

Python

Core Python

Setting up and running Python

Python Implementations – CPython, Iron

Python, Jython, pypy

Distributions – python.org, anaconda python,

pypy

Shells and IDEs – python, Ipython, Jupyter,

Eclipse, Pycharm

Python 2 vs 3

Hello World!

No braces only indentation

Interpretation and .pyc, .pyo files

Types and conversion functions

Int()

float ()

bool()

str()

id ()

type ()

None

sys. getsizeof ()

Operators

arithmetic

relational

logical

assignment

bitwise

membership

identity operators

Control Structures

Conditionals

If

If-else

If-elif

Looping statements

while

for

range () and xrange ()

Break and Continue

Example Problems

Container Types

tuple

Strings

Array or List in Python

List slicing techniques

Collections

- set ()
- dict ()
- Defaultdict()
- OrderedDict ()

Counter ()

heapq ()

deque ()

Exercise Problems

Functions

Positional arguments

Variable Arguments

Keyword arguments

Default Arguments

Call-by-object-reference (call-by-reference
and call-by-value)

passkeyword

Built-in methods

Iterators and Generators

range () and xrange () re-visited

next ()vs.next ()

yield and Custom Generators

custom iterators `__iter__`, `__next__`

Generator expressions

Miscellaneous

zip()

enumerate()

Unpacking in python 3

Copy(), deepcopy()

Modules and Namespaces

import statement

`if __name__ == '__main__'`

python files

Package vs Folder

`__init__.py`and `__all__`

locals ()and globals ()

File Handling

Files and directories

Text files

Binary files

Encoding – UTF-8 & ASCII

Byte order (big-endian, little-endian)

Text file processing

Pickling - Binary file handling

Advanced Python

Functional programming

- Lambdas
- map (), filter () and reduce ()
- List comprehension revisited

Object Orientation

- Abstraction
- Encapsulation
- Data binding
- Data hiding

Classes and POD

Inheritance

Re-Usability

MRO

Single Inheritance

Multiple Inheritance

Multi-level Inheritance

Hierarchical Inheritance

Hybrid Inheritance

Diamond Problem

Polymorphism I: overloading

Method overloading in python

Operator overloading

Polymorphism II: Overriding

Static Methods

ClassMethods

Self & Cls

Decorators using functions

local functions

passing and returning functions

function decorators

Decorators using Classes

Context managers

Callable objects (functors) `__call__`

Descriptors and Properties

`__get__`, `__set__`, `__del__`

@property decoration

property function

Exception handling

System Continuity

try -except

finally
else
writing custom exceptions

Useful modules

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

Regular Expressions

- match ()
- search ()
- findall()
- Greedy and Non-Greedy Matching

Serialization

- pickle module
- CSV and EXCEL – csv module
- XML parsing – lxml module
- JSON Parsing –json and simplejson modules
- Universal conversion - odo module

Multi-threading

Concurrency and Parallelism

Multi-processing

Multi-threading

Creating thread

Joining thread

Daemon threads

Cancelling threads

Critical section and locking

deadlock

GIL

Asynchronous Programming

Database connectivity

Connectors – MySQL, SQLite, Mongo

dB&PostgreSQL

Querying and cursors

Cursor unpacking

Logging

Logging setup

Log Levels

Console Logger

File Logger



Rotating File Logger

Data analysis

Numpy Intro

Arrays, Datatypes

Matrices

Broadcasting

Pandas Intro

series

Data Frames

Merge, Join, Concatenate

Handling missing data

IO Tools, Text, CSV, HDF5, JSON, EXCEL

plotting

Matplotlib Intro

Line plot

Bar charts

Histograms

Scatter Plot

Unit testing(optional)

Writing unit tests

Assertions

nosetests

mocking

coverage

ORM Object relational mapping(optional)

SQLAlchemy

Creating, dropping tables

Updating records

Querying

Networking(optional)

TCP/IP Basics

Socket programming

Simple TCP Client – Server

Simple UDP Client – Server

Emailing

FTP

Web frameworks

▪ Django Intro

▪ Flask Intro