PYTHON v3.7

Core Python

Setting up and running Python

- Distributions – python.org, anaconda python
- Shells – python, Jupyter,
- IDES – Pycharm, spyder, eclipse
- Editors - Visual Studio Code, Atom
- Python 2 vs 3
- First program - ‘Hello World!’
- Interpretation and .pyc, .pyo files
- Python Implementations – CPython, Ironpython, Jython, pypy

Introduction

- Values and variables
- Python data types
- type(), id(), sys.getsizeof()
- Python labeling system
- Object pooling
- Conversion functions
- The language which knew infinity
- Console input, output
- Operators in python
  - Arithmetic operators
  - Relational operators
  - Logical operators
  - Assignment operators
  - Bitwise operators
  - Membership operators
  - Identity operators
  - Ternary Operator
- Strings
  - Define a string - Multiple quotes and Multiple lines
  - String functions
  - String slicing - start, end & step
  - Negative indexing
- Built-in functions
- Type conversions
  - int()
  - float()
  - bool()
  - str()
  - complex()
- Interview questions
- Exercise Programs
- Summary

Control structures:

- if statement
- if - else statement
- if - elif statement
- Nest if-else
- Multiple if
- Which control structure to choose?
- Looping statements
  - while loop
  - for loop
    - range()
    - xrange()
    - Iterator and generator Introduction
  - for - else
  - When to use for-else ?

Interview Questions

- Exercise Programs
- Summary

Data structures

- List
  - Introduction to List
  - Purpose of a List
  - Iterating through a List
  - List slicing, -ve indexing
  - Internals of list
  - List Operations
  - Searching for an element
    - In and count()
  - Adding an element
    - append()
    - insert()
  - Removing an element
    - remove()
    - pop()
  - Merging two lists
    - + operator
    - extend()

  - Ordering a list
    - sort()
    - reverse()
  - Finding index of an element - index()
  - List of lists
  - Comparing lists

- Homogeneous data
  - Built-in array.array()
  - numpy.array()
- Tuple
  - Introduction of Tuple
  - Tuple Slicing
  - -ve indexing
  - Iterating through a Tuple
  - List of tuples Vs Tuple of Lists
  - Purpose of a tuple

- List Vs Tuple - An interviewer's question
- Set
  - Introduction of set
  - How to remove duplicates in list?
  - How set removes duplicates?
  - Set functions
    - Searching for an element
      - In - The fastest
    - Adding an element
      - add()
    - Removing an element
      - remove()
      - discard()
      - pop()
  - Relation between two sets
    - intersection()
    - union()
    - difference()
    - isdisjoint()
    - issubset()
    - issuperset()
  - Merging two sets
    - update()

- Sets are hashable but Lists or unhashable
- Set Use-Cases
- Dictionary
  - Introduction of Dictionary - Associative data structure
  - Creating a Dictionary
  - Adding elements to Dictionary
  - Deleting key value pair
  - Updating / extending a Dictionary
  - Iterating through a Dictionary
  - Tuple unpacking method
  - Converting list/tuples of tuples/lists into Dictionary
  - Converting Dictionary to List of tuples
  - Lambda introduction
  - Sorting List of tuples and dictionaries
  - Finding max(), min() in a dict
  - Wherever you go, dictionary follows you!
- Counter() - simplest counting algorithm
- DefaultDict - Always has a value
- OrderedDict - Maintains order
- Dequeue - Short time memory loss
- Forzenset() – hashable set
- namedtuple() – hashable dict
- Heapq - efficient in-memory min-heap()
  - heapify()
  - nlargest()
  - nsmallest()
  - heappush()
  - heappop()

- Importance of Hashability
- Packing and Unpacking
  - Swapping two values
  - List packing and Unpacking
  - Tuple packing and Unpacking
  - String packing and Unpacking
  - Set packing and Unpacking
  - Iterator using iter() and next()

