Test Técnico de Evaluación en Python para Automatización de Procesos

1.1 Explica la diferencia entre una lista y una tupla en Python. Proporciona un ejemplo

de cada una.

Las listas y tuplas permiten almacenar varios elementos en una sola variable. La principal diferencia es que las listas son mutables y las tuplas inmutables.

Ejemplo: //Código

1.2 ¿Qué es la indexación negativa en Python? Da un ejemplo de cómo se utiliza.

Colecciones de elementos como las listas o las tuplas pueden ser accedidos iniciando desde el final en lugar desde el principio, por ejemplo -1 se refiere al último elemento de la colección.

Ejemplo://Codigo.

2.1 Escribe un código en Python que imprima los números del 1 al 10 utilizando un

bucle for.

//Codigo

2.2 ¿Cuál es la diferencia entre if, elif y else en Python? Proporciona un ejemplo de su

uso combinado.

If ejecuta un bloque de código cuando se cumple una condición, por ejemplo, el numero X > 5, elif tiene la misma funcionalidad excepto que tiene que ser usado en conjunto con if y la condición solo se evalúa cuando una sentencia if o elif anteriores falsa, por último, else se ejecuta cuando ninguna de las condiciones codificadas se cumple.

Ejemplo://Codigo

3.1 Define una función llamada suma\_lista que reciba una lista de números como

argumento y devuelva la suma de todos los elementos de la lista.

//Codigo

3.2 Explica qué son los módulos en Python y cómo se utilizan.

Los módulos son archivos que contiene funcionalidad que deseamos incluir a nuestro código, pueden ser clases, funciones, sentencias, etc.

Para incluir un modulo hacemos uso de la sentencia import como se muestra en el ejemplo.

Ejemplo: //Codigo

4.1 Escribe un programa en Python que lea el contenido de un archivo llamado

datos.txt y lo imprima por pantalla.

//Codigo

4.2 ¿Qué es la serialización de datos en Python? ¿Para qué se utiliza?

La serialización es el proceso de convertir estructuras de datos como objetos o diccionarios a un formato que es mas sencillo de transmitir o almacenar, algunos formatos para la serialización que existen son JSON, XML o pickle en Python.

La serialización es usada a través de internet para transmitir datos (JSON) de forma que no requieran un gran ancho de banda, por ejemplo, la gran mayoría de las API sirven sus datos en este formato.

Por otro lado, Django REST framework utiliza la serialización de datos para convertir querys de base de datos a tipos de datos de Python que luego pueden ser renderizados como JSON o XML.

5.1 ¿Cuál es la función principal de la librería selenium en Python? Proporciona un

ejemplo básico de cómo automatizar una tarea en un navegador web usando

selenium.

La principal función es automatizar pruebas de aplicaciones web o tareas repetitivas como rellenar formularios, hacer clics o navegar entre páginas.

Ejemplo://Codigo

5.2 ¿Qué función tiene la librería openpyxl en Python? Proporciona un ejemplo de

cómo usarla para manipular una hoja de cálculo Excel.

La función de openpyxl es leer y escribir archivos de Excel, por ejemplo, se pueden escribir funciones, formatear hojas de cálculo o construir gráficos.

Ejemplo:

//Código por crear

6.1 Explica qué es una clase en Python y cómo se define.

Una clase es un tipo de plantilla o molde para crear objetos que contienen atributos y funcionalidad, pero cuyos valores pueden variar.

Una clase en Python se define con la palabra clave class seguida del nombre de la clase y dos puntos.

Ejemplo://Codigo

6.2 Describe el concepto de herencia en POO. Proporciona un ejemplo de cómo se

implementa en Python.

La herencia es un mecanismo que le permite a una clase (padre) compartir atributos o métodos a otras clases (hijas), en otras palabras, la herencia relaciona objetos de forma jerárquica para reutilizar y simplificar el código.

Ejemplo://Codigo

7.1 Imagina que necesitas automatizar la descarga de archivos CSV desde una URL

utilizando Python. Describe brevemente cómo abordarías este problema.

//Por contestar

7.2 ¿Qué son las PEP 8 en Python y por qué son importantes? Menciona al menos tres

pautas de codificación que recomienden las PEP 8.

Son una serie de convenciones sobre el cómo escribir código en Python. Tiene el objetivo de que el código sea legible. Son importantes por que ayudan a las personas que leen el código a comprenderlo con facilidad, saber cual es su utilidad y poder identificar errores rápidamente si se presentan.

Algunas pautas son:

* Los imports siempre deben de estar al principio del archivo.
* Las líneas deben de limitarse hasta 79 caracteres.
* Usar espacios alrededor de operadores aritméticos.