Python FSD

HTML 5, CSS 3 with Bootstrap

HTML 5

HTML Basics

- Understand the structure of an HTML page.
- New Semantic Elements in HTML 5
- Learn to apply physical/logical character

Tables

- Understand the structure of an HTML table.
- Learn to control table format like cell spanning, cell spacing, border

List

- Numbered List
- Bulleted List

Working with Links

- Understand the working of hyperlinks in web pages.
- Learn to Create hyperlinks in web pages.
- Add hyperlinks to list items and table contents

Image Handling

- Understand the role of images in web pages
- Learn to add images to web pages
- Learn to use images as hyperlinks

Frames

- Understand the need for frames in web pages.
- Learn to create and work with frames.

HTML Forms and User Input

- Understand the role of forms in web pages
- Understand various HTML elements used in forms.
- Single line text field
- Text area

- Check box
- Radio buttons
- Password fields
- Pull-down menus
- File selector dialog box

New Form Elements

- Understand the new HTML form elements such as date, number, range, email, search and data list
- Understand audio, video, article tags

CSS3

Introduction to Cascading Style Sheets 3.0

- What CSS can do
- CSS Syntax
- Types of CSS

Working with Text and Fonts

- Text Formatting
- Text Effects
- Fonts

CSS Selectors

- Type Selector
- Universal Selector
- ID Selector
- Class selector

Colors and Borders

- Background
- Multiple Background
- Colors RGB and RGBA
- HSL and HSLA
- Borders
- Rounded Corners
- Applying Shadows in border

Bootstrap

Introduction to Bootstrap

- Introduction
- Getting Started with Bootstrap

Bootstrap Basics

- Bootstrap grid system
- Bootstrap Basic Components

Bootstrap Components

- Page Header
- Breadcrumb
- Button Groups
- Dropdown
- Nav & Navbars

JavaScript

- ES6
- Var, Let and Const keyword
- Arrow functions, default arguments
- Template Strings, String methods
- Object de-structuring
- Create,apply,prototype,bind method
- Spread and Rest operator
- Typescript Fundamentals
- Types & type assertions, Creating custom object types, function types
- Typescript OOPS - Classes, Interfaces, Constructor, etc
- Decorator & Spread Operator
- Difference == & ===
- Asynchronous Programming in ES6
- Promise Constructor

- Promise with Chain
- Promise Race

Core Python + PDBC

Python Introduction

- What's Python?
- Why do people use Python?
- Some quotable quotes
- A Python history lesson
- Advocacy News
- What's Python good for?
- What's Python not good for?
- The features list
- Python portability
- Summary

Using the Interpreter

- Python's Interactive Prompt
- Scripting
- Program Execution Model
- Program Architecture: modules
- How to run Python programs
- Using Python IDEs

working with Variables in Python

- Python Variables
- Naming Conventions & Rules
- Types as Objects
- Variable References & Garbage Collection
- Sequence Types
- Membership Statements
- List Iteration
- Sequence Assignments
- Mutable vs Immutable Objects
- Multi Target Assignments

Numeric Operations in Python

- More About Python's Numeric Types
- Numeric Tools
- The Decimal Module
- Operator
- Arithmetic
- Logical
- Relational
- Bitwise

- Common Tuples Methods
- Tuples Operations
- Tuples Indexing
- Tuples Slicing
- Tuples Iteration
- Multi-Dimensional Tuples (Matrices)

Python's Lists

- Common List Methods
- The range() Function
- List Operations
- String Indexing
- String Slicing
- String Iteration
- Multi-Dimensional Lists (Matrices)

Python List Comprehension

- Basic List Comprehensions
- Compound List Comprehensions

Python set data type

- Understanding & using set data type

Python Dictionaries

- Python Dictionaries
- Assigning Values to Dictionaries
- Dictionary Methods
- Dictionaries vs Lists & Tuples
- Dictionary Indexing
- Dictionary Iteration

Basic Input/Output with Files

- Opening Files
- Working with Files
- Controlling Output Location

Creating Python Functions

- Function Basics
- Defining Functions
- Function Polymorphism
- Argument Defaults
- Lambdas
- Local Variables
- Understanding __built-in__
- Preventing Variable Modifications
- Argument Matching Methods
- Keyword Argument Method

Classes and Objects

- Introduction to OOP using python
- Classes and class attributes
- Instances and instance attributes
- Binding and method invocation
- Composition, Sub-classing and Derivation
- Inheritance
- Built-in functions for classes, instances and other objects
- Privacy and Delegation
- An overview of built-in python classes and modules

