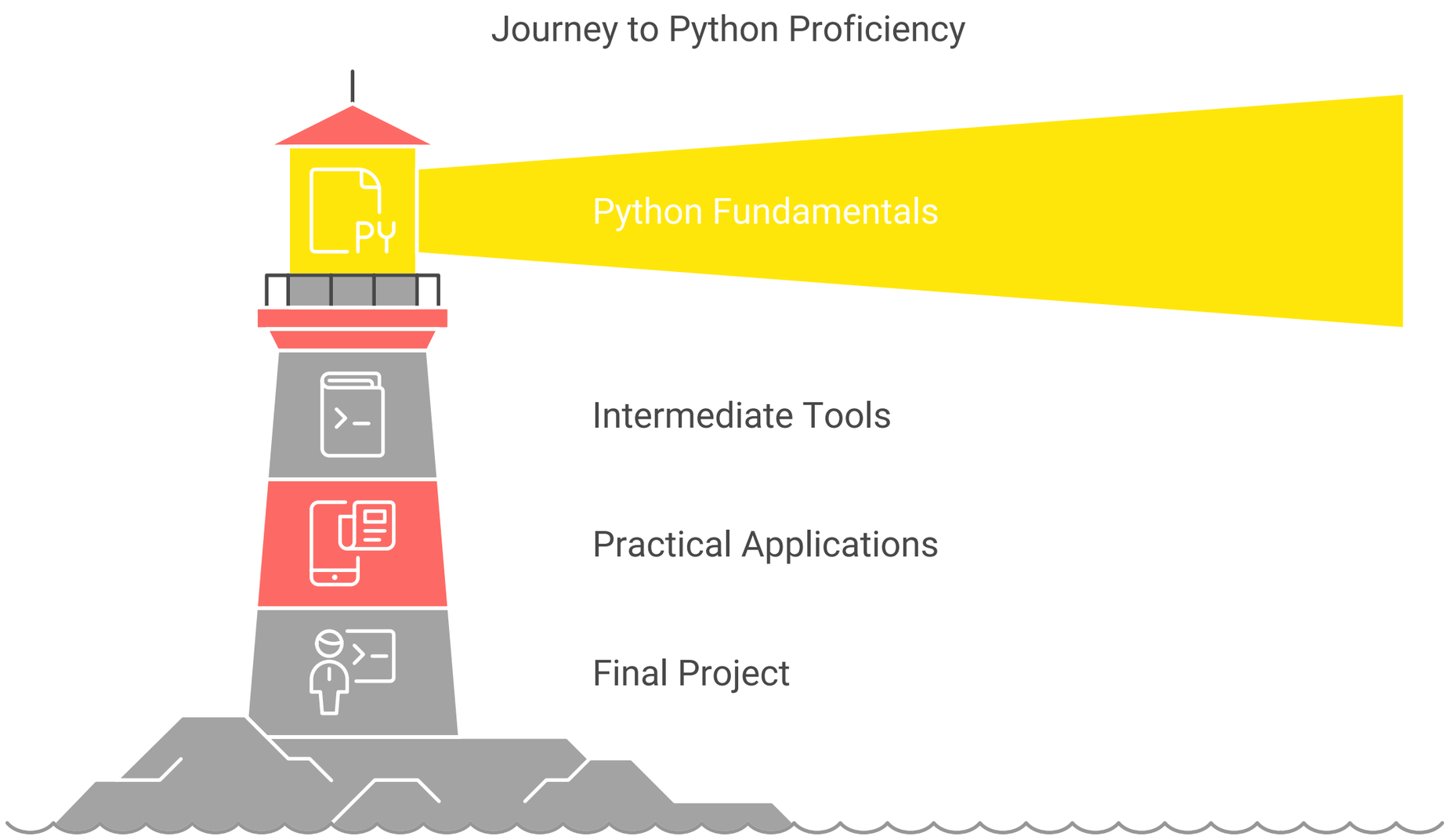
Python



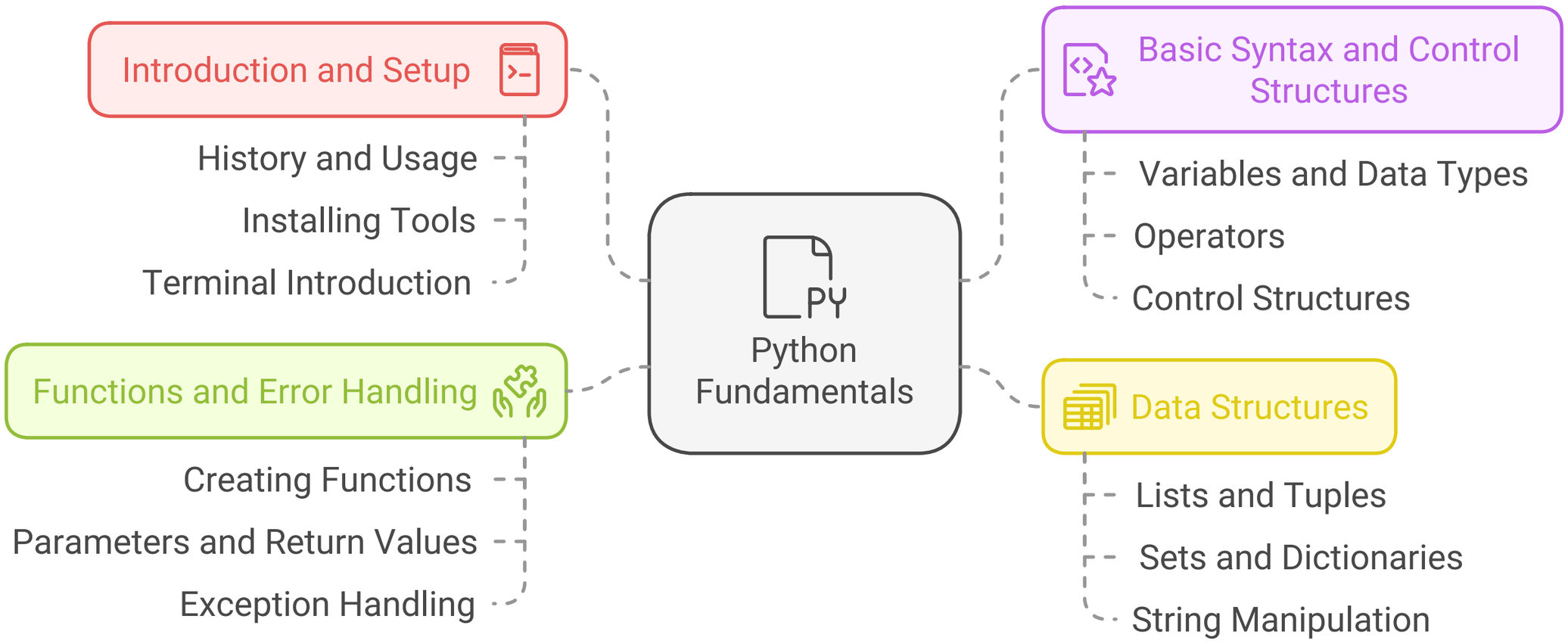
Month 1: Python FundamentalsGoal: Understand basic syntax, fundamental programming concepts, and work with simple data.

Week 1: Introduction and Setup
History and usage of Python:Background of the language.Current applications (web development, data science, AI, etc.).Installing tools:Install Python (from python.org).Set up an IDE (VS Code, PyCharm, or Jupyter Notebook).Introduction to the terminal and how to run Python scripts.Week 2: Basic syntax and control structures

Variables and data types: strings, numbers, booleans.Arithmetic, logical, and relational operators.Control structures:Conditionals (if, elif, else).Loops (for, while).Week 3: Data structures

Lists, tuples, sets, and dictionaries.Main methods for manipulating these structures (add, remove, search, etc.).Working with strings: common methods and text formatting.Week 4: Functions and error handling

Creating and using functions (def).Parameters, arguments, and return values.Exception handling (try, except, finally).



