



**Bundle Course - Full Stack Web Development  
(Java - JS - MongoDB - ReactJS - HTML5 - CSS3)  
Course Curriculum**

**JavaScript**

**Course Content**

***Introduction to JavaScript***

- Introduction to Scripting
- JavaScript Characteristics
- JavaScript and Common Programming Concepts
- JavaScript Flavors and Versions
- Technologies Based on JavaScript

- Server-Side vs. Client-Side Applications
- Annotating Your Code with Comments
- Expanding the Role of JavaScript

***Working with Variables and Data in JavaScript***

- Using JavaScript to Communicate with the User
- Using Data More Than Once: Variables
- JavaScript Operators
- JavaScript Expressions
- JavaScript Alerts and Prompts
- Inline Scripting, Simple User Events and Basic Event Handlers
- JavaScript Keywords and Reserved Words

***Functions, Methods and Events in JavaScript***

- Introduction to Functions
- Defining a Function
- Calling a Function
- Methods as Functions
- Types of Functions
- User Events and JavaScript Event Handlers

***Debugging and Troubleshooting JavaScript***

- What Is Debugging?
- Tools for Debugging Code
- Testing Code in Various Browsers
- Debugging Logic Errors

- JavaScript and Mobile Devices

***Controlling Program Flow in JavaScript***

- Controlling Decisional Program Flow
- The if...else Statement
- The while Statement
- The do...while Statement
- The for Statement
- The break Statement
- The continue Statement
- The switch Statement

***The JavaScript Document Object Model (DOM)***

- The JavaScript Document Object Model (DOM)
- The window Object
- The document Object
- The image Object
- The history Object
- The location Object
- The navigator Object
- The Virtual DOM

***JavaScript Language Objects***

- Introduction to JavaScript Language Objects
- The String Object
- Evaluating Strings

- JavaScript Regular Expressions
- The Array Object
- The for Each Method
- The Map Method
- The Date Object
- Setting and Extracting Time Information
- The Math Object

***Custom JavaScript Objects***

- Creating Custom Objects
- Creating a JavaScript Object: The Constructor
- Creating an Instance of a Custom Object
- Creating Object Methods
- Creating Functions for Your Objects
- Concept of Classes
- Iterators and Generators
- Call, apply and bind

***Changing HTML on the Fly***

- Changing HTML on the Fly
- Using the getElementById Method
- Using the getElementsByTagName Method
- Using the getElementsByTagName Method
- Using the getElementsByTagName Method
- Modifying Attributes within the DOM

- Appending Text to the DOM

***Developing Interactive Forms with JavaScript***

- Interactive Forms
- Overview of Form Elements
- Referring to Form Elements
- The form Object
- The input Elements and text Objects
- The text area object
- The select Object
- Form Validation
- Common form Security Errors

***JavaScript Security***

- Introduction to JavaScript Security Issues
- Browser vs. Operating System
- Browser-Specific Security Issues
- Browser Compatibility and Client-Side Detection
- Script Blocking
- Differences in document write Among Browsers
- Malicious and Accidental Coding
- Frame-to-Frame URL Changing
- Signed Scripts
- Cross-Site Scripting (XSS)
- Cookies and Security in JavaScript

- Ethics in Collecting, Storing, Using and Protecting User Data
- Creating Secure JavaScript Code

***JavaScript Libraries and Frameworks***

- Introduction to JavaScript Libraries and Frameworks
- Choosing a Code Library
- External and Pre-made Scripts
- Loading Your First Library
- Using JavaScript Library Plug-ins
- Using Frameworks
- Best Practices with JavaScript Libraries and Frameworks
- Copyright Issues and JavaScript

***JavaScript and AJAX***

- Introduction to AJAX
- Handling Data and Content Types
- Interactions Among JavaScript, Data and HTML
- Using AJAX Scripts
- AJAX and Servers
- AJAX and Databases
- Usability Issues with AJAX
- Combining AJAX with Libraries

