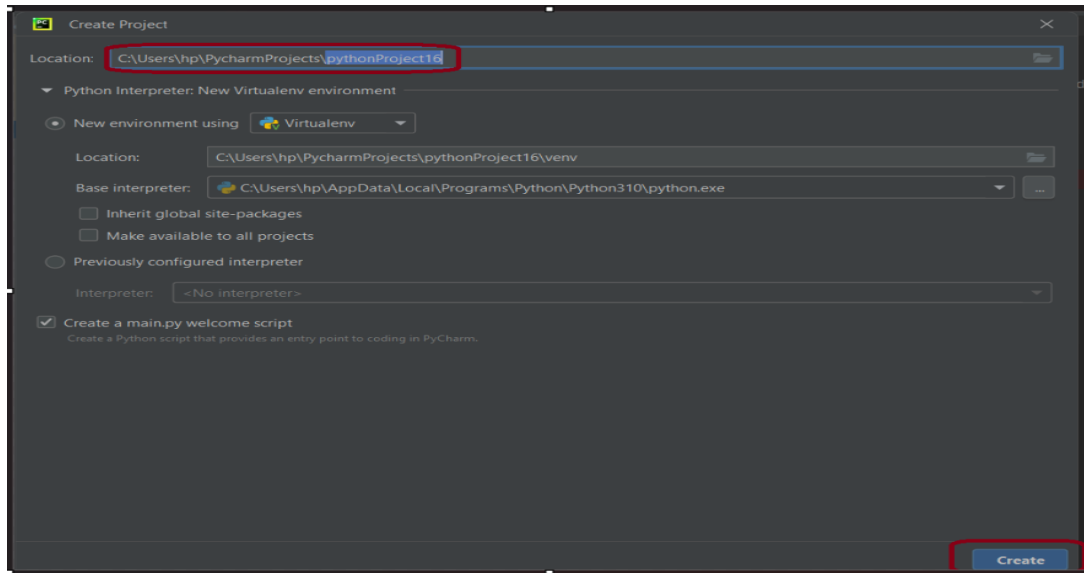


For creating new project click on “File” → click on new project → add file location and create.

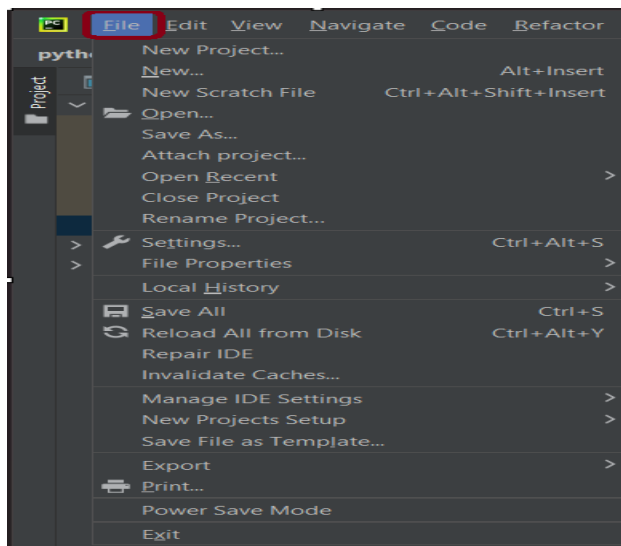
- From this file option you can create multiple new project.
- You can open our previous files just click on “open “.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

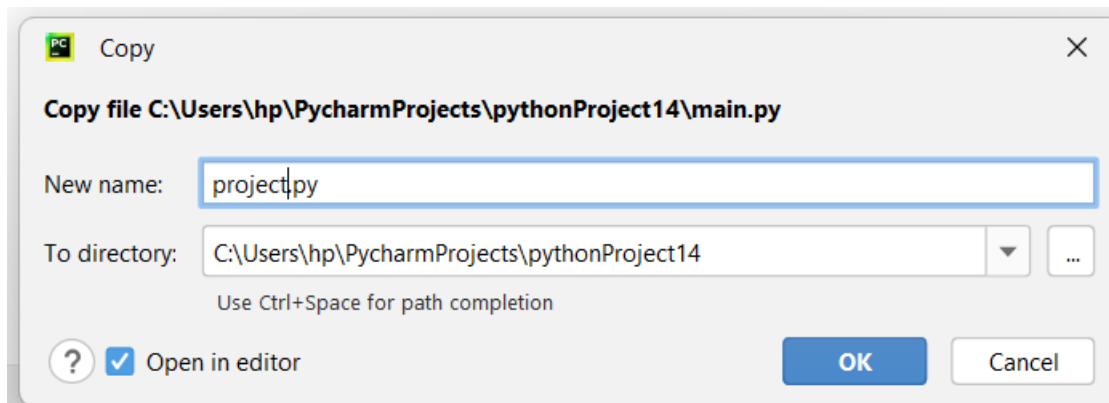
- You can attach another project to recent one. You can rename the project using rename button.



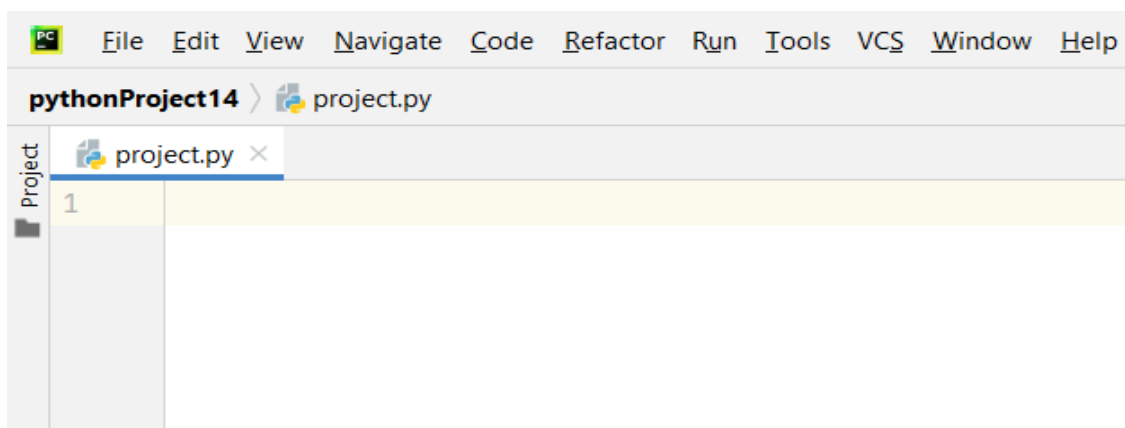
- To save current window click on file → click on “save as” option to save → Now you have to put the file name whatever you want to put with extension “.py” → syntax: filename.py → ok.



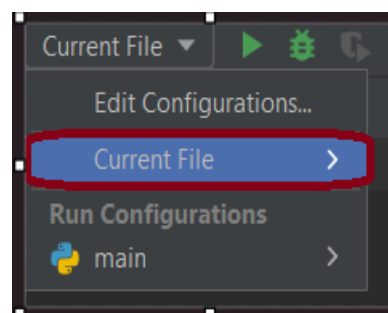
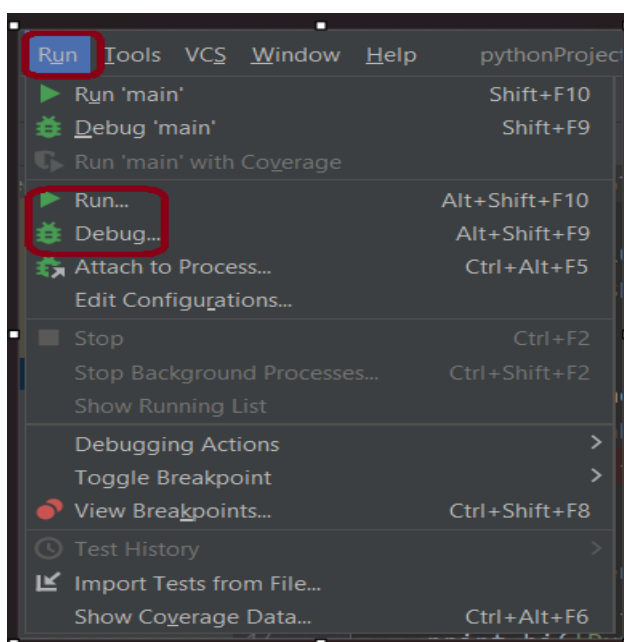
“Efficient Development with PyCharm: Mastering the PyCharm IDE”



- This is the window where the user can run multiple tasks of projects.