Functions

- Purpose of a function
- Defining a function
- Calling a function
- Function parameter passing
  - Formal arguments
  - Actual arguments
  - Positional arguments
  - Keyword arguments
  - Variable arguments
  - Variable keyword arguments
  - Use-Case *args, **kwargs
- Function call stack
  - locals()
  - globals()
  - Stackframe
- Call-by-object-reference
  - Shallow copy - copy.copy()
  - Deep copy - copy.deepcopy()

Decorators and Generators

- Passing one function to another function
- Defining one function within another function
- Returning a function from another function
- Passing a function to another function along with its arguments
- Call-back functions and delegation
- Decorators
  - Creating decorators
  - Multiple decorators
  - Use case - TimeIt
- Generator
  - Creating custom generators
  - Use Case - lazy evaluation

Modules

- Python Code Files
- Importing functions from another file
- __name__: Preventing unwanted code execution
- Importing from a folder
- Folders Vs Packages
- __init__.py
- Namespace
- __all__
- Import *
- Private global variables and functions
- __butiltins__
- Recursive imports
- Use Case: Project Structure

Comprehensions

- List comprehension
- Tuple comprehension
- Set comprehension
- Dictionary comprehension
- enumerate
- Zip and unzip

Functional programming

- Procedural vs Functional
- Pure functions
- Map()
- Reduce()
- Filter()
- Lambdas
- Loop vs Comprehension vs Map

File - IO

- Creating file
- File reading
- File writing
- File modes
- Line by line file reading
- Writing multiple lines
- seek()
- tell()
- Binary files
  - Pickling
- Use Case - Cleaning text

Pet Project – Students and Faculties

Advanced Python

Object Orientation

- Purpose of Object Orientation
- Design starts with Data Binding
- Abstraction - What the world sees
- Data hiding - What is hidden
- Encapsulation - Boundary between Abstraction and Data hiding
- Class - Classification of type
- Creating a class type
- Creating multiple instances of a functionality
- Object - The physical existence of a class
- __init__() - the initializer
- Data members
- Member functions(methods)
- Method invocation
- Printing objects
- __str__()
- __repr__()
- Inheritance
    - Use Case - 4 wheeler
    - Types of inheritance
    - Diamond problem
    - MRO
- Private members
- Creating inline objects, classes, types
- Class method
- Static method and static variables
- Function Objects(Functor) - Callable Objects
- Class as decorator and Context manager
- Polymorphism - Incorporating changes
- Operator Overloading
    - __lt__()
    - __add__()
    - __hash__()
    - __eq__()

- Function Overloading in python
- Sorting objects
- Hashing objects

Exception Handling

- Purpose of Exception Handling
- try block
- except block
- Else block
- finally block
- Built-in exceptions
- Order of ‘except’ statements
- Exception - mother of all exceptions
- Writing Custom exceptions
- Stack Unwinding
- Use Case - finally
- Interview Questions
- Summary

Descriptors

- Abstract
- Definition and Introduction
- Descriptor Protocol
- Invoking Descriptors
- Descriptor Example
- Properties
- Functions and Methods
- Static Methods and Class Methods

Multi-Threading

- Program Memory Layout
- Concurrency
- Parallelism
- Process
- Thread of execution
- Creating a thread
- Joining a thread
- Critical section
- Lock and Conditional variable
- Wait, notify, notify all
- How much concurrency is required?
- GIL
- Multiprocessing
- Python on JVM - jython and threading
- Producer - consumer: 1 dad - 2 Sons
- Lock-free Programming Intro
- Interview Questions
- Exercise Programs
- Summary

Database connections

- Database introduction
- **MYSQL** database connection setup
- Installing connector
- Cursor
- Running a query
- Iterating a cursor
- Fetching data
- Closing a connection
- **Mongo DB** setup
- Creating collections
- CRUD operations
- Interview Questions
- Exercise Programs

Regular Expressions

- Functions
  - re.match()
    - start()
    - end()
    - group()
  - re.search()
  - re.findall()
  - Regex symbols
- Greedy and non-greedy
- Interview Questions
- Exercise Programs
- Summary

