

CHECKPOINT 3

QUESTION 1

Statement

¿Cuáles son los tipos de datos en Python?

Answer

- Booleans (bool): Representan valores de verdad (True, False).
- Numbers (int, Float, complex): Números enteros, flotantes y complejos.
- Strings (str): Cadenas de texto.
- Bytes (bytes): Secuencias de bytes.
- None (NoneType): Representa la ausencia de un valor.
- Lists (list): Colección ordenada y mutable de elementos.
- Tuples (tuple): Colección ordenada e inmutable de elementos.
- Sets (set): Conjunto no ordenado de elementos únicos.
- Dictionaries (dict): Colección no ordenada de pares clave-valor.

Examples

```
myBool = True
myInt = 5
myFloat = 5.5
myString = "hello"
myByte = b"hello"
myNone = None
myList = [1,2,3,4,5]
myTuple = (1,2,3,4,5)
mySet = {1,2,3,4,5}
myDictionary = {"name": "Python", "type": "language"}
```

QUESTION 2

Statement

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

Answer

Python sigue la convención de snake_case para los nombres de variables y funciones, y CamelCase para los nombres de clases. Dependiendo del ámbito (global, local) y estado (público, privado) en el que se encuentran estos elementos, la nomenclatura también puede variar.

Estas son las reglas a la hora de nombrar variables:

- Los nombres deben comenzar con una letra o un guion bajo (_)
- Se usa guion bajo (_) para separar palabras
- No deben comenzar con números ni usar caracteres especiales

Examples

```
variable = "language"  
another_variable = "Python"  
_private_variable = "C#"  
1variable = "Error" # No puede comenzar con un número
```

QUESTION 3

Statement

¿Qué es un Heredoc en Python?

Answer

Un heredoc es en definitiva una cadena de texto multilinea. El concepto Heredoc se usa en muchos lenguajes de programación y la idea es la utilización de varias líneas de texto.

Examples

```
lorem = """  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
  
Curabitur ullamcorper quam a ex efficitur, non faucibus leo luctus.  
  
Pellentesque bibendum vitae ipsum vitae aliquam.  
"""
```

QUESTION 4

Statement

¿Qué es una interpolación de cadenas?

Answer

La interpolación de cadenas es la forma de insertar valores dentro de un string. La principal ventaja es que permite usar variables dentro de la cadena de texto.

Se puede hacer de varias formas:

- f-strings
- format()

Examples

```
name = "Iker"
nickname = "Karko"

# f-string
print(f"Mi nombre es {name} pero me suelen llamar {nickname}.")

# format()
print("Mi nombre es {1} pero me suelen llamar {2}.".format(name, nickname))
```

QUESTION 5

Statement

¿Cuándo deberíamos usar comentarios en Python?

Answer

Los comentarios se usan para documentar el código y explicar su funcionamiento. A pesar de que son muy útiles, lo principal es siempre poner nombres que sean identificativos a las clases, funciones y variables, de tal forma que se expliquen por si mismas.

Algunos usos útiles de los comentarios:

- Describir en profundidad funciones y clases
- Marcar TODOs o FIXMEs.
- Deshabilitar temporalmente partes del código

Examples

```
# Comentario explicativo general de la función

def suma(a, b):
    """Tareas específicas de la función."""
    sum = a + b # Suma las variables
    return sum # Devuelve el valor
```

QUESTION 6

Statement

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

Answer

Aplicaciones monolíticas: Toda la aplicación está en un solo bloque de código. Esto hace que sean más fáciles de desarrollar, pero menos escalables. Recomendado para aplicaciones pequeñas y que no requieran de un cambio constante en el código.

Aplicaciones microservicios: La aplicación está dividida en múltiples servicios independientes. Permite una mayor escalabilidad y es más flexible. Más usado en aplicaciones grandes, en las que es recomendable mantener los servicios por separado para que si algo va mal no afecte a toda la app.