

Prueba de Python

Cristhiam Daniel Campos Julca

12 de agosto de 2021

- Preguntas teóricas

1. ¿Qué es Python?

Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible. [1].

Las principales características de Python son, es un lenguaje **interpretado**, es decir, no es necesario compilar el código para su ejecución, ya que existe un intérprete que se encarga de leer el fichero fuente y ejecutarlo. Además, es un lenguaje **multiplataforma**. [5].

2. ¿Por qué Python es mejor que Java?

No se puede decir que un lenguaje es mejor que el otro, sino que depende de las necesidades del programador en ese momento.

Aunque Java es algo complejo a comparación de Python en la estructura, proporciona una mejor comprensión de la gestión de la memoria y es más seguro. Por otro lado, Python es corto, simple y fácil. Un principiante puede entender fácilmente un programa de Python porque está escrito en un inglés simple. Además tiene una sintaxis más legible, por lo que la curva de aprendizaje será menor en comparación que con Java. [3].

3. ¿Cuántos tipos de datos existen en el lenguaje Python?

Los tipos de datos básicos son los *booleanos*, los *numéricos* y las *cadenas de caracteres*. Dentro de los numéricos podemos clasificarlos en: *enteros*, *flotantes* y los *complejos*.

Además de los tipos básicos, otros tipos fundamentales de Python son las secuencias (list y tuple), los conjuntos (set) y los mapas (dict). Todos ellos son tipos compuestos y se utilizan para agrupar juntos varios valores.

- Las listas son secuencias mutables de valores.
- Las tuplas son secuencias inmutables de valores.
- Los conjuntos se utilizan para representar conjuntos únicos de elementos, es decir, en un conjunto no pueden existir dos objetos iguales.
- Los diccionarios son tipos especiales de contenedores en los que se puede acceder a sus elementos a partir de una clave única.

4. ¿Cuál es la diferencia entre *tuple* y *lista*?

En python, una tupla es una estructura de datos que representa una colección de objetos, pudiendo estos ser de distintos tipos. Para declarar una tupla se utilizan paréntesis, entre los cuales deben separarse por comas los elementos que van a formar parte de ella.

Por otro lado, una lista es una colección ordenada de objetos, similar al *array dinámico* empleado en otros lenguajes de programación. Para definir una lista se utilizan corchetes entre los cuales aparecen diferentes valores separados por comas.

A diferencia de las tuplas, los elementos de las listas pueden ser reemplazados accediendo directamente a través del índice que ocupan en la lista [5].

5. ¿Qué es PEP8?

PEP, Python Enhancement Proposal (Propuesta de Mejora de Python), es un conjunto de normas de estilo para estandarizar la escritura en este lenguaje, cuyo objetivo es ayudar a escribir un código más legible y abarcar desde cómo nombrar variables, al número máximo de caracteres que una línea debe tener.

“Code is read much more often than it is written”., Guido van Rossum.

6. ¿Qué es “pickling” y “unpickling”?

Con los algoritmos del módulo de *pickle* podemos serializar y deserializar estructuras de objetos de Python. “Pickling” se refiere

al proceso de convertir una jerarquía de objetos de Python en un flujo de bytes, y “unpickling” por otro lado es la operación inversa, es decir, el flujo de bytes se convierte de nuevo en una jerarquía de objetos.

7. ¿Qué es “lambda”?

Son funciones anónimas definidas en línea, que no serán referenciadas posteriormente. Se construyen mediante el operador **lambda**, sin usar el paréntesis para indicar los argumentos. Estos van directamente después del nombre de la función, que finaliza su declaración con dos puntos (:). Justo después, en la misma línea se escribe el código de la función [6].

8. ¿Cómo se administra la memoria dentro del lenguaje Python?

El administrador de memoria es el responsable del proceso de asignar, desasignar y coordinar la memoria de manera eficiente para los diferentes procesos que se ejecuten en Python. El administrador de memoria realiza un seguimiento del número de referencias a cada objeto en el programa; cuando el recuento de referencias de objeto cae a cero, el recolector de basura automáticamente libera la memoria de ese objeto en particular.

9. ¿Qué es “pass”?

Es una declaración que provoca que no se produzca ninguna operación, es decir, no pasará nada cuando se ejecute. A diferencia de los comentarios, éstos últimos son ignorados y no se ejecutan; en cambio, la sentencia pass se ejecutará sin que resulte nada.

10. ¿Puedes copiar un objeto en el lenguaje Python?

La operación de copiar objetos en Python varía según el tipo de objeto, teniendo en cuenta que las declaraciones de asignación siempre establecen un valor de referencia para un objeto, en lugar de copiar el objeto.

