

PREGUNTAS TEORICAS CHECKPOINT 3

¿Cuáles son los tipos de Datos en Python?

-Boolean → representan verdadero o falso

-Numérico → hay 3 tipos: int, float y complex

Int: es un número entero, positivo o negativo, sin decimales y con longitud ilimitada. EJ: 1997

Float: es un número positivo o negativo que tienen uno o más decimales. También pueden ser números científicos llevando una “e” para indicar potencia de 10 EJ: 25.95, 10e2 (1000.0)

Complex: los números complejos están compuestos por dos partes, una real (55) y otra imaginaria(5j), van acompañados de una “j”.

-Strings → puede ser desde un nombre hasta un documento completo de html. Un string siempre va “rodeado” de comillas, estas comillas pueden ser simples ‘ ’ o dobles “ ” EJ: “Esto es un ejemplo”.

-None → asigna un valor “ninguno”. Es un tipo de dato propio y solo None puede ser None. Este valor se puede modificar mas adelante en caso de ser necesario.

-Estructura de datos → hay 4 tipos: lista, tuple y sets y diccionario

Lista: las listas se utilizar para almacenar varios elementos en una única variable. Es una colección ordenada. Se crean utilizando corchetes [“coche”, “moto”, “bici”, “patín”]. Estos elementos son modificables y permiten valores duplicados. El primer elemento de la lista tendrá el índice(index) [0]

Tupla: las tuplas se utilizar para almacenar varios elementos en una única variable. Es una colección ordenada. Se crean utilizando paréntesis (“coche”, “moto”, “bici”, “patín”). Esta lista es inmutable, es decir, no se puede modificar y permite valores duplicados. El primer elemento de la tupla tendrá el índice(index) [0].

Sets: los sets se utilizar para almacenar varios elementos en una única variable. Es una colección desordenada, es decir, los elementos pueden aparecer en un orden diferente cada vez que los usemos, por lo tanto, no podemos buscarlos mediante índice(index). Se crean utilizando llaves {“coche”, “moto”, “bici”, “patín”}. Esta lista es inmutable y no permite valores duplicados.

Diccionario: los diccionarios se utilizan para almacenar valores de datos en pares clave: valor. Es una colección ordenada, modificable y no permite duplicados. Se crean utilizando llaves {"marca": "MSI",
"modelo": "GP Leopard",
"color": "negro"
}

Se puede hacer referencia a ellos utilizando la clave, que en este caso sería la marca, el modelo y el color.

-Tipos binarios → bytes y byte arrays. Ambos se usan para trabajar a nivel de bytes pero tienen algunas diferencias.

Bytes: es una secuencia inmutable de bytes. Solo permite caracteres ASCII

Bytes array: es un tipo mutable de bytes

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

Snake case.

Snake case es una convención de nombres de variables en la que cada palabra va escrita en minúscula y separada por guiones bajos (_).

¿Qué es un Heredoc en Python?

Son strings de múltiples líneas. Se crean poniendo 3 comillas (simples o dobles) al inicio y al final de la variable EJ: heredoc= """

```
    Lorem ipsum dolor sit amet,  
    consectetur adipiscing elit,  
    sed do eiusmod tempor incididunt  
    ut labore et dolore magna aliqua.  
    """
```

Podemos usar .strip tras las comillas finales para evitar dejar línea de código en blanco ("".strip) o si necesitamos dar espacio podemos usar \n. EJ: (Lorem ipsum dolor sit amet,\nconsectetur adipiscing elit,).

¿Qué es una interpolación de cadenas?

Es una técnica que permite insertar valores o variables a un string si hay marcadores de posición presentes. Los valores de las variables reemplazan los marcadores de posición. Esto ayuda a