Python

Backend

Development

Training Cirriculum

GEEKONIK

www.geenonik.com | ☎ 9990161025

1)LANGUAGE FUNDAMENTALS

Introduction

Features of Python

- Simple and Easy to Learn
- Freeware and Open Source
- High Level Programming Language
- Platform Independent
- Portability
- Dynamically Typed
- Both Procedure Oriented and Object Oriented
- Interpreted
- Extensible
- Embedded
- Extensive Library

Limitations of Python

Flavors of Python

- CPython
- Jython OR JPython
- IronPython
- PyPy
- RubyPython
- AnacondaPython

Python Versions

Identifiers

Reserved Words

Data Type

- int Data Type
  - Decimal Form
  - Binary Form
  - Octal Form
  - Hexa Decimal Form
- Float Data Type
- Complex Data Type
- bool Data Type
- str Data Type
- bytes Data Type
- bytearray Data Type
- List Data Type
- Tuple Data Type
- Range Data Type
- Set Data Type
- frozenset Data Type
- dict Data Type
- None Data Type

Base Conversions

Slicing of Strings

Type Casting

- int()
- float()
- complex()
- bool()
- str()

Fundamental Data Types vs Immutability

Escape Characters

Constants

2) OPERATORS

- Arithmetic Operators
- Relatio nal Operators OR Comparison Operators
- Equality Operators
- Logical Operators
- Assignment operators
- Ternary Operator OR Conditional Operator
- Special operators
  - Identity Operators
  - Membership operators
- Operator Precedence
- Mathematical Functions (math Module)
- Command Line Arguments
- Output Statements

3) FLOW CONTROL

Conditional Statements

- if
- if-elif
- if-elif-else

Iterative Statements

- for
- while

Transfer Statements

- break
- continue
- pass

Loops with else Block

del Statement

Difference between del and None

4)STRING DATA TYPE

- What is String?
- How to define multi-line String Literals?
- How to Access Characters of a String?
  - Accessing Characters By using Index
  - Accessing Characters by using Slice Operator
- Behaviour of Slice Operator
- Slice Operator Case Study
- Mathematical Operators for String
- len() in-built Function
- Checking Membership
- Comparison of Strings
- Removing Spaces from the String
  - rstrip()
  - lstrip()
  - strip()
- Finding Substrings
- Counting substring in the given String
- Replacing a String with another String
- Splitting of Strings
- Joining of Strings
- Changing Case of a String
- Checking Starting and Ending
  - s.startswith(substring)
  - s.endswith(substring)
- To Check Type of Characters
- Formatting the Strings
- Important Programs regarding String Concept

5) LIST DATA STRUCTURE

- Creation of List Objects
- Accessing Elements of List
    - By using Index
    - By using Slice Operator
- List vs Mutability
- Traversing the Elements of List
    - By using while Loop
    - By using for Loop
    - To display only Even Numbers
    - To display Elements by Index wise
- Important Functions of List
- To get Information about List
    - len()
    - count()
    - index()
- Manipulating Elements of List
    - append()
    - insert()
    - extend()
    - remove()
    - pop()
- Ordering Elements of List
    - reverse()
    - sort()
- Using Mathematical Operators for List Objects
    - Concatenation Operator (+)
    - Repetition Operator (*)
- Comparing List Objects
- Membership Operators
    - in Operator
    - not in Operator

- clear() Function
- Nested Lists
- Nested List as Matrix
- List Comprehensions

6)TUPLE DATA STRUCTURE

- Tuple Creation
- Accessing Elements of Tuple
    - By using Index
    - By using Slice Operator
- Tuple vs Immutability
- Mathematical Operators for Tuple
    - Concatenation Operator (+)
    - Multiplication Operator OR Repetition Operator (*)
- Important Functions of Tuple
    - len()
    - count()
    - index()
    - sorted()
- min() And max()
- cmp()
- Tuple Packing and Unpacking
- Tuple Comprehension
- Differences between List and Tuple

7)SET DATA STRUCTURE

- Creation of Set Objects
- Important Functions of Set
    - add(x)
    - update(x,y,z)
    - copy()
    - pop()
    - remove(x)
    - discard(x)
    - clear()
- Mathematical Operations on the Set
    - union()
    - intersection()
    - difference()
    - symmetric_difference()
- Membership Operators: (in, not in)
    - Set Comprehension

