1. Intro to Python

* Setup (windows + Linux)
* Virtual environments + virtual environment wrapper
* Language fundamentals (operator, tokens, keywords, etc.)
* Functions
* Conditional statements
* Loops
* Primitive data types and internal functions

1. Data structures and comprehensions

* List
* Queue
* Tuples + named tuple
* Dictionaries + Ordered dictionaries
* Sets + frozen sets

1. Internal useful libraries

* Functions
* OS
* Sys
* Collection
* Itercools
* Functools
* Lru cache
* Str, math (internal function)

1. Oops and exceptions

* Class and objects
* Methods
* Abstraction
* Inheritance (pending - using private method + variable of parent class)
* Polymorphism
* Static methods and class methods
* Operator and method Overloading and method overriding
* Exception handling + User defined exceptions
* Lambda functions

1. Advanced concepts

* Comprehension (list, tuple, dictionary, set)
* Decorators (Built in - @property, @staticmethod, @classmethod
* Generators, iterators and iterables
* File handling
* Duck typing
* Monkey patching
* Deep copy and sallow copy
* Object slicing
* Regular expressions
* Logging
* Linting
* Debugging
* Profiling

1. Concurrency and parallelism

* GIL
* Multithreading
* Multiprocessing
* Mutex
* Jupyter notebook

1. Python communication

* Basic python Redis and python memcache usage
* Python postgressql connection and usage
* Python mongo dB connection and usage

1. Web programming (Introductory level)

* Monolithic vs micro service.
* Rest APIs and creating some projects in most famous frameworks.
* Django + Django rest framework.
* Twisted Klein
* Flask
* Swagger
* Binding back ends with Databases (Any 1)
* Binding rest APIs with front end (Angular 4) at very intro level.