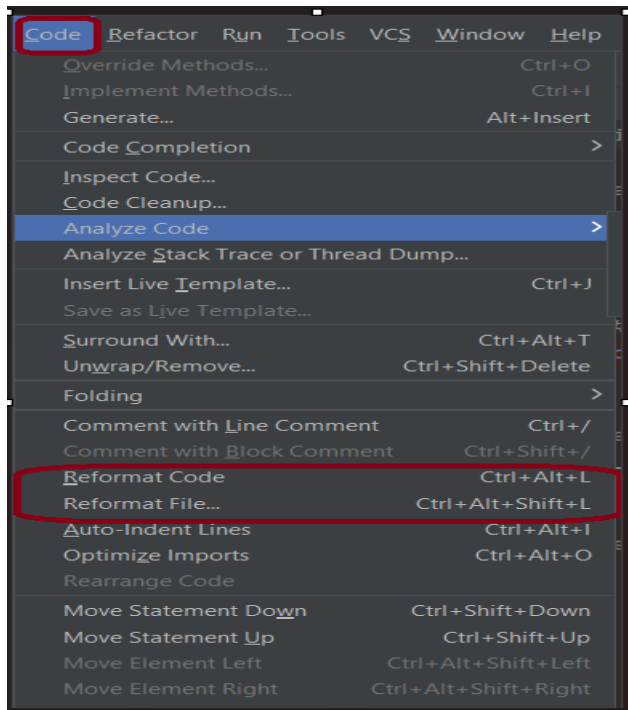
- Run project file using Run button either you can set a current file at the of window as default to run automatically after click run.



“Efficient Development with PyCharm: Mastering the PyCharm IDE”

Why Reformat code/file?

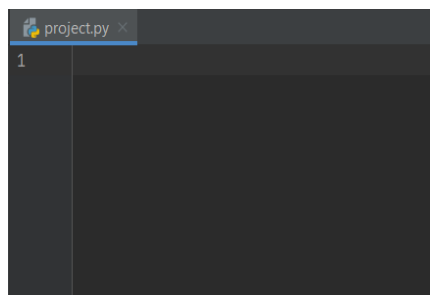
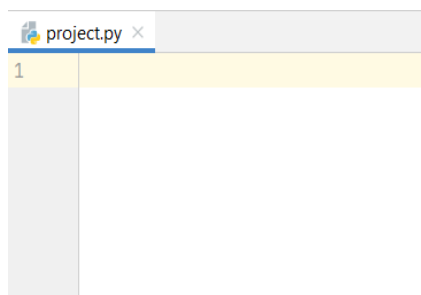
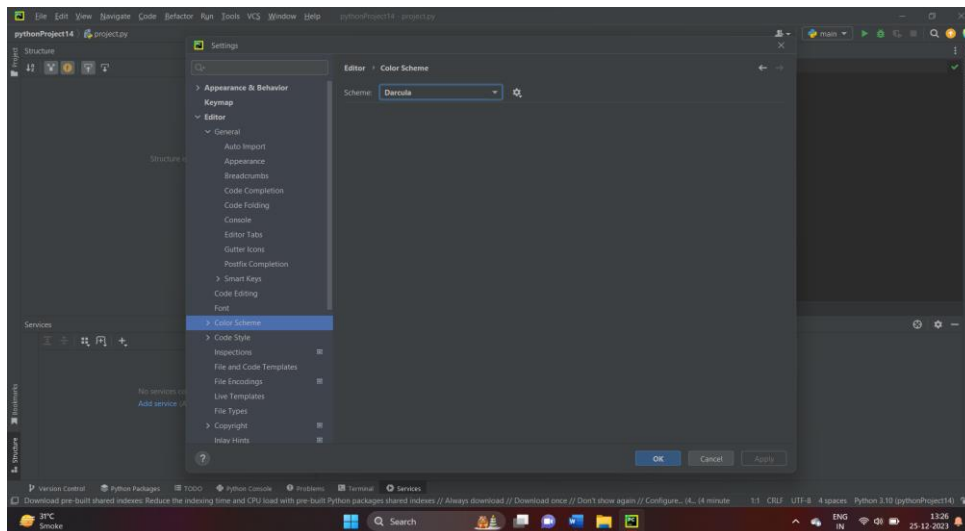
- Code→reformat file→reformat code.
- Reformat Code option in PyCharm is a valuable tool for maintaining a uniform and professional coding style, promoting collaboration. It helps the quality and readability of your Python codebase.



To change PyCharm Theme:

- Click top right corner of page→configure editor tabs→color scheme→click “Dacula” option to change PyCharm theme to black screen.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”



2. Basic Python Development in PyCharm

2.1. Creating and managing Python projects.

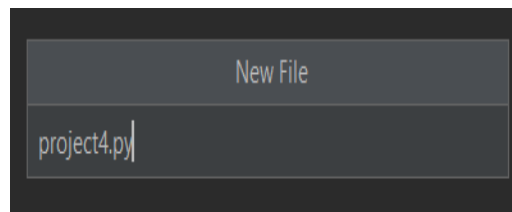
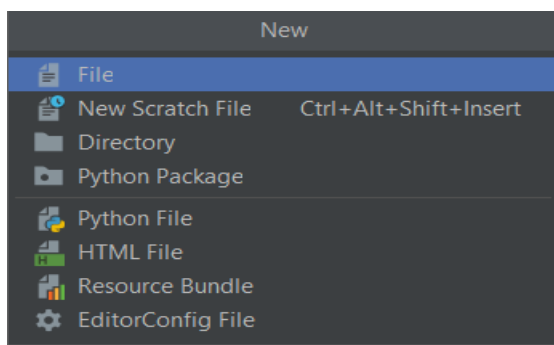
In PyCharm Python scripts play a central role as they are the files containing the actual Python code you write to perform specific tasks or functions.

How to create new project?

- Open PyCharm>File →Create new project

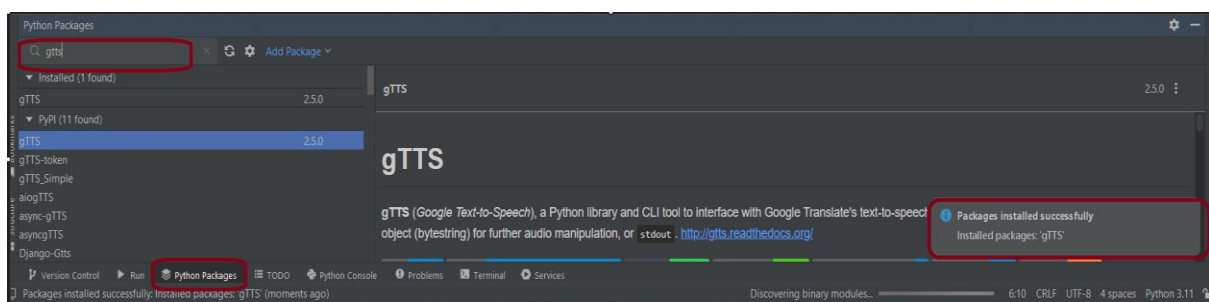
How to create new file within project?

