Certificación JSA – Certified Associate JavaScript Programmer

Calle de la Basílica, 19
28020 Madrid
(34) 915 53 61 62
www.cas-training.com

Nivel
Intermedio

Duración
65 minutos

Modalidad
Presencial

Learning by doing

Curso Oficial

Certificación JSA – Certified Associate JavaScript Programmer

Objetivos:

Obtener la **certificación JSA – Certified Associate JavaScript Programmer**

Requisitos:

- No existen requisitos previos para tomar este examen de certificación.
- Sin embargo, es recomendable haber obtenido la certificación JSE – Certified Entry-Level JavaScript Programmer y haber completado el curso JavaScript Essentials 2.

Metodología:

- "Learning by doing" se centra en un contexto real y concreto, buscando un aprendizaje en equipo para la resolución de problemas en el sector empresarial.
- Aulas con grupos reducidos para que el profesional adquiera la mejor atención por parte de nuestros instructores profesionales.
- El programa de estudios como partners oficiales es confeccionado por nuestro equipo de formación y revisado por las marcas de referencia en el sector.
- La impartición de las clases podrá ser realizada tanto en modalidad Presencial como Virtual.

Examen y Certificación:

Certificación JSA – Certified Associate JavaScript Programmer

Contenidos:

Sección 1: Object-Oriented Programming - Objects
- be able to create objects using a variety of techniques, including literals, constructors and factories;
- be able to refer to object fields, including nested ones, with both dot notation and bracket notation;
- be able to alter objects as required, including by adding, deleting, and modifying properties and methods;
- know how to verify the presence of object fields, enumerate them, and influence the configuration of an object and individual fields (e.g. by blocking the possibility to modify them)
- understand that objects are stored in variables as references and know how this affects manipulating them in practice (e.g. when comparing objects)
- know what the context of a method call is and be able to use the keyword this in practice;
- understand what an object prototype is, what inheritance using a chain of prototypes is and how it can be used, for example, to modify the properties of a group of related objects.

Sección 2: Object-Oriented Programming - Classes
- be able to declare classes, including using the class expression technique, and create objects using them;
- be able to define properties both from the class methods and directly in the class body;
- understand the idea of class inheritance and be able to use it in practice;
- be able to create and use static methods and properties of classes;
- be able to define getters and setters in classes.

Sección 3: Object-Oriented Programming - Built-in Objects
- be familiar with the basic set of built-in objects of the JS language;
- understand the difference between primitive types and their corresponding wrapper objects, know the basic properties and methods of these objects, and be able to use autoboxing in practice;
- be able to handle Array data to an advanced degree, using such methods as filtering, sorting, reducing, mapping, searching, merging, etc.
- be able to use the destructuring assignment and spread operator in their work with arrays;
- know in which situations objects of the Map and Set types can be used for data storage as an alternative to Array and Object, declare them, and manipulate their elements;
- understand the JSON format and be able to convert objects and arrays to and from this format;
- be able to use the methods provided by the built-in Math object to perform basic mathematical calculations;
- be able to take advantage of basic regular expressions (using the RegExp type) to parse character strings;
- be able to extend the built-in JS types with new properties and methods.

Sección 4: Advanced Functions
- know and be able to use in practice such mechanisms related to functions as: Extended Parameter Handling (default parameter values, rest parameter, spread operator), Recursion (closure, first-class functions), Forwarding calls (apply, call, bind), Decorating functions (wrappers, higher order functions)
- understand the concept of lazy evaluation, and be able to use generators and iterators in practice;
- clearly understand when asynchronous programming techniques should be used;
- know how to use callback functions in practice to solve problems that require asynchronous operations;
- understand the Promise mechanism and be able to use it as an alternative to callback functions;
- understand the operation of the async function in conjunction with the await keyword and be able to use it to handle promises.

CAS TRAINING

UN ESPACIO PARA CRECER

cas-training.com

OpenEDG
JS Institute

Certified Associate
JavaScript Programmer

JSA-41-01

Certificación JSA – Certified Associate JavaScript Programmer

JSA