El módulo estándar `copy` permite crear copias de distintos objetos de Python, generalmente colecciones mutables (como las listas y los diccionarios) e instancias de clases, también mutables.

11. ¿Cómo borrar un archivo dentro de Python?

Para eliminar un archivo, se debe importar el módulo `OS` y ejecutar la función `os.remove()`. Se debe tener en cuenta, para evitar errores, la comprobación de la existencia del archivo antes de tratar eliminarlo.

12. ¿Qué es un “diccionario”?

Un Diccionario es una estructura de datos y un tipo de dato en Python con características especiales que nos permite almacenar cualquier tipo de valor como enteros, cadenas, listas e incluso otras funciones. Estos diccionarios nos permiten además identificar cada elemento por una clave (Key).

13. ¿Es Python un lenguaje de programación interpretado?

Sí. Un lenguaje interpretado es aquél que se puede ejecutar sin necesidad de ser compilado. Para ello, en lugar de un compilador, tenemos lo que se llama un *intérprete*, que lee el código y ejecuta las instrucciones en él contenidas. Al no haber compilación no existe un binario, y los programas escritos en lenguajes interpretados se suelen llamar *scripts*.

Como ventajas tiene el que son portables entre plataformas (siempre que dispongan del intérprete adecuado) y que no necesitan ser compilados cada vez que se modifican. [2].

14. ¿Cuál de ellos es un error?

Hay dos tipos de errores: errores de sintaxis y excepciones. Los primeros son conocidos como errores de interpretación; mientras que las excepciones son los errores detectados durante la ejecución, y no son incondicionalmente fatales.

15. ¿Cómo Python se considera un lenguaje orientado a objetos?

Según la definición, la programación orientada a objetos es un paradigma de programación en el que los programas se ven como formados por entidades llamadas objetos que recuerdan su propio estado interno y que se comunican entre sí mediante el paso de mensajes que se intercambian con la finalidad de:

- cambiar sus estados internos,
- compartir información y
- solicitar a otros objetos el procesamiento de dicha información.

El sistema de objetos de Python proporciona una sintaxis cómoda para promover el uso de estas técnicas de organización de programas [4].

16. ¿Qué es “slicing”?

El slicing es la operación por el cuál se extraen elementos de una secuencia, tal como una lista o una cadena de caracteres. Dependiendo del caso, los elementos podrían ser consecutivos o podrían estar separados dentro de la secuencia original.

17. Según la guía de estilos PEP8 ¿cómo deben escribirse las constantes?

Las constantes deben escribirse en mayúscula y usar la raya al piso para evitar espacios entre palabras.

18. ¿Qué es virtualenv y cómo lo inicias con un interpretador de Python específico?

Virtualenv es una herramienta que nos permite tener entornos Python totalmente separados y aislados, de manera que el intérprete y las librerías de proyectos diferentes no entren en conflicto.

Virtualenv guarda cada entorno virtual en un directorio con el nombre de ese entorno. Dentro de ese directorio se guardarán todos los archivos necesarios (de ellos nos interesan, en particular,

los módulos que instalemos en el entorno y el script que lo inicia).

■ Ejercicio práctico

En el correo, se adjunta un archivo JSON con tweets. Debes leer el archivo en Python y obtener las siguientes métricas

1. Numero de tweets
2. Lista de los usuarios autores de los tweets (tip: el nombre de usuario está en `user > screen_name`)
3. Totalizador del numero de seguidores de todos los usuarios (tip: los seguidores están en `user > followers_count`)
4. Obtén una lista de todos los usuarios mencionados en todos los tweets
5. Obtén el total de tweets con tendencia 0, 1 y 2 respectivamente

Bibliografía

- [1] CHALLENGER-PÉREZ, I., DÍAZ-RICARDO, Y., AND BECERRA-GARCÍA, R. A. El lenguaje de programación python. *Ciencias Holguín* 20, 2 (2014), 1–13.
- [2] GUTIÉRREZ, Á. P. H. *Python paso a paso*. Grupo Editorial RA-MA, 2016.
- [3] KHOIROM, S., SONIA, M., LAIKHURAM, B., LAISHRAM, J., AND SINGH, T. D. Comparative analysis of python and java for beginners. *Int. Res. J. Eng. Technol* 7 (2020), 4384–4407.
- [4] LÓPEZ, L. Programación orientada a objetos, 2006.
- [5] MONTORO, A. F. *Python 3 al descubierto*. Rc Libros, 2012.
- [6] RODÓ, D. M. El lenguaje python.