- File→new>save with .py extension



How to install packages in PyCharm ?

- Bottom of page→click on python packages→search for it→you'll get the result.

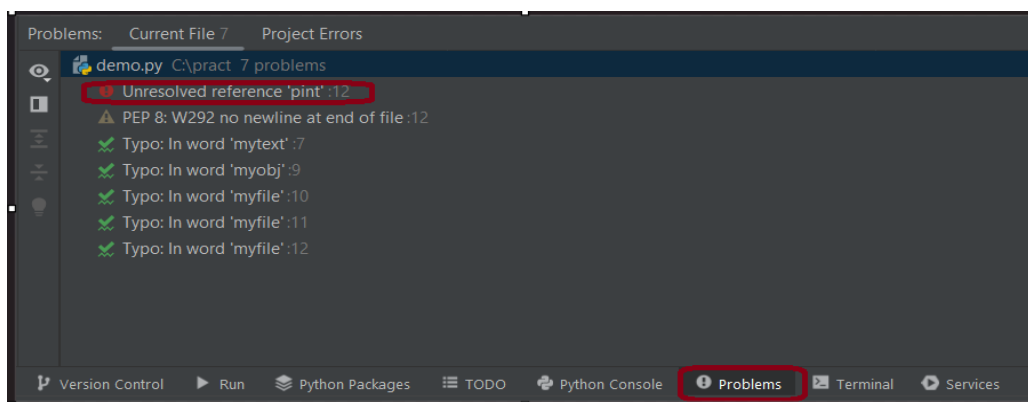


2.2. Writing and executing Python scripts

- Python scripts help organize your code into logical units. Each script can represent a module, class, or a specific functionality, making it easier to manage and maintain your codebase.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

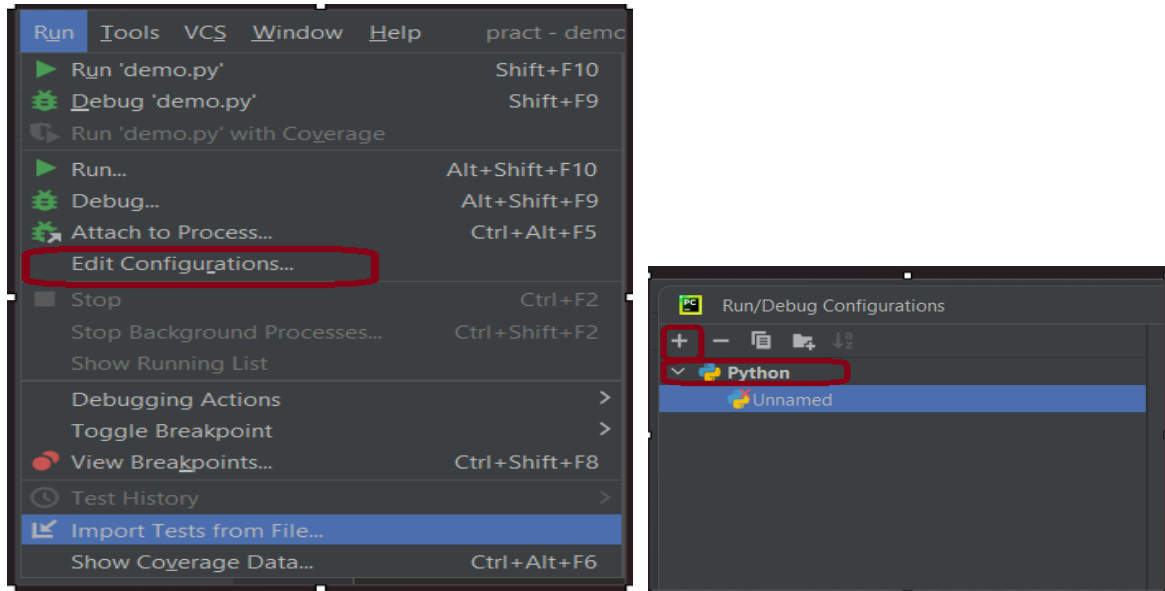
- Python scripts contain the instructions and logic for performing tasks. Whether it's data analysis, web development, automation, or any other Python-related task, scripts encapsulate the implementation details.
 - This tool provides a dedicated editor for writing and editing Python scripts. This editor offers features like syntax highlighting, code completion, and error checking. It is beneficial to users.
 - Python scripts contribute to the overall project structure. In PyCharm, you organize your code into files and directories, and Python scripts. Python scripts are the building blocks of your Python projects in PyCharm. They contain the logic, functionality, and instructions that define how your application works.
 - Working with Python scripts in PyCharm provides a structured and efficient approach to software development
- To see problems within file Go bottom of page→problems.



How to add python scripts?

- Click on Run→edit configuration→click ‘+’ this sign →python→add script path →apply.

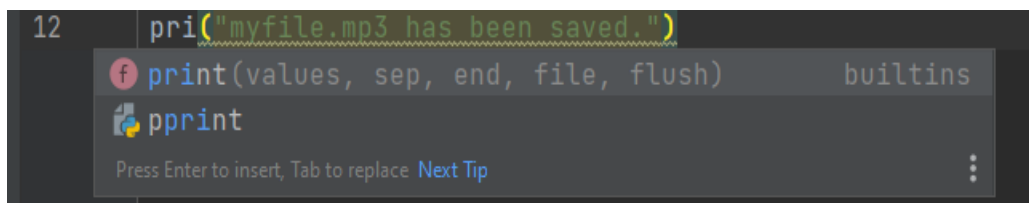
“Efficient Development with PyCharm: Mastering the PyCharm IDE”



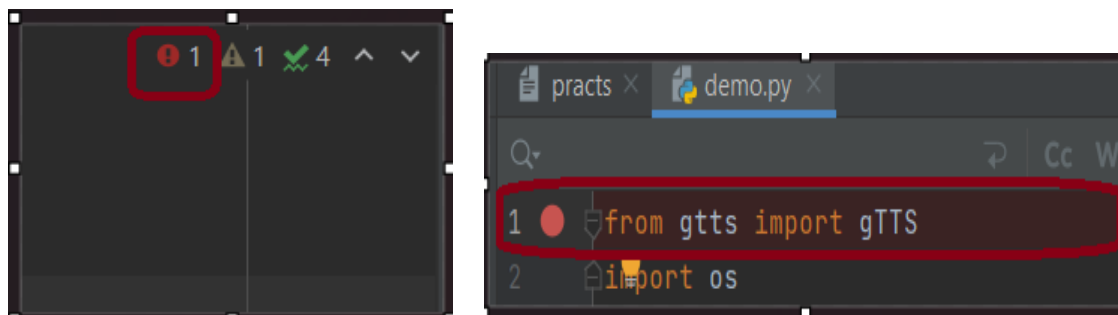
- If there is need to debug your script, you can set breakpoints in the code and use the ‘Debug’ button instead of Run.
- To stop the execution of the script, click the red square "Stop" button in the toolbar.

2.3. Understanding the editor and its features like syntax highlighting, code completion, and quick navigation.

- Just write one/ two letters you will get suggestion of it→double click on suitable suggestion.

