



JOIN THE
PYTHON
REVOLUTION AND
CODE YOUR DREAMS!



CORE PYTHON

INTRODUCTION TO PYTHON

- What is python
- Python use cases
- The Birth of Python
- Python Timeline
- Features of Python
- Versions of Python
- Python distributions

INTRODUCTION TO SCRIPT

- What is script
- What is program
- Types of script
- Types of programming languages

THE PYTHON ENVIRONMENT

- Installation of python on windows, MAC, LINUX
- Setting path for python
- Python Documentation
- Getting Help
- Python Command line shell, Editors and IDE's
- Basic Syntax
- Running Python Script on WINDOWS
- Running Python Script on LINUX
- Working with IDLE

GETTING STARTED

- Keywords v Data Types
- Static Data + Types versus Dynamic Data Types
- Fundamental Data Types
- Collection Types
- Number Systems
- Mutable objects versus Immutable Objects
- Variables
- Naming Conventions
- Print(), Type() and Id() Functions
- Input and raw_input() functions
- Type Conversion functions

OPERATIONS

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Short Hand Assignment Operators
- Bitwise Operators

- Membership Operators
- Identify Operators
- Precedence of Operators
- Evaluating expressions

FLOW CONTROL

- About Flow Control
- Elements of Flow Control
- Block / Clause
- Conditional Statements
 - > Simple if
 - > if....else
 - > if....elif...else
- Looping Statements
 - > While loop
 - > whileelse
 - > for loop
 - > for..else
 - > using range() in for loop
 - > working with infinite loops and nested loops
- Break Statement
- Continue Statement
- Pass Statement

STRING HANDLING

- What is String?
- Single –quoted string Literals
- Triple-quoted string literals
- String Indexing
- String Slicing
- Working with String Functions
- Working with String Methods
- Reversing a String
- String multiplication and concatenation

PYTHON

COLLECTION LIST:

- Lists are mutable
- Getting to Lists
- List indices
- Traversing a list
- List operations
- List Slices
- List methods
- Map, Filter and Reduce
- Deleting elements
- Lists and Strings

TUPLE:

- Advantages of Tuple over list
- Packing and Unpacking
- Comparing Tuples
- Creating nested tuple
- Using tuples as keys in dictionaries
- Deleting tuples
- Slicing of Tuple
- Tuple Membership Test
- Built-in function with Tuple
- Dotted Charts

SETS:

- How to create a set?
- Iteration Over sets
- Python set methods
- Python set Operations
- Union Of sets
- Built –in Functions with set
- Python Frozenset

DICTIONARY

- How to create a dictionary?
- Python Hashing
- Python Dictionary Methods
- Copying Dictionary
- Updating Dictionary
- Delete keys from the dictionary
- Dictionary items () Method

Functions

- Defining a Function
- Calling a function
- Function Parameters
- Types of Parameters
- > default parameters
- > non default parameters
- > keywords arguments
- > non key word arguments
- > arbitrary arguments

- Return statement in functions
- Handling return values
- Global variables and Local Variables
- Scope of global variables and local variables
- Passing collections to a function

LAMBDA FUNCTIONS

- Lambda functions / anonymous functions
- Filter()
- Map()
- Reduce()

MODULES

- What is Module?
- Types of Modules
- The import statement
- Module Aliases/renaming a module
- From... Import
- Reloading a Module
- Built in properties of a Module
- Dir() Functions
- Creating user defined modules
- Module search path
- Command line arguments
- Working with pre defined Standard modules (Math,Random,Datetime, Os, Sys,String,...)
- Packages
- Introduction to packages
- __init__.py file
- Defining packages
- Importing from packages
- Defining sub packages
- Importing from sub packages
- Difference between 2.X and 3.X packages

PYTHON



◆ ADVANCE PYTHON

OOPS Concepts

- Introduction to OOPS programming
- OOPS principles
- Encapsulation
- Defining Classes
- Creating Objects
- Class Variables
- Instance Variables
- Parameters
- Local Variables
- Defining Methods
- Difference between functions and methods
- Instance Methods
- Static Method
- Class Method
- Difference static and class methods
- Constructors
- Destructors
- Inheritance
- Types of inheritances
- Polymorphism (overloading & over riding)
- Super() Statement
- Built in properties of class
- Inner classes

EXCEPTION HANDLING

- Syntax Errors
- Runtime Errors
- What is BUG?
- What is Exception?
- Need of Exception handling
- Predefined Exceptions
- Predefined Exceptions Hierarchy
- Try, except and finally clauses
- Named except block
- Default except block
- Handling Multiple Exceptions
- Nested try, except and Finally blocks
- User Defined Exceptions
- Raise, assert statements.

