JavaScript Essentials 1, developed by OpenEDG JavaScript Institute

Last updated: October 29, 2021

Scope & Sequence

1. Target Audience

The *JavaScript Essentials 1* curriculum is designed for **students with little or no prior knowledge of programming**: students of secondary school, university, vocational school, or simply anyone interested in learning programming.

2. Prerequisites

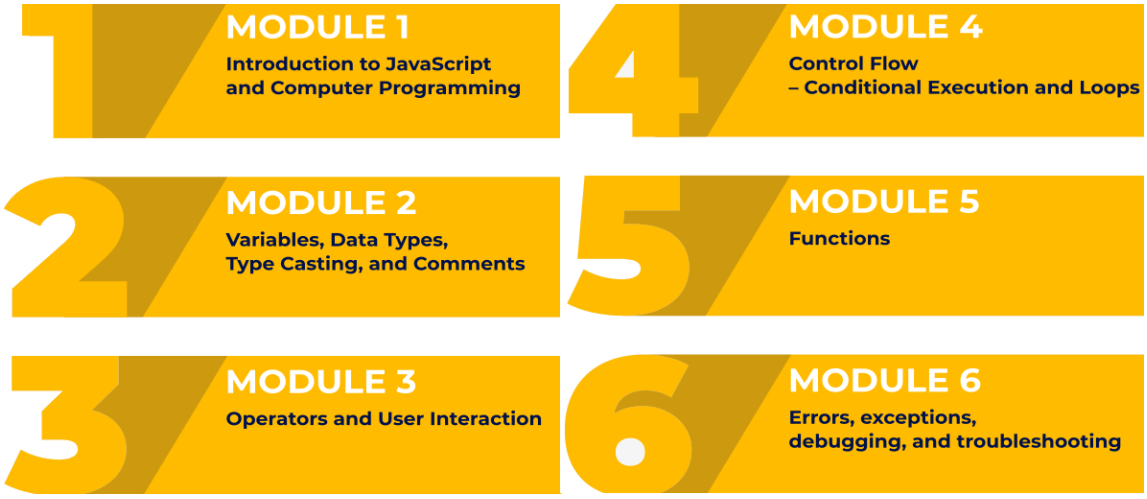
There are **no specific prerequisites** for this course except motivation and very basic knowledge in mathematics. The main goal of the course is to introduce the student to computer programming using JavaScript from the state of complete programming illiteracy to a level that will allow them to start their own studies at an intermediate level and continue their journey with programming.

3. Curriculum Description

The *JavaScript Essentials 1* course **introduces the student to some universal computer programming** concepts, such as **data types**, **type casting**, **containers**, **comments**, **operators**, **conditional execution**, **loops**, **functions**, **errors**, **exceptions**, **troubleshooting**, and **code debugging**. The course guides the student step-by-step to understanding and using the core JavaScript language to **design**, **build**, and **improve** simple JavaScript programs.

4. Course Outline

The *JavaScript Essentials 1* course is divided into **six modules**, each of them consisting of multiple sections. Every module concludes with a brief **quiz** and a **Module Test**. At the end of each section, the student can perform various programming-related tasks, and put in practice the concepts they have learned. Additionally, modules 2-5 include **labs**, which cover the topic areas discussed in multiple sections.



The course ends with the **Final Test** which wraps up all the most important questions covered in modules 1 through 6.

Topics covered:

Module 1

Introduction to JavaScript and Computer Programming

- About JavaScript (how to communicate with the computer, what is JS, advantages and limitations of JS, where is JS used today)

- Setting up the programming environment (development tools, online development environment, local development environment)
- First JS program – Hello, World! (a few words about HTML, how to run your JavaScript code, executing the code directly in the console)

Module 2

Variables, Data Types, Type Casting, and Comments

- Variables (naming, declaring and initializing variables, declarations and strict mode, changing variable values, constants, scope)
- Primitive data types (Boolean, Number, BigInt, String, undefined, null, type casting – primitive construction functions and primitive conversions, implicit conversions)
- Complex data types (Object, Array, basic Array properties and methods)
- Comments (single-line comments, multi-line comments, documentation)

Module 3

Operators and User Interaction

- Assignment, arithmetic, and logical operators (what are operators, assignment operators, arithmetic operators, logical operators, compound assignment operators)
- Strings, JS operators including comparison operators (string concatenation and compound assignments, comparison operators, conditional operators, typeof, instanceof and delete operators, operator precedence)
- Interacting with the user (dialog boxes – alert, confirm, prompt)