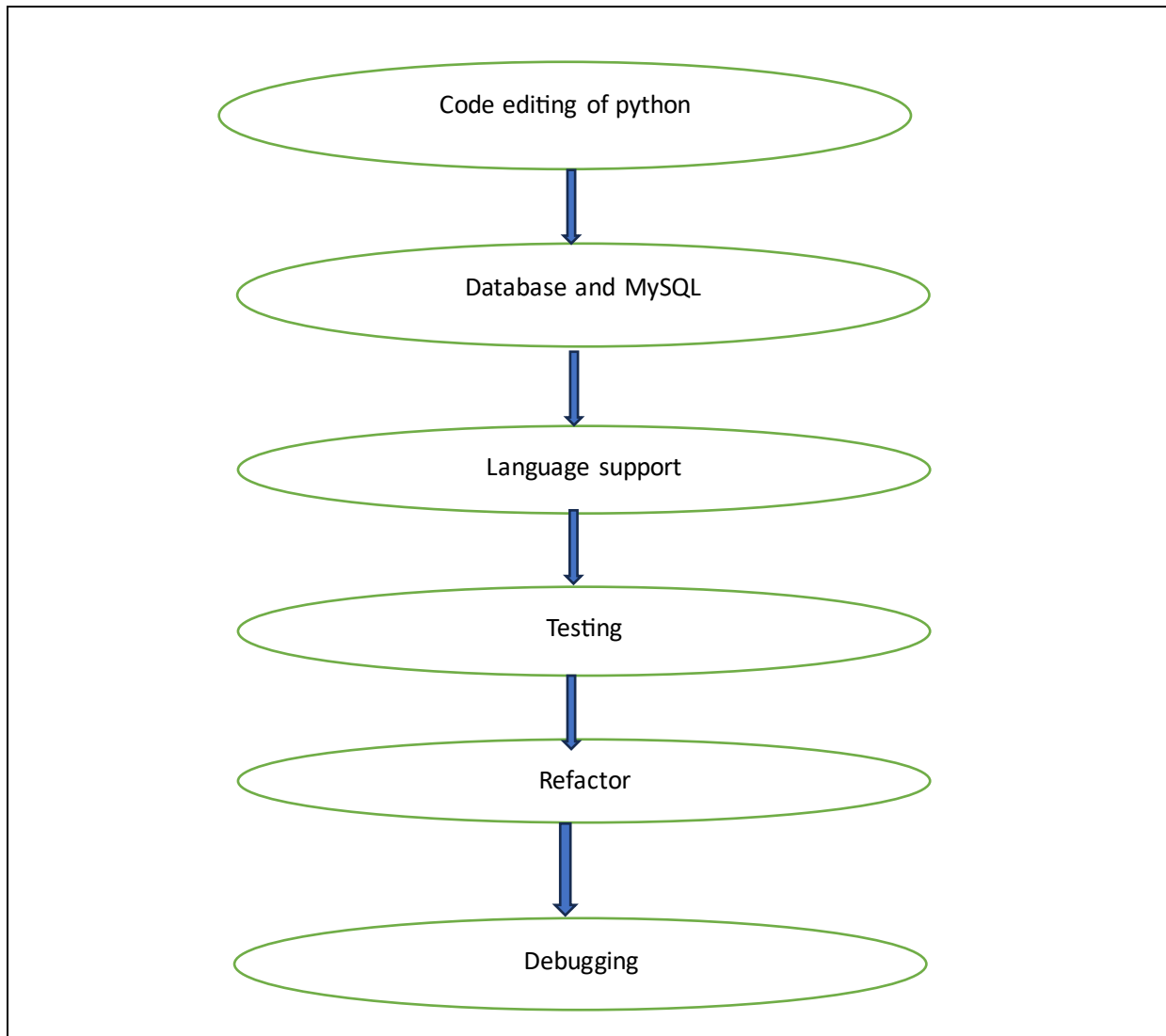


- Red line shown the errors in code.



“Efficient Development with PyCharm: Mastering the PyCharm IDE”

Features of PyCharm:



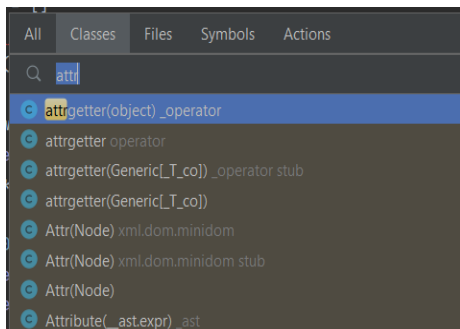
Quick Navigation in PyCharm:

In PyCharm, "Quick Navigation" refers to the ability to swiftly move around your codebase, files, and various parts of the IDE using keyboard shortcuts and commands.

1. Navigation class/file/symbol

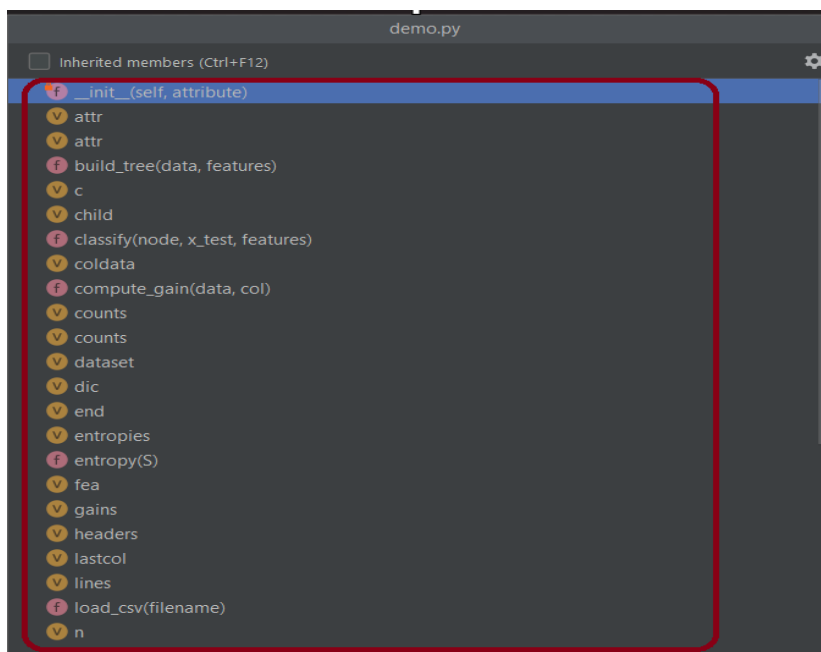
- Ctrl + N to open a dialog
- search for and navigate to a specific class or file by typing its name.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”



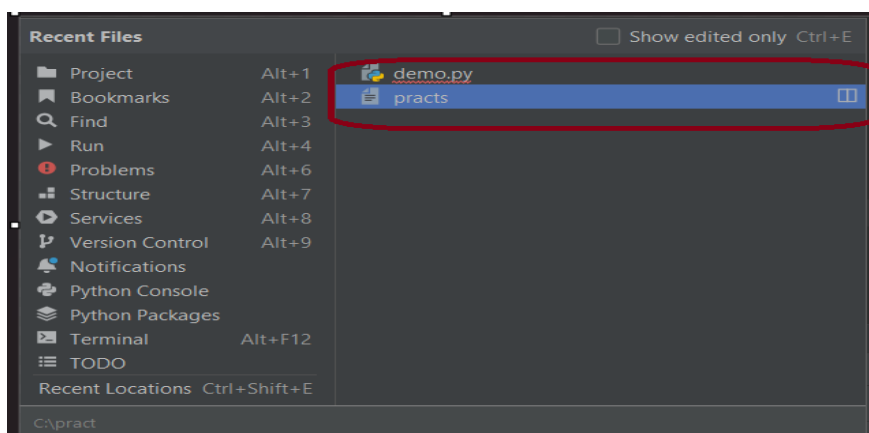
2. Navigate list

- Ctrl + F12
- To quickly list and navigate to methods or fields within the current file.