FILE HANDLING

- What is file?
- Opening a file?
- Reading data from a file
- Writing data to a file
- Closing a file
- Working with the methods of file object

- Replacing the content of file
- Working with Directories
- Handling IO Exceptions

REGULAR EXPRESSIONS

- Introduction to regular expression
- Simple character matches
- Special characters
- Characters Classes
- Quantifiers
- Forming regular expressions
- Matching at beginning or end
- Greedy matches
- Compiling regular expressions
- Grouping
- Match Objects
- Match(), Search() and sub() functions
- Matching versus searching
- Splitting a string
- Replacing text
- Flags

DATABASE ACCESS

- Introduction to RDBMS
- Cursor object
- Executing SQL queries
- Executing SQL queries with bind variables
- Execution of PL/SQL procedures and functions
- Installation of mysql database
- Creating databases in mysql
- Creating users and assignment privileges to the users in mysql
- Installation of mysql python modules
- Establishing connection with mysql
- Closing mysql database connections
- Execution of insert, update, delete and select queries
- Handling db errors

PYTHON

ADVANCED CONCEPTS

- Python Iterator
- Python Generator
- Python Closure
- Python Decorators
- Python Property
- PIP
- Installation of external modules using PIP
- Working with csv,xml,and jsonfiles
- Debugging using IDE
- Test cases implementations

DATA ANALYTICS

- Introduction to big data
- Pandas
- Numpy Matplotlib
- Hadoop word count example

PYTHON WEB DEVELOPMENT

- Introduction to web application
- Architecture of web application
- Introduction to DJANGO framework
- Introduction to FLASK framework

D JANGO

- Introduction to Web Development
- Introduction to DJANGO
- MVT pattern
- Flow in Django Project
- Virtualenv
- Installing Django
- Creating a Django project
- What is a view
- Function based views
- Class based views
- Http Response
- URL route configuration
- CSRF
- HTML AND Django
- About templates
- Django template syntax
- Static Files
- Loading templates
- Context in Templates
- Sending Data
- Forms overview
- GET and POST
- The form Class
- Processing the form
- Validation

- Forms in templates
- Database Setup
- Creating Models
- Activation Models
- Activating Models
- Admin site
- One-to-Many Relation
- Many to one Relation
- One-to-one Relation
- Pagination
- Generic Views
- List Vies
- Detail View
- Create View
- Update View
- Delete View
- Cookies in Django
- Django Session Framework
- Caching
- File Uploading & Downloading
- Deployment on Apache
- Sending Email
- Other Email Functions
- Security with user Authentication
- Automated Testing
- When to create tests
- Using Django's test Framework
- Using the test Client
- Running tests
- Checking Code Coverage

PYTHON

Django

Introduction to Django

- Introduction
- About Django
- Django Components
- Installing & Configuring Django Components
- Django Pre-Requisites
- Downloading & Installing Django
- Choosing a Database
- Creating a New Project

Module 2: Generating Simple Django Views

- Generating Simple Django Views
- About View Functions
- Using Django's HttpResponse Class
- Understanding HttpRequest Objects
- Using QueryDict Objects

Module 3: Configuring URLconf's

- Configuring URLconf's
- About URLconf
- Regular Expressions
- Expression Examples
- Simple URLConf Examples
- Using Multiple URLConf's
- Passing URL Arguments

Module 4: Django Templates

- About Templates
- Template Fundamentals
- Creating Template Objects
- Loading Template Files
- Filling in Template Content (Context Objects)
- Template Filters Template Tags
- More on For Loops
- Template Inheritance
- Easy Rendering of Templates
- RequestContext Processors
- Global Context Processors

Module 5: Forms

- Getting Data From the Request Object
- A Simple Form-Handling Example
- Making a Contact Form
- Tying Form Objects Into Views

Module 6: Database Models with Django

- About Database Models
- Configuring Django for Database Access
- Understanding Django Apps

- About Django Models
- Defining Django Models
- Understanding Model Fields & Options
- Table Naming Conventions
- Creating A Django Model
- Adding the App to Your Project
- Validating the App
- Generating & Reviewing the SQL
- Adding Data to the Model
- Primary Keys and the Model
- Simple Data Retrieval Using a Model
- Understanding QuerySets
- Applying Filters
- Specifying Field Lookups
- Lookup Types
- Slicing QuerySets
- Specifying Ordering in QuerySets
- Common QuerySet Methods
- Deleting Records
- Managing Related Records
- Retrieving Related Records
- Using Q Objects
- Creating Forms from Models

Module 7: Using the Django Admin Interface

- Using the Django Admin Interface
- Enabling the Admin Interface
- Creating an Admin User

Module 8: Access Control with Sessions and Users

- Access Control with Sessions and Users
- Cookies & Django
- The Django Session Framework
- Sessions in Views
- Session Tuning
- Installing Django User Authentication
- Using Authentication in Views
- Login and Logout
- Building your Own Login/Logout Views
- Authentication Decorators
- Adding & Deactivating Users
- Asynchronous Messaging
- Managing Permissions

PYTHON