Git and GitHub, JavaScript, TypeScript and Node.js

OVERVIEW

Git and GitHub – 0.25 days, JavaScript including ES2015 – 2.5 days, TypeScript – 1 day, Node.js – 1.25 days
5 days
7.5 hours/day (excluding breaks)

JavaScript (JS) is the language for scripting web pages – to enable user interactions on a web page, communicate with the backend etc. The latest versions of JavaScript like ES2015 (ES6) have introduced a plethora of great new features that have found adoption in modern frontend and backend frameworks. A good understanding of JS, especially ES2015 features, lays a strong foundation to get started with frameworks like React and Angular, as also Node.js and Express.

TypeScript, a language developed at Microsoft, is a typed superset of JavaScript. It introduces explicit types for variables, eliminating errors due to JavaScript's implicit typing, and improving code quality.

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

The Node.js package ecosystem, npm, is the largest ecosystem of open source libraries in the world. The Node.js built-in modules along with the plethora of third-party modules and frameworks make it a good choice to quickly create backend web applications.

PREREQUISITES

- Sound knowledge of programming (with basic knowledge of algorithms & data structures)
- Working knowledge of HTML, and basic knowledge of CSS
- Knowledge of Object Oriented Programming (OOP) concepts is desirable, but not required

CAPSTONE PROJECT

Add the end of this bootcamp, participants would have built a product catalog application. They shall be provided a backend server. The application will involve communicating with the backend and listing products, adding, editing and removing products, posting product reviews etc.

CHAPTERS AND TOPICS

Git and GitHub (0.25 days)

Introduction to Git and GitHub

- Types of Version Control Systems
- Introduction to Git
- Installation and tooling
- Setting up a Git repository
- Staging files and committing locally
- Setting up remote repository on GitHub
- Pushing to remote and Pulling from it
- Checking status
- Diffing commits
- Branching and merging
- Working on features and in teams

JavaScript including ES2015 (2.5 days)

Quick Introduction to JavaScript Language Fundamentals

- Variables and Primitive Data Types
- Strict Mode Execution
- Variable Scopes, Scope chain
- Using Arrays
- Expressions, Operators and Operator Precedence
- Control flow - Branching and Looping

Introduction to Functions

- Function Declaration and Usage
- Function context (the *this* keyword)
- Handling Variable Number of Arguments
- Inner Functions
- Callbacks - Passing Functions as Arguments
- Returning Functions
- Higher-order Functions and Functional Programming Paradigm
- Closures

Introduction to Objects

- Object Declaration using Literal Syntax
- Accessing Properties and Methods
- Adding and Deleting Properties

The "Class" in JavaScript (ES5)

- The new Operator and the Constructor Function
- Introduction to the Object Prototype and Prototype Chain
- Establishing Inheritance using Function.prototype

Basic Introduction to Built-in Classes and Objects in the JavaScript Language

- Function - Functions as Objects, call(), apply() and bind() as Methods of Functions
- Array methods
- Functional Programming revisited - Array iterator methods
- Date methods
- JSON

Introduction to Objects the Browser Creates

- Window (coverage includes setTimeout and setInterval)
- The Event Loop
- Synchronous (blocking) and Asynchronous (non-blocking) methods
- The document object
- The navigator object
- The location object
- The history object

The Document Object Model (DOM)

- Nodes and the DOM Tree
- Node Relationships and DOM Tree Traversal
- Methods for DOM Manipulation

Event Handling

- Various Browser Events
- Different Ways to Handle Events
- Event Object Properties and Methods

Features of ES2015+

- Installing the Babel transpiler and using it
- Block-level scoping and the use of let, const
- Object and Array Destructuring
- Arrow Functions
- Default Parameters
- Rest and spread operators
- Strings and Templating
- Classes, Class Inheritance
- Modules
- Promises

TypeScript (1 day)

Introduction to TypeScript

- Why Typescript?
- Features of TypeScript
- Installation and getting started
- DefinitelyTyped and using type definition files
- The tsc compiler options and configuration using tsconfig.json file

Basic types

- Primitive types and the any type
- Static type checking and type inference
- Type assertion (type-casting)
- Arrays
- Tuples
- Type Aliases
- Union types
- Intersection types

Functions

- Defining function argument and return types
- Function signatures involving callback functions
- Using interfaces to define function signatures
- Overloaded function types

Interfaces

- Using interface to define structure for an object (properties and methods)
- Extending an interface
- Implementing interfaces in classes

Classes

- Access modifiers – public, protected, private and readonly
- Shortcut syntax for creating properties via constructor

Generics

- Creating a generic class
- Using Generics
- Generic type aliases
- Generic type defaults

Miscellaneous topics

- Creating and using Decorators
- Introduction to creating type definition files
- Using libraries that don't have type definition files

Node.js (1.25 days)

Getting Started

- About Node.js
- Downloading and installing Node.js
- Creating a simple web server
- Node.js architecture

Setting up Development Workflows

- Integrating TypeScript into the development workflow
- Using TypeScript in Node.js (shall be done throughout the development process)

Packages and npm

- Node packages, and npm registry
- Introduction to package.json and npm CLI tool
- Introduction to using npm scripts

How Node works

- Blocking vs Non-blocking I/O
- The event loop
- Error-first callbacks

Modules and Using the Module System

- Overview of Built-in modules
- Using built-in modules
- Creating your own modules and using them
- Using third-party modules from npm registry

Working with the Local System

- Global properties related to the local system
- Working with the Node.js process - the process object
- Basic introduction to the filesystem module
- The path module

Using Express along with Node.js

Building web application using Express

Setting up project structure using Express Generator

Serving static files

Parsing request body

Routing in Express

Setting up and using Middleware

Templating using EJS

Managing DB schemas, validation & CRUD operations in Mongo DB using Mongoose