

Python

Los candidatos a este examen deben poder reconocer y escribir código Python 3 bien documentado y sintácticamente correcto que resuelva lógicamente un problema determinado, utilizar correctamente los tipos de datos admitidos por Python y utilizar bibliotecas comunes para escribir un programa que resuelva un problema complejo.

Se espera que los candidatos hayan tenido al menos 150 horas de instrucción y/o experiencia práctica con el lenguaje de programación Python, estén familiarizados con sus características y capacidades y comprendan cómo escribir, depurar y mantener archivos bien formados y bien desarrollados. -Código Python documentado.

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

- Habilidades de lectura de octavo grado
- Conocimientos básicos de informática
- Álgebra I

1. Operaciones utilizando tipos de datos y operadores

1.1 Evaluar expresiones para identificar los tipos de datos que Python asigna variables

- str, int, float y bool

1.2 Realizar y analizar datos y operaciones de tipo de datos.

- Conversión de tipos de datos, indexación, división, construcción de estructuras de datos, listas, operaciones de listas (incluidas ordenar, fusionar, agregar, insertar, eliminar, encontrar máximos y mínimos e invertir)

1.3 Determinar la secuencia de ejecución según la precedencia del operador

- Asignación (=, +=, -=, /=, %=, //=, **=), comparación (==, >=, <=, !=), lógica (y, o, no), lógico, aritmético (+, -, /, //, %, **, unario + y -), identidad (es), contención (en)

1.4 Seleccionar operadores para lograr los resultados previstos

- Asignación (=, +=, -=, /=, %=, //=, **=), comparación (==, >=, <=, !=), lógica (y, o, no), lógico, aritmético (+, -, /, //, %, **, unario + y -), identidad (es), contención (en)

2. Control de flujo con decisiones y bucles

2.1 Construir y analizar segmentos de código que utilizan declaraciones de ramificación

- expresiones condicionales if, elif, else, anidadas y compuestas

2.2 Construir y analizar segmentos de código que realizan iteraciones

- while, for, break, continue, pass, bucles anidados, bucles que incluyen expresiones condicionales compuestas

3. Operaciones de entrada y salida

3.1 Construir y analizar segmentos de código que realizan operaciones de entrada y salida de archivos

- abrir, cerrar, leer, escribir, agregar, verificar existencia, eliminar, con declaración

3.2 Construir y analizar segmentos de código que realizan operaciones de entrada y salida de la consola

- Leer entradas desde la consola, imprimir texto formateado (método `string.format()`), método `f-String`), use argumentos de línea de comando

4. Documentación y estructura del código

4.1 Segmentos de código de documento

- Utilice sangría, espacios en blanco, comentarios y cadenas de documentos; generar documentación usando `pydoc`

4.2 Construir y analizar segmentos de código que incluyan definiciones de funciones

- Firmas de llamada, valores predeterminados, retorno, definición, pase

5. Solución de problemas y manejo de errores

5.1 Analizar, detectar y corregir segmentos de código que tienen errores

- Errores de sintaxis, errores lógicos, errores de tiempo de ejecución

5.2 Analizar y construir segmentos de código que manejen excepciones

- intentar, excepto, de lo contrario, finalmente, subir

5.3 Realizar pruebas unitarias

- Prueba unitaria, funciones, métodos y métodos de afirmación (`assertIsInstance`, `afirmarEqual`, `afirmarTrue`, `afirmarIs`, `afirmarIn`)

6. Operaciones utilizando módulos y herramientas

6.1 Realizar operaciones básicas del sistema de archivos y de la línea de comandos mediante el uso de módulos integrados

- `io`, `os`, `os.path`, `sys` (importación de módulos, uso de módulos para abrir, leer y comprobar la existencia de archivos, argumentos de línea de comandos)

6.2 Resolver problemas informáticos complejos mediante el uso de módulos integrados

- Matemáticas (`fabs`, `ceil`, `floor`, `trunc`, `fmod`, `frexp`, `nan`, `isnan`, `sqrt`, `isqrt`, `pow`, `pi`)
fecha y hora (`ahora`, `strftime`, día de la semana), aleatorio (`randrange`, `randint`, `aleatorio`, `aleatorio`, `elección`, `muestra`)

