Open Education and Development Group 2022. All Rights Reserved.

JSA – Certified Associate JavaScript Programmer

(Exam JSA-41-01)

Exam Objectives

Last updated: 2022-09-27

JSA – Certified Associate JavaScript Programmer (Exam JSA-41-01)
Aligned with *JavaScript Essentials 2*

Objective ID	Exam Block and Objectives
JSA-41-01 1	**Exam block #1: Classless objects**
	Weight: 25% | 11 exam items
JSA-41-01 1.1	**JSA 1.1 Create individual objects**
	• basic object creation using literals
JSA-41-01 1.2	**JSA 1.2 Explain and use object properties**
	• adding, modifying, and deleting properties • nested properties
JSA-41-01 1.3	**JSA 1.3 Compare and contrast dot and bracket notations**
	• dot notation as the primary way to refer to object fields • using bracket notation to allow the use of multi-word and computed keys
JSA-41-01 1.4	**JSA 1.4 Test property existence and perform property enumeration**
	• testing for the presence of a field in an object (the **in** keyword) • using the **for ... in** statement to pass the keys of an object • the **Object.key** method as an alternative to **for ... in**
JSA-41-01 1.5	**JSA 1.5 Compare and contrast objects**
	• the idea of object references • reference comparison vs. field comparison (deep comparison)
JSA-41-01 1.6	**JSA 1.6 Implement object copying mechanisms**
	• copying references, cloning, and merging (the **Object.assign** method) • the spread operator and shallow cloning • the concept of deep cloning
JSA-41-01 1.7	**JSA 1.7 Explain and implement methods in code**
	• function as an object property • defining methods in the object body and adding methods to existing objects • using the **this** keyword inside methods

Open Education and Development Group 2022. All Rights Reserved.

JSA-41-01 1.8	**JSA 1.8 Explain and implement getters and setters**
	• methods as properties • defining and using getters and setters (the **get** and **set** keywords)
JSA-41-01 1.9	**JSA 1.9 Organize and configure objects and properties**
	• modifying attributes of objects and fields • using the methods **Object.defineProperty**, **Object.preventExtensions**, **Object.seal**, and **Object.freeze**
JSA-41-01 1.10	**JSA 1.10 Demonstrate different ways to create classless objects**
	• the Factory pattern • the constructor function and the **new** operator • the **Object.create** method
JSA-41-01 1.11	**JSA 1.11 Explain and apply the concept of prototypes**
	• prototype-based inheritance • object property **__proto__** • constructor function prototype property • using the **setPrototypeOf** method
JSA-41-01 2	**Exam block #2: Classes and class-based approach**
	Weight: 23% | 7 exam items
JSA-41-01 2.1	**JSA 2.1 Design classes and implement class declarations**
	• normal class declarations (the **class** keyword) • class body (constructor, properties, methods) • a class as a first-class citizen – storing classes in variables and class expression
JSA-41-01 2.2	**JSA 2.2 Create objects from classes**
	• creating an object based on a class (the **new** operator) • looking for a connection between an object and a source class (the **instanceof** operator)
JSA-41-01 2.3	**JSA 2.3 Explain and use class properties**
	• defining properties inside class methods (constructor and regular methods) • direct declaration inside the class body
JSA-41-01 2.4	**JSA 2.4 Create and implement getters and setters**
	• defining and using getters and setters (the **get** and **set** keywords)
JSA-41-01 2.5	**JSA 2.5 Explain and apply the concept of inheritance**
	• class inheritance (the **extends** keyword) • shadowing methods and properties • using the **super** keyword in the constructor and in methods
JSA-41-01 2.6	**JSA 2.6 Explain and apply static members in code**
	• the concept of static members • defining and using methods and properties associated with a class instead of an object

Open Education and Development Group 2022. All Rights Reserved.