3. Navigate to Recent Files

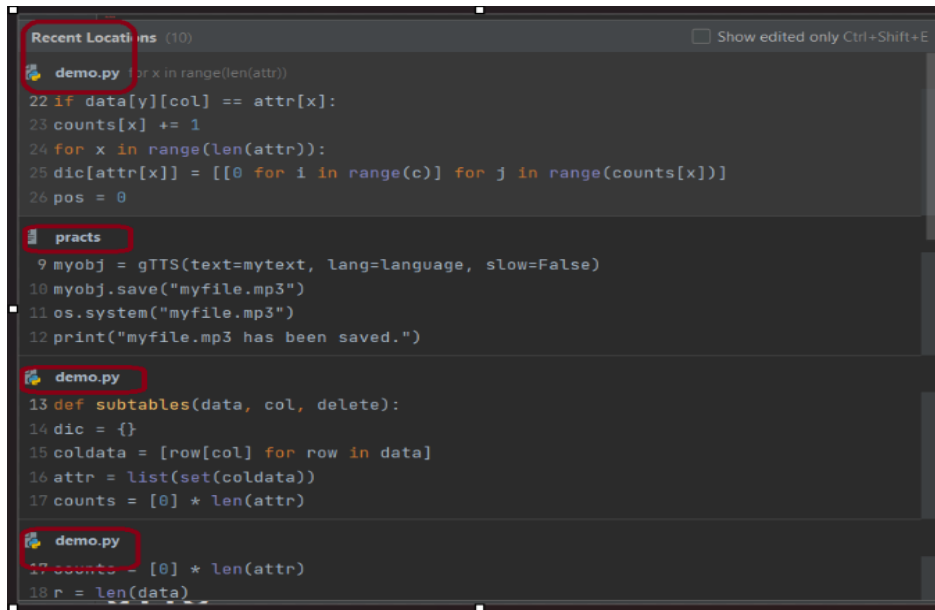
- Ctrl + E
- To open a popup showing a list of recently opened files. Select a file to jump to it.



“Efficient Development with PyCharm: Mastering the PyCharm IDE”

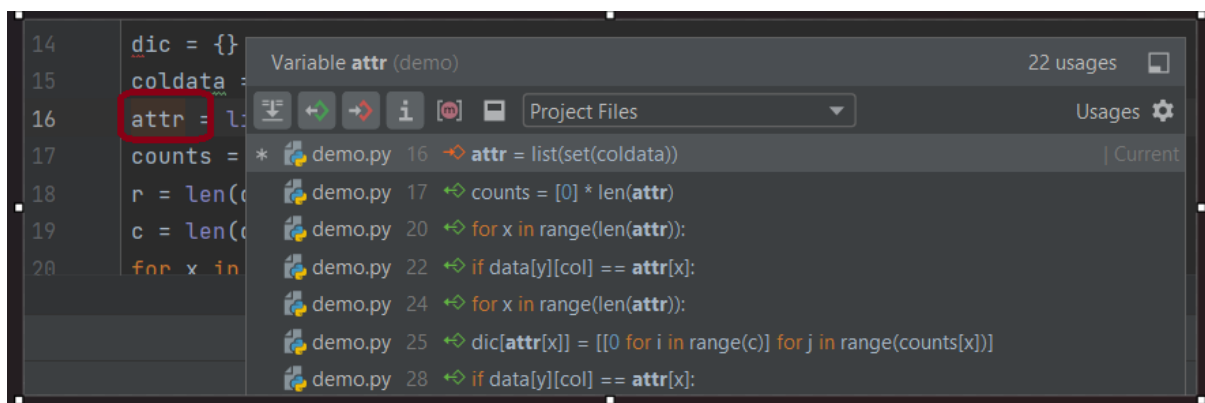
4. Navigate to Recent Edits:

- Ctrl + Shift + E
- To open a popup with a list of recently edited files. Select a file to jump to the location of your recent edits.



5. Navigate to Declaration/Usage:

- Place on a variable, method, or class name → use Ctrl + B
- To jump to its declaration or usages.



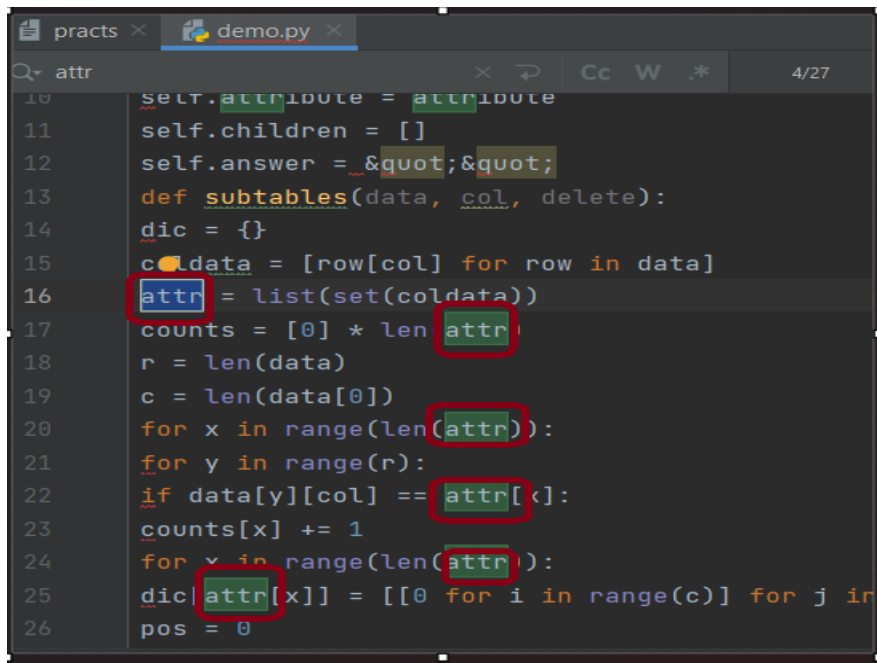
6. Navigate Back/Forward:

- Ctrl + Alt + Left and Ctrl + Alt + Right
- To navigate back and forward through your navigation history.

7. Navigate to File Structure:

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

- Ctrl + F3
 - To open a popup displaying the file structure. Select a class or method to jump to its location.

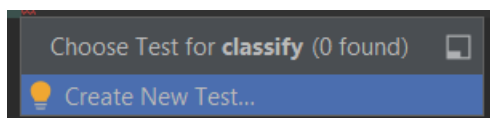


8. Navigate to Super Method/Class:

- Ctrl + U
 - To jump to the super method or class of the current method or class.

9. Navigate to Test:

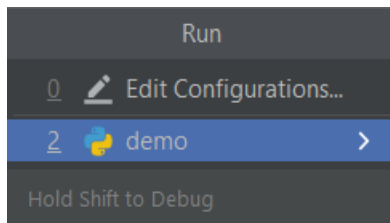
- Ctrl + Shift + T
 - To navigate to the test for the current class or method.



10. Navigate to Run Configuration:

- Alt + Shift + F10
 - To open the Run/Debug configuration popup. Select a configuration and press Enter to run/debug it.

“Efficient Development with PyCharm: Mastering the PyCharm IDE”

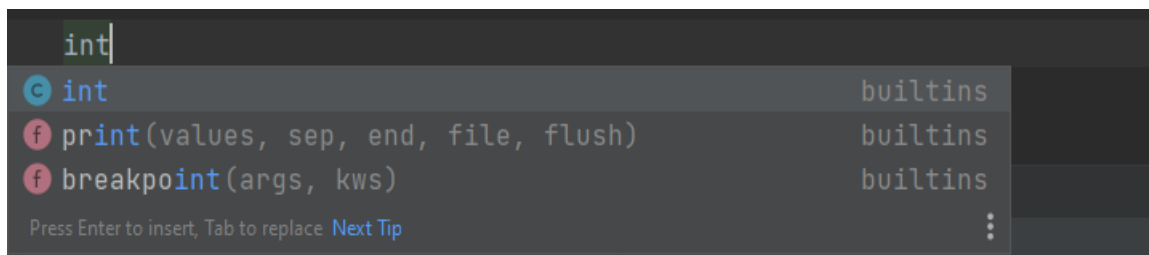


These shortcuts can improve navigation speed within PyCharm, allowing you to focus more on coding and less on manually finding files or declarations.

