

✓

¡Felicitaciones! ¡Aprobaste!

Calificación recibida

100 %

Para Aprobar

80 % o más

Ir al siguiente elemento

i

Estás viendo una versión traducida automáticamente de esta evaluación

Puedes volver a ver el contenido en su idioma original si lo prefieres. No perderás el progreso que hayas conseguido si cambias el idioma.

Mostrar la versión en Inglés

1.

¿Cuál de las siguientes NO se considera una operación cara?

1 / 1 punto

Análisis de un archivo

Descarga de datos a través de la red

Repasar una lista

Utilizar un diccionario

✓

Correcto

¡Impresionante! Utilizar un diccionario es más rápido para buscar elementos que recorrer una lista.

2.

¿Cuál de las siguientes puede ser la más costosa de llevar a cabo en la mayoría de las tareas de automatización de un script?

1 / 1 punto

Bucles

Listas

Vector

Hash

✓

Correcto

¡Gran trabajo! Los bucles que se ejecutan indefinidamente e incluyen subtareas que hay que completar antes de seguir adelante pueden resultar muy costosos para la mayoría de las tareas de automatización.

3.

¿Cuál de las siguientes afirmaciones representa el consejo más acertado a la hora de escribir guiones?

1 / 1 punto

Apunte a todas las ventajas de velocidad que pueda conseguir en su código

Utilizar a menudo operaciones costosas

Empiece escribiendo un código claro, luego acélerelo sólo si es necesario

Utilice los bucles con la mayor frecuencia posible

✓

Correcto

¡Impresionante! Si no notamos ninguna ralentización, entonces no tiene mucho sentido intentar acelerarlo.

4.

En Python, ¿qué es una estructura de datos que almacena varios datos, en orden, que pueden modificarse posteriormente?

1 / 1 punto

Un hash

Diccionarios

Listas

Tuplas

✓

Correcto

Muy bien Las listas son eficientes, y si estamos iterando a través de toda la lista o estamos accediendo a los elementos por su posición, las listas son el camino a seguir.

5.

¿Qué comando, palabra clave, módulo o herramienta puede utilizarse para medir el tiempo que tarda en ejecutarse una operación o un programa? (Marque todas las que correspondan)

1 / 1 punto

tiempo

✓

Correcto

¡Excelente! Podemos preceder el nombre de nuestros comandos y scripts con el "tiempo" shell builtin y el shell mostrará las estadísticas de tiempo de ejecución cuando se completen.

kcachegrind

✓

Correcto

¡Buen trabajo! La herramienta kcachegrind se utiliza para la visualización de datos de perfiles que, si podemos insertar algo de código en el programa, puede decirnos cuánto tiempo tarda la ejecución de cada función.

cPerfil

✓

Correcto

Gran trabajo! cProfile proporciona un perfil determinista de los programas Python, incluyendo la frecuencia y la duración de la ejecución de las distintas partes del programa.

romper