Useful modules

- datetime
- time
- pytz
- sys
- os
- random

Serialization pickling, XML & JSON

- Introduction to Serialization
- Structure and Container
- Pickle Module
- pickling built-in data structures
  - byte strings
  - binary
- xml parsing and construction - xml,
- json parsing and construction - json, simplejson
- Interview Questions
- Exercise Programs
- Summary

Unit testing

- Purpose of Unit testing
- Unittest2 module
- Test case
- Test Suit
- assert()
- nosetests module
- Coverage module - Code coverage
- Mocking – faking
- Profiling
- Interview Questions
- Exercise Programs
- Summary

Logging

- Purpose of logging
- Logging levels
- Types of logging
- Logging format
- Logging Handlers
- Use-Case- Rotating File Logger
- Disadvantages of excessive logging
- Custom loggers
- Interview Questions
- Exercise Programs
- Summary

ORM Object relational mapping (optional)

- Purpose
- Creating engine
- Create a schema
- Declare mapping
- Connecting
- Create session
- Adding and Updating records
- Rolling back
- Building a relationship
- Querying
- Deleting

Networking (optional)

- TCP/IP Basics
- 3 way and 4 way Handshake
- Socket programming
- Simple TCP Client – Server
- Simple UDP Client – Server
- Emailing - smtplib
- FTP

Pet Project – Students & Faculties with OOPS

Data analysis

Numpy

- Numpy arrays
  - Double dimension arrays
    - Resizing, reshaping
    - Vector multiplication
    - Boolean filtering
    - Querying using where() function
    - Indexing
    - Slicing
    - Mean, median, standard deviation, average
    - Transpose
    - broadcasting
- Numpy matrix
  - Addition, multiplication
  - Transpose, inverse
- Numpy random module

Pandas

l. Series

Constructing from dictionaries

Custom index

Data filtering

m. Data Frames

Constructing from a dictionary with values as lists

Custom indexing

Rearranging the columns

Accessing values loc(), iloc(), at()&iat()

Setting values

Sum

umulative sum

Assigning a column to the data frame

Adding a new column

Deleting a column

Slicing

Indexing and Advanced indexing

Boolean indexing

Transposing

Sort by

Concatenate

Merge

1. Inner join
2. Outer join
3. Left outer join
4. Right outer join
5. Merge on columns

Join

Group By- Aggregation

Data Munging

6. Working with missing data

Reading Data from CSV, Excel, JSON

Writing Data to CSV, Excel, JSON, HTML

Reading data from database and storing in data frame

Writing data frame to database

Handling PDF files - tabula-py

Matplotlib

b. Basic Plotting
   i. Colors
   ii. Styles
   iii. Seaborn themes

c. labels

d. Title

e. Legend

f. Axis

g. Bar chart

h. Histogram

i. Scatter Plot

j. Box Plot

k. Pie Chart

Interview Questions

Exercise Programs

Summary

Pet Project – 911 Data Analysis & Visualization

Django with REST Webservices

- Client Server architecture
- MVC and MVT
- Web Application
- Website
- Web server – Apache, nginx, node
- Web client
- Web framework
- Introduction to Ubuntu Linux
- Installaing mysql and virtualenv
- Creating virtual environments for python3
- Installing Django modules
- Creating first basic Django Project – Web-news Use-Case
- Process of Migrations
- Request response cycle
- Understanding ORM
- Creating Model classes
- Admin Interface
- Adding Data through Admin Interface
- Templating and rendering
- Creating Django Apps
- URL Routing
- Django Template tags and template programming
- Django rest framework
- Understanding REST Architecture
- HTTP GET, PUT, POST, DELETE and UPDATE
- Serializer classes
- JSON serialization
- Writing REST API
- Testing with Postman
- Nginx Setup & AWS Deployment walkthrough
- Working with domain names setting public IP walkthrough
- Real-Time Project Explanation
- Summary