***Introduction to Web APIs***

- Web Application Programming Interface (API)
- The Web Storage API

- The Geo location API
- The Canvas API

### **HTML5 & CSS3**

#### ***HTML5***

##### **Introduction**

- History of HTML
- What you need to do to get going and make your first HTML page
- What are HTML Tags and Attributes?
- HTML Tag vs. Element
- HTML Attributes
- How to differentiate HTML Document Versions

##### **HTML - Basic Formatting Tag**

- HTML Basic Tags
- HTML Formatting Tags
- HTML Color Coding

##### **HTML - Grouping Using Div Span**

The HTML <div> and <span> element is the generic container for flow content and does not inherently represent anything. Use it to group elements for purposes such as styling (using the class or id attributes), and inline level and block level separation.

- Div and Span Tags for Grouping

##### **HTML - Lists**

The HTML <li> element is used to represent an item in a list. It must be contained in a parent

element: an ordered list (<ol>), an unordered list (<ul>), or a menu (<menu>). In menus and unordered lists, list items are usually displayed using bullet points, now we are going to check how to use html list.

- Unordered Lists
- Ordered Lists
- Definition list

#### **HTML - Images**

- Creating an image
- Choosing alt text
- Responsive image using the secret attribute
- Responsive image using picture element

#### **HTML - Hyperlink & Anchors**

- Link to another site
- Link to an anchor Link to a page on the same site Link that dials a number
- Open link in new tab/window
- Link that runs JavaScript Link that runs email client

#### **HTML - Tables**

- Simple Table
- Spanning columns or rows
- Column Groups
- Table with thead, tbody, tfoot
- Heading scope

#### **HTML - iframe**



The HTML <iframe> element represents a nested browsing context, effectively embedding another HTML page into the current page. This module will help you make use of iframe tag.

- Attributes Using
- iframe as the Target

#### **HTML - Forms**

- Submitting
- Target attribute in form tag
- Uploading Files
- Grouping a few input fields

#### **HTML - Headers**

The HTML <head> element provides general information (metadata) about the document, including its title and links to its scripts and style sheets.

- Title
- Base
- Link
- Styles
- Script

#### **HTML - Miscellaneous**

The HTML <Meta> element represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>. This module will help you learn Meta

- Tags, xhtml and deprecated tags
- HTML Meta Tag

- XHTML
- HTML Deprecated Tags & Attributes

### **CSS3**

#### **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 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

**Selectors and Pseudo Classes**

- Attribute Selectors
- The Target Pseudo-Class
- UI Element States Pseudo-Classes
- Negation Pseudo-Class
- Structural Pseudo-Classes

**MongoDB**

***Introduction to NoSQL Architecture with MongoDB***

- What Is MongoDB?
- Downloading the required Software
- Installation and Configuration
- MongoDB Advantages
- MongoDB Data Modelling
- MongoDB Tools, Collection and Documents
- Configuration Files
- Touring the File Structure
- Securing the Installation

***CRUD and the MongoDB Shell***

- Introduction to CRUD
- Introduction to the MongoDB API
- Creating a Database, Collection and Documents

***Data Modelling and Schema Design***

- MongoDB Database References
- Model Tree Structures
- MongoDB Analyzing Queries
- MongoDB Atomic Operations
- MongoDB Map Reduce
- MongoDB Text Search
- MongoDB Regular Expression
- MongoDB Capped Collections
- Data Storage
- Working with Datatypes
- Collections
- Document Datatypes
- Creating \_id Fields

***Querying Data***

- Databases and Collections
- Querying Collections
- Working with Operators
- Referencing a Database
- Querying Dates

***Manipulating Data***

- Inserting Data into Collections

- Updates
- Deletes
- Atomic Operations
- Removing Data
- Capped Collections

***High Performance Options***

- Creating Indexes
- Manipulating Index Behaviour
- Index Properties
- Specialised Index Types
- Replication
- Sharding

