

Javascript

Los candidatos a este examen deben poder reconocer y escribir código JavaScript sintácticamente correcto que resolverá lógicamente un problema determinado y utilizará tipos de datos compatibles con JavaScript.

Se espera que los candidatos tengan al menos 150 horas de instrucción o experiencia práctica con el lenguaje de programación JavaScript. Los candidatos deben estar familiarizados con las características y capacidades de JavaScript y comprender cómo escribir, depurar y mantener código JavaScript bien formado y documentado.

Para aprobar el examen, también se espera que el candidato tenga los siguientes conocimientos y habilidades previos:

- Habilidades de lectura de octavo grado
- Álgebra I
- Conocimientos fundamentales de HTML.
- Conocimientos fundamentales de CSS

1. Operadores, métodos y palabras clave de JavaScript

1.1 Código completo y depurado que utiliza asignación y aritmética operadores

- Asignación, incremento, decremento, suma, resta, división, operadores de multiplicación, módulo y asignación compuesta (+=, -=, *=, /=, %=)

1.2 Aplicar las mejores prácticas de JavaScript

- Comentarios, sangría, convenciones de nomenclatura, noscript, constantes, palabras clave reservadas, palabra clave de depuración, configuración de puntos de interrupción, console.log

1.3 Evaluar el uso de scripts internos y externos

- Cuándo usarlo, cómo usarlo y qué sucede cuando los scripts se usan en múltiples niveles.

1.4 Implementar manejo de excepciones

- intentar, atrapar, finalmente

1.5 Código completo y depurado que interactúa con el modelo de objetos del navegador (BOM)

- Mostrar cuadros de diálogo, determinar el tamaño de la pantalla

2. Variables, tipos de datos y funciones

2.1 Declarar y utilizar variables de tipos de datos primitivos

- Número, booleano, cadena, nulo, indefinido, tipo de operador, verificación de tipo funciones, uso estricto, conversión entre tipos de datos (parseInt, parseFloat), formato de números, operaciones de cadena, eval(), toFixed(), toLocaleString(), toPrecision(), comillas simples versus comillas dobles (anidamiento), inicialización

2.2 Declarar y usar matrices

- Matrices unidimensionales; matrices multidimensionales; iteración; inicialización; definir, ordenar y buscar una matriz; métodos push, pop, shift y unshift; propiedad de longitud; acceder a un elemento de matriz

2.3 Código completo y depurado que utiliza objetos

- Propiedades, métodos, creación de instancias, objeto de fecha, recuperación de fecha y hora. partes, localización del formato de fecha (MM/DD vs DD/MM), suma y resta de fechas

2.4 Código completo y depurado que utiliza funciones matemáticas integradas

- aleatorio, redondo, abs, piso, techo, min, max, pow, sqrt

2.5 Funciones completas y de depuración que aceptan parámetros y devuelven valores

- Código reutilizable, alcance local versus global, redefinición de variables, transferencia parámetros, valor versus referencia, valores de retorno

3. Decisiones y bucles

3.1 Evaluar expresiones que utilizan operadores lógicos y de comparación.

- !=, <, >, <=, >=, !, ==, &&, ||

3.2 Completar y depurar declaraciones de decisión

- Alternativa única (si), alternativa dual (si no), alternativa múltiple (cambiar), anidado si

3.3 Ciclos completos y depurados

- para, para dentro, mientras, hacer mientras, romper, continuar

4. Modelo de objetos de documento

4.1 Identificar y construir el árbol del Modelo de objetos de documento (DOM)

- ventana, documento, cuerpo y otros elementos HTML

4.2 Identificar y manejar eventos de documento, formulario, teclado y mouse

- onload, onfocus, onblur, onchange, onkeydown, onkeyup, onkeypress, al hacer clic, al mouse sobre, al mouse fuera

4.3 Código completo y depurado que genera un documento HTML

- documento.escribir, HTML interno, contenido de texto

4.4 Código completo y depurado que localiza, modifica y agrega HTML elementos y atributos de los documentos

- getElementById, getElementsByTagName, getElementsByClassName, establecerAtributo, crearElemento

4.5 Crear eventos utilizando controladores y oyentes de eventos

- Eventos DOM, evento de atributo HTML, addEventListener

5. Formularios HTML

5.1

Código completo y depurado que recupera la entrada del formulario y configura el formulario valores de campo

- Recuperar valores de formulario; identificar la ruta DOM; obteniendo valores de diferentes tipos de elementos; rellenar previamente, enmascarar y actualizar valores

5.2 Código completo y depurado que realiza la validación de entrada

- Caso, comparaciones de cadenas, Not-A-Number (NaN), no en blanco

5.3 Describir el proceso de envío del formulario

- onsubmit, POST vs. GET, objetivos potenciales para el envío