Modules & Packages

- Module Basics
- Packages
- Package Creation and Importing
- Using __all__ and _ Variables
- Using __name__
- Using third party modules

Exceptions

- About Exceptions

- Learning how exceptions work in Depth
  - Handling exceptions
  - Raising exceptions
  - Catching exceptions
- Python's Default Exception Handler
- Using Try/Except/Else/Finally Exceptions
- Generating User Defined Exceptions
- Using Asserts
- Exception Classes

Regular Expression in Python

- Using the re module
- Searching with regular expressions
- Replacing with regular expressions
- Reusing regular expressions with recompile
- The match Function
- The search Function
- Regular-expression patterns

Multi-threaded Programming

- Starting a New Thread
- The Threading Module
- Creating Thread Using Threading Module

Database & SQL

Introduction

- The Relational Model

Understanding Basic SQL Syntax

- The Relational Model
- Basic SQL Commands - SELECT
- Basic SQL Commands - INSERT
- Basic SQL Commands - UPDATE
- Basic SQL Commands – DELETE

Querying Data with the SELECT Statement

- The SELECT List
- SELECT List Wildcard (*)
- The FROM Clause
- How to Constrain the Result Set
- DISTINCT and NOT DISTINCT

Filtering Results with the Where Clause

- WHERE Clause
- Boolean Operators
- The AND Keyword
- The OR Keyword

- Other Boolean Operators BETWEEN, LIKE, IN, IS, IS NOT

Shaping Results with ORDER BY and GROUP BY

  - ORDER BY
  - Set Functions
  - Set Function And Qualifiers
  - GROUP BY
  - HAVING clause

PDBC

Introduction to JDBC

  - Introduction to PDBC
  - Architecture of PDBC

Performing Basic CRUD operations using JDBC

DJANGO / Flask

Installation and Introduction to Django

  - Learn about python programming (inbuilt data structure & Object oriented
  - Programming)

- Learn and explore various web dev tools like VsCode Editor, Git & Github,
- Sqlite3, Postman, Virtual Environment, DBeaver, and Other Essential Tools
- Install Django framework and it's dependencies
- Setup Django environment
- Create your first sample Django project
- Learn about project structure in Django

Django Admin, Commands and Shell

- Acquire the knowledge to reuse initial built-in Django applications
- Learn about Django admin
- Learn about various Django Commands
- Learn about Django Shell

Routing and Views in Django

- Map web URLs to view functions
- Familiarize with various HTTP methods, including GET, POST, PUT/PATCH, and DELETE

- Create simple views in Django with HTTP response or JSON response and learn about different status codes

Typescript & Angular 6

Introduction to Angular Framework

- Introduction to Angular Framework, History & Overview
- Environment Setup, Angular CLI, Installing Angular CLI
- NPM commands & package.json
- Bootstrapping Angular App, Components, AppModule
- Project Setup, Editor Environments
- First Angular App & Directory Structure
- Angular Fundamentals, Building Blocks

Essentials of Angular

- Component Basics
- Setting up the templates
- Creating Components using CLI
- Nesting Components

- Data Binding - Property & Event Binding, String Interpolation, Style binding
- Two-way data binding
- Input Properties, Output Properties, Passing Event Data

Templates, Styles & Directives

- Template, Styles, View Encapsulation, adding bootstrap to angular app
- Built-in Directives, Creating Attribute Directive
- Using Renderer to build attribute directive
- Host Listener to listen to Host Events
- Using Host Binding to bind to Host Properties

Pipes, Services & Dependency Injection

- In-built Pipes, Creating a Custom Pipes
- Services & Dependency Injections
- Creating Data Service
- Understanding Hierarchical Injector

Template-Driven and Reactive Forms

- Template-Driven vs Reactive Approach

- Understanding Form State
- Built-in Validators & Using HTML5 Validation
- Grouping Form Controls
- FormGroup, FormControl, FormBuilder
- Forms with Reactive Approach
- Predefined Validators & Custom Validators
- Showing validation errors

Components Deep Dive / Routing

- Component Life Cycle Hooks
- Reusable components in angular using <ng-content>
- Navigating with Router links
- Understanding Navigation Paths
- Navigating Programmatically
- Passing Parameters to Routes
- Passing Query Parameters and Fragments
- Setting up Child (Nested) Routes
- Outsourcing Route Configuration (create custom module)

Http Requests / Observables

- HTTP Requests
- Sending GET Requests
- Sending a PUT Request
- Using the Returned Data
- Catching Http Errors
- Basics of Observables & Promises

For More :