an initiative by affimintus

Programmers Point SM
Earning Trust In Education

PYTHON | DJANGO WEB
FULL STACK DEVELOPER
Placement TRAINING MODULE

JavaScript/ React/Python/Django/ REST API/ GITHUB

ISO 9001 International Organization for Standardization

an initiative by affimintus
Programmers Point SM
Earning Trust In Education

ICL INTEGRAL CERTIFICATION PVT LTD
EGAC ACC. 112001
Certified ISO 9001:2015

affimintus TECHNOLOGIES

JAVA | SPRING| HIBERNATE| BIG DATA| HADOOP| DATA ANALYTICS| WEB DEVELOPMENT| ANDROID| ANGULAR JS|CC++| DATA STRUCTURE| SQL DBMS | MACHINE LEARNING| DATA SCIENCE WITH PYTHON | PLACEMENT PROVIDER

VIJAY NAGAR | BHAWARKUA | GEETA BHAWAN | Helpline No- 7974250782

Full Placement JOB Oriented
Corporate PYTHON Full Stack Web Development Training

- **For Logic Improvement and Programming basics**
  - C programming basics
  - C++ with Object Oriented Programming (OOPS)
  - Data structure and Algorithms for Product based companies
- **Corporate Python Complete Road Map step by step-**
  - Core Python
  - Advance Python with Object Oriented Programming
  - CGI Programming
  - Python Database Connectivity with ORACLE & MONGO DB
  - Python Server Pages
  - MVC Architecture
- **Python Frameworks**
  - ➔ DJANGO MVC
  - ➔ DJANGO ORM
  - ➔ DJANGO with REST API Development
  - ➔ DJANGO Live Server side deployment
  - ➔ DJANGO with Front End UI

Front End UI Web Designing

- HTML 5
- CSS 3
- JavaScript complete

- JQuery
- Bootstrap
- Ajax

REACT JS- JavaScript Front End Frameworks

- React JS
  - Project Development Training on Python projects
  - Internship on Python Live projects
  - Knowledge of Project Development tools
  - PyCharm and Jupyter
  - Major Project development using Django
  - Front End & Back End Integration on Server Live Application
  - Naukri.com, Linked IN & Other Job Portal Profile Building
  - Live Project Server hosting Python
  - Technical Interview Preparation Python
  - Placement Support after training completion
  - Technical Resume/CV creation
  - After Placement support for Verification
  - GITHUB uses for code repository

Introduction to Python Programming

Python is a high-level general purpose programming language. It has been one of the most popular programming languages of the recent years and has many areas of application from web applications to machine-learning and data science.

Python is easy to learn because of its intuitive and natural syntax. It is also a highly productive programming language, which allows you to build complex applications quickly with minimal lines of code.

We at Programmers Point, designed this course to teach students to program in Python in a practical and hands-on manner using the industry standard methods, tools and technologies.

Trainers at Programmers Point not only teaches students the Python programming language but also improves their algorithmic thinking and problem solving capabilities so that they can write code that actually works and produces the desired functional results. Giving students enough well thought coding exercises ensures this.

➔ Teaching Strategies

The course material is designed to appeal to a variety of students, students belongs to **Non-Technical/ Technical background**. All lectures is delivered **LIVE** through an OFFLINE and ONLINE both system that allows students to work practically & Live interaction with faculty. Students can ask any no of doubts from trainer without any hesitation.

Core Python Syllabus/Curriculum

Module 1:- Introduction & Course Overview

- ✓ Introduction to Python Programming
- ✓ Why become a Python Developer
- ✓ Python Overview
- ✓ Need of Python
- ✓ Features of Python
- ✓ Python 2 Vs Python 3
- ✓ Python Vs Java
- ✓ Python Setup Installation
- ✓ Working on Command Line
- ✓ Working with IDEs
- ✓ Example of Python Program
- ✓ Writing comments in Python

Module 2:- Setup & Installation

- ✓ Command Line Basics
- ✓ Installing Python setup
- ✓ Running Python Code

Module 3:- Python Data Types

- ✓ Data Types in Python
  - Numeric
  - Boolean
  - Sequence
- ✓ Variables
- ✓ Types of Variables

Module 4:- Operators in Python

- ✓ Arithmetic operators
- ✓ Assignment operators
- ✓ Comparison operators
- ✓ Logical operators
- ✓ Identity operators
- ✓ Membership operators
- ✓ Bitwise operators

Module 5:- Python Conditional Statements

- ✓ If Statement
- ✓ If Else Statement
- ✓ If Else If Statement
- ✓ Nested If Else Statement

Module 6:- Python Looping Statements

- ✓ While Loop
- ✓ For Loop
- ✓ Range in Python
- ✓ Nested Loops in Python

Module 7:- Python Casting

- ✓ Int() function
- ✓ Float() function
- ✓ Str() function

Module 8:- LIST in Python

- ✓ Introduction to List

- ✓ Accessing List
- ✓ List Iteration using Loops
- ✓ Insertion in List
- ✓ Delete a List
- ✓ Searching & Sorting in List
- ✓ Reverse a List
- ✓ Functions & Methods

Module 9:- TUPLES in Python

- ✓ Introduction to Tuples
- ✓ How to create Tuples
- ✓ Accessing Tuples
- ✓ Functions & Methods
- ✓ Nested Tuples

Module 10:- ARRAY in Python

- ✓ Create Array
- ✓ Accessing Array
- ✓ Array Operations
- ✓ Diff. between Array & List

Module 11:- STRING MANIPULATION in Python

- ✓ Create String
- ✓ String Operations
    - Concatenation

- Splitting & Joining String
- String Trimming
- Reverse String

✓ Functions & Methods String

✓ Substring

Module 10:- Functions in Python

✓ Defining a Function

✓ Calling a Function

✓ Types of Functions

✓ Function Arguments

✓ Anonymous Functions

✓ Global & Local Variables

Module 11:- Dictionary in Python

✓ Introduction

✓ Create Dictionaries

✓ Access Dictionaries

✓ Working with Dictionaries

✓ Properties of Dictionaries

✓ Dictionaries Functions

✓ Delete Dictionaries

Module 12:- Modules in Python

✓ Introduction to Python Modules

- ✓ Importing Modules
- ✓ Types of Modules
- ✓ Math Module
- ✓ Random Module
- ✓ User define Module
- ✓ Packages
- ✓ Composition

Advance Python-Object Oriented Programming with Python

- ✓ Class & Objects
- ✓ Overview of OOPS
- ✓ Create Objects
- ✓ Constructors
- ✓ Destructors
- ✓ Program for Class & Objects

Inheritance

- ✓ Overview of Inheritance
- ✓ Inheritance in Python
- ✓ Types of Inheritance
- ✓ Method Overloading
- ✓ Method Overriding
- ✓ Constructors in Inheritance
- ✓ Abstract Class
- ✓ Abstract Methods
- ✓ Program for Inheritance

