

M2C1 QUESTIONS

-¿Cuáles son los tipos de Datos en Python?

Palabras -> str “Hola que tal”

Numericos -> int (enteros 32), float(racionales 42.2) y complex(complejos 3+2j)

Listas -> list(lista flexible), tuple(lista inmutable), range(lista de números consecutivos)

Diccionarios -> dict(pares de elementos)

Sets -> set(estructuras no ordenadas sin índice) y frozenset(similar al set son inmutables y pueden usarse para elementos en otros conjuntos)

Booleanos -> True o false

Bytes y bytes arrays-> conjuntos de tipo binario.

Memoryview -> Elementos que se suelen usar con bytes para hacer operaciones y procesos compartidos.

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 “”” o ”” 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 que ya conocía de %s.

-¿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.