

# MasterProgram in Python FullStackWebDevelopment

COURSE

BROCHURE&SYLLABUS

## HTML

- WhatisMarkupLanguage
- BasicStructure of HTML
- Meta Tags
- ExternalLinkTags
- HTMLStructure Tags
- DifferencebetweenHTML&XHTML
- IntroductiontoHTML5
- Whatsnew inHTML5
- HTML5SemanticTag
- HTML5Multimedia
- CreateTable,Div, andFrameTag
- Contentand HeaderTags
- Paragraph,Span,PreTags
- AnchorLinksandNamedAnchors
- IframeanflmageTagImplementingiFrameinRealttime
- WorkingwithForms
- FormTagand Attributes
- POSTandGET Method
- TextInput,TextArea,CheckboxandRadio
- PasswordField
- SelectOption,OptionGroup
- FileFieldand HiddenFields
- Submit,Reset,ImageButtons
- HTML5Forms
- Canvas&SVGs
- Convertingsvgsintofont-icons
- HTML5Apis

## CSS

- **IntroductiontoCSS2& CSS3**
- **TypesofStyle Sheets**
- **MediaType**
- **DefaultCSSProperties**
- **CSSSelectors**
- **BasicCSSSelectors**
- **AdvancedCSSSelectors**
- **CSSproperties**
- **BackgroundProperties**
- **BlockProperties**
- **BoxModelProperties**
- **ListProperties**
- **BorderProperties**
- **PositioningProperties**
- **Flex**
- **Addingourownfontstowebsites**
- **CSSOptimizationandCompressionTips**
- **New CSS3Properties**
- **CSSRoundedCorners**
- **BorderImage**
- **BoxandTextShadow**
- **MultipleBackgrounds**
- **BackgroundOrigin**
- **BackgroundResize**
- **CSSClip,Gradients,Opacity**
- **Transitions&Transform**

## JAVASCRIPT

- **IntroductiontoClient-SideScripting**
- **IntroductiontoJavaScript**
- **JavascriptTypes**
- **VariablesinJS**
- **OperatorsinJS**
- **ConditionsStatements**
- **JavaScriptLoops**
- **JSPopupBoxes**
- **JSEvents**
- **JSArrays**
- **JSObjects**
- **JSFunctions**
- **UsingJavaScript inRealtime**
- **ValidationofForms**
- **RelatedExamples**

## JQUERY

- **IntroductiontojQuery**
- **jQueryFeatures**
- **InstallingjQuery**
- **jQuerySyntax**
- **jQueryReadyFunction**
- **jQuerySelectors**
- **jQueryActions**
- **jQueryCustom Functionality**
- **jQueryLibraries**
- **jQueryValidation**
- **jQuerySlideshow**

- **jQueryDropdown**
- **JSFunctions**
- **jQueryUI**
- **jQueryAccordions**
- **jQueryTabs**
- **jQueryTooltips**
- **jQueryAutocomplete**

## **MySQLDatabase**

- **UnderstandingBasicSQLSyntax**
- **QueryingDatawiththeSELECTStatement**
- **Filtering ResultswiththeWhereClause**
- **ShapingResultswithORDER BYand GROUPLY**
- **MatchingDifferentData TableswithJOINS**

CreatingDatabaseTable stamp

## ReactJS

- ReactJSBasics
- ReactJSFeatures
- ReactJSSetupandHelloWorldApplication
- ReactJSJSX
- ReactJS Component
- ReactJSState
- ReactJS Props
- ReactJSConstructor
- ReactJSLifecycle
- ReactJSEvents
- ReactJSRouter
- ReactJSForms
- ReactJSTables
- ReactJSPortals
- ReactJSES6
- ReactJS CSS
- ReactJS Hook
- ReactJSandBackEndIntegration
- ReactJSUsingBackEndCRUDApplication
- ReactJS,JDK1.8,SpringBoot,Hibernate, Maven,MySQL,Microservices,

## ProjectLombok-ProjectImplementation

### MongoDB

- RestApi
- AuthenticationandAuthorization
- Miniproject-QuizApp
- MongoDB
- Introduction&Overview ofMongoDB
- MongoDBInstallation
- CRUDOperationinMongoDB
- Data Modeling
- StorageClasses
- IndexingandPerformanceConsiderations
- Aggregation
- MongoDBReplication

# AngularJs

## Module1-Introductionto AngularWhatisAngular?

- **BriefHistoryofAngularPrerequisitefor AngularBasicUnderstandingofCSS**
- **BasicUnderstandingonJavascript**
- **Introduction: Typescript**
- **BuildingBlocksof Angular**
- **WhatisLinting?Whydoes it mattertous?**
- **More on TSLint**
- **ExploringTSLint**
- **Playing With**