Polymorphism

- ✓ Introduction
- ✓ Practical Implementation in Python
- ✓ Method Overloading
- ✓ Method Overriding
- ✓ Static and Dynamic Polymorphism

Multithreading

- ✓ Introduction to Threads
- ✓ Starting a Thread
- ✓ Use of Threads
- ✓ Thread Class
- ✓ Synchronizing Threads

Exception Handling

- ✓ Introduction to Exceptions
- ✓ How to Handle Exceptions
- ✓ Try, catch in finally
- ✓ User Defined Exceptions
- ✓ Types of Exceptions

File Handling

- ✓ Concept of Files
- ✓ Create, Open & Close Files
- ✓ File Modes
- ✓ Text Files
- ✓ Binary Files
- ✓ Read & Write operations in Files
- ✓ Delete a File

Python MySQL database Programming

- ✓ SQL/MySQL Introduction
- ✓ Overview of RDBMS
- ✓ MySQL Create Database
- ✓ MySQL Create Table
- ✓ MySQL Insert
- ✓ MySQL Select
- ✓ MySQL Where
- ✓ MySQL Order By

- ✓ MySQL Delete
- ✓ MySQL
- ✓ Drop Table
- ✓ MySQL Update
- ✓ MySQL Limit
- ✓ Operations on Database
- ✓ Operations of Tables
- ✓ Data Manipulations

Basics of Data Analysis with Python

- ✓ Introduction to NumPy
- ✓ How to use NumPy
- ✓ Introduction to Pandas
- ✓ How to use Pandas
- ✓ Application of NumPy and Pandas

Interview + Placement Preparation

- ✓ Exercise Problems
- ✓ Interview Questions on Core Python
- ✓ Interview Questions on Advance Python
- ✓ Python Quiz Questions

Web development using Django Programming

WHAT YOU WILL LEARN:

- Best practices in Django.
- Create a fully functional web site using the Full-Stack with Django
- Use CSS to create beautifully styled sites.
- Use Javascript to interact with sites on the Front-End.
- Understand HTTP requests.
- Learn the power of Python to code out your web applications.
- Implement a full Models-Views-Templates structure for your site.
- Learn how to use HTML to create website content.
- Learn how to take advantage of Bootstrap to quickly style sites.
- Learn how to use jQuery to quickly work with the DOM.
- Create fantastic landing pages.
- Use Django as a back end for the website.

Course Objective:

- To understand why Python is a useful Web developers.
- To learn how to design and program Python Web applications.
- To learn how to use python libraries in Web Development
- To learn Django- Python Web Framework
- To learn how to Integrate Front end and back end
- To define the structure and components of a web applications
- To learn Database connectivity with python
- To learn how to generate UI on front end with Web designing
- To learn Python web development tools
- Server side dynamic web pages response generation to end user
- To learn how to design object-oriented programs with Python classes.
- To learn how to use class inheritance in Python for reusability.
- To learn how to use exception handling in Python applications for error.

SYLLABUS

- Introduction to WEB
- Client/Server Application Architecture
- Web Technology & Protocols
- Web Application Architecture
- Static / Dynamic Resources
- HTTP
- HTTP Status Code
- HTTP Methods

SERVER SIDE PROGRAMMING USING PYTHON DJANGO FRAMEWORK

- Introduction
- Design Patterns
- Front Controller
- MVC (Model-View-Controller)
- MVT (Model-View-Template)
- ORM (Object Relational Mapping)
- Framework Introduction
- **Django Framework**
    - Django Architecture
    - Django API
    - Django Server
- **Virtual Environment**
    - Why Virtual Environment
    - Install pipenv
    - Activate pipenv
    - Virtual environment Files

- **DJango installation**
  - Environment configuration
  - Working With VS Code IDE
- **Django Project**
  - Introduction
  - Creating First Django Project
  - Project Directory Structure
  - Auto generated Components of Project
  - Manage.py
  - Setting.py
  - Run Server
  - Access Application on Browser
  - Change Server Port

- **Apps in Django Project**
  - App Concept
  - Create **App** in Django Project
  - App Directory Structure
  - Auto generated Components of App
  - Three Core Component: model.py, view.py & urls.py
  - App configuration with in Project
- **Understanding URL**
  - Path in URL
  - Path argument
    - Route
    - View
    - Name
  - Project URLs
  - App URLs

- Default URLs
- URL Configuration
    - Project UrlConf
    - App UrlConf
- URL Patterns
- Slug in URL

- **Views in Django**
    - View Types
    - Method View
    - Class View
    - Multiple Views
    - View Decorators

- Cache Control
- HttpRequest
- GET/POST
- Query String
- Read Slug
- CSRF Token
- Read GET Data
- Read POST Data
- HttpResponse
- Text Response
- Render HTML Page as Response
- Bind Data with HTML Response
- Rendering Dynamic Data in HTML
- Response Redirect
- Resolve URL
- get_object_or_404

- get_list_or_404
- raise Http404
- **Static Resource Management**
    - Images
    - CSS
    - Java Script
- **HTML Template Pattern**
- **Base Template Design**
    - Header
    - Navigation bar and Menus
    - Content Blocks
    - Footer
- Bootstrap Integration
- Extending Template
- Override Template Blocks
- **Django template language**
    - Tags
    - Using Variables
    - Filters
    - if statements
    - for loop
    - block
    - url
    - load static
    - comments
- **Django ORM**
- **Django Models**
    - Understanding Models
    - Models v/s Database

  - Model Data Types
  - Primary Key Configuration
  - Auto Increment Field
  - Unique Key Configuration
  - Index Column
  - Foreign Key Relations
  - Multiple Choice
  - Null Value
  - Blank Data
  - Default Value
  - Field Label for Form
  - Column Name
  - Non Editable Field
  - Validators
  - Error Messages
  - Required/ Optional
  - Meta Class
  - Table Name
- Default Database Configuration
- Database Configuration (MySQL)
- Migrations With Database
- **Django Admin Interface**
  - Create super user
  - Login to Admin Panel
  - Manage Users
  - Create Model
  - Register Model with Admin
  - Model CRUD Operations

- Add New
- Show All
- Update
- Delete
- Multiple Delete

- **Model Query Language**
  - Save Model
  - Update Model
  - Delete Model
  - QuerySet
  - Select All
  - Get Model for Primary Key
  - Query with field criteria
  - Query With Relational Operators
  - Like Query
  - Query for Null
  - Logical operations : AND | OR
  - Order By
  - Limit Query
  - Aggregate Functions
  - Result as Objects
  - Result as fields List
  - Result as field Dict
  - Get Selected Field Only
  - Native Raw Query
  - Model Query on Python Shell
- **Django Form Support**