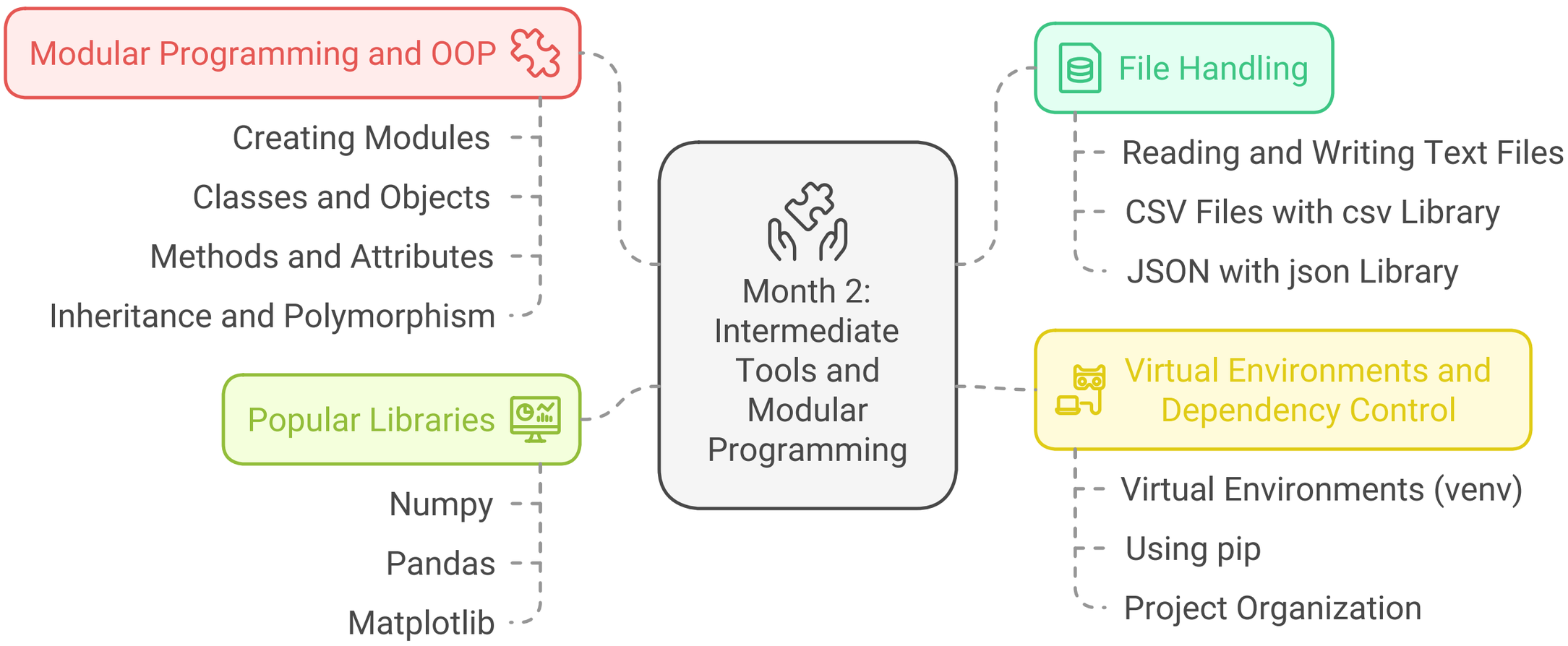
Month 2: Intermediate tools and modular programmingGoal: Learn advanced concepts, file manipulation, and the basics of structured programming.

Week 5: File handling
Reading and writing text files (open, read, write).Introduction to CSV files with the csv library.
Basic concepts of JSON and its use in Python (json).Week 6: Modular programming and OOP

Importing and creating modules.Introduction to Object Oriented Programming (OOP):Classes and objects.Methods and attributes.Inheritance and polymorphism.Week 7: Virtual environments and dependency control

Introduction to virtual environments (venv).Using pip to install external libraries.Best practices for organizing Python projects.Week 8: Introduction to popular libraries

Numpy (basic numerical calculations).Pandas (tabular data manipulation).Matplotlib (simple graphs).