## TSLintModule 2-Modules

- **Modules:What makes itsocool?**
- **Modules:What doesa Modulelooklike?**
- **Modules:What doesimportdo?**
- **Modules:WhyDeclarations &imports?**
- **Modules:WhyProviders&Bootstrap?**
- **Modules:WhatisRootModule?**
- **Creatingnewmodulein Angular Application...**
- **CreatingnewComponentinnewmodule...**

## Module3-Directives

- Directives:Whatmakesthemsowesome?
- Directives:WhatreallyisaDirective?
- Directives:TypesofDirectives
- Directives:FlavoursofDirectives
- Directives:HowtcreateDirectives
- Directives:HowdoesDirectivelooklike?

## Module4 -Components

- WhatmakestheComponentsocool?
- ComponentLifecycle
- Components: ComponentsinComponents
- Components: Component Interaction
- Components:MoreonComponents

## Module 5-Pipes

- Pipes:Whatmakesthemsowesome?
- Pipes:WhatreallyisaPipes?
- Pipes: JourneyfromfilterstoPipesSyntaxofPipesandthe Connectionwith
- Pipes:TypesofPipes
- DivingintoBuilt-inPipes
- Pipes:How tocreatePipes
- Pipes:How doespipeslooklike?UsingMultiplePipes...



## Module6-Services

- **Services:Whatmakesthemsowesome?**
- **Services:WhatreallyisaServices?**
- **Services:TypesofServices**
- **Services:HowtocreateServices**
- **Services:HowdoesServiceslooklike?**

## Module 7–Routing

- **Routing:What doesittaketodoRouting?**
- **Routing:Whatreallyis a Routing?**
- **Routing:WhodoestherealRouting?**
- **AngularRouter: Whatis it?**
- **Routing:HowtoCreateRouting**
- **Routing : How does Router look**

## LikeModule 8-TestingInAngular

- **WhatisTestDrivenDevelopment(TDD)AndBehavioralDrivenDevelopment(BDD)?**
- **TestingFrameworksinJS**
- **Whytotestangularcode?**
- **TypesofTesting**
- **Howtoruntest?**
- **Testing:Jasmine**
- **Testing:FundamentalConceptsDependencyInjection(DI)**
- **Testing:Karma**
- **Testing: TestEntryFile**
- **ComponentTesting**

- **ServiceTesting**
- **DirectiveTesting**
- **Pipe Testing**
- **Test debugging**

## 10. **Introductiontoforms in Angular**

## 11. **MiniProject**

- **MiniProject:UserDashboard**
- **UsingBootstrap 4inAngular Project**

## **Python and Django Course Syllabus**

### **Overview**

- **Why do we need Python?**
- **Program structure**

### **Environment Setup Python Installation**

- **Execution Types**
- **What is an interpreter?**
- **Interpreters vs Compilers**
- **Using the Python Interpreter**
- **Interactive Mode**
- **Running python files**
- **Working with Python shell**
- **Integrated Development Environments (IDES)**
- **Interactive Mode Programming**
- **Script Mode Programming**

## Basic Operators in Python Types of Operator

- Python Arithmetic Operators
- Python Comparison Operators
- Python Assignment Operators
- Python Bitwise Operators
- Python Logical Operators
- Python Membership Operators (in, not in)
- Python Identity Operators (is, is not)
- Python Operators Precedence

## Basic Concepts Data Types

- Data Types
- Variables
- Assigning Values to Variables
- Multiple Assignment
- Python Numbers
- Python Strings
- Accessing Values in Strings
- String Special Operators
- Triple Quotes
- Built-in String Operations

## Python Lists

- Accessing Values in Lists
- Updating Lists
- Delete List Elements
- Basic List Operations
- Indexing, Slicing, and Matrixes
- Built-in List Functions & Methods

## Python Tuples

- **Accessing Values in Tuples**
- **Updating Tuples**
- **Delete Tuple Elements**
- **Basic Tuples Operations**
- **Indexing, Slicing, and Matrixes**
- **No Enclosing Delimiters**
- **Built-in Tuple Functions**

## Python Dictionary

- **Accessing Values in Dictionary**
- **Updating Dictionary**
- **Delete Dictionary Elements**
- **Properties of Dictionary Keys**
- **Built-in Dictionary Functions & Methods**

