Course duration

- 4 days

Course Benefits

- Leverage OS services
- Add enhancements to classes
- Code graphical interfaces for applications
- Understand advanced Python metaprogramming concepts
- Create easy-to-use and easy-to-maintain modules and packages
- Implementand run unit tests
- Create multithreaded and multi-process applications
- Interact with network services
- Design professional scripts
- Query databases

Course Outline

1. Introduction
2. Python Refresher
    1. Built-in data types
    2. Lists and tuples
    3. Dictionaries and sets
    4. Program structure
    5. Files and console I/O
    6. If statement
    7. for
    8. Built-in functions
    9. User-defined functions
    10. Modules and packages
    11. Basic OOP
3. OS Services
    1. The os and os.path modules
    2. Environment variables
    3. Launching external commands with subprocess
    4. Walking directory trees
    5. Paths, directories, and filenames
    6. Working with file systems
4. Dates and Times
    1. Basic date and time classes
    2. Different time formats

Phoca PDF

    3. Converting between formats
    4. Formatting dates and times
    5. Parsing date/time information
5. Binary Data
    1. What is Binary Data?
    2. Binary vs text
    3. Using the Struct module
6. Pythonic Programming
    1. The Zen of Python
    2. Tuples
    3. Advanced unpacking
    4. Sorting
    5. Lambda functions
    6. List comprehensions
    7. Generator expressions
    8. String formatting
7. Functions, Modules, and Packages
    1. Four types of function parameters
    2. Four levels of name scoping
    3. Single/multi-dispatch
    4. Relative imports
    5. Using __init__ effectively
    6. Documentation best practices
8. Enhancing Classes
    1. Class/static data and methods
    2. Inheritance (or composition)
    3. Abstract base classes
    4. Creating attributes with attr
    5. Implementing protocols (context, iterator, etc.)
9. Metaprogramming
    1. Implicit properties
    2. globals() and locals()
    3. Working with object attributes
    4. The inspect module
    5. Callable classes
    6. Decorators
    7. Monkey patching
10. Developer Tools
    1. Analyzing programs with pylint
    2. Using the debugger
    3. Profiling code
    4. Testing speed with benchmarking
11. Unit Testing with PyTest
    1. What is a unit test
    2. Creating test cases
    3. Writing and running tests
    4. Test harnesses

Phoca PDF

    5. Working with fixtures
12. Database Access
    1. The DB API
    2. Available Interfaces
    3. Connecting to a server
    4. Creating and executing a cursor
    5. Fetching data
    6. Parameterized statements
    7. Using Metadata
    8. Transaction control
    9. ORMs and NoSQL overview
13. PyQt
    1. Overview
    2. Qt Architecture
    3. Using designer
    4. Standard widgets
    5. Event handling
    6. Extras
14. Network Programming
    1. Built-in classes
    2. Using requests
    3. Grabbing web pages
    4. Sending email
    5. Working with binary data
    6. Consuming RESTful services
    7. Remote access (SSH)
15. Multiprogramming
    1. The threading module
    2. Sharing variables
    3. The queue module
    4. The multiprocessing module
    5. Creating pools
    6. About async programming
16. Scripting for System Administration
    1. Running external programs
    2. Parsing arguments
    3. Creating filters to read text files
    4. Logging
17. Serializing Data
    1. Working with XML
    2. XML modules in Python
    3. Getting started with ElementTree
    4. Parsing XML
    5. Updating an XML tree
    6. Creating a new document
    7. About JSON
    8. Reading JSON

Phoca PDF

    9. Writing JSON
    10. Reading/writing CSV files
    11. YAML, other formats as time permits
18. Advanced Data Handling [as time permits]
    1. Discover the collections module
    2. Use defaultdict, Counter, and namedtuple
    3. Create dataclasses
    4. Store data offline with pickle
19. Type Hinting [as time permits]
    1. Annotate variables
    2. Learn what type hinting does NOT do
    3. Use the typing module for detailed type hints
    4. Understand
    5. Write stub interfaces
20. Conclusion

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Python class:

- Introductory-level experience with Python.

Prerequisite Courses

Courses that can help you meet these prerequisites:

- Introduction to Python Training

Phoca PDF