

M2C3 QUESTIONS

- ¿Cuáles son los tipos de Datos en Python?

1. Palabras: str "Hola que tal"
2. Numéricos:
 - a. int (enteros 32)
 - b. float(racionales 42.2)
 - c. complex(complejos 3+2j)
3. Listas:
 - a. list(lista flexible) ["jose", 23, True]
 - b. tuple(lista inmutable)
 - c. range(lista de números consecutivos) range(6) -> [0,1,2,3,4,5]
4. Diccionarios: dict(pares de elementos) { "Name": "Pepe", "edad": 64}
5. Sets:
 - a. set (estructuras no ordenadas sin índice)
 - b. frozenset (similar al set pero son inmutables y pueden usarse para elementos en otros conjuntos)
6. Booleanos: True o false
7. Bytes y bytes arrays: conjuntos de tipo binario (01000111).
8. Memoryview: Elementos que se suelen usar con bytes para hacer operaciones y procesos compartidos.
9. None: Se define como la falta de un valor en el tipo.

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

Se recomienda seguir el PEP 8 pero no es una obligación. Principalmente utilizar _ y lowercase para variables y capitalización en clases (CapitalizedWords) para tener cierto orden.

También evitar el uso de los caracteres "I" (i mayúscula), "l" (L minúscula) y "O" (o mayúscula) ya que pueden dar confusión con otros caracteres.

- ¿Qué es un Heredoc en Python?

Es una palabra multilínea que crea con """ (tres dobles comillas) o '' (tres simple comillas) al principio y al final, permitiendo añadir todo un texto y mantener su formato.

- ¿Qué es una interpolación de cadenas?

Es un sistema para poder introducir variables de manera dinámica en palabras. Para Python hay 3 estilos. Los de este curso f-string y .format() y el sistema %s para sustituir variables muy parecido a .format(), más antiguo heredado de otros lenguajes.

- ¿Cuándo deberíamos usar comentarios en Python?

Cuando se consideren beneficiosos, lo cual es mayormente de tipo opinión y dividido en la comunidad. Como normal general, para organizar partes del código suele ser positivo y para hacer algún apunte sobre alguna función en específico.

También es muy típico hacer al principio de una función para explicar la acción que realiza.

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

Las aplicaciones monolíticas están construidas en el mismo entorno y por lo tanto muy interconectadas entre sí. Las de microservicios se dividen en diferentes “apartados” (servidores, aplicaciones, lenguajes, etc.) y requieren de comunicación entre ellas.