

# JSE – Certified Entry-Level JavaScript Programmer

## (Exam JSE-40-01)

### Exam Objectives

Last updated: 2022-09-27

<b>JSE – Certified Entry-Level JavaScript Programmer (Exam JSE-40-01)</b> Aligned with <i>JavaScript Essentials 1</i>	
Objective ID	Exam Block/Objective
JSE-40-01 1	<b>Exam block #1: Basics of JavaScript and Computer Programming</b>
	Weight: 8%   3 exam items
JSE-40-01 1.1	<b>JSE 1.1 Understand fundamental terms and definitions</b>
	<ul style="list-style-type: none"> <li>interpreting and the interpreter</li> <li>compilation and the compiler</li> <li>client-side vs. server-side programming</li> </ul>
JSE-40-01 1.2	<b>JSE 1.2 Set up the programming environment</b>
	<ul style="list-style-type: none"> <li>basic development tools</li> <li>online vs. local development environment (code editor, interpreter, debugger)</li> </ul>
JSE-40-01 1.3	<b>JSE 1.3 Explain client-side execution of JavaScript code</b>
	<ul style="list-style-type: none"> <li>HTML fundamentals</li> <li>embedding JavaScript code</li> <li>executing code directly in the console</li> </ul>
JSE-40-01 2	<b>Exam block #2: Variables, Data Types, and Type Casting</b>
	Weight: 20%   6 exam items
JSE-40-01 2.1	<b>JSE 2.1 Use variables effectively</b>
	<ul style="list-style-type: none"> <li>naming, declaring and initializing variables</li> <li>changing values assigned to variables</li> <li>constants</li> <li>scope (blocks, shadowing, hoisting)</li> </ul>
JSE-40-01 2.2	<b>JSE 2.2 Define, identify, and use primitive data types in code: <i>boolean, number, bigint, undefined, null</i></b>
	<ul style="list-style-type: none"> <li>the <b>number</b> range and its formats</li> <li>the <b>bigint</b> range</li> <li><b>boolean</b> values</li> <li>empty and unknown values (<b>null</b> or <b>undefined</b>)</li> </ul>
JSE-40-01 2.3	<b>JSE 2.3 Define, identify, and use primitive data types in code: <i>string</i></b>

	<ul style="list-style-type: none"> <li>string literals – single and double quotes</li> <li>the escape character</li> <li>string interpolation</li> <li>basic properties and methods: <b>length</b>, <b>charAt</b>, <b>slice</b>, <b>split</b></li> </ul>
JSE-40-01 2.4	<b>JSE 2.4 Perform type-casting operations</b>
	<ul style="list-style-type: none"> <li>primitive construction functions: <b>String</b>, <b>Number</b>, <b>BigInt</b>, <b>Boolean</b></li> <li>primitive conversions;</li> <li>implicit conversions</li> </ul>
JSE-40-01 2.5	<b>JSE 2.5 Define, identify, and use complex data types in code: Object</b>
	<ul style="list-style-type: none"> <li>object as a record</li> <li>object literals</li> <li>getting and setting properties</li> </ul>
JSE-40-01 2.6	<b>JSE 2.6 Define, identify, and use complex data types in code: Array</b>
	<ul style="list-style-type: none"> <li>array as an example of a complex data type</li> <li>basic properties and methods of arrays: <b>length</b>, <b>indexOf</b>, <b>push</b>, <b>unshift</b>, <b>pop</b>, <b>shift</b>, <b>reverse</b>, <b>slice</b>, <b>concat</b></li> </ul>
JSE-40-01 3	<b>Exam block #3: Operators and User Interaction</b>
	Weight: 18%   5 exam items
JSE-40-01 3.1	<b>JSE 3.1 Recognize and proficiently utilize assignment, arithmetic, and string operators</b>
	<ul style="list-style-type: none"> <li>definition of operators</li> <li>assignment operators</li> <li>arithmetic operators</li> <li>compound assignment operators</li> </ul>
JSE-40-01 3.2	<b>JSE 3.2 Identify and proficiently use logical and comparison operators</b>
	<ul style="list-style-type: none"> <li>logical operators</li> <li>compound assignment operators</li> <li>comparison operators</li> </ul>
JSE-40-01 3.3	<b>JSE 3.3 Recognize and effectively use the ternary and other JavaScript operators</b>
	<ul style="list-style-type: none"> <li>ternary (conditional) operator</li> <li>other operators: <b>typeof</b>, <b>instanceof</b>, and <b>delete</b></li> </ul>
JSE-40-01 3.4	<b>JSE 3.4 Understand and implement operator precedence</b>
	<ul style="list-style-type: none"> <li>precedence and associativity of basic operators</li> <li>using brackets to force a specific order of operations (grouping)</li> </ul>
JSE-40-01 3.5	<b>JSE 3.5 Implement the mechanism for interacting with the user</b>
	<ul style="list-style-type: none"> <li>dialog boxes: <b>alert</b>, <b>confirm</b>, <b>prompt</b></li> <li>actions based on user decisions</li> </ul>
JSE-40-01 4	<b>Exam block #4: Control Flow – Conditional Execution and Loops</b>
	Weight 21%   6 exam items
JSE-40-01 4.1	<b>JSE 4.1 Implement the decision-making mechanism and organize the control flow with the <i>if</i> instruction</b>