JSA-41-01 2.7	**JSA 2.7 Compare and contrast classes and constructors**
	• similarity of classes to constructors • conversion of a class into an adequate constructor function and vice versa
JSA-41-01 3	**Exam block #3: Built-in objects**
	Weight: 27% | 12 exam items
JSA-41-01 3.1	**JSA 3.1 Explain and use the Number constructor**
	• creating **Number** objects from data of different types, including various string formats • converting numbers into different string formats • static properties and methods of the **Number** constructor (i.e. properties defining the basic ranges)
JSA-41-01 3.2	**JSA 3.2 Explain and use the String constructor**
	• the string as an array of characters • case conversion methods • splitting the string • searching for and replacing substrings • padding and trimming • comparison of strings
JSA-41-01 3.3	**JSA 3.3 Explain and use the Date constructor**
	• creating a **Date** object (constructor) • time zones and local time handling • getting current time • timestamp and using it to measure the time of code execution • time specification • operating on individual date and time components
JSA-41-01 3.4	**JSA 3.4 Describe and implement the concept of arrays**
	• basic methods for managing an **Array** type collection (creating, merging, adding and removing items, passing through, the **slice** method, the **splice** method) • using the **spread** operator • destructuring assignments
JSA-41-01 3.5	**JSA 3.5 Explain and implement advanced array methods**
	• methods using the functional approach: **find**, **every**, **some**, **filter**, **sort**, **map** and **reduce**
JSA-41-01 3.6	**JSA 3.6 Explain, implement, and process Set data type collections**
	• the concept of set data structure and the **Set** object • basic methods and properties of **Set** objects (**constructor**, **add**, **has**, **delete**, **clear**, **size**) • walking through elements (**for ... of**, iterators) • the **spread** operator
JSA-41-01 3.7	**JSA 3.7 Explain, implement, and process Map data type collections**

Open Education and Development Group 2022. All Rights Reserved.

	• the concept of the map data structure and the **Map** object • basic methods and properties of Map objects (constructor, **set**, **get**, **has**, **delete**, **clear**, **size**) • walking through elements (**for ... of**, iterators) • the **spread** operator
JSA-41-01 3.8	**JSA 3.8 Implement objects as data structures**
	• using an object as a regular dictionary-type data structure • item management • walking through data structures • the **spread** operator
JSA-41-01 3.9	**JSA 3.9 Use the JSON object to process data**
	• basic concept of the JSON format (JavaScript Object Notation) • converting data to JSON (the **JSON.stringify** method) • parsing the JSON format and retrieving data (the **JSON.parse** method)
JSA-41-01 3.10	**JSA 3.10 Use the Math object to perform mathematical operations**
	• basic methods: **ceil**, **floor**, **round**, **random**, **min**, **max**, **abs**, **pow**, **log** • trigonometric functions
JSA-41-01 3.11	**JSA 3.11 Explain and apply the concept of regular expressions**
	• basic rules for creating regular expressions • using the RegExp object • abbreviated notation of a **RegExp** object declaration • using methods of **RegExp** and **String** objects to efficiently search patterns in text: **test**, **exec**, **match**, **search**, **replace**
JSA-41-01 3.12	**JSA 3.12 Explain and implement the concept of extending built-in types**
	• using prototypes to extend built-in types (adding new properties and methods)
JSA-41-01 4	**Exam block #4: Advanced functions**
	Weight 25% | 10 exam items
JSA-41-01 4.1	**JSA 4.1 Organize and implement extended function parameter handling**
	• using default parameter values, the **rest** parameter, and the **spread** operator • simulating named parameters
JSA-41-01 4.2	**JSA 4.2 Explain and use closures and IIFEs**
	• the use of the closure – execution environment of the function • Immediately Invoked Function Expressions (IIFE)
JSA-41-01 4.3	**JSA 4.3 Implement the mechanism for call forwarding**
	• the **this** keyword in functions • function call methods: **apply, call, bind**
JSA-41-01 4.4	**JSA 4.4 Implement the mechanism for decorating functions**

Open Education and Development Group 2022. All Rights Reserved.

	• wrappers and higher-order functions • functions as first-class citizens • passing functions as arguments and returning functions as results • decorating functions and adding new functionalities using a wrapper function
JSA-41-01 4.5	**JSA 4.5 Explain, create, and implement generators and iterators in code**
	• creating and using generators • the concept of iterable objects • generators as elements of iterable objects • iterators
JSA-41-01 4.6	**JSA 4.6 Explain, organize, and handle asynchronous events using callback functions**
	• the concept of asynchronous programming • using callback functions to handle asynchronous events
JSA-41-01 4.7	**JSA 4.7 Explain and apply the concept of promises**
	• the concept of promises as an alternative method of asynchronous programming • defining user-created promises • using the **then**, **catch,** and **finally** methods to handle promises
JSA-41-01 4.8	**JSA 4.8 Explain and apply advanced promise chaining techniques**
	• promise chaining techniques • basic methods for handling sets of promises: **Promise.all**, **Promise.any**, **Promise.race**
JSA-41-01 4.9	**JSA 4.9 Use async and await to handle promises**
	• alternative method for handling promises – asynchronous functions and event waiting (the **async** and **await** keywords)
JSA-41-01 4.10	**JSA 4.10 Understand and implement asynchronous handling of network requests**
	• practical use of asynchronous techniques to retrieve data from the network • the **XMLHttpRequest** object • the **Fetch** API