  - Form
  - ModelForm
  - Form Fields
  - Form Widgets
  - Form Widgets Arguments
  - Form Input Types
  - Form in Template
  - Form Validation
  - Form Data processing in Views
- **Database Operations (CRUD) in Views**
  - Save Form Data
  - Update Form Data
  - Show All in Template
  - Search Data
  - Delete Data
- URL Management
- **Session Management**
  - Understanding HTTP Session
  - Access session in Views
  - Bind Data with Session
  - Remove Data from Session
  - Access Session in Template
  - Clear Session
  - Session in Middleware
- **Session Application**
  - Login/Logout
  - Authentication
  - Profile Data Management
  - Change Password

- Session in navigation bar
- Shopping Cart

- **File Handling in Django**
  - FileField
  - ImageField
  - File Object
  - File uploading
  - File Downloading
  - File Delete
  - Profile Photo Management
  - Create Photo Album

- **Generic View Classes & Uses**
  - TemplateView
  - CreateView
  - DetailView
  - UpdateView
  - DeleteView
  - ListView
  - FormView
- Download Data as CSV
- Download Data as PDF

Django – Sending E-mails

- Sending a Simple E-mail
- Sending Multiple Mails with send_mass_mail
- Sending HTML E-mail
- Sending HTML E-mail with Attachments

Application Development (Project)

- ✓ Project Requirement analysis
- ✓ Identify Project Users
- ✓ Project Modules
- ✓ Use Case
- ✓ Database Modeling
- ✓ UI Design
- ✓ Create Django Project
- ✓ Create Django Apps
- ✓ Write Models
- ✓ Write Views
- ✓ Add Urls
- ✓ Create HTML Master Templates
- ✓ Create Child Templates
- ✓ Manage Static Resources
- ✓ Create Forms
- ✓ Applying Session
- ✓ Test Application

an initiative by affimintus

Programmers Point SM
Earning Trust In Education

ORACLE CERTIFIED ASSOCIATE EXAM PREPARATION

ORACLE®

DATABASE

PL/SQL

an initiative by affimintus

DBMS PROGRAMMING

Programmers Point SM
Earning Trust In Education

(Become ORACLE Certified Working Professional)

Oracle SQL & PL-SQL Syllabus

affimintus TECHNOLOGIES
81090 50011

CERTIFIED ISO 9001:2015 COMPANY

Programmers Point
Bhawarkua | Geeta Bhawan
0731 4701122

Writing Basic SQL SELECT Statements

- ✓ Basic SELECT Statement
- ✓ Selecting All Columns
- ✓ Selecting Specific Columns
- ✓ Writing SQL Statements
- ✓ Column Heading Defaults
- ✓ Arithmetic Expressions
- ✓ Using Arithmetic Operators
- ✓ Defining a Null Value
- ✓ Null Values in Arithmetic Expressions
- ✓ Defining a Column Alias
- ✓ Using Column Aliases
- ✓ Concatenation Operator
- ✓ Using the Concatenation Operator
- ✓ Literal Character Strings
- ✓ Using Literal Character Strings
- ✓ Duplicate Rows
- ✓ Eliminating Duplicate Rows

Restricting and Sorting Data

- ✓ Limiting Rows Using a Selection
- ✓ Limiting the Rows Selected
- ✓ Using the WHERE Clause
- ✓ Character Strings and Dates
- ✓ Comparison Conditions
- ✓ Using Comparison Conditions
- ✓ Other Comparison Conditions
- ✓ Using the BETWEEN Condition
- ✓ Using the IN Condition
- ✓ Using the LIKE Condition
- ✓ Using the NULL Conditions
- ✓ Logical Conditions
- ✓ Using the AND Operator

- ✓ Using the OR Operator
- ✓ Using the NOT Operator
- ✓ Rules of Precedence
- ✓ ORDER BY Clause
- ✓ Sorting in Descending Order
- ✓ Sorting by Column Alias
- ✓ Sorting by Multiple Columns

Single-Row Functions

- ✓ SQL Function
- ✓ Two Types of SQL Functions
- ✓ Single-Row Functions
- ✓ Character Functions
- ✓ Case Manipulation Functions
- ✓ Using Case Manipulation Functions
- ✓ Character-Manipulation Functions
- ✓ Using the Character-Manipulation Functions
- ✓ Number Functions
- ✓ Using the ROUND Function
- ✓ Using the TRUNC Function
- ✓ Working with Dates
- ✓ Arithmetic with Dates
- ✓ Using Arithmetic Operators with Dates
- ✓ Date Functions
- ✓ Using Date Functions

Displaying Data from Multiple Tables

- ✓ Obtaining Data from Multiple Tables
- ✓ Cartesian Products
- ✓ Generating a Cartesian Product
- ✓ Types of Joins
- ✓ Joining Tables Using Oracle Syntax
- ✓ What is an Equijoin?
- ✓ Retrieving Records with Equijoins
- ✓ Additional Search Conditions Using the AND Operator

- ✓ Qualifying Ambiguous Column Names
- ✓ Using Table Aliases
- ✓ Joining More than Two Tables
- ✓ Non-Equijoins
- ✓ Retrieving Records with Non-Equijoins
- ✓ Outer Joins Outer Joins Syntax
- ✓ Using Outer Joins
- ✓ Self Joins
- ✓ Joining a Table to Itself
- ✓ Creating Cross Joins
- ✓ Creating Natural Joins
- ✓ Retrieving Records with Natural Joins
- ✓ Creating Joins with the USING Clause
- ✓ Retrieving Records with the USING Clause
- ✓ Creating Joins with the ON Clause
- ✓ Retrieving Records with the ON Clause
- ✓ Creating Three-Way Joins with the ON Clause
- ✓ INNER Versus OUTER Joins
- ✓ LEFT OUTER JOIN
- ✓ RIGHT OUTER JOIN
- ✓ FULL OUTER JOIN
- ✓ Additional Conditions

Aggregating Data Using Group Functions

- ✓ What Are Group Functions?
- ✓ Types of Group Functions
- ✓ Group Functions Syntax
- ✓ Using the AVG and SUM Functions
- ✓ Using the MIN and MAX Functions
- ✓ Using the COUNT Function
- ✓ Using the DISTINCT Keyword
- ✓ Group Functions and Null Values
- ✓ Using the NVL Function with Group Functions
- ✓ Creating Groups of Data
- ✓ Creating Groups of Data: The GROUP BY Clause Syntax
- ✓ Using the GROUP BY Clause

- ✓ Grouping by More Than One Column
- ✓ Using the GROUP BY Clause on Multiple Columns
- ✓ Illegal Queries Using Group Functions
- ✓ Excluding Group Results
- ✓ Excluding Group Results: The HAVING Clause
- ✓ Using the HAVING Clause
- ✓ Nesting Group Functions

Subqueries

- ✓ Objectives
- ✓ Using a Subquery to Solve a Problem
- ✓ Subquery Syntax
- ✓ Using a Subquery
- ✓ Guidelines for Using Subqueries
- ✓ Types of Subqueries
- ✓ Using the ANY Operator in Multiple-Row Subqueries
- ✓ Using the ALL Operator in Multiple-Row Subqueries

