



# Core and Advanced Python Programming

3<sup>rd</sup> Floor, Surviba Towers, Plot No.247/3RT, Near Umesh Chandra Statue, Sanjeeva Reddy Nagar,  
Hyderabad, Telangana. 500038 Email Us : [info@invictusengineers.com](mailto:info@invictusengineers.com) Phone: +91 99666 33097

[www.invictusengineers.com](http://www.invictusengineers.com)

# I. Core Python: 3 weeks

## Week 1

### 1. Introduction to Python Programming

- Introduction
- Variables and Assignment Statements
- Arithmetic
- Function print and an Intro to Single- and Double-Quoted Strings
- Triple-Quoted Strings
- Getting Input from the User
- Decision Making: The if Statement and Comparison Operators
- Objects and Dynamic Typing
- Intro to Data Science: Basic Descriptive Statistics

### 2. Control Statements and Program Development

- Introduction
- Algorithms
- Pseudocode
- Control Statements
- if Statement
- if...else and if...elif...else Statements
- while Statement
- for Statement: Iterables, Lists and Iterators & Built-In range Function
- Augmented Assignments
- Program Development: Sequence-Controlled Repetition: Requirements Statement, Pseudocode for the Algorithm, Coding the Algorithm in Python & Introduction to Formatted Strings
- Program Development: Sentinel-Controlled Repetition
- Program Development: Nested Control Statements
- Built-In Function range: A Deeper Look
- Using Type Decimal for Monetary Amounts
- break and continue Statements
- Boolean Operators and, or and not
- Intro to Data Science: Measures of Central Tendency—Mean, Median and Mode

### 3. Functions

- Introduction
- Defining Functions
- Functions with Multiple Parameters
- Random-Number Generation
- Case Study: A Game of Chance
- Python Standard Library
- math Module Functions
- Using IPython Tab Completion for Discovery
- Default Parameter Values
- Keyword Arguments
- Arbitrary Argument Lists
- Methods: Functions That Belong to Objects
- Scope Rules
- import
- Passing Arguments to Functions
- Function-Call Stack
- Functional-Style Programming
- Intro to Data Science: Measures of Dispersion

### 4. Sequences: Lists and Tuples

- Introduction
- Lists
- Tuples
- Unpacking Sequences
- Sequence Slicing
- del Statement
- Passing Lists to Functions
- Sorting Lists
- Searching Sequences
- Other List Methods
- Simulating Stacks with Lists

- List Comprehensions
- Generator Expressions
- Filter, Map and Reduce
- Other Sequence Processing Functions
- Two-Dimensional Lists
- Intro to Data Science: Simulation and Static Visualizations

## **Week 2**

### **5. Dictionaries and Sets**

- Introduction
- Dictionaries: Creating a Dictionary, Iterating through a Dictionary, Basic Dictionary Operations, Dictionary Methods keys and values, Dictionary Comparisons, Dictionary Method update, & Dictionary Comprehensions
- Sets: Comparing Sets, Mathematical Set Operations, Mutable Set Operators and Methods, & Set Comprehensions
- Intro to Data Science: Dynamic Visualizations

### **6. Array-Oriented Programming with NumPy**

- Introduction
- Creating arrays from Existing Data
- array Attributes
- Filling arrays with Specific Values
- Creating arrays from Ranges
- List vs. array Performance: Introducing %timeit
- array Operators
- NumPy Calculation Methods
- Universal Functions
- Indexing and Slicing
- Views: Shallow Copies
- Deep Copies
- Reshaping and Transposing
- Intro to Data Science: pandas Series and DataFrames

## **7. Strings: A Deeper Look**

- Introduction
- Formatting Strings: Presentation Types, Field Widths and Alignment, Numeric Formatting, & String's format Method
- Concatenating and Repeating Strings
- Stripping Whitespace from Strings
- Changing Character Case
- Comparison Operators for Strings
- Searching for Substrings
- Replacing Substrings
- Splitting and Joining Strings
- Characters and Character-Testing Methods
- Raw Strings
- Introduction to Regular Expressions: re Module and Function fullmatch, Replacing Substrings and Splitting Strings, & Other Search Functions; Accessing Matches
- Intro to Data Science: Pandas, Regular Expressions and Data Munging

