

## Contents

1. About Python
  - 1.1 Python Overview
  - 1.2 History of Python
  - 1.3 Features of Python
  - 1.4 Python Environment
  - 1.5 Getting Python
  - 1.6 Case studies
  - Exercise
2. Context of Software Developments
  - 2.1 Software
  - 2.2 Development Tools
3. Installing Python
  - 3.1 Unix & Linux Installation
  - 3.2 Windows Installation
  - 3.3 Macintosh Installation
  - 3.4 Setting up a path
    - 3.4.1 Setting up a path at Unix/Linux
    - 3.4.2 Setting up a path at Windows
  - 3.5 Python Environment Variables
4. Running Python Programs
  - 4.1 Interactive Interpreter
  - 4.2 Script from the command line
  - 4.3 Integrated development Environment(IDE)
5. Python Basic Syntax
  - 5.1 First Python Program
  - 5.2 Python Identifiers
  - 5.3 Reserved words
  - 5.4 Lines and Indentation
  - 5.5 Multi-Line Statements
  - 5.6 Quotation in python
  - 5.7 Comments in python
  - 5.8 Using blank Lines
  - 5.9 Waiting for the users
  - 5.10 Multiple statements on a single line
  - 5.11 Multiple statement groups as suites
  - 5.12 Command line arguments
    - 5.12.1 Accessing command line arguments
    - 5.12.2 Parsing command line arguments
6. Python data types, values and Identifiers
  - 6.1 Assigning values to variables
  - 6.2 Multiple assignment
  - 6.3 Naming rules
  - 6.4 Understanding reference semantics
  - 6.5 Python's standard data types
    - 6.5.1 Python Integer(Number)
    - 6.5.2 Sequence types(Lists, Tuples and Strings)
    - 6.5.3 Operations on sequence types
    - 6.5.4 Lists and Tuples (Mutability vs immutability)
    - 6.5.5 Dictionary
  - 6.6 Data type conversion
  - 6.7 Exercises
7. Statements, Expressions and operators
  - 7.1 Statements and Expressions
  - 7.2 What are operators and operands
  - 7.3 Arithmetic operators

- 7.4 Relational operators
- 7.5 Assignment operators
- 7.6 Bitwise operators
- 7.7 Logical operators
- 7.8 Membership operators
- 7.9 Identity operators
- 7.10 Precedence and associativity of python operators
- 7.11 Exercises
- 8. Python's flow control tools
  - 8.1 if statements
  - 8.2 if-else statements
  - 8.3 if..elif..else statements
  - 8.4 Python's for loop
    - 8.4.1 The range( ) method
    - 8.4.2 for / else statement
  - 8.5 Python's while loop
    - 8.5.1 while / else statement
    - 8.5.2 Infinite loops
  - 8.6 Break and Continue statements
    - 8.6.1 The break statement
    - 8.6.2 The continue statement
  - 8.7 Python's pass statement
  - 8.8 Exercises
- 9. Functions in python
  - 9.1 Standard mathematical functions
  - 9.2 Time functions
  - 9.3 Random functions
  - 9.4 Why write your own functions
  - 9.5 Functions basics
    - 9.5.1 Docstring
    - 9.5.2 The return statement
    - 9.5.3 Flow of control at function call and return
    - 9.5.4 The scope and lifetime of variables
    - 9.5.5 Arguments
    - 9.5.6 Recursive functions
    - 9.5.7 Lambda functions
    - 9.5.8 Global, local and non local variables
  - 9.6 Exercises
- 10. Python Modules
  - 10.1 Packages
- 11. Python classes and objects
  - 11.1 Introduction to object oriented programming
    - 11.1.1 Classes
    - 11.1.2 Class objects
    - 11.1.3 Instance objects
    - 11.1.4 Method object
    - 11.1.5 Class and instance variables
    - 11.1.6 Inheritance
    - 11.1.7 Multiple inheritance
    - 11.1.8 Private variables
    - 11.1.9 Encapsulation
    - 11.1.10 Polymorphism and method overriding
  - 11.2 The object Class
  - 11.3 Operator overloading in python
    - 11.3.1 Special functions in python
    - 11.3.2 Overloading the '+' operator in python
    - 11.3.3 Overloading comparison operators in python

- 11.4 Exercises
- 12. Errors And Exceptions
  - 12.1 What Are Exceptions?
  - 12.2 Exceptions in Python
  - 12.3 Detecting and Handling Exceptions
  - 12.4 \*Exceptions as Strings
  - 12.5 \*Exceptions as Classes
  - 12.5 Raising Exceptions
  - 12.6 Assertions
  - 12.7 Standard Exceptions
  - 12.8 \*Creating Exceptions
  - 12.9 Why Exceptions (Now)?
  - 12.10 Why Exceptions at All?
  - 12.11 Exceptions and the sys Module
  - 12.12 Case Studies
  - 12.13 Exercises
- 13. Files and Input/Output
  - 13.1 File Objects
  - 13.2 File Built-in Function [ open() ]
  - 13.3 File Built-in Methods
  - 13.4 File Built-in Attributes
  - 13.5 Standard Files
  - 13.6 Command-line Arguments
  - 13.7 File System
  - 13.8 File Execution
  - 13.9 Persistent Storage Modules
  - 13.10 Related Modules
  - 13.11 Case Studies
  - 13.12 Exercises
- 14. Introduction to machine learning
  - 14.1 How does a machine learning model work
  - 14.2 Data exploration
  - 14.3 Creating a model
  - 14.4 testing a model
  - 14.5 underfitting and overfitting
- 15. Data visualization
  - 15.1 Introduction to seaborn
  - 15.2 Line charts
  - 15.3 heatmaps and Bar charts
  - 15.4 scatter plots
  - 15.5 Distributions
  - 15.6 styling a chart
- 16. Intermediate machine learning
  - 16.1 Handling Missing values
  - 16.2 Using categorical variables
  - 16.3 Using Pipelines
  - 16.4 Applying cross validation
  - 16.5 More accurate algorithms
  - 16.6 Handling data leakage
- 17. Case study: Solving a real world problem