Manipulating Data

- ✓ Data Manipulation Language
- ✓ Adding a New Row to a Table
- ✓ The INSERT Statement Syntax 8-5
- ✓ Inserting New Rows
- ✓ Inserting Rows with Null Values
- ✓ Inserting Special Values
- ✓ Inserting Specific Date Values
- ✓ Creating a Script
- ✓ Copying Rows from Another Table
- ✓ Changing Data in a Table
- ✓ The UPDATE Statement Syntax
- ✓ Updating Rows in a Table
- ✓ Updating Two Columns with a Subquery
- ✓ Updating Rows Based on Another Table
- ✓ Updating Rows: Integrity Constraint Error
- ✓ Removing a Row from a Table

- ✓ The DELETE Statement
- ✓ Deleting Rows from a Table
- ✓ Deleting Rows Based on Another Table
- ✓ Deleting Rows: Integrity Constraint Error
- ✓ Using a Subquery in an INSERT Statement
- ✓ Using the WITH CHECK OPTION Keyword on DML Statements
- ✓ Overview of the Explicit Default Feature
- ✓ Using Explicit Default Values
- ✓ The MERGE Statement
- ✓ The MERGE Statement Syntax
- ✓ Merging Rows
- ✓ Database Transactions
- ✓ Advantages of COMMIT and ROLLBACK Statements
- ✓ Controlling Transactions
- ✓ Rolling Back Changes to a Marker
- ✓ Implicit Transaction Processing
- ✓ State of the Data Before COMMIT or ROLLBACK
- ✓ State of the Data after COMMIT
- ✓ Committing Data
- ✓ State of the Data After ROLLBACK
- ✓ Statement-Level Rollback
- ✓ Read Consistency
- ✓ Implementation of Read Consistency
- ✓ Locking
- ✓ Implicit Locking
- ✓ Read Consistency Example

Creating and Managing Tables

- ✓ Database Objects
- ✓ Naming Rules
- ✓ The CREATE TABLE Statement
- ✓ Referencing Another User?s Tables
- ✓ The DEFAULT Option
- ✓ Creating Tables
- ✓ Tables in the Oracle Database
- ✓ Querying the Data Dictionary 9-10
- ✓ Data Types

- ✓ DateTime Data Types
- ✓ TIMESTAMP WITH TIME ZONE Data Type
- ✓ Creating a Table by Using a Subquery Syntax
- ✓ Creating a Table by Using a Subquery
- ✓ The ALTER TABLE Statement
- ✓ Adding a Column
- ✓ Modifying a Column
- ✓ Dropping a Column
- ✓ The SET UNUSED Option
- ✓ Dropping a Table
- ✓ Truncating a Table
- ✓ Adding Comments to a Table

Including Constraints

- ✓ What are Constraints?
- ✓ Constraint Guidelines
- ✓ Defining Constraints
- ✓ The NOT NULL Constraint
- ✓ The UNIQUE Constraint
- ✓ The PRIMARY KEY Constraint
- ✓ The FOREIGN KEY Constraint
- ✓ FOREIGN KEY Constraint Keywords
- ✓ The CHECK Constraint
- ✓ Adding a Constraint Syntax
- ✓ Adding a Constraint
- ✓ Dropping a Constraint
- ✓ Disabling Constraints
- ✓ Enabling Constraints
- ✓ Cascading Constraints
- ✓ Viewing Constraints
- ✓ Viewing the Columns Associated with Constraints

Creating Views

- ✓ Database Objects
- ✓ What is a View?
- ✓ Why use Views?

- ✓ Simple Views and Complex Views
- ✓ Creating a View
- ✓ Retrieving Data from a View
- ✓ Querying a View
- ✓ Modifying a View
- ✓ Rules for Performing DML Operations on a View
- ✓ Removing a View
- ✓ Inline Views

Controlling User Access

- ✓ Objectives
- ✓ Controlling User Access
- ✓ Privileges
- ✓ System Privileges
- ✓ Creating Users
- ✓ User System Privileges
- ✓ Granting System Privileges
- ✓ What is a Role?
- ✓ Creating and Granting Privileges to a Role
- ✓ Changing Your Password
- ✓ Object Privileges
- ✓ Granting Object Privileges
- ✓ Using the WITH GRANT OPTION and PUBLIC Keywords
- ✓ Confirming Privileges Granted
- ✓ How to Revoke Object Privileges
- ✓ Revoking Object Privileges
- ✓ Database Links

Enhancements to the GROUP BY Clause

- ✓ Review of Group Functions
- ✓ Review of the GROUP BY Clause
- ✓ Review of the HAVING Clause
- ✓ GROUP BY with ROLLUP and CUBE Operators
- ✓ ROLLBACK Operator
- ✓ ROLLBACK Operator Example
- ✓ GROUPING Function
- ✓ GROUPING Function: Example

- ✓ GROUPING SETS
- ✓ GROUPING SETS: Example
- ✓ Concatenated Groupings
- ✓ Concatenated Groupings Example

Extensions to DML and DDL Statements

- ✓ Review of the INSERT Statement
- ✓ Review of the UPDATE Statement
- ✓ Multitable INSERT Statements
- ✓ Unconditional INSERT ALL
- ✓ Conditional INSERT ALL
- ✓ Conditional FIRST INSERT
- ✓ CREATE INDEX with CREATE TABLE Statement

PL/SQL Oracle certification syllabus

Part I: Programming in PL/SQL

- ✓ Introduction to PL/SQL
- ✓ What Is PL/SQL?
- ✓ The Origins of PL/SQL
- ✓ About PL/SQL Versions
- ✓ Resources for PL/SQL Developers

Creating and Running PL/SQL Code

- ✓ SQL*Plus
- ✓ Performing Essential PL/SQL Tasks
- ✓ Calling PL/SQL from Other Languages
- ✓ Language Fundamentals
- ✓ PL/SQL Block Structure

- ✓ The PL/SQL Character Set
- ✓ Identifiers
- ✓ Literals
- ✓ The Semicolon Delimiter
- ✓ Comments
- ✓ Part II: PL/SQL Program Structure
- ✓ Conditional and Sequential Control
- ✓ IF Statements
- ✓ CASE Statements and Expressions
- ✓ The NULL Statement
- ✓ Iterative Processing with Loops

Loop Basics

- ✓ The Simple Loop
- ✓ The WHILE Loop
- ✓ The Numeric FOR Loop
- ✓ The Cursor FOR Loop
- ✓ Loop Labels
- ✓ Tips for Iterative Processing

Exception Handlers

- ✓ Exception-Handling Concepts and Terminology
- ✓ Defining Exceptions
- ✓ Raising Exceptions
- ✓ Handling Exceptions
- ✓ Building an Effective Error Management Architecture
- ✓ Making the Most of PL/SQL Error Management
- ✓ Part III: PL/SQL Program Data
- ✓ Working with Program Data
- ✓ Naming Your Program Data
- ✓ Overview of PL/SQL Datatypes
- ✓ Declaring Program Data
- ✓ Programmer-Defined Subtypes
- ✓ Conversion Between Datatypes