8)DICTIONARY DATA STRUCTURE

- How to Create Dictionary?
- How to Access Data from the Dictionary?
- How to Update Dictionaries?
    - How to Delete Elements from Dictionary?
    - del d[key]
    - d.clear()
    - del d
- Important Functions of Dictionary

- Important Functions of Dictionary
  - dict()
  - len()
  - clear()
  - get()
  - pop()
  - popitem()
  - keys()
  - values()
  - items()
  - copy()
  - setdefault()
  - update()
- Dictionary Comprehension

9) FUNCTIONS

- Built in Functions
- User Defined Functions
- Parameters
- Return Statement
- Returning Mul tiple Values from a Function
- Types of Arguments
- Posi tional Arguments
- Keyword Arguments
- Default Arguments
- Variable L ength Arguments
- Case Study
- Types of Variables

  - Global Variables
  - Local Variables
  - global Keyword
  - Recursive Functions
  - Anonymous Functions
  - Normal Function
  - Lambda Function
  - filter() Function
  - map() Function
  - reduce() Function
  - Everything is an Object
  - Function Aliasing
  - Nested Functions

10)MODULES

- Renaming a Module at the time of import (Module Aliasing)
- from ... import
- Various Possibilties of import
- Member Aliasing
- Reloading a Module
- Finding Members of Module
- The Special Variable __name__
- Working with math Module
- Working with random Module
- random() Function
- randint() Function
- uniform() Function
- randrange ([start], stop, [step])
- choice() Function

PACKAGES
PATTERN PROGRAMS
[Adv.Python

OOP's Part – 1

- What is Class?
- How to define a Class?
- What is Object?
- What is Reference Variable?
- Self Variable
- Constructor Concept
- Differences between Methods and Constructors
- Types of Variables
    - Instance Variables (Object Level Variables)
    - Static Variables (Class Level Variables)
    - Local variables (Method Level Variables)
- Where we can declare Instance Variables
    - Inside Constructor by using self variable
    - Inside Instance Method by using self variable
    - Outside of the class by using object
- How to Access Instance Variables
- How to delete Instance Variable
- Static Variables
- Instance Variable vs Static Variable
- Various Places to declare Static Variables
- How to access Static Variables
- Where we can modify the Value of Static
- How to Delete Static Variables of a Class
- Local Variables
- Types of Method

  - Instance Methods
  - Class Methods
  - Static Methods
- Setter and Getter Methods
- Passing Members of One Class to Another Class
- Inner Classes
- Garbage Collection
- How to enable and disable Garbage Collector in our Program
- Destructors
- How to find the Number of References of an Object

OOP’s Part – 2

- Inheritance
  - By Composition (Has-A Relationship)
  - By Inheritance (IS-A Relationship
- IS-A vs HAS-A Relationship
- Composition vs Aggregation
- Types of Inheritance
  - Single Inheritance
  - Multi Level Inheritance
  - Hierarchical Inheritance
  - Multiple Inheritance
  - Hybrid Inheritance
  - Cyclic Inheritance
- Method Resolution Order (MRO)
- Head Element vs Tail Terminology
- How to find Merge?
- Finding mro(P) by using C3 Algorithm
- super() Method
- How to Call Method of a Particular Super Class?
- Various Important Points about super()

OOP’s Part – 3

- Polymorphism
- Duck Typing Philosophy of Python
- Overloading
- Operator Overloading
- Method Overloading
- Construc tor Overloading
- Overriding
- Method Overriding
- Constructor Overriding

OOP’s Part – 4

- Abstract Method
- Abstract class
- Interface
- Concreate class vs Abstract Class vs Inteface
- Public,Private and Protected Members
- __str__() Method
- Difference between str() and repr() functions
- Small Banking Application

Exception Handling

- Syntax Errors
- Runtime Errors
- What is Exception
- Default Exception Handing in Python
- Python's Exception Hierarchy
- Customized Exception Handling
- Control Flow in try-except

- How to Print Exception Information
- try with Multiple except Blocks
- Single except Block that can handle Multiple Exceptions
- Default except Block
- finally Block
- Control Flow in try-except-finally
- Nested try-except-finally Blocks
- Control Flow in nested try-except-finally
- else Block with try-except-finally
- Various possible Combinations of try-except-else-finally
- Types of Exceptions
  - Predefined Exceptions
  - User Definded Exceptions
- How to Define and Raise Customized Exceptions

