

98-382

Introduction to Programming Using JavaScript

Exam number: 98-382

Exam title: Introduction to Programming Using JavaScript

Publish date:

GUID:

Language(s) this exam will be available in: ENU, JPN, CHT, CHS, ESM, PTB, FRA, DEU, KOR

Audience (IT professionals, Developers, Information workers, etc.): Beginning programming students

Technology: JavaScript (ECMAScript v5)

Credit type (example: MCSA): MTA

Exam provider (VUE, Certiport, or both): Both

Exam Design

Audience Profile

Candidate should be able to recognize and write syntactically correct JavaScript code, use data types supported by JavaScript, and be able to recognize and write JavaScript code that will logically solve a given problem.

Candidates are expected to have had, at minimum, instruction and/or hands-on experience of approximately 100 hours with the JavaScript programming language, be familiar with its features and capabilities, and understand how to write, debug, and maintain well-formed, well documented JavaScript code.

Prerequisite skills for the 98-382: Introduction to Programming Using JavaScript Exam are:

Core Algebra (Algebra I) (typical US 9th/10th grade level)

Fundamental knowledge of HTML

Skills measured

Program with JavaScript Operators, Methods, and Keywords

Complete or debug code that uses assignment and arithmetic operators

Assignment; increment; decrement; addition; subtraction; division; multiplication; modulus; compound assignment operators

Apply JavaScript best practices

Comments; indentations; naming conventions; noscript; constants; reserved keywords; debugger keyword; setting breakpoints; console.log

Evaluate the use of inline and external scripts

When to use, how to use, and what happens when both are used

Implement exception handling

try; catch; finally

Complete and debug code that interacts with the Browser Object Model (BOM)

Manage state; display dialogs; determine screen size

Program with Variables, Data Types, and Functions

Declare and use variables of primitive data types

Number; Boolean; String; Null; Undefined; typeof operator; type checking functions; use strict; converting between data types; formatting numbers; string operations; single quote vs double quote (nesting); initialization

Declare and use arrays

Single-dimensional: iteration; initialization; array definition; sorting and searching; push and pop; shift/unshift; length; accessing an element; understanding multi-dimensional arrays

Complete and debug code that uses objects

Properties; methods; instantiation; Date object: time; retrieving date parts; localization (MM/DD vs DD/MM); adding and subtracting dates

Complete and debug code that uses built-in Math functions

Random; round; abs; floor; ceiling; min/max; pow; sqrt

Complete and debug a function that accepts parameters and returns a value

Reusable code; local vs global scope, redefining variables, passing parameters, value vs. reference, return values

Implement and Analyze Decisions and Loops

Evaluate expressions that use logical and comparison operators

==; !=; <; >; <=; >=; !; &&; ||

Complete and debug decision statements

if; else if; switch; nested if

Complete and debug loops

for; while; do; break; continue

Interact with the Document Object Model

Identify and construct the Document Object Model (DOM) tree

 window; document; body; other HTML elements

Identify and handle HTML events

 onchange; onmouseover; onload; onclick; onmouseout; onkeydown

Complete and debug code that outputs to an HTML document

 innerHTML; document.write

Complete and debug code that locates, modifies, and adds HTML elements and attributes

 getElementById; getElementsByTagName; getElementsByClassName; setAttribute;
 createElement

Interact with HTML Forms

Complete and debug code that retrieves input from forms and sets form field values

 Retrieving form values; the DOM path to form field and then to the value property; getting
 values from different types of elements; prepopulating values; masking values

Complete and debug code to perform input validation

 Case (upper and lower); string comparisons; validation; NaN

Describe the form submission process

 onsubmit; understand post vs get; understand potential targets for submission