

Core Python Programming (version 3):

1 Getting started:

- An explanation of the Python programming language
- Installing Python
- Python interpreter
- Editor vs IDE
- First Python Program (Hello World)
- Linting, formatting and running Python code
- Execution

2 Primitives:

- Variables and name variables
- Strings
- Escape sequences
- Formatted Strings
- String methods
- Numbers
- Type conversions

3 Control flow:

- Comparison operators
- Conditional Statements
- Ternary operators
- Logical operators
- Short circuit evaluations
- Chaining operators
- For loops and For-else
- Nested loops
- Iterables
- While loops
- Infinite loops and break statements

4 Functions:

- Defining functions
- Arguments
- Types of functions

- Keyword arguments
- Default arguments
- Extended arguments (*args and **kwargs)
- Scope

5 Data structures:

- Lists
- Accessing, unpacking, looping, adding and removing, finding and sorting items in a list
- Lambda functions (anonymous functions)
- Map functions
- Filter functions
- List comprehensions
- Zip functions
- Stacks and queues
- Tuples
- Arrays
- Sets
- Dictionaries and dictionary comprehensions
- Generator expressions
- Unpacking operator

6 Exceptions:

- An explanation of exceptions and their need
- Handling exceptions
- The with statement
- Exceptions with external resources
- Raising exceptions and the cost of raising exceptions

7 Parsing data serialization formats into Python structures:

- Parsing XML with Python
- XML libraries
- Parsing JSON with Python
- Parsing YAML with Python

8 Object oriented programming (OOP):

- Introduction to OOP

- Classes –general syntax and structure
- Instances and instance attributes
- Understanding instantiation using self
- Creating instances
- Default arguments
- Accessing and modifying instance attributes
- Class attributes
- Class attributes vs instance attributes
- Defining class attributes
- Accessing and modifying class attributes
- Encapsulation and abstraction
- Public and private attributes
- Inheritance
- Introduction to inheritance methods
- Polymorphism
- Decorators
- Methods
- Objects in memory
- Aliasing, mutation and cloning
- Docstrings –how to document your code
- Special methods

9 Modules:

- Creating modules
- Compiled Python files
- Module search path
- Packages
- Sub-packages
- Intra-package references
- The dir function
- Executing modules as scripts

10 Python standard library:

- Introduction
- Working with paths
- Working with directories

- Working with files (zip, CSV, JSON)

Optional:

- REST API fundamentals, requests and response codes
- Basic Unix commands and fundamentals for Git Control
- Git Control and collaborating with Git
- Automating networks with Python scripts