## **8. Files and Exceptions**

- Introduction
- Files
- Text-File Processing: Writing to a Text File, Reading Data from a Text File
- Updating Text Files
- Serialization with JSON
- Focus on Security: pickle Serialization and Deserialization
- Additional Notes Regarding Files
- Handling Exceptions: Division by Zero and Invalid Input, try Statements, Catching Multiple Exceptions in One except Clause, What Exceptions Does a Function or Method Raise?, & What Code Should Be Placed in a try Suite?
- finally Clause
- Explicitly Raising an Exception
- (Optional) Stack Unwinding and Tracebacks
- Intro to Data Science: Working with CSV Files

### Week 3

#### 9. Object-Oriented Programming

- Introduction
- A Custom Class: Composition: Object References as Members of Classes
- Controlling Access to Attributes
- Properties for Data Access
- Simulating “Private” Attributes
- Case Study: Card Shuffling and Dealing Simulation
- Inheritance: Base Classes and Subclasses
- Building an Inheritance Hierarchy; Introducing Polymorphism
- Duck Typing and Polymorphism
- Operator Overloading
- Exception Class Hierarchy and Custom Exceptions
- Named Tuples
- A Brief Intro to Python 3.7’s New Data Classes: Creating a Data Class, Using the Data Class, Data Class Advantages over Named Tuples, & Data Class Advantages over Traditional Classes
- Unit Testing with Docstrings and doctest
- Namespaces and Scopes
- Intro to Data Science: Time Series and Simple Linear Regression

## II. Advanced Python: 3 weeks (Project development centric)

### Week 4

#### 1. Starting Your Project

- Laying Out Your Project
- Version Numbering
- Coding Style and Automated Checks

#### 2. Modules, Libraries, and Frameworks

- The Import System
- Useful Standard Libraries
- External Libraries
- Package Installation: Getting More from pip
- Using and Choosing Frameworks

### **3. Documentation and Good API Practice**

- Documenting with Sphinx

### **4. Handling Timestamps and Time Zones**

- The Problem of Missing Time Zones
- Building Default datetime Objects
- Time Zone–Aware Timestamps with dateutil
- Serializing Time Zone–Aware datetime Objects
- Solving Ambiguous Times

## **Week 5**

### **5. Distributing Your Software**

- The setup.py History
- Packaging with setup.cfg
- The Wheel Format Distribution Standard
- Sharing Your Work with the World
- Entry Points

### **6. Unit Testing**

- The Basics of Testing
- Virtual Environments
- Testing Policy

### **7. Methods and Decorators**

- Decorators and When to Use Them
- How Methods Work in Python
- Static Methods
- Class Methods
- Abstract Methods
- Mixing Static, Class, and Abstract Methods

## Week 6

### 8. Functional Programming

- Creating Pure Functions
- Generators
- List Comprehensions
- Functional Functions Functioning

### 9. The Abstract Syntax Tree, Hy, and Lisp-like Attributes

- Looking at the AST
- Extending flake8 with AST Checks
- Introduction to Hy

### 10. Performances and Optimizations

- Data Structures
- Understanding Behavior Through Profiling
- Defining Functions Efficiently
- Ordered Lists and bisect
- namedtuple and Slots
- Memoization
- Faster Python with PyPy
- Achieving Zero Copy with the Buffer Protocol

### 11. Scaling and Architecture

- Multithreading in Python and Its Limitations
- Multiprocessing vs. Multithreading
- Event-Driven Architecture
- Other Options and asyncio
- Service-Oriented Architecture
- Interprocess Communication with ZeroMQ

### 12. Managing Relational Databases

- RDBMSs, ORMs, and When to Use Them
- Database Backends
- Streaming Data with Flask and PostgreSQL