Ejercicios teóricos Python

1.- ¿Cuáles son los tipos de Datos en Python?

Los tipos de datos en Python son: Números, Booleanos, String, Bytes (también byte arrays), tipo None, Listas, Tuplas, Sets y diccionarios (muy usados en la gestión de datos para análisis, etc.)

2.- ¿Qué tipo de convención de nomenclatura deberíamos utilizar para las variables en Python?

En Python se usa la nomenclatura “snake case”, también conocido como “underscore case”. Se escriben todas las letras en minúsculas y se separan mediante el guion bajo.

3.- ¿Qué es un Heredoc en Python?

Los “Heredoc” en Python, y en prácticamente en el resto de lenguajes de programación, son strings multi líneas que pueden servir para mandar comandos a dispositivos remotos y ejecutarlos.

4.- ¿Qué es una interpolación de cadenas?

La interpolación de cadenas es cuando se añade un valor dinámico a, usualmente, un string fijo, de forma que el string puede ir variando dependiendo del valor de la variable externa. Usualmente se emplea cuando a un valor fijo hay que añadirle un input que ha insertado el usuario.

5.- ¿Cuándo deberíamos usar comentarios en Python?

Aunque a primera vista se esperaría que los comentarios deberían de estar por todo el programa, la realidad es hay gran cantidad de desarrolladores que piensa que el número de comentario en un programa debería de ser mínimo. El programa de debería entender mediante los nombres de variables, métodos, funciones, páginas, etc y los comentarios se deberían de usar de forma temporal para “subrayar” errores y futuros fixes del programa.

6.- ¿Cuáles son las diferencias entre aplicaciones monolíticas y de microservicios?

Las aplicaciones monolíticas son aquellas en las que la propia aplicación incluye todos los servicios que gestiona. Destacan por ser más rápidas tanto al desarrollar como en el funcionamiento del mismo. Pero el mantenimiento de la aplicación puede ser más costosa, ya que un pequeño cambio, puede llevar a la caída de todos los servicios.

Por otro lado, las aplicaciones de microservicios simplemente realizan la gestión de diferentes servicios independientes entre sí, de forma que minimizan los servicios a “hospedar”. Los basados en microservicios son más fáciles de mantener y si hay una caída de uno de los servicios, no afecta al resto por lo que se podrá seguir usando el programa. Como contra, se requiere de un mayor tiempo de inversión al desarrollarlo, ya que la compatibilidad

de los independientes no suele ser la mejor e incluso llevando a cabo un script de compatibilidad, seguramente tarde más que la aplicación monolítica.

Ninguna de las candidatas es un claro favorito para el desarrollo de las aplicaciones. Se debe de tomar la elección teniendo en cuenta las necesidades de la app a desarrollar. Pero, en términos generales, desde mi punto de vista a las aplicaciones grandes les beneficiaría un desarrollo de microservicio y a las pequeñas, en cambio, una monolítica.