Fundamentals of Javascript Training

Duration: **3 Day**

BILGINC IT Academy
info@bilginc.com
bilginc.com

Exclusive - Begin your journey in programming by learning coding skills with JavaScript

Fundamentals of JavaScript will equip you with the essential programming skills necessary to excel in the dynamic field of web development. Throughout this course, you will gain proficiencies in using industry-standard tools such as Visual Studio Code to run JavaScript files locally, ensuring you have a strong foundation in practical programming. By the end of this course, you will have a solid understanding of JavaScript programming fundamentals, empowering you to build dynamic and interactive web applications.

Who Should Attend?

This is considered a foundational course. Best for individuals who are new to programming and starting their full-stack career.

Prerequisites

Basic computer skills and a desire to learn programming

What You Will Learn

- Utilize tools used in programming such as VS Code and run JavaScript files locally.
- Understand the basic concepts of programming and JavaScript's role in web development.
- Create variables that store different data types and manipulate the values using operators in JavaScript.
- Use JavaScript built-in libraries, like `console`, to work with primitive data types.
- Implement control flow using `if`, `else if`, `else`, `switch`, and ternary syntax in JavaScript.
- Create functions using all of JavaScript's different syntax that use parameters — and call them with arguments to return values.
- Interpret common JavaScript errors and interpret error messages.
- Purposefully define values and objects in a scope that is accessible to the objects that require said values and other objects in JavaScript.
- Store, access, and manipulate elements in an array in JavaScript.
- Use loops that have defined starting, stopping, and continuing conditions to iterate over elements in a collection in JavaScript.

Training Outline

Introduction to JavaScript

- Exploring what programming is

Fundamentals of Javascript Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Exploring JavaScript usage
- Demo: Setting up a local environment
- Preparing to write a JavaScript file (.js)
- Reviewing statement construction in JavaScript
- Reviewing basic syntax rules
- Printing to console with a basic statement
- Running a JavaScript file in VS Code

Basics of JavaScript Programming

- Exploring data types and their real life counterparts
- Exploring primitive data types in JavaScript
- Working with JavaScript strings
- Adding special string characters (\n, \\, ', ")
- Exploring built-in string methods
- Combining strings with string concatenation
- Working with numbers
- Using arithmetic operators
- Working with booleans
- Data type properties and the dot notation
- Tracking data types with typeof operator

JavaScript Variables
Creating Variables

- Exploring the function and purpose of storing data
- Observing JavaScript variable syntax
- Creating a variable with var
- Creating a variable with let and const
- Variable naming best practice
- Mentally mapping variables storage in memory

Manipulating Variables

- Updating variable value
- Combining variables and arithmetic operations
- Using the increment and decrement operator
- Working with variable string concatenation
- String interpolating with variables
- Data type coercing with variables

JavaScript Conditional Branching

- Exploring the function and purpose of conditions in a program

Basic Control Flow

- Creating an if-statement
- Using comparison operators
- Understanding == and ===
- Using logical operators
- Using Truthy and Falsy
- Adding the else-clause
- Chaining with else-if

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Nesting if-statements
- Condensing with ternary operator

Switch Case

- Creating a switch case
- Controlling switch case execution behavior with fall-through and break
- Managing data typing in a switch case
- Defining a default cas

JavaScript Functions

- Exploring the function and purpose of reusing blocks of code
- Function syntax construction
- Defining a function
- Calling a function
- Adding parameters into a function declaration
- Calling a function with arguments
- Adding default parameters
- Returning values
- Exploring function declaration hoisting
- Using functions inside other functions
- Considerations for defining functions and naming functions

Debugging

- Exploring basic JavaScript errors (syntax errors, runtime errors, logical errors)
- Reading and interpreting error messages
- Using basic debugging technique with console.log
- Common beginnner mistakes

JavaScript Scope

- Exploring the concept of "scoping" in a program
- Understanding "global" scope
- Understanding "block" scope
- Understanding "local" scope
- Exploring differences between "block" scope and "local" scope
- Exploring scope pollution
- Scoping best practice

JavaScript Arrays
Array Creation

- Exploring the purpose of organizing data
- Creating an array with singular data types using an array literal
- Creating an array with mixed data types using an array literal
- Creating multi-dimensional arrays

Array Manipulation

- Accessing elements
- Using element index
- Updating elements in place
- Exploring mutability of an array declared with const

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Exploring array built-in methods
- Accessing array length with .length
- Resetting an array with .length
- Using push()
- Using pop()
- Using sort()

Array Usage Pattern

- Exploring passing arrays to functions
- Exploring manipulating arrays inside of functions
- Trying array equality
- Analyzing references vs value in memory

JavaScript Loops

- Understanding repetition in a program
- Constructing loops semantically
- Using the while loop
- Using the do while loop
- Using the for loop
- Looping in reverse
- Looping through arrays
- Constructing nested loops
- Using the break keyword

JavaScript Objects

- Exploring object data type
- Exploring the purpose and function of objects
- Exploring organizing data into objects
- Creating an object with object literals
- Accessing object properties
- Accessing with bracket notation
- Mutating objects with property reassignment
- Creating an object with methods
- Trying object equality
- Analyzing object references in memory
- Creating nested objects
- Accessing nested properties by chaining
- Using for..in to loop through objects
- Choosing between arrays and objects