***Aggregation Framework***

- Aggregating Results
- Single Purpose Functions
- The Aggregation Pipeline
- Date Aggregation Operators

***Indexing***

- Indexing and Aggregation
- Indexing, query profiling and the query optimizer
- Geospatial Indexes
- Index types, Index Properties
- MongoDB Advanced Indexing
- MongoDB Indexing Limitations
- Aggregation Introduction

***Replication***

- Replication Concept
- Failover & Recovery

***Administration***

- MongoDB Deployment and Cluster setup
- MongoDB GridFS
- Trident Spout
- Working with Replica Sets
- MongoDB Sharding
- MongoDB Create Backup

***Using an API***

- Introducing Drivers
- Java
- PHP
- Ruby
- Python

**JAVA Programming Basics**

**Java Programming Fundamentals course syllabus**

**Java Overview**

- How the Java Environment Works
- HelloWorld Program
- Launch Single-File Source Code
- jShell REPL
- Comments and Terminators
- Identifiers

### **Syntax and Types**

- Java Variables
- var declarations
- Java Types
- Basic Java Types
- Reference types
- Conditional Expressions
- Logical operators
- If statement
- While Statement
- Do Statement
- For statement
- Special flow of control operators
- Switch statement
- Numerical Operators
- Casting
- Strings

### **Classes and Objects**

- What is an Object?
- Objects and Encapsulation
- What is a Class?
- Class Object Relationship
- The Person class
- Variable types
- Constructors
- Working with the class Person

### **Working with Methods**

- Method Definitions

- Method Arguments & Return Types
- Overloading Methods
- Method Invocations
- Defining methods
- Constructors Versus Methods
- toString() method
- Adding behaviour to the class Person
- The PersonApp program

**Class Inheritance**

- Inheritance in Java
- Implementing Inheritance
- Rule for overriding methods
- Rules for Polymorphic variables
- Casting and Inheritance
- The super variable
- Constructors and Inheritance

**Abstract Classes, Class Side Behaviour and Final**

- Abstract Classes
- Abstract Classes in Java
- Defining an Abstract Class
- Extending an Abstract Class
- Using Concrete Subclasses
- Class Side Information
- Class Side Data
- Class Side Behaviour
- Final Keyword

**Java Interfaces and Enumerations**

- What is an Interface in Java?
- Basic Interface Definitions
- Implementing an Interface



- Interface
- Using an Interface in a Contract
- Inheritance by Interfaces and Types
- Classes and multiple Interfaces
- Default Interface Methods
- Static Interface Methods
- Enumeration Support
- Implementing Enumerated Types

### **Packages**

- Packages
- Class-Package Relationship
- Declaring Packages
- Packaging the Person class
- Role of the Classpath
- JAR Files
- Encapsulation and Packages
- Class modifier
- Constructor Modifier
- Variable Modifier
- Method Modifiers
- Package Summary

### **Java 9+ Modules**

- Introduction the JPMS
- Java Platform Module System
- Why we need modules
- Look at what modules are
- How to define a module
- How to create a single module application
- How to link modules

### **Arrays**

- What is a Java Array?
- Creating arrays of Objects
- Accessing Array Elements
- Main method args array
- Short hand from
- Ragged Arrays in Java
- Working with Ragged Arrays
- Implications of Inheritance for Arrays
- Integer Array Example

#### **Java Generics**

- Generics and Basic Types
- Generics and Their Types
- Adding Generics to your classes
- A simple user defined Generic class (the Bag class)
- Type Equality
- Generic Collection Assignment
- Generics and Inheritance

#### **Collections Classes**

- Collections API
- ArrayList
- Interfaces v Concrete Classes
- HashMap
- Iteration and Enumeration
- Queues
- Generics and Collections
- For Loops
- Boxing and Unboxing
- Raw Collections

#### **Java 9+ Immutable Collections**

- Why Immutability?