3. Code Management and Version Control

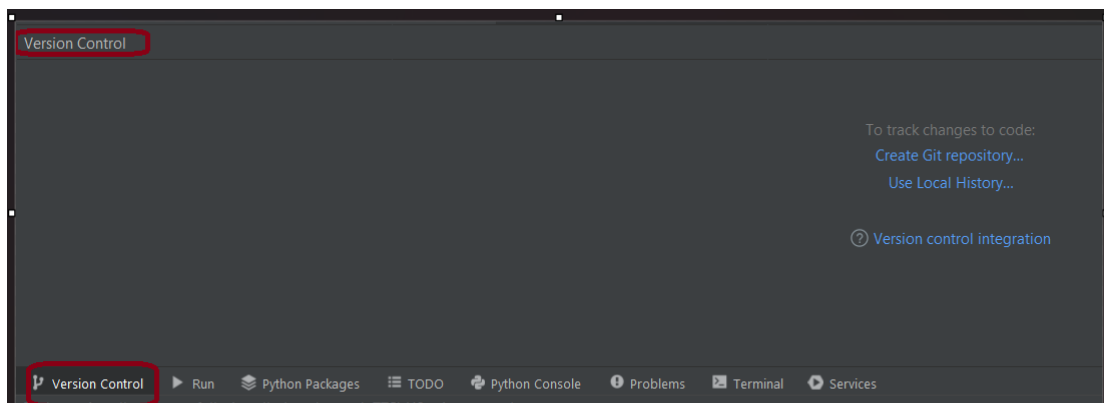
Why use code management?

- Code management in PyCharm is essential for creating an organized, maintainable, and collaborative development environment. It improves code quality, streamlines collaboration, and enhances the overall efficiency of the development process.
- With code management, developers can quickly navigate through the project. PyCharm provides features like "Navigate to" and quick navigation shortcuts, making it efficient to move between files, classes, and methods.
- When you typing the function name it suggest a functions below.



Why use version control ?

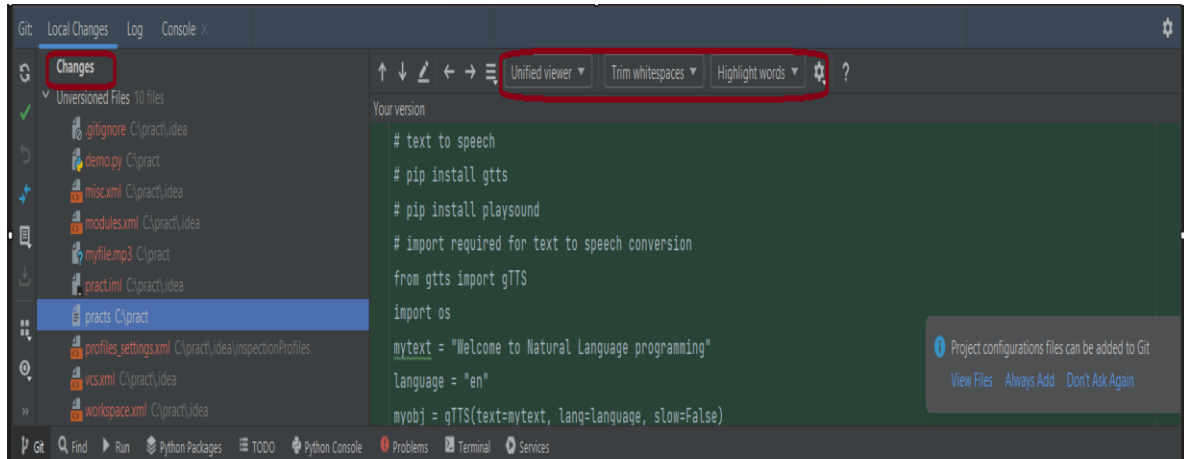
- Using version control systems like Git within PyCharm, Python scripts are the files you commit, push, and pull. They are the primary components tracked by version control.
- Using version control features in PyCharm, developers can effectively manage code changes, collaborate with team members, and ensure the stability of the codebase throughout the development lifecycle.
- PyCharm allows developers to view annotations directly in the code, showing who last modified a specific line. This helps in understanding the origin of changes.



"Efficient Development with PyCharm: Mastering the PyCharm IDE"

3.1. Integrating with Git and other version control systems.

- Open Your Project → Enable Version Control → Create a Branch → Make Code Changes → Review Changes.



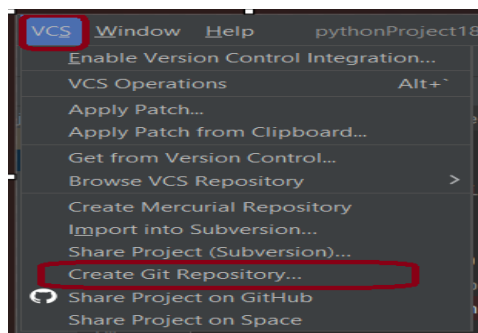
3.2. Using PyCharm for Committing, pushing, and pulling code.

Git setting:

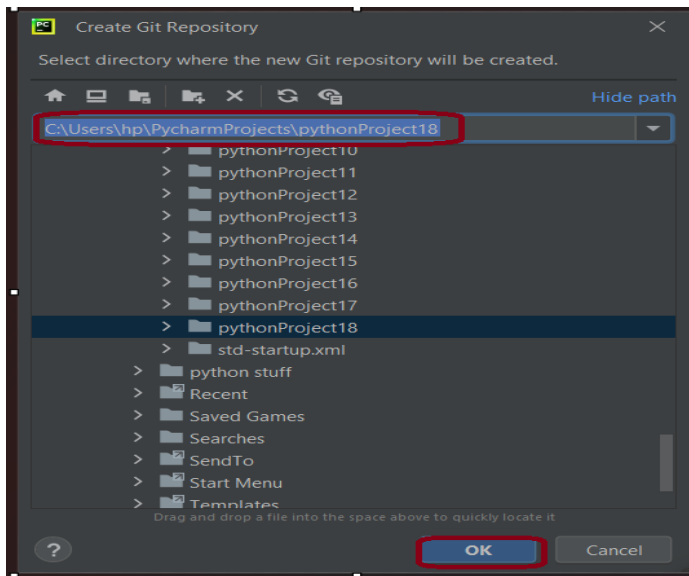
- Go to "File" → "Settings" → "Version Control" → "Git" to configure your Git username, email, and other settings.

How to create repository?

- VCS → create new repository → yes.