## Loops and Decision Making

- **if statements**
- **else statements**
- **nested if statements**
- **while loop**
- **for loop**
- **nested loops**
- **Loop Control Statements**
- **1 break statement**
- **2 continue statement**
- **3 pass statement**

## Functions

- Defining a Function
- Syntax
- Calling a Function
- Pass by reference vs value
- Function Arguments
- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments
- The return Statement
- Scope of Variables

## Global vs. Local variables

### Built-in module and functions Os module

- mkdir()
- chdir()
- getcwd()
- rmdir()
- listdir()

### Math module and functions

- Constants
- Trigonometry
- Logarithmic
- Representation

### Random module and functions

- random()
- randint()
- randrange()
- choice () shuffle ()

### Python Modules and Packages

- Framework vs Packages
- Folium Introduction
- Why are modules used?
- Creating modules
- The import Statement
- The from...import Statement
- The from...import \* Statemen
- Locating Modules
- The PYTHONPATH Variable
- Namespaces and Scoping
- The dir( ) Function
- The globals() and locals() Functions
- The reload() Function
- Packages in Python
- Constructing user defined packages
- Importing user defined packages

### Basic OOPs Concept

- Creating class in Python Private Identifier
- Constructor
- Inheritance
- Polymorphism

## Decorator, Iterator and Generator

### Anonymous Function

- Lambda
- Map
- Filter
- Reduce

### File Manipulation

- Opening Text File
- Working with a File on Python
- The open function
- File modes
- The file object attributes
- close() method
- write() method
- read() method
- Files: Input
- Files: Output
- Reading files
- Renaming & deleting files
- Writing into a file
- remove() method

### Python GUI

- Basic Operations using Tkinter
- Buttons and Textbox
- Menu Bar
- Message Box and Radio Button

- **Checkbox and Event Creating**
- **Creating Application in GUI SQL and Python**
- **Overview of SQLite**
- **Integrating Python with SQLite**

### **Project Demonstration Tkinter**

- **Project Demonstration Tkinter**

### **Other Concepts**

- **Errors and Exception Handling**
- **Standard exceptions**
- **Assertions in Python**
- **The assert Statement**
- **What is Exception?**
- **Handling an exception**
- **Syntax**
- **The except Clause with No Exceptions**
- **The except Clause with Multiple Exceptions**
- **The try-finally Clause**
- **Argument of an Exception**
- **Example with Tkinter Application**
- **Regular Expression**

### **Advanced Concept – Overviews**

- **Basics of Pandas and Numpy**
- **How to use Anaconda**
- **How to create dashboard**
- **Overview of Django**



## Django Course Syllabus

### SECTION – I

- Introduction to Django
- What Is a Web Framework?
- The MVC Design Pattern
- Django's History

### 2. Installation of Django

- Installing Python
- Installing Django
- Setting Up a Database
- Starting a Project.
- The Development Server
- Django Commands Overview

### 3. The Basics of Dynamic Web Pages

- Your First View: Dynamic Content
- Mapping URLs to Views
- How Django Processes a Request
- URL configurations and Loose Coupling
- 404 Errors
- Your Second View: Dynamic URLs
- A Word About Pretty URLs
- Wildcard URL patterns
- Django's Pretty Error Pages

#### 4. The Django Template System

- **Template System Basics**
- **Using the Template System**
- **Creating Template Objects**
- **Rendering a Template**
- **Multiple Contexts, Same Template**
- **Context Variable Lookup**
- **Playing with Context Objects**
- **Basic Template Tags and Filters Tags**
- **Philosophies and Limitations**
- **Using Templates in Views**
- **Template Loading**
- **render\_to\_response()**
- **The locals() Trick**
- **Subdirectories in get\_template()**
- **The include Template Tag**
- **Template Inheritance**

#### 5. Interacting with a Database: Models

- **The “Dumb” Way to Do Database Queries in Views**
- **The MTV Development Pattern**
- **Configuring the Database**
- **Your First App**
- **Defining Models in Python**
- **In Your First Model**
- **Installing the Model**
- **Basic Data Access**
- **Adding Model String Representations**
- **Inserting and Updating Data**
- **Selecting Objects**

- **Filtering Data**
- **Retrieving Single Objects**
- **Ordering Data**
- **Chaining Lookups**
- **Slicing Data**
- **Deleting Objects**
- **Making Changes to a Database Schema**
- **Adding Fields**
- **Removing Fields**
- **Removing Many to Many Fields**
- **Removing Models**