	<ul style="list-style-type: none"> <li>conditional statements: <b>if, if-else</b></li> <li>multiple conditional statements</li> <li>nesting conditional statements</li> </ul>
JSE-40-01 4.2	<b>JSE 4.2 Implement the decision-making mechanism and organize the control flow with the <i>switch</i> instruction</b>
	<ul style="list-style-type: none"> <li>conditional statement: <b>switch-case</b></li> </ul>
JSE-40-01 4.3	<b>JSE 4.3 Implement the decision-making mechanism and organize the control flow with the <i>do-while</i> and <i>while</i> loops, and the <i>break</i> and <i>continue</i> instructions</b>
	<ul style="list-style-type: none"> <li>repeating a code fragment in a loop</li> <li>iterations</li> <li>conditions that (don't) depend on the number of iterations</li> <li>basic statements related to loops: <b>while, do ... while, break, case</b></li> </ul>
JSE-40-01 4.4	<b>JSE 4.4 Implement the decision-making mechanism and organize the control flow with the <i>for</i> statement</b>
	<ul style="list-style-type: none"> <li>repeating a code fragment using the <b>for</b> statement</li> </ul>
JSE-40-01 4.5	<b>JSE 4.5 Implement the decision-making mechanism and organize the control flow with the <i>for-in</i> statement</b>
	<ul style="list-style-type: none"> <li>using the <b>for ... in</b> statement for enumerating object keys</li> </ul>
JSE-40-01 4.6	<b>JSE 4.6 Implement the decision-making mechanism and organize the control flow with the <i>for-of</i> statement</b>
	<ul style="list-style-type: none"> <li>using the <b>for ... of</b> statement for traversing arrays</li> </ul>
JSE-40-01 5	<b>Exam block #5: Functions</b>
	Weight 21%   6 exam items
JSE-40-01 5.1	<b>JSE 5.1 Organize and decompose the code using functions</b>
	<ul style="list-style-type: none"> <li>declaring, calling, and passing arguments</li> <li>returning function results</li> </ul>
JSE-40-01 5.2	<b>JSE 5.2 Organize and implement interactions between the function and its environment</b>
	<ul style="list-style-type: none"> <li>parameters</li> <li>local variables</li> <li>shadowing</li> </ul>
JSE-40-01 5.3	<b>JSE 5.3 Apply function expressions and functions as first-class members in code</b>
	<ul style="list-style-type: none"> <li>storing a function in a variable</li> <li>passing a function as a parameter</li> <li>named and anonymous function expressions</li> </ul>
JSE-40-01 5.4	<b>JSE 5.4 Understand and implement the concept of recursion</b>
	<ul style="list-style-type: none"> <li>using recursion to solve simple computational problems</li> </ul>
JSE-40-01 5.5	<b>JSE 5.5 Explain and implement callback functions</b>
	<ul style="list-style-type: none"> <li>synchronous callbacks</li> <li>asynchronous callbacks</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>setTimeout, setInterval</b></li> </ul>
JSE-40-01 5.6	<b>JSE 5.6 Explain and implement arrow functions</b>
	<ul style="list-style-type: none"> <li>• notation of function expressions in the arrow form</li> <li>• passing arguments</li> <li>• shortened form for a function body with one statement</li> </ul>
JSE-40-01 6	<b>Exam block #6: Errors, exceptions, debugging, and troubleshooting</b>
	Weight: 12%   4 exam items
JSE-40-01 6.1	<b>JSE 6.1 Classify and distinguish between different programming errors</b>
	<ul style="list-style-type: none"> <li>• error classification: syntax, semantics, and logical errors</li> <li>• runtime errors</li> </ul>
JSE-40-01 6.2	<b>JSE 6.2 Compare and contrast different JavaScript errors and exceptions</b>
	<ul style="list-style-type: none"> <li>• basic types of JavaScript errors: Syntax Error, ReferenceError, TypeError, RangeError</li> </ul>
JSE-40-01 6.3	<b>JSE 6.3 Understand, examine, and design an exception-handling mechanism in JavaScript</b>
	<ul style="list-style-type: none"> <li>• handling: <b>try ... catch ... finally</b></li> <li>• throwing custom exceptions: <b>throw</b></li> </ul>
JSE-40-01 6.4	<b>JSE 6.4 Use different debugging and troubleshooting techniques</b>
	<ul style="list-style-type: none"> <li>• step-by-step execution</li> <li>• viewing and modifying variables</li> <li>• measuring code execution time</li> </ul>