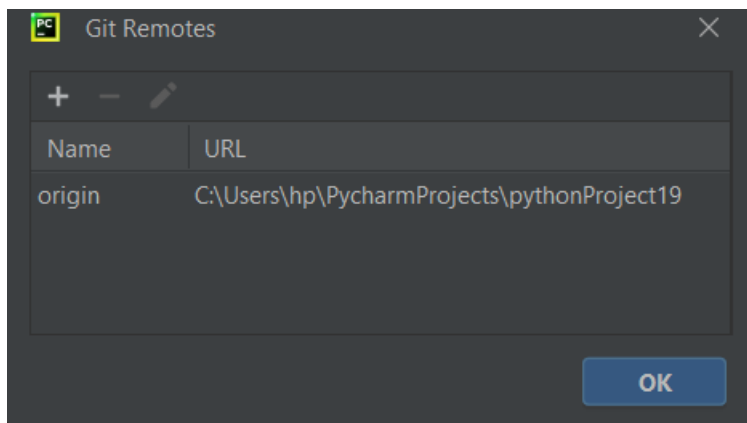


“Efficient Development with PyCharm: Mastering the PyCharm IDE”

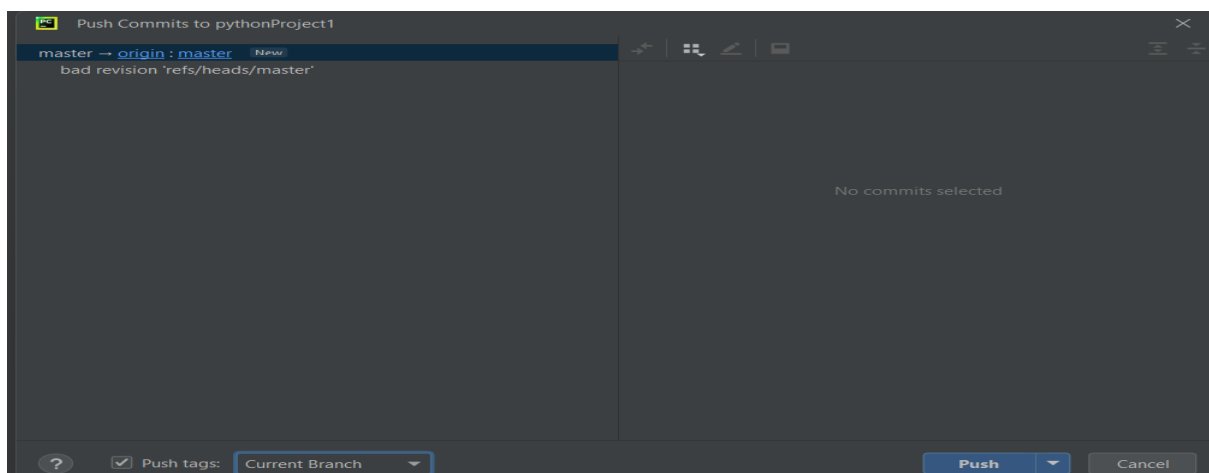


How to push code on git?

- Open PyCharm project → Git → manage remote → ok



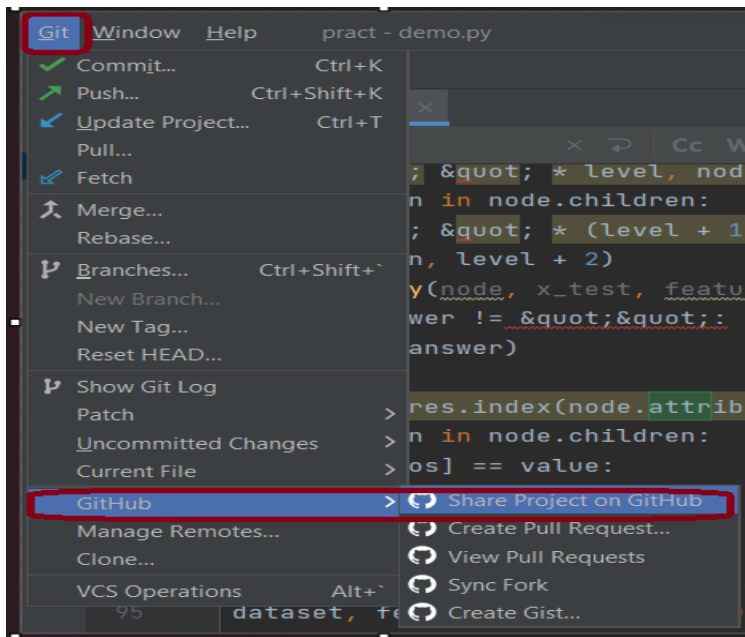
- Git → push → ok.



“Efficient Development with PyCharm: Mastering the PyCharm IDE”

How to share code on git?

- Go Git→github →Share project on Github →add account→ok →create.



4.Debugging and Problem Solving

- Open Your Python File→Add Breakpoints→To Create a Debug Configuration→Go to the top-right corner of the PyCharm window→Click on the dropdown menu →Select Edit Configurations→Click on the "+" icon to add a new configuration, choose "Python," and give a name.

How to Specify Script and Parameters?

- configuration settings→specify the Python script→Set the working directory if needed→Apply and Close.

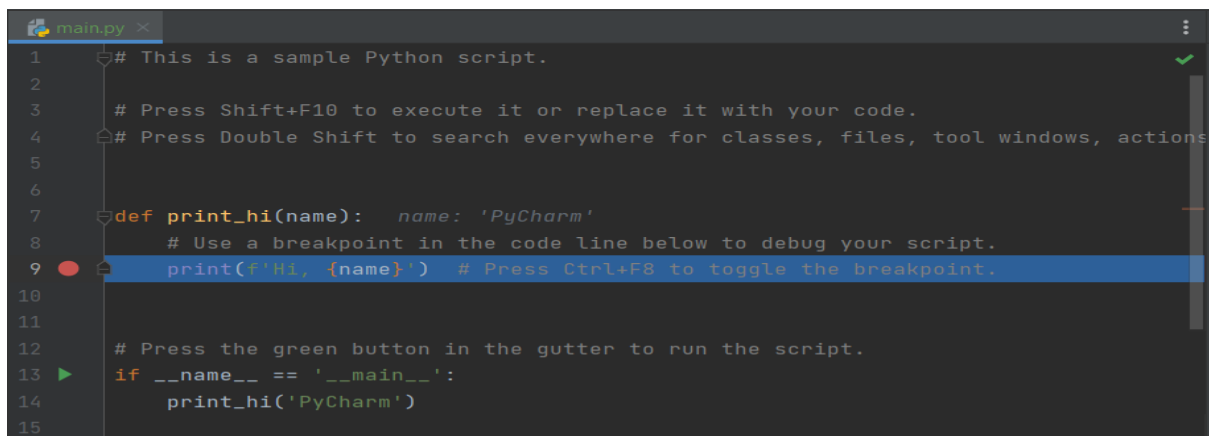
How to Start the Debugger?

- Run the Debug Configuration or use shift+F9
- PyCharm opens Console at the bottom of the window→The execution stops at the breakpoints you set.

5. Advanced Python Development

5.1. Exploring advanced PyCharm features such as code refactoring, database support, and remote development.

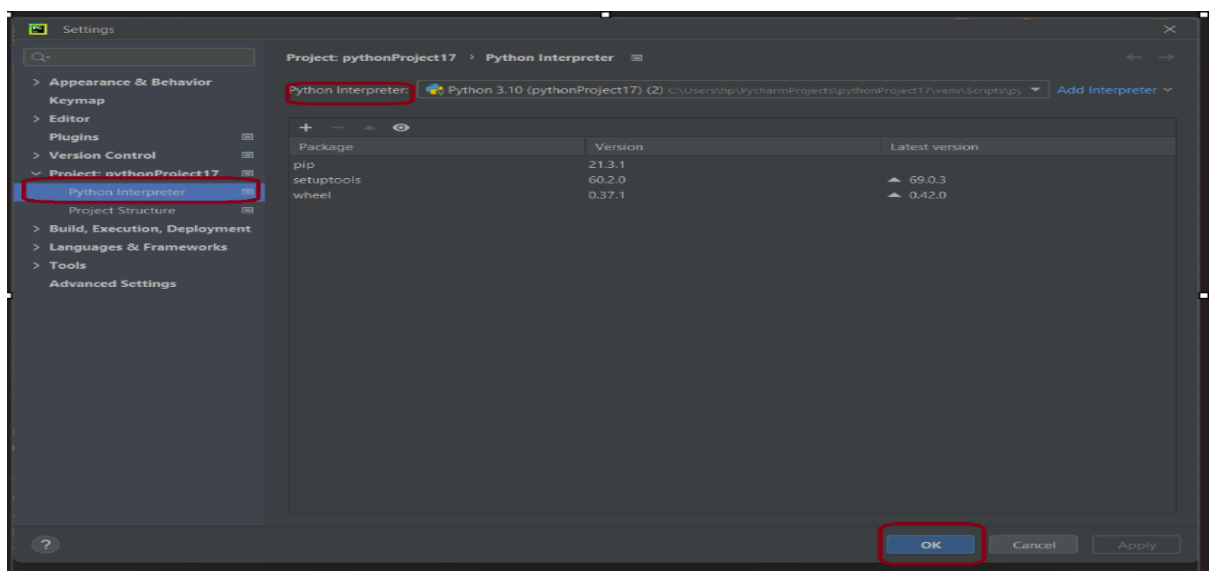
- Set up run configurations to execute scripts on the remote interpreter.
- Debugging works seamlessly, allowing you to set breakpoints and inspect variables as if you were working locally.