- Immutability
- List.of Factory method
- Set.of Factory method
- Map.of Factory method
- Immutable Collection Nodes
- Java 10 Enhancement

#### **Error and Exception Handling**

- Errors & Exceptions
- Exception types in Java
- Part of the Exception Hierarchy
- Exception Handling
- Local Handling
- Exception Handling Example
- Passing the Buck
- Try with Resources
- Defining new Exceptions
- Chained Exceptions

#### **Nested / Inner Classes**

- Four types of Nested / Inner Class
- Properties of Member level inner classes
- Properties of Method Inner classes
- Anonymous Method Inner classes
- Java 11 Nest-based Access

#### **Java Functions**

- Functional Programming
- Functional Programming in Java
- Functional Interfaces
- Using Functional Interfaces
- Lambdas in Java
- Closing / Capturing Variables

- Returning Functions
- Combining Functions
- Higher Order Functions
- Defining Lambdas
- Method References

**Java Optional Type**

- Null considered Harmful
- Java Optional Type
- Optional Variables
- Creating Optional values
- Method Summary

**Java 9 Streams**

- Streams
- Streams from Collections
- Terminal / Non Terminal Operators
- Creating a Stream
- Map Operation
- Collectors
- Filter operation
- Sorted operation
- For Each
- Pipelining Operations
- Parallel Streams
- Not just collections

**Files, Paths and IO Streams**

- Introducing NIO.2
- Paths class and Path Interface
- The Files class
- File Attributes
- The File Watcher Service

- IO Streams
- Scanners

### **ReactJS Syllabus**

- **Introduction**
- What is ReactJS?
- Installation or Setup
- Hello World with Stateless Functions
- Absolute Basics of Creating Reusable Components
- Create React App
- Hello World
- Hello World Component
- **Components**
- Creating Components
- Basic Components
- Nesting Components
- Props
- Component states - Dynamic user-interface
- Variations of Stateless Functional Components
- setState pitfalls
- **Using ReactJS with TypeScript**
- ReactJS component written in TypeScript
- Installation and Setup
- Stateless React Components in TypeScript
- Stateless and property-less Components
- **State in React**
- Basic State
- Common Antipattern

- setState()
- State, Events And Managed Controls
- **Props in React**
- Introduction
- Default props
- PropTypes
- Passing down props using spread operator
- Props.children and component composition
- Detecting the type of Children components
- **React Component Lifecycle**
- Component Creation
- Component Removal
- Component Update
- Lifecycle method call in different states
- React Component Container
- **Forms and User Input**
- Controlled Components
- Uncontrolled Components
- **React AJAX call**
- HTTP GET request
- HTTP GET request and looping through data
- Ajax in React without a third party library - a.k.a with Vanilla
- **Communication Between Components**
- Child to Parent Components
- Not-related Components
- Parent to Child Components

- **Stateless Functional Components**
- Stateless Functional Component
- **Performance**
- Performance measurement with ReactJS
- React's diffs algorithm
- The Basics - HTML DOM vs Virtual DOM.
- **Introduction to Server-Side Rendering**
- Rendering components
- **Setting Up React Environment**
- Simple React Component
- Install all dependencies
- Configure webpack
- Configure babe
- HTML file to use react component
- Transpile and bundle your component
- **Using React with Flow**
- Using Flow to check prop types of stateless functional components
- Using Flow to check prop types
- **JSX**
- Props in JSX
- Children in JSX
- **Keys in react**
- Using the id of an element
- Using the array index
- **Higher Order Components**
- Higher Order Component that checks for authentication
- Simple Higher Order Component
- **React with Redux**

**Uplatz**

**training.uplatz.com**

Training Provider for IT and Certification Courses

- Using Connect.

<https://training.uplatz.com>

info@uplatz.com

+44 7836 212635