File Handling

- Types of Files
  - Text Files
  - Binary Files
- Opening a File
- Closing a File
- Various Properties of File Object
- Writing Data to Text Files
  - write(str)
  - writelines(list of lines)

- Reading Character Data from Text Files
- read() à To Read Total Data from the File
- read(n) à To Read 'n' Characters from the File
- readline() à To Read only one Line
- readlines() à To Read all Lines into a List
- The with Statement
- The seek() and tell() Methods
- How to check a particular File exists OR not
- Handling Binary Data
- Handling CSV Files
- Writing Data to CSV File
- Reading Data from CSV Files
- Zipping and Unzipping Files
- To Create Zip File
- Working with Directories
- Running Other Programs from Python Program
- How to get Information about a File
- Pickling and Unpickling of Objects

Multi Threading

- Multi T asking
  - Process based Multi Tasking
  - Thread based Multi Tasking
- The ways of Creating Thread in Python
- Creating a Thread without using any class
- Creating a Thread by extending Thread class
- Creating a Thread without extending Thread
- Setting and Getting Name of a Thread
- Thread Identification Number (ident)

- enumerate() Function
- isAlive() Method
- join() Method
- Daemon Threads
- Default Nature
- Synchronization
  - Lock
  - RLock
  - Semaphore
- Synchronization By using Lock Concept
- Problem with Simple Lock
- Demo Program for Synchronization by using RLock
- Difference between Lock and RLock
- Synchronization by using Semaphore
- Bounded Semaphore
- Difference between Lock and Semaphore
- Inter Thread Communication
- Inter Thread Communication by using Event Objects
- Methods of Event Class
  - set()
  - clear()
  - isSet()
  - wait()|wait(seconds)
- Inter Thread Communication
- Methods of Condition
- acquire()
- release()
- wait()|wait(time)
- notify()
- notifyAll()
- Case Study

- Inter Tread Communic ation by using Queue
- Important Methods of Queue
    - put()
    - get()
- Types of Queues
    - FIFO Queue
    - LIFO Queue
    - Priority Queue
- Good Programming Practices with usage of Locks

Python Database Programming

- Storage Areas
    - Temporary Storage Areas
    - Permanent Storage Areas
- File Systems
- Databases
- Python Database Programming
- Working with Oracle Database
- Installing cx_Oracle
- How to Test Installation
- Working with MySQL Database
- Commonly used Commands in MySQL
- Driver/Connector Information
- How to Check Installation

Regular Expressions & Web Scraping

- Character Classes
- Pre defined Character Clas ses
- Qunatifiers
- Important Functions of Remodule
  - match()
  - fullmatch()
  - search()
  - findall()
  - finditer()
  - sub()
  - subn()
  - split()
  - compile()
- Web Scraping by using Regular Expressions

Decorator Functions

Decorator Chaining

Generator Functions

Advantages of Generator Functions

Generators vs Normal Collections wrt Performance

Generators vs Normal Collections wrt Memory Utilization

Assertions

Debugging Python Program by using ass ert Keyword

Types of assert Statements

Simple Version

Augmented Version

Exception Handling vs Assertions

Regular Expressions & Web Scraping

- Character Classes
- Pre defined Character Clas ses
- Qunatifiers
- Important Functions of Remodule
  - match()
  - fullmatch()
  - search()
  - findall()
  - finditer()
  - sub()
  - subn()
  - split()
  - compile()
- Web Scraping by using Regular Expressions

Decorator Functions

Decorator Chaining

Generator Functions

Advantages of Generator Functions

Generators vs Normal Collections wrt Performance

Generators vs Normal Collections wrt Memory Utilization

Assertions

Debugging Python Program by using assert Keyword

Types of assert Statements

Simple Version

Augmented Version

Exception Handling vs Assertions

Django & Atom Installation and Development of First Web Application

- How to install django
- ATOM IDE/Editor
- Speciality of ATOM IDE
- How to configure Atom for Python
- Django Project vs Django Application
- How to create Django Project
- How to run Django Development server
- How to send first request
- Role of Web Server
- Creation of First web application
- Activities required for Application
- Http Request flow in Django Application
- Summary of sequence of activities related to Django Project
- How to change Django Server Port
- Various Practice Applications
- Defining urlpatterns at application level instead of project level

Django Templates and Static Files

- Django Templates
- Python stuff required to develop Template
- Steps to develop Template Based Application
- Template Tags
- Application to send date and time from views.py
- Application To display date ,time and student information
- Application to wish end user based on time
- Working with Static Files
- Process to include static files inside template
- How to include css file