Strings

- ✓ String Datatypes

- ✓ Working with Strings
- ✓ String Function Quick Reference

Numbers

- ✓ Numeric Datatypes
- ✓ Number Conversions
- ✓ Numeric Functions

Collections

- ✓ Collections Overview
- ✓ Part IV: SQL in PL/SQL
- ✓ DML and Transaction Management
- ✓ DML in PL/SQL
- ✓ Transaction Management

Data Retrieval

- ✓ Cursor Basics
- ✓ Working with Implicit Cursors
- ✓ Working with Explicit Cursors
- ✓ Cursor Variables and REF CURSORs
- ✓ Cursor Expressions

Procedures, Functions, and Parameters

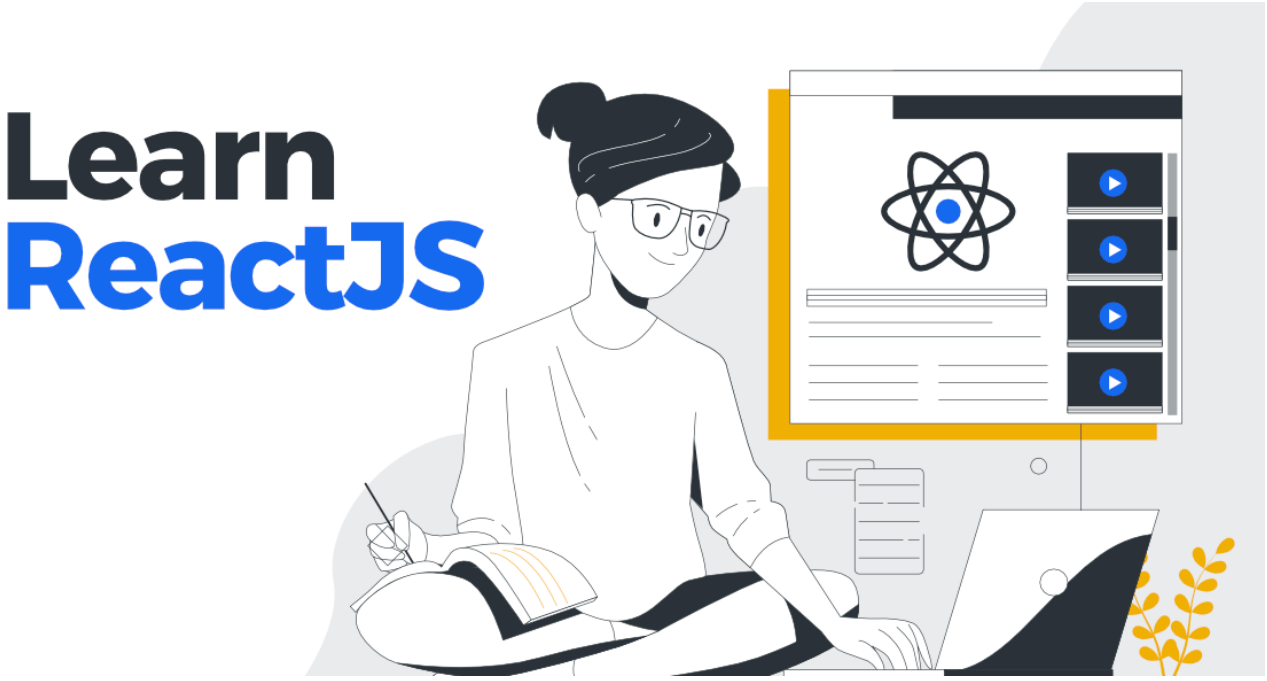
- ✓ Procedures
- ✓ Functions
- ✓ Parameters
- ✓ Local Modules
- ✓ Module Overloading
- ✓ Forward Declarations
- ✓ Advanced Topics

Triggers

- ✓ DML Triggers
- ✓ DDL Triggers
- ✓ Database Event Triggers

✓ INSTEAD OF Triggers

FRONT END (WEB DESIGNING)



Front END **React JS** Training

With REDUX & **JavaScript**

JAVA | SPRING| HIBERNATE| BIG DATA| HADOOP| DATA ANALYTICS| WEB DEVELOPMENT| ANDROID| ANGULAR JS|CC++| DATA STRUCTURE| SQL DBMS | MACHINE LEARNING| DATA SCIENCE WITH PYTHON | PLACEMENT PROVIDER

REACT JS FRONT END Developer (JavaScript)
Job Oriented Course for Job Switch/Fresher's/Expeienced/Non-Tech people

- Starting from HTML
- CSS
- JavaScript Core for Client side validation
- JavaScript Advance with Object Oriented Programming
- Bootstrap Framework for Responsive web apps
- jQuery for Animation and Slider in Websites
- Ajax (Asynchronous JavaScript)
- ECMA Script (ES6) before React JS
- React JS

- Landing Home page creation for websites
- Project Development Training on React JS
- Internship on JavaScript & React JS Real time applications
- Knowledge of Project Development tools
  - Sublime Text 3
  - Notepad++
- Live Project Server hosting for REACT JS web apps
- Technical Interview Preparation for REACT JS web development
- Written Guaranteed Placement Support after training completion
- Technical Resume/CV creation

Web Designing

HTML

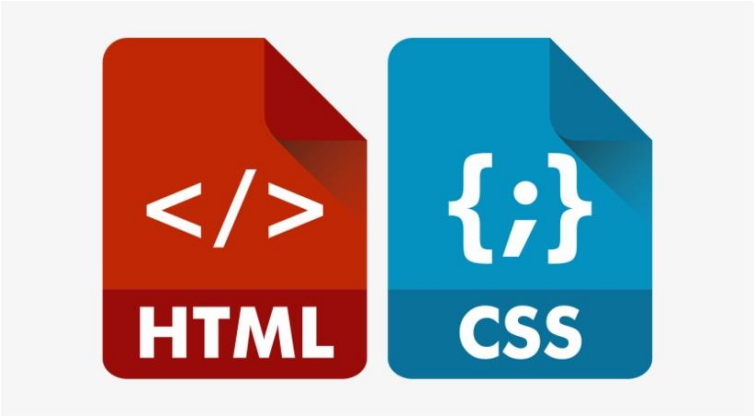
- ✓ Introduction of different Web Technology
- ✓ Introduction
- ✓ HTML Elements
- ✓ HTML Attributes
- ✓ HTML Headings
- ✓ HTML Paragraphs
- ✓ HTML Formatting
- ✓ HTML Fonts
- ✓ HTML Styles
- ✓ HTML Links
- ✓ HTML Images
- ✓ HTML Tables
- ✓ HTML Lists
- ✓ HTML Forms
- ✓ HTML Frames
- ✓ HTML Iframes
- ✓ HTML Colors
- ✓ HTML Colornames
- ✓ HTML Colorvalues
- ✓ HTML Quick List