Module 4

Control Flow – Conditional Execution and Loops

- Conditional execution (what is conditional execution, the if–else statement, the conditional operator, the switch–case statement)
- Loops (what are loops, the while loop, the do–while loop, the for loop, the for–of loop, the for–in loop, the break and continue statements)

Module 5

Functions

- Function basics (what are functions, declaring functions, calling functions, local variables, the return statement, function parameters, shadowing)
- Functions as first-class members (function expressions, passing a function as a parameter, callbacks)
- Arrow functions (declaring and calling)
- Recursion (basic idea)

Module 6

Errors, exceptions, debugging, and troubleshooting

- Errors and exceptions – introduction (natural languages and communication errors, errors vs. exceptions, errors without exceptions, limited confidence)
- Basic types of errors in JS (SyntaxError, ReferenceError, TypeError, RangeError)
- Exception handling (the try–catch statement, the finally statement, the throw statement, and custom errors)
- Code debugging and troubleshooting (what is debugging, step-by-step execution, viewing and modifying variables, the step out option, measuring code execution time)

5. Curriculum Objectives

Completing the course ensures that **the student is equipped with the most essential means provided by the core JavaScript language** to enable them to start their own studies at an intermediate level and continue their professional development.

Objectives:

Module 1

After completing Module 1, the student will:

- understand the fundamental programming concepts, such as: interpreting and the interpreter, compilation and the compiler, client-side vs. server-side programming;
- have basic knowledge of how to set up and use the basic programming environment (online or local)
- gain skills allowing them to run their first JavaScript program on the client side (both as an element embedded in the HTML page and directly in the browser console).

Module 2

After completing Module 2, the student will:

- have the knowledge and skills to work with variables (i.e. naming, declaring, initializing and modifying their values)
- understand concepts such as scope, code blocks, shadowing, and hoisting;
- know the basic properties of primitive data types such as boolean, number, bigint, undefined, null, and be able to use them;
- be familiar with the basic properties of the primitive data type string, including string literals – single or double quotes, the escape character, string interpolation, basic properties and methods;
- know the basic properties of complex data types such as Array and Object (treated as a record) and be able to use them in practice.

Module 3

After completing Module 3, the student will:

- know what operators are and how to classify them (by type of operand, by number of operands, etc.)
- be able to use assignment, arithmetic, logical, and comparison operators in practice;
- understand the operation of the conditional operator and the typeof, instanceof, and delete operators;
- understand what the precedence and associativity of basic operators are and be able to influence them by means of bracket grouping;
- be able to perform basic two-way communication with the program user using the alert, confirm, and prompt dialog boxes.

Module 4

After completing Module 4, the student will:

- be able to force conditional execution of a group of statements (make decisions and branch the flow) using if-else and switch commands;
- be able to force a group of statements to repeat in a loop using the for, while, and do-while commands, using both dependent and independent conditions on the number of iterations;
- understand and be able to use loop-specific break and continue instructions;
- be able to use the for-in statement to iterate over the properties of an object;
- be able to use the for-of statement to walk through the elements of an array.

Module 5

After completing Module 5, the student will:

- be able to declare and call functions;
- know how to pass call arguments to a function and return the result of its operation from it;
- understand the concept of a local variable and the effect of shadowing variables with the same names within a function;
- know that a function in JS is a first-class member and be able to take advantage of this by declaring functions using function expressions and passing functions as arguments to calls of other functions;
- understand the concept of recursion in the context of functions and be able to solve simple programming problems by using it;
- have a basic understanding of the callback function and be able to use it asynchronously in conjunction with the setTimeout and setInterval methods;
- have a clear understanding of arrow function notation and be able to write alternative functions as regular declarations, function expressions, and arrow functions.

Module 6

After completing Module 6, the student will:

- gain an understanding of the differences between syntactic, semantic, and logical errors;
- understand the concept of an exception and distinguish between the basic exceptions generated by JS when an error occurs: SyntaxError, ReferenceError, TypeError, RangeError;
- have the ability to handle exceptions using the try-catch-finally statement;
- be able to generate their own exceptions using the throw statement;
- have the skills to use the debugger for basic analysis of their own code, including: step-by-step execution, viewing and modifying variables, measuring code execution time.

