



PYTHON ADVANCED Course (PEPs)

PEP courses refer to a series of courses that explore and explain proposed enhancements and changes to the Python programming language. These courses are aimed at improving and advancing Python and facilitating communication and collaboration within the Python community. They cover various topics, including coding conventions, standard library modules, development processes, and related guidelines.

Course Overview

You will learn about PEP 8 and how to implement the conventions for code comprising the standard library in the main Python distribution. You will also learn about the Zen of Python (PEP 20) and the principles that influence the design of Python code. Finally, you will become familiar with PEP 257 and the docstring conventions.

What you'll learn

After the course you will know how to implement the best coding practices and conventions using the Python language professionally. You will be able to write better and more effective code, and know how to avoid the most common errors and mistakes.

Course outline

- Python
- Computer Programming
- Algorithmic Thinking
- Analytical Thinking
- Design, Develop, Refactor, and Debug Multi-Module applications
- Best Practices and Python Coding Conventions
- Solutions Architecture
- Advanced Object-Oriented Programming
- GUI Programming
- Network Programming

Who should attend?

PEP courses are generally suitable for individuals who have a solid understanding of the Python programming language and are actively engaged in Python development or related fields. The following individuals may benefit from attending PEP courses:

 info@gandotech.com

 Lisbon, Sintra, Mirante St, No3, 2/C

 +351 911970800

 P.Code: 27 45-039



1. **Python Developers:** If you are a Python developer looking to deepen your understanding of the language and actively contribute to its development, attending PEP courses can be highly beneficial. It allows you to stay up-to-date with the latest enhancements, understand the rationale behind language decisions, and actively participate in shaping the future of Python.
2. **Open Source Contributors:** PEP courses can be valuable for individuals interested in contributing to open-source Python projects. Understanding the PEP process and guidelines enables you to propose and implement new features or improvements, fostering collaboration within the Python community.
3. **Technical Leads and Architects:** Technical leads and architects responsible for designing and implementing Python-based systems can benefit from PEP courses. They provide insights into upcoming changes in the language, best practices, and design principles, enabling you to make informed decisions and leverage new features effectively.
4. **Python Enthusiasts:** Even if you are not directly involved in Python development, attending PEP courses can enhance your knowledge of the language and its evolution. It allows you to stay informed about new features, understand the reasoning behind language design choices, and participate in discussions around Python enhancements.

Prerequisites

To make the most of PEP courses and fully understand the concepts and discussions, it is beneficial to have the following prerequisites:

1. **Python Programming:** A strong understanding of the Python programming language is essential. Familiarize yourself with the syntax, data types, control flow structures, functions, and modules. Practical experience in writing Python code will help you grasp the concepts discussed in PEPs.
2. **Object-Oriented Programming (OOP):** Many PEPs cover topics related to object-oriented programming in Python. Having a good understanding of OOP concepts such as classes, objects, inheritance, and polymorphism will be helpful in comprehending the content of PEPs.
3. **Python Development Experience:** Experience in developing Python applications or working on Python projects will provide you with practical context and insights. It helps in understanding the challenges and considerations addressed by various PEPs.
4. **Familiarity with Python Community and Ecosystem:** Being familiar with the Python community, forums, and development processes will provide you with a better understanding of the collaborative nature of PEPs. Engaging in discussions, reading mailing lists, and participating in open-source projects can enhance your learning experience.
5. **Reading Comprehension and Technical Writing Skills:** PEPs are technical documents, and they require a certain level of reading comprehension and technical writing skills. The ability to understand complex technical concepts and express your ideas effectively will greatly aid in understanding and contributing to PEP discussions.

🌐 info@gandotech.com

📍 Lisbon, Sintra, Mirante St, No3, 2/C

☎ +351 911970800

✉ P.Code: 27 45-039