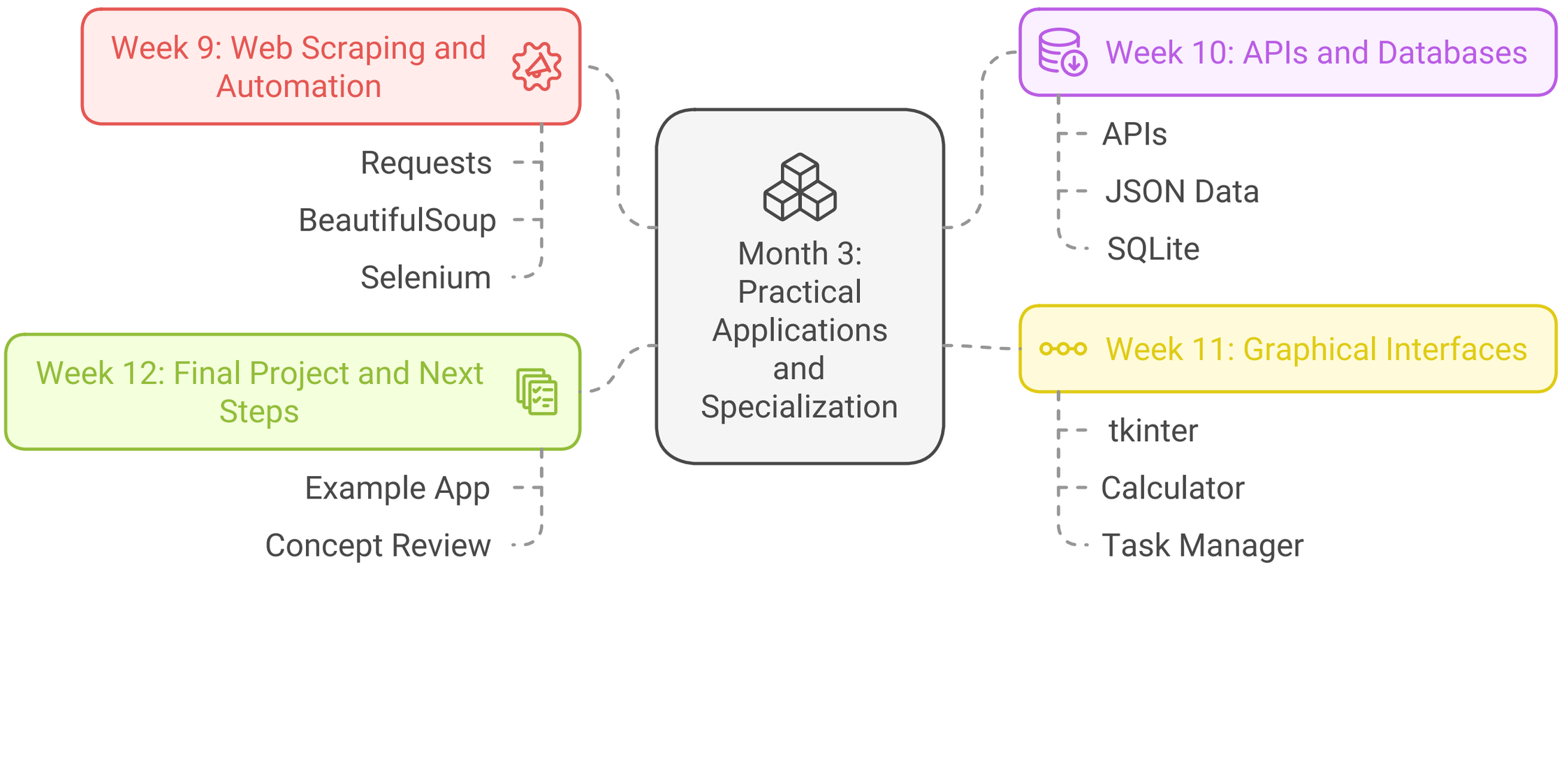


Month 3: Practical applications and specializationObjective: Integrate basic and intermediate knowledge to develop practical applications.