## **6. The Django Administration Site**

- **Activating the Admin Interface**
- **Using the Admin Interface**
- **Users, Groups and Permissions**
- **Customizing the Admin Interface**
- **Customizing the Admin Interface's Look and Feel**
- **Customizing the Admin Index Page**
- **When and Why to Use the Admin Interface**

## **SECTION - II**

## **7. Form Processing**

- **Search**
- **The “Perfect Form”**
- **Creating a Feedback Form**
- **Processing the Submission**
- **Custom Validation Rules**
- **A Custom Look and Feel**
- **Creating Forms from Models**

## 8. Advanced Views and URL configurations

- **URL configuration Tricks.**
- **Streamlining Function Imports**
- **Using Multiple View Prefixes**
- **Special-Casing URLs in Debug Mode**
- **Using Named Groups**
- **Understanding the Matching/Grouping Algorithm**
- **Passing Extra Options to View Functions**
- **Using Default View Arguments**
- **Special-Casing Views**
- **Capturing Text in URLs**
- **Determining What the URL configuration Searches Against**
- **Including Other URL configurations**
- **How Captured Parameters Work with include()**
- **How Extra URL configurations Options Work with include()**
- **Section II : Django Sub Framework**

## 9. Generic Views

- **Using Generic Views**
- **Generic Views of Objects**
- **Extending Generic Views**
- **Making “Friendly” Template Contexts**
- **Adding Extra Context**
- **Viewing Subsets of Objects**
- **Complex Filtering with Wrapper Functions**
- **Performing Extra Work**

## 10. Extending the Template Engine

- **Template Language Review**
- **Request Context and Context Processors**
- **django.core.context\_processors.auth**
- **django.core.context\_processors.request**
- **Guidelines for Writing Your Own Context Processors**
- **Inside Template Loading**
- **Extending the Template System**
- **Creating a Template Library**
- **Writing Custom Template Filters**
- **Writing Custom Template Tags**

## 11. Users and Registration

- **Cookies**
- **Getting and Setting Cookies**
- **The Mixed Blessing of Cookies**
- **Setting Test Cookies**
- **Users and Authentication**
- **Enabling Authentication Support**
- **Using Users**
- **Logging In and Out**
- **Limiting Access to Logged-in Users**
- **Managing Users, Permissions and Groups**
- **Using Authentication Data in Templates**
- **Permissions**
- **Groups**
- **Messages**
- **Profiles**

# AWS

## 1: Introduction to AWS

- **Navigate the AWS Management Console**
- **Recognise AWS Global Infrastructure**
- **Describe the security measures AWS provides**
- **Create an Amazon EC2 instance**
- **Remotely connecting to an EC2 instance**
- **Setting up Amazon Linux and Apache web server**
- **Browsing Amazon Machine Images (AMI)**
- **Specifying security groups and key pairs**
- **Creating new images off of running instances**

## 2: AWS Storage

- **Identify key AWS storage options**
- **Describe Amazon EBS**
- **Use Amazon EBS with Amazon EC2**
- **Working with Volumes and snapshots**
- **Transmitting data in/out of the Amazon cloud**

## 3: Installing Software in your Amazon Instance

- **Implementing a Web server in an Amazon Linux/Windows Instance**
- **Configure the firewall to access a Web server**

## 4: Security in Public Cloud

- **Security issues in Public Cloud**
- **Securing the Access : Creating a RSA Public/Private Key for VMs**
- **Creating a software firewall**
- **Configuring firewall rules**
- **Securing the access with an Elastic IPs**

## 5: Alternate access

- **Using EC2 Command Line APIs [from Windows/Linux machines]**

## 6: AWS-IAM

- **Creating Groups and Defining access policy**

- Creating Users
- Login to AWS account using new users

# Linux

## 1: Overview to Linux & Unix based operating systems

- Unix, Linux, Debian
- Virtualization, software, diff b/w reg and virt
- OS Architecture
- Windows VS Linux
- Gnome and shell
- Types of shell
- FS Hierarchy

## 2: Working with Basic Linux Commands

- Access a shell prompt and issue commands with correct syntax Listing and navigating
- Log in and switch users in multiuser targets
- Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2
- Create and edit text files
- Create, delete, copy, and move files and directories
- Create hard and soft links
- Locate, read, and use system documentation including man, info, and files in /usr/share/doc
- Access remote systems using SSH
- List, set, and change standard ugo/rwx permissions
- Create and configure set-GID directories for collaboration

## 3: System Configuration

- Use input-output redirection
- Use grep and regular expressions to analyze text
- Identify CPU/memory intensive processes and kill processes
- Manage tuning profiles
- Locate and interpret system log files
- Securely transfer files between systems

**\*Thank You\***