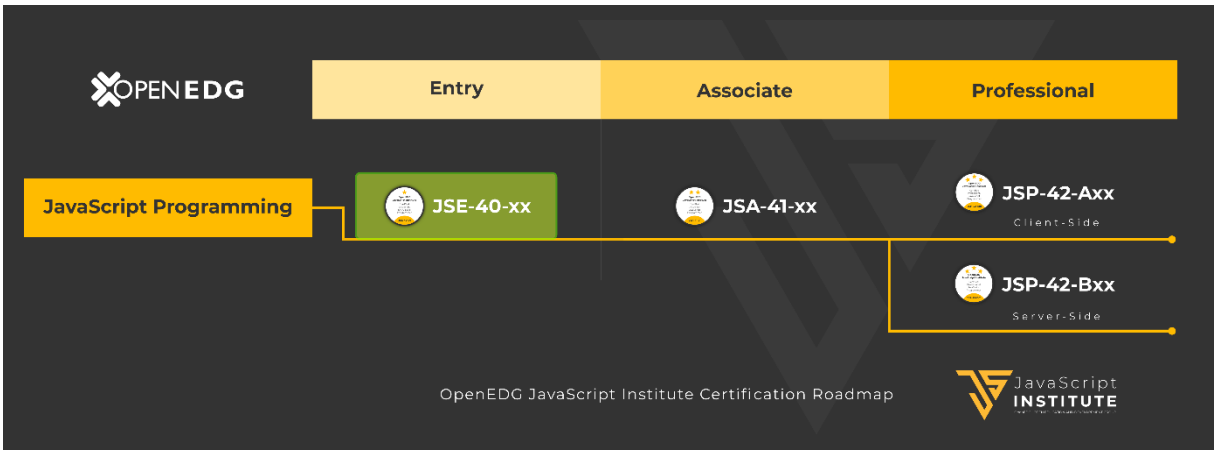
6. Graduate Profile

The graduate of the course:

- knows the syntax of the core JavaScript language to a degree that allows for working with variables, operators, flow control, and functions;
- knows the basics of the JavaScript data types system, distinguishing between primitive and complex types, and is able to choose a type adequate to their needs;
- thinks algorithmically and can analyze a problem using a programmatic conceptual apparatus;
- can choose a data type adequate to the problem being solved and use suitable flow control means;
- can design, develop, and improve very simple JavaScript programs;
- can interpret and handle basic exceptions related to errors in program execution
- understands a programmer's work in the software development process and the role of fundamental development tools;
- knows how a program is interpreted and executed in an actual computer environment, local or remote;
- can create and develop their own programming portfolio;
- is prepared to attempt the qualification JSE – Certified Entry-Level JavaScript Programmer certification (Exam JSE-40-0x) from the JavaScript Institute.

7. Target certification

This course is aligned with the JSE-40-0x exam.



The JSE – Certified Entry-Level JavaScript Programmer certification is a professional credential that demonstrates the candidate's understanding of the **JavaScript language core syntax and semantics**, as well as their **proficiency** in using the most **essential** elements of the **language**, **tools**, and **resources** to **design**, **develop**, and **refactor** simple JavaScript **programs**.

The certification holder knows the syntax of the core JavaScript language to a degree that allows them to work with **variables**, **operators**, **control flow mechanisms**, and **functions**, as well as understands the fundamentals of the JavaScript **data type system**, **exception handling**, **troubleshooting**, **debugging**, and the **runtime environment**.

Moreover, the certified individual understands the **universal concepts of computer programming** and a **programmer's work and role** in the software development process; knows how to apply **best coding practices and conventions**; **thinks algorithmically** and is able to analyze a problem using a programmatic conceptual apparatus.

Becoming JSE certified ensures that the individual is acquainted with the most essential means provided by the **core JavaScript language** to enable them to start their own studies at an intermediate level, and to continue their professional development.

8. Minimum Requirements

The course contents, including the course modules, labs, quizzes, and assessments can be **accessed online through any Internet browser**. For the best learning experience, we recommend using the most recent versions of Mozilla Firefox, Internet Explorer/Microsoft Edge, or Google Chrome (recommended).

Minimum Technical Recommendations:

- a desktop computer with a mouse/touchpad and a regular keyboard;
- an Internet browser (e.g. Google Chrome)
- Internet connection;
- screen size: 13 inches or more.