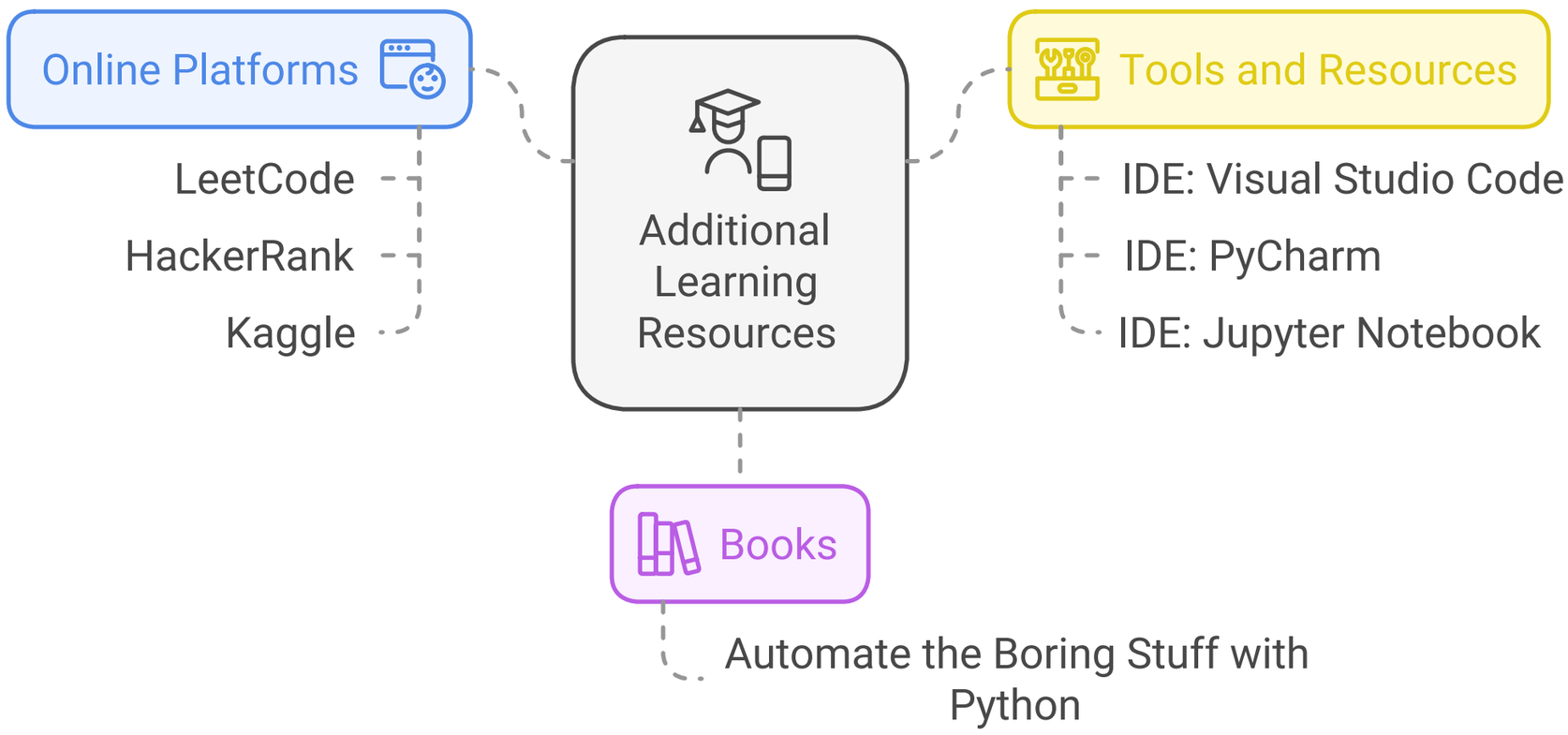
Week 9: Web scraping and automation
Introduction to requests and BeautifulSoup to extract data from websites.Using Selenium to automate browsers.Week 10: Working with APIs and databases

Introduction to APIs:Making requests to APIs with requests.Working with JSON data from public APIs.Basic use of SQLite with the sqlite3 library.Week 11: Creating graphical interfaces