Working with Models and Databases

- Database configuration
- How to check django database connection
- Configuration of MySQL database
- Configuration of Oracle database
- Model Class
- Converting Model class into database specific sql code
- How to see corresponding sql code of migrations
- How to execute generated SQL code (migrate command)
- What is the advantage of creating tables with ‘migrate’ command
- How to check created table in django admin interface
- Creation of super user to login to admin interface
- Difference between makemigrations and migrate
- To display data in admin interface in browser
- MVT Diagram

Working with Django Forms

- Django Forms
- Advantages of Django Forms over HTML forms
- Process to generate Django forms
- CSRF(Cross Site Request Forgery) Token
- How to process input data from the form inside views.py file
- Form Validations
  - Explicitly by the programmer by using clean methods
  - By using Django inbuilt validators
- How to implement custom validators by using the same parameter

Working with Django Model Forms

- Model Forms(Forms based on Model)
- How to develop Model based forms
- How to save user's input data to database in Model based forms
- How to add date widget

Working with Advanced Template Features

- Template Inheritance
- How to implement template inheritane
- Demo program: advtempproject
- Advantages of Template Inheritance
- How to add seperate css files to child templates
- Tempalte Filters
- Syntax of Template Filter
- How to create our own filter
- Template Tags for urls

Session Management

- Session Management
- Session Management By using Cookies
- How to test our browser supports Cookies
- Limitations of Cookies
- Temporary vs Permanent Cookies
- Session Management By using Session API Framework)
- Useful Methods for session Management
- Important Methods related to Session
- Browser Length Sessions and Persistent Sessions

User Authentication and Authorization

- User Authentication and Authorization
- Difference between Static and Media Folders
- How to configure media folder in settings.py file

Class Based Views and CRUD Operations by using both CBVs and FBVs

- Class Based Views(CBVs)
- HelloWorld Application By using ClassBasedViews
-
- How to send context paramters
- ListView
- How to create template file for ListView
- How to provide our own context object name
- How to configure our own template file at project level
- DetailView
- Django CRUD Operations
- CreateView class
- UpdateView class
- DeleteView class

Django ORM

- Django ORM
- How to find Query associated with QuerySet
- How to filter records based on some condition
- Various possible Field Look ups
- How to implement OR Queries in Django ORM
- How to implement AND Queries in Django ORM
- How to implement NOT Queries in Django ORM
- How to perform Union operation for query sets of the same or different models
- How to select only some columns in the queryset
- Aggregate Functions
- How to Create, Update, Delete Records
- How to add multiple records at a time
- How to Delete a Single Record
- How to Delete Multiple Records
-
- How to Update Field of a Particular Record
- How to Order queryset in Sorting Order

Working with Advanced Model Concept

- Model Inheritance
- Abstract Base Class Model Inheritance
- Multi table Inheritance
- Multilevel Inheritance
- Multiple Inheritance
- Model Manager
- How to define our own Custom Manager
- Proxy Model Inheritance

Working with Django Middleware

- Middleware
- Middleware Structure
- Demo Application for Custom Middleware Execution Flow
- Execution Process for a Single Middleware Class
- Middleware application to show information saying application under maintenance
- Middleware application to show meaningful response if view function raises any error
- Configuration of multiple middleware classes

Real Time Project

- Introduction to Web application Development by using Flask
- How to install Flask
- How to Develop Application
- How to Run Development Server
- How to send Request
- Rest

DJango Rest Framework:[Rest API]

- API
- Web API/Web Service
- RESTFul API
- HTTP Verbs
- HTTP Verbs vs Database CRUD Operations
- How to install Django Rest Framework
- Types of Web Services:
    - SOAP Based WebServices
    - RESTful WebServices

- Differences between SOAP and REST
- HTTPie Module
- Class Based View(CBV) to send JSON Response
- Differences between Mixins and Multiple Inheritance
- Serialization
- Error Handling in the API
- Exception Handling in Partner Application (Python Script)
- How to add Status Code to HttpResponse explicitly
- How to disable CSRF Verification:
- Creating Model Based Form to hold Employee Data
- Developing WEB APIs by using 3rd Party Django REST Framework:
- API Functionality Testing by using POSTMAN:
- JWT (Json Web Token) Authentication

GEEKONIK

CONTACT

🌐 www.geenonik.com

☎ 9990161025

GEEKONIK