

Curso Hackea Tu Futuro – Python Granada (Módulo 4)

Nombre: Sergio

Apellidos: Caruncho Muñoz

Fecha: 06/10/23

Responde a las siguientes preguntas. Justifica la respuesta.

1. Tipos de archivos: ¿Qué es un archivo JSON? ¿Y un archivo CSV?

-Archivos de formato abierto y estructurados para compartir datos entre un servidor y una app. Es el formato estándar de transferencia de archivos.

-Archivo de texto estructurado para usarse en Excel y pandas.

2. ¿Qué es un ETL?

ETL es un proceso, que proviene del anglicismo y acrónimo **“extract, transform and load (extracción, transformación y carga)”**, utilizado en el campo de la informática con la idea de integrar y consolidar datos para un único sistema de destino.

En este proceso, se extraen datos de una o varias fuentes, las cuales pueden estar en diferentes formatos y almacenados en distintos tipos de repositorios. El objetivo principal del proceso es asegurar que **los datos sean precisos, completos y coherentes entre sí antes de ser utilizados** y tratados en análisis o replicados para el uso de aplicaciones.

Una buena transformación de los datos durante el proceso ETL debe incluir la conversión de los formatos, la eliminación de duplicados, la normalización, la aplicación de reglas y/o la agregación de nuevas variables.

3. ¿Para qué sirve el comando pip? ¿Y el comando import?

PIP: Para instalar una librería en el ordenador. Import: Para indicar que quieres usar esa librería.

4. Enumera las características de Python explicando cada una de ellas (menciona 4 al menos).

Python es un **lenguaje de programación de alto nivel**, que se ha convertido en una de las opciones más populares para desarrolladores de software en todo el mundo. Sus principales características son:

1. Es un **lenguaje simple y claro**, esto hace que Python sea más fácil de entender y de usar para programadores a cualquier nivel.

2. Es un **lenguaje expresivo y legible**, los desarrolladores pueden escribir código que se asemeja mucho al lenguaje natural, lo que hace que sea más fácil para otros programadores que no trabajan en este sistema entender lo que otros están haciendo.
3. Tiene una **sintaxis elegante y concisa**, lo que hace que su código sea fácil de leer y de escribir. Python ha sido diseñado para ser intuitivo y fácil de usar.
4. Tiene un **tipado dinámico y fuerte**, lo que significa que los tipos de datos se comprueban en tiempo real de ejecución así como posibles errores a la hora de generar código.
5. Es **multiparadigma**, soporta varios estilos de programación. Los programadores pueden usar Python para escribir código orientado a objetos, funcional o procedimental, lo que lo hace ideal para diferentes tipos de proyectos.
6. Es **interpretado**, lo que significa que no es necesario compilar su código antes de ejecutarlo. Esto hace que el proceso de desarrollo sea más rápido, ya que los desarrolladores ven los resultados del código inmediatamente.
7. Es **multiplataforma**, es compatible con diferentes sistemas operativos y plataformas, haciéndolo ideal para desarrolladores que trabajan en diferentes entornos sin necesidad de modificar el código.

5. **¿Cuáles son los tipos primitivos en Python y qué valores pueden contener cada uno de ellos? Pista: Son 3 tipos.**

En Python, existen tres tipos primitivos de datos:

- **Numéricos**, este tipo de dato representa números enteros sin decimales. Pueden ser positivos o negativos y no tienen límites de tamaño en Python. Dentro de ellos contamos con,
 - **enteros (int) sin decimales (1, 10, -15)**. Este tipo de dato representa números enteros sin decimales. Pueden ser positivos o negativos y no tienen límites de tamaño en python,
 - **y flotantes (float) con decimales, (3.14, 2.718, -0.5)**. Este tipo de dato representa números con decimales. Los flotantes en python se escriben con un punto decimal y pueden ser positivos o negativos.
- **Booleanos (bool)**, se representan con los valores, **verdadero (True) y falso (False)**. Los booleanos son útiles en la programación para tomar decisiones y controlar el flujo del programa.
- **String (str)**, este tipo de dato representa cadenas de texto. Las cadenas de texto en Python se escriben entre comillas simples o dobles, y pueden contener letras, números, signos de puntuación y espacios. Ejemplos de cadenas de texto son **"hola", "Python me mola", "123"**.

6. Menciona alguna estructura de datos más compleja que los tipos primitivos que conozcas.

Una de las estructuras de datos más compleja en Python es el **diccionario (dictionary)**, que se utiliza para almacenar una colección de pares clave-valor. Un diccionario en Python es una estructura de datos no ordenada y mutable que se compone de una serie de elementos, cada uno de los cuales tiene una clave única y un valor asociado.

7. Escribe la sintaxis para crear variables.

-viejos_del_pueblo : ["Aureliano", "Rigoletto", "Hermenegilda", "Canuta"]

```
1 nombre_variable = valor_variable
```

8. Escribe la sintaxis para crear funciones.

-nombre_funcion():

9. Escribe la sintaxis para llamar a variables.

-print(nombre_variable)

10. Escribe la sintaxis para llamar a funciones.

-nombre_funcion(nombre_variable)

11. Explica con tus palabras para qué sirven los condicionales y los bucles en Python.

-Condicionales: ejecutan un código cuando se cumplan las condiciones propuestas.

-Bucles: una función que recorre el código tantas veces como sea necesarias hasta que encuentre el motivo dado para que salga del bucle (*break*) o hasta el infinito si no se da ningún *break*.

12. Explica con tus palabras para qué sirven las librerías: Pandas y Numpy.

Numpy es una librería que proporciona soporte para cálculos científicos y matemáticos en Python. Es muy utilizada para la manipulación de matrices y arreglos multidimensionales. Con Numpy, se pueden realizar operaciones numéricas eficientes y complejas, como la multiplicación de matrices, operaciones estadísticas y algebraicas, y otras operaciones matemáticas.

Por otro lado, **Pandas es una librería de análisis de datos que proporciona estructuras de datos de alta eficiencia, como dataframes y series para manipular y analizar datos.** Pandas es muy útil para explorar, analizar y transformar datos, y se puede utilizar para tareas como la limpieza de datos, el filtrado, la transformación y la agregación de datos. Además, Pandas tiene una serie de herramientas para la

visualización de datos y para el manejo de datos faltantes. Pandas también permite integrar los datos de múltiples fuentes de datos, como bases de datos, archivos CSV, archivos Excel, etc.

13. ¿Cómo representamos el valor vacío en Python?

None.

14. ¿Qué es un IDE?

Un IDE en Python **es un programa que proporciona herramientas para facilitar el desarrollo de software**. En particular, el IDE de Python ofrece características como la edición de código, la depuración, la ejecución y la gestión de proyectos, todo en una sola interfaz gráfica de usuario.

15. ¿Qué es el CRISP-DM?

Es una **metodología** de procesos estándar **para la minería de datos**, es decir, para el proceso de descubrir patrones y relaciones en grandes conjuntos de datos.