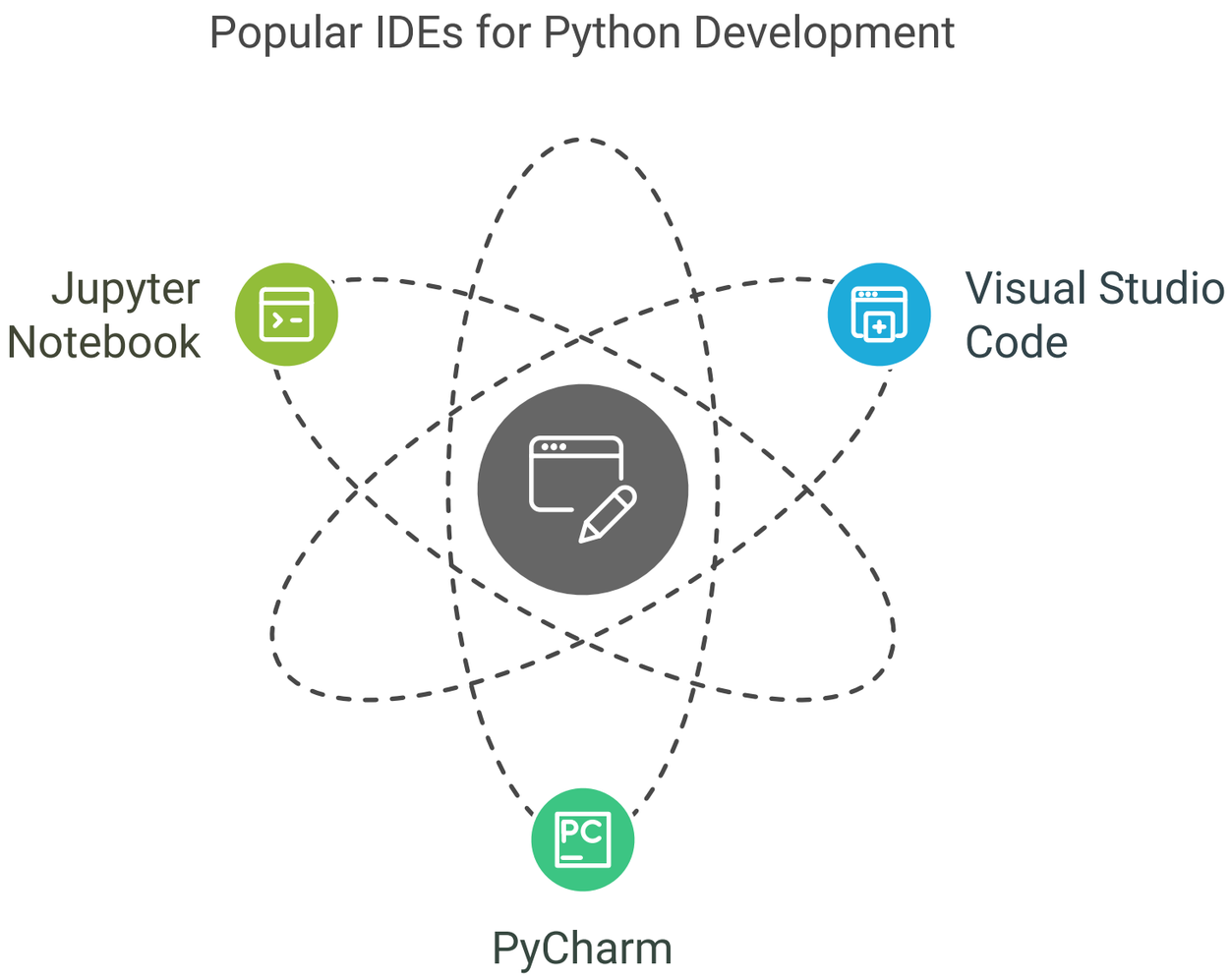
Introduction to tkinter to create simple interfaces.Practical exercise: creating a calculator or a task manager.Week 12: Final Project and Next Stepsearned:Example: an app that takes data from an API, processes it, and visualizes it.Review of concepts.



Additional resources to continue learning:Platforms like LeetCode, HackerRank, or Kaggle. Recommended books (e.g., "Automate the Boring Stuff with Python").Recommended tools and resources



IDE: Visual Studio Code, PyCharm, or Jupyter Notebook.



Learning platforms:Real PythonFreeCodeCampKaggleOfficial Python documentation: docs.python.org