5.2. Using virtual environments and managing Python packages.

How to work Virtual environment in PyCharm?

- Go to File → Settings → project → Project_Name > Python Interpreter → Click on the gear icon and select Add → Virtualenv Environment/Pipenv Environment → Specify the location and click OK → PyCharm will set up the virtual environment for your project.



"Efficient Development with PyCharm: Mastering the PyCharm IDE"

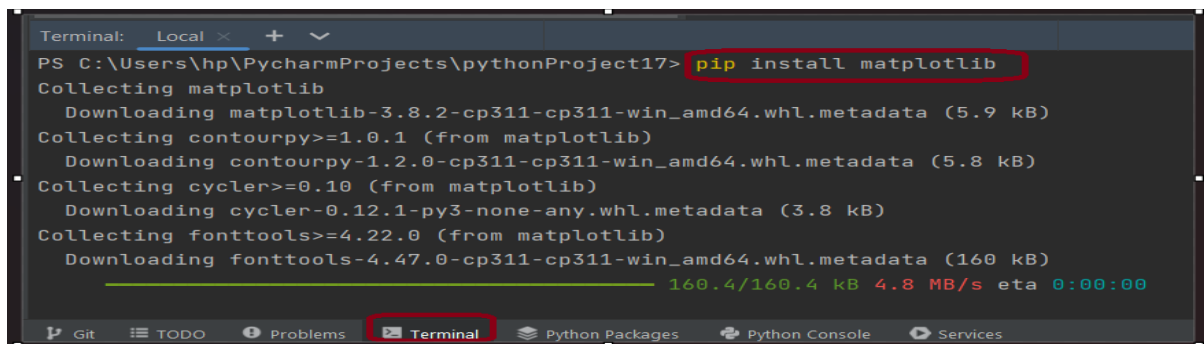
How to Activate/Deactivate Virtual Environment?

Hence once virtual environment is created then it is automatically activated. You can see the active virtual environment in the status bar at the bottom of your page.

- terminal in PyCharm → run command as deactivate.

Package management:

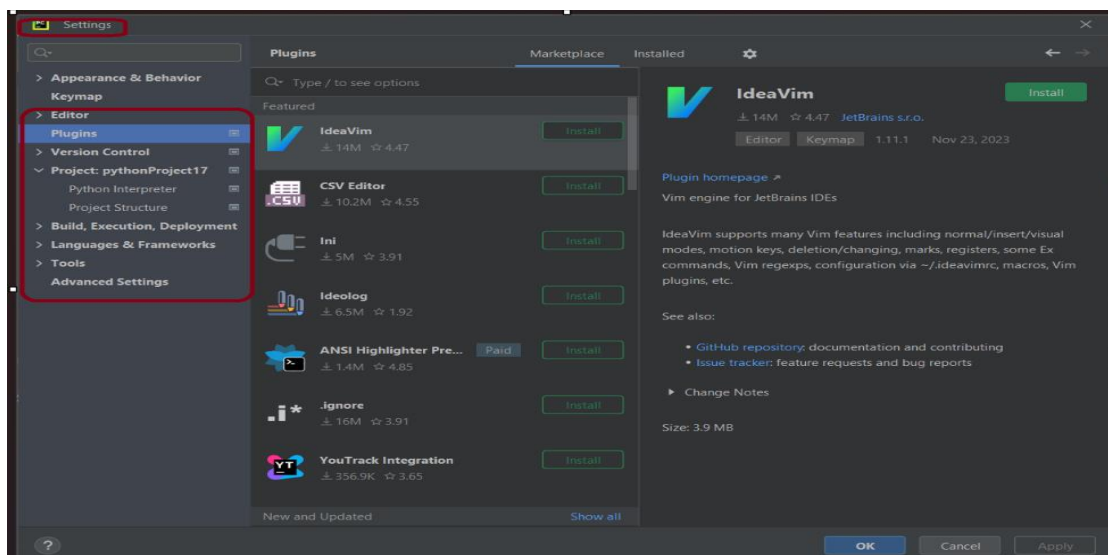
Open terminal in PyCharm → use pip to install packages → syntax: `pip install package_name`



```
Terminal: Local x + v
PS C:\Users\hp\PycharmProjects\pythonProject17> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.8.2-cp311-cp311-win_amd64.whl.metadata (5.9 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.2.0-cp311-cp311-win_amd64.whl.metadata (5.8 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.47.0-cp311-cp311-win_amd64.whl.metadata (160 kB)
160.4/160.4 kB 4.8 MB/s eta 0:00:00
```

5.3. Customizing the IDE with plugins and themes.

- PyCharm, being a feature-rich Integrated Development Environment (IDE) for Python, comes with various settings and configurations to customize the development environment according to the user's preferences.
- Right Bottom of page → interpreter Setting → settings → set plugins → version control → and so on as need.

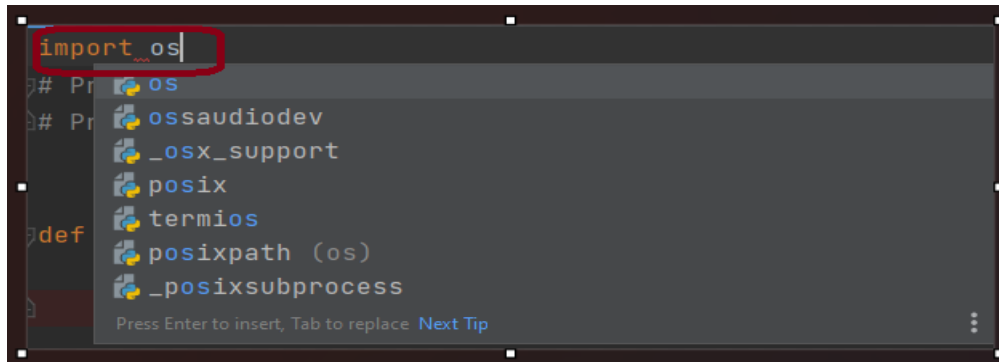


6. Python Development Challenge

6.1. Focus on code efficiency, readability, and use of PyCharm's advanced features.

Autocompletion feature:

- For autocompletion used this advanced features → `import os / os.path .`



Evaluate Expression:

- Evaluate Expression feature to execute Python expressions during debugging.

Conclusion:

PyCharm is rich set of features and intuitive interface make it a powerful Integrated Development Environment (IDE) for Python developers. The editor provides comprehensive tools for writing, organizing, and managing Python code efficiently. Gain hands-on experience in debugging Python code with built-in tools, enabling the identification and resolution of errors step by step. Explore features such as code analysis, inspections, and tools for code refactoring to enhance the quality and readability of Python code. Acquire the ability to interact with databases directly from PyCharm, simplifying tasks related to managing data in Python applications.

Learning Outcomes:

- Comprehensive understanding and hands-on experience with PyCharm.
- Enhanced efficiency in writing, testing, and debugging Python code.
- Proficiency in code management and version control integration.
- Advanced skills in Python development and IDE customization.