- ✓ HTML Layout
- ✓ HTML Doctypes
- ✓ HTML Head
- ✓ HTML Meta
- ✓ HTML Scripts
- ✓ HTML Entities
- ✓ HTML URLs
- ✓ HTML URL Encode
- ✓ HTML Media
- ✓ HTML Audio
- ✓ HTML Object
- ✓ HTML Video
- ✓ HTML YouTube
- ✓ HTML Media Tags
- ✓ HTML Summary

CSS

- ✓ Introduction
- ✓ CSS Syntax
- ✓ CSS Id & Class
- ✓ CSS Styling
- ✓ Styling Backgrounds

- ✓ Styling Text

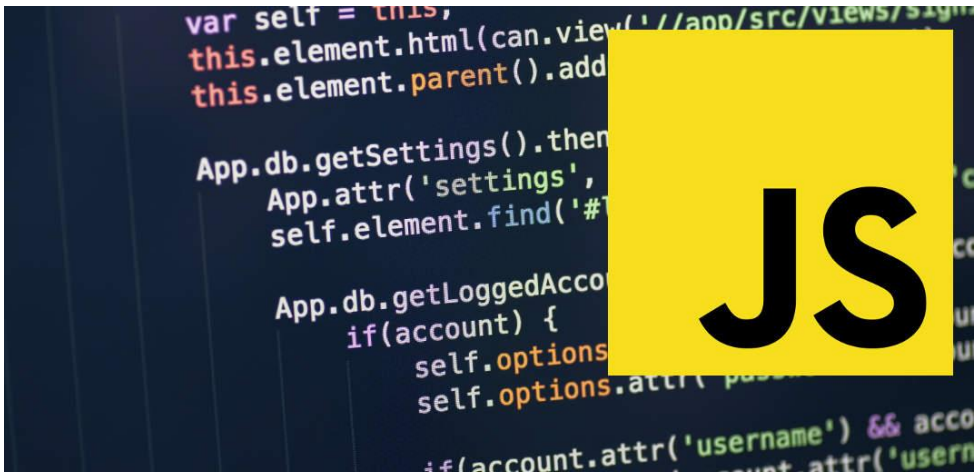


- ✓ Styling Fonts
- ✓ Styling Links
- ✓ Styling Lists
- ✓ Styling Tables
- ✓ CSS Box Model
- ✓ CSS Box Model
- ✓ CSS Border
- ✓ CSS Outline
- ✓ CSS Margin
- ✓ CSS Padding
- ✓ CSS Advanced
- ✓ CSS Grouping/Nesting
- ✓ CSS Dimension
- ✓ CSS Display
- ✓ CSS Positioning
- ✓ CSS Floating
- ✓ CSS Align
- ✓ CSS Navigation Bar
- ✓ CSS Image Gallery
- ✓ CSS Image Opacity
- ✓ CSS Image Sprites
- ✓ CSS Media Types

JavaScript

Why Learn JavaScript?

JavaScript is among the most powerful and flexible programming languages of the web. It powers the dynamic behavior on most websites, including this one.



Take-Away Skills:

You will learn programming fundamentals and basic object-oriented concepts using the latest JavaScript syntax. The concepts covered in these lessons lay the foundation for using JavaScript in any environment.

Up Next:

After learning JavaScript basics (up through the Objects lesson), try applying JavaScript to:

- Build Interactive Websites
- Build Animated websites & Animated applications
- Build web applications with Angular js after JavaScript

Module 1: Introduction to JavaScript-

- Introduction of client side script
- Introduction of javascript
- Cross browser issues.
- Declaration syntax of javascript
- Statements
- Comments
- Popup Boxes
- Alert
- Confirm
- Prompt
- Variables, Arrays and Operators
- Variables
- Operators
- Arithmetic
- Assignment
- Comparison
- Logical

Module 2: JavaScript Basics

- The HTML DOM
- JavaScript Syntax
- Literals, identifiers and reserved words
- Basic Rules
- Dot Notation

- Square Bracket Notation
- Expressions and expression evaluation
- JavaScript Objects, Methods and Properties
- Statements

Module 3: JavaScript Core

- Objects and Arrays
- JSON
- Functions
- Scope
- Closure

Module 4: Variables, Arrays and Operators

- JavaScript Variables
- Working with Numbers and Strings
- A Loosely-typed Language
- Storing User-Entered Data
- Arrays
- Associative Arrays
- Array Properties and Methods
- JavaScript Operators

Module 5: JavaScript Functions

- Built-in Functions
- User-defined Functions
- Function Syntax

- Passing Values to Functions
- Returning Values from Functions
- Built-In JavaScript Objects
- Typeof Operator

Module 6: Conditionals and Loops

- if - else if - else Conditions
- Switch / Case
- while Loop Syntax
- do...while Loop Syntax
- for Loop Syntax
- for...in Loop Syntax

Module 7: Advanced JavaScript

- Date object
- This object
- Event object
- State management
- Cookie
- Form validation
- Expressions
- Email validation
- Dynamic functionalities of html controls

Module 8: JavaScript and the Browser

- The Implicit window Object
- The getElementById() Method
- The getElementsByTagName() Method
- The getElementsByClassname() method
- The querySelector() and querySelectorAll() methods
- Event Handlers
- Creating, Inserting and Deleting Nodes
- Element Position Manipulation
- Scrolling
- Manipulating CSS
- Scripting Inline Styles
- Scripting CSS Classes
- Scripting Style Sheets
- Working with cookies

Module 9: JavaScript Form Validation

- Accessing Form Data
- Basics of Form Validation
- The this Object
- Validating Radio Buttons
- Validating Checkboxes
- Validating Select Menus
- Focus, Blur, and Change Events
- Focus and Blur
- Validating Textareas

Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development.

Module 1:-Introduction to Bootstrap

In this module, you will learn about Bootstrap Introduction, how to design web page look and feel good by using Bootstrap and the basics of Bootstrap Framework using which you can create web projects

- What is Bootstrap Framework
- Why Bootstrap
- History of Bootstrap
- Advantages of Bootstrap Framework
- What is Responsive web page
- How to remove Responsiveness
- Major Features of Bootstrap

- What is Mobile-First Strategy
- Setting up Environment
- How to apply Bootstrap to Applications

Module 2:- Bootstrap Grid

In this module, you will learn about the Bootstrap Grids in web design organize and structure content, makes the websites easy to scan and reduces the cognitive load on users. How to create page layouts through a series of rows and columns that house your content and how the Bootstrap grid system works

- What is Bootstrap Grid
- How to apply Bootstrap Grid
- What is Container
- What is Offset Column
- How to Reordering Columns
- Advantages of Bootstrap Grid
- How to Display responsive Images
- How to change class properties
- How to use readymade themes
- How to customize Bootstrap's components, Less variables, and jQuery plug-in.
- What is Bootstrap Typography
- How to use Typography
- What is Bootstrap Tables
- What is Bootstrap Form Layout
- What is Bootstrap Button
- How display images in different styles like Circle shape etc
- How to display text like muted and warning etc

- What is Carets Classes
- How to hide or show the text in Bootstrap

Module 3:- Bootstrap Components

In this module, you will get knowledge on over a dozen reusable components built to provide iconography, dropdowns, input groups, navigation, alerts, and much more. Advantages of button groups and toolbars and how to use that

- What is Bootstrap Components
- Why Bootstrap Components
- Advantages of Bootstrap Components
- What are the different types of Bootstrap Components
- What is Glyphicons Component
- How to use Glyphicons Component
- What is Bootstrap Dropdown Menu Component
- What is Button Groups and Button Toolbar
- How to use Button Groups and Button Toolbar
- What are different Input Groups Components
- What is Navigation Pills & Tabs Components
- How to use Navigation Pills and Tabs Components
- What is Navbar Component
- How to build a Responsive Navbar
- How to Add Forms and other controls to Navbar
- How to Fix the position of navbar
- What is Breadcrumb Component
- What is Pagination Component
- How to apply Pagination in Application
- What is Labels / Badge Components

- What is Jumbotron / Page Header Components
- What is Thumbnail Component
- What is Alerts & Dismissible Alerts
- How to Create Progress Bar
- What is Media Objects Component
- Why Media Objects Component
- How to use Media Objects Component
- What is Bootstrap List Group Component
- What is Bootstrap Panel Component

jQuery

Introduction to jQuery

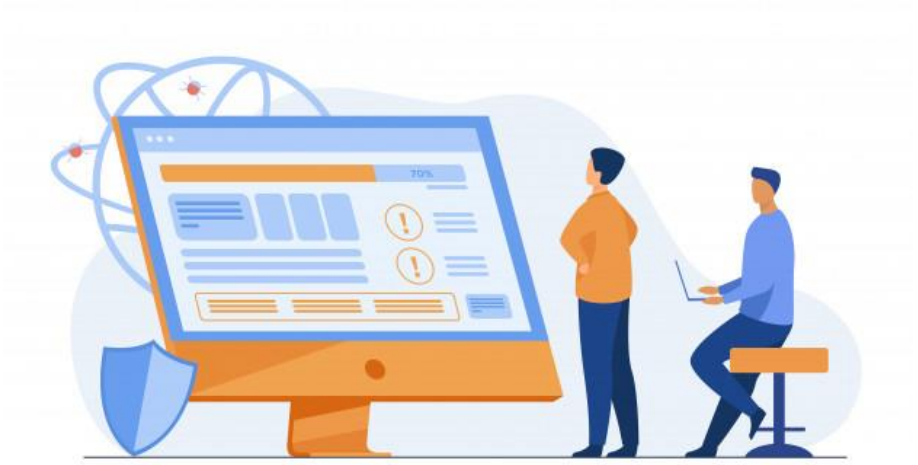
- jQuery Syntax
- jQuery Selectors
- jQuery Events
- jQuery Effects
- jQuery HTML
- jQuery Traversing

- Traversal
- Traversal Functions
- jQuery AJAX & Misc
- Showing/Hiding elements
- Sliding elements
- Fading elements
- Deleting animation elements

React JS Syllabus

Starting from ES6 & JavaScript Tour before React JS:

- History of Javascript
- Object based JavaScript
- Introduction to ES6
- Arrow Function
- What is ES6
- A word on bable
- Block scope, let &const



- Template literals
- Arrow functions
- Spread and Rest operators
- Object literal improvements
- Destructuring
- Classes
- Inheritance
- Static properties and methods

- Promises
- Iterators and Iterables
- Generators
- Modules

1. Introduction to React JS:

Learning Objective:

Understand how react makes things perform, learn how to set up, run and debug a react app.

- What is React?
- Why React?
- React version history
- React 16 vs React 15
- Just React - Hello World
- Using create-react-app
- Anatomy of react project
- Running the app
- Debugging first react app

Hands-on: Install create-react-app and create a new react project.

2. Basic Features of React JS

Learning Objective: React JS and features.

- React Concepts
- JSX
- TSX
- Render elements

- Components and props
- State and Lifecycle
- Handling events
- Project on above topics

3. Templating using JSX

Learning Objective:

Understand the significance of JSX and know its syntax and features.

- Working with React. createElement
- Expressions
- Using logical operators
- Specifying attributes
- Specifying children
- Fragments

Hands-on: Create JSX expressions with different javascript expression, apply css via className and styles, use conditionals.

4. React JS important components

Learning Objective: Understand the significance of component architecture and learn how to decompose UI into components and compose them back to make UI.

- Significance of component architecture
- Types of components
- Functional
- Class based
- Pure

- Component Composition

Hands-on: Create class based and functional components.

5. Working with state and props

Learning Objective: Learn how to manage state in class based react components and how to make communication between components using props.

- What is state and it significance
- Read state and set state
- Passing data to component using props
- Validating props using propTypes
- Supplying default values to props using defaultProps

Hands-on: Create a stateful component and stateless component. Pass data from parent component to child component using props. Implement child to parent communication using callbacks.

6. Rendering lists

Learning Objective:

Learn how to render lists and use key prop.

- Using react key prop
- Using map function to iterate on arrays to generate elements

Hands-on: Create component which renders lists iteratively using map function of array

7. Event handling in React

Learning Objective:

Learn about React's synthetic event system and its working.

- Understanding React event system
- Understanding Synthetic event
- Passing arguments to event handlers

Hands-on: Handle different synthetic events.

8. Understanding component lifecycle and handling errors

Learning Objective:

Understand the significance of lifecycle methods and application in real time use cases. Also learn how to handle errors declaratively.

- Understand the lifecycle methods
- Handle errors using error boundaries

Hands-on: Create a stateful component and implement lifecycle methods. Implement try catch mechanism using error boundaries.

9. Working with forms

Learning Objective:

Understand how to handle forms in react.

- Controlled components
- Uncontrolled components

- Understand the significance to default Value prop
- Using react ref prop to get access to DOM element

Hands-on: Create a component that uses different form controls.

10. Context

Learning Objective:

Understand how to work with global state using context API.

- What is context
- When to use context
- Create Context
- Context.Provider
- Context.Consumer
- Reading context in class

Hands-on: Create components that get applied with multiple themed styles using context to store theme info globally and apply to all components

11. Hooks

Learning Objective:

Understand the need for hooks and implement hooks to access state and effects hook in functional components.

- What are hooks
- Why do you need hooks
- Different types of hooks
- Using state and effect hooks
- Rules of hooks

Hands-on: Create a functional component that uses the ability of state and life cycle features

12. Routing with react router

Learning Objective:

Understand the significance of routing, configure routing for SPA.

- Setting up react router
- Understand routing in single page applications
- Working with BrowserRouter and HashRouter components
- Configuring route with Route component
- Using Switch component to define routing rules
- Making routes dynamic using route params
- Working with nested routes
- Navigating to pages using Link and NavLink component
- Redirect routes using Redirect Component
- Using Prompt component to get consent of user for navigation
- Path less Route to handle failed matches

Hands-on: Install and setup router, configure routing rules, implement declarative and imperative navigation.

13. Just Redux

Learning Objective:

Understand how to manage state in just redux in plain vanilla JS app.

- What is redux
- Why redux
- Redux principles
- Install and setup redux
- Creating actions, reducer and store

Hands-on: Create actions, reducer and store. Dispatch actions and subscribe to store changes.

14. Immutable.js

Learning Objective: Understand the challenges of mutability and how immutable.js helps over the mutability challenges.

- What is Immutable.js?
- Immutable collections
- Lists
- Maps
- Sets

Hands-on: Create immutable List, map and set. Perform CRUD operations.

15. React Redux

Learning Objective: Understand how to integrate redux into react application.

- What is React Redux
- Why React Redux
- Install and setup
- Presentational vs Container components
- Understand high order component
- Understanding mapStateToProps and mapDispatchtToProps usage

Hands-on: Install and setup react redux. Configure Provider component as top level component. Migrate react statefull component to connected component.

16. Redux middleware

Learning Objective: Understand the significance of middleware and learn how saga middleware works.

- Why redux middleware
- Available redux middleware choices
- What is redux saga
- Install and setup redux saga
- Working with Saga helpers

- Sagas vs promises

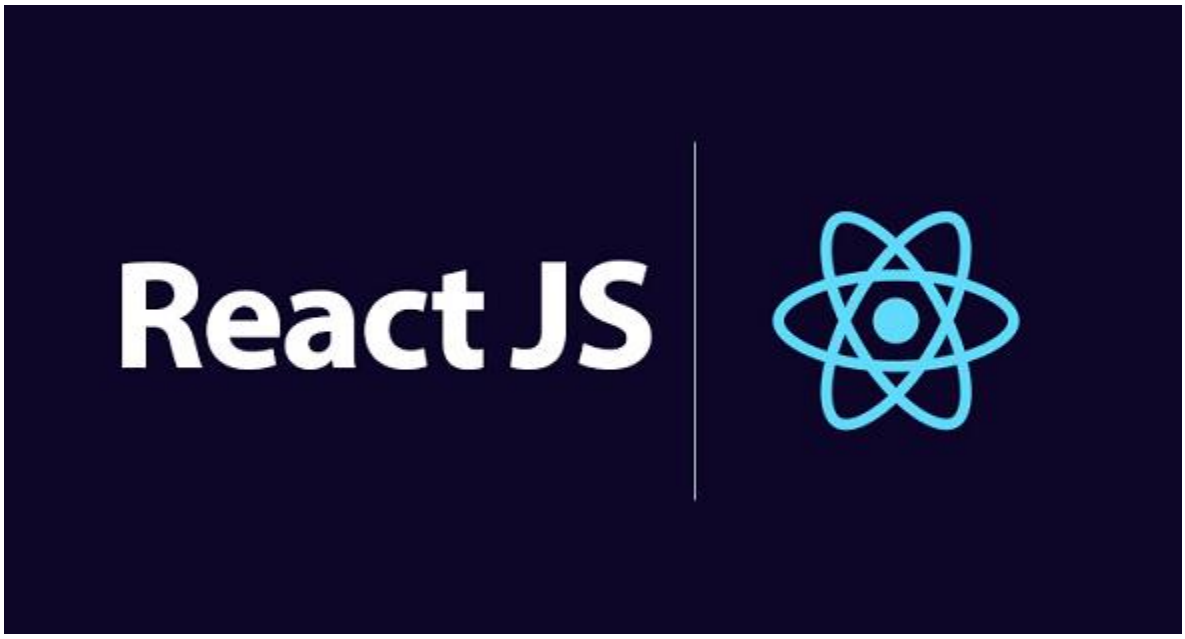
Hands-on: Install and setup logger and saga middleware. Develop sagas use different side effects. Code sagas to fetch data from remote API using fetch/axios.

17. Unit Testing

Learning Objective: Understand significance of UI testing and learn how to unit test components, reducers using jest and enzyme.

- Understand the significance of unit testing
- Understand unit testing jargon and tools
- Unit testing react components with Jest
- Unit testing react components with enzyme

Hands-on: Understand jest configuration. Install and setup enzyme. Write unit test to components and reducers.



Why Students join PROGRAMMERS POINT for Job Oriented PYTHON Training in Indore

PROGRAMMERS POINT provides complete flexibility to the students, to explore the technology and learn based on real-time projects. Our trainers help the candidates for completing their projects and even prepare them for **PYTHON** interview questions and answers. Candidates are free to ask any questions at any time

- ✓ More than **15+ Years** of Experience.in IT/Software training & Placements
- ✓ Trained more than **1 Lac+** students
- ✓ Strong Theoretical & Practical Knowledge.
- ✓ Certified Professionals experienced in IT companies
- ✓ Well connected with Hiring HRs in multinational IT companies.
- ✓ Expert level technical Knowledge and fully up-to-date on real-world IT software applications.
- ✓ Trainers have experienced on multiple real-time **PYTHON** projects.
- ✓ Programmers Point having a team of **PYTHON** Experts who are currently working in Top MNC with Extensive Knowledge on **PYTHON** and **DJANGO**. Our Trainers will guide you to learn the **DJANGO** Framework Step by Step with unique Course Syllabus. End of this **PYTHON** Course you will get a good exposure on using **DJANGO** framework for developing dynamic Web applications.

an initiative by affimintus
Programmers Point SM
Earning Trust In Education

DETAILS-

Total duration	6-8 months
Weekly Days	6 days
Application development	At the end of module
Trainer Profile	Experienced PYTHON Developer
Live Project Development	Live Project application
Placement Location	Indore or anywhere in INDIA
Best For	Non Tech (MBA/ MECH/COMMERCE/CS) students

Upcoming Centers in India-

1. UJJAIN (NOW IN UJJAIN, FREEGANJ)
2. VADODARA
3. JAIPUR
4. RAIPUR
5. PUNE
6. JABALPUR
7. GWALIOR
8. AND MANY MORE IN COUNT….

1. Training by Professional Python Developers
2. After Training 100% Guaranteed placement Interviews
3. Complete Python Interview Preparation
4. Everyday 6 hours training with Practical's
5. Live Programming concepts Implementation
6. Institute running by ORACLE CERTIFIED people
7. Central India's No. 1 IT/SOFTWARE training institute
8. Indore's most loved Programming training institute
9. Highly rated institute by **Google**
10. Just Dial **1st Ranking Institute** for Java/Python training
11. And many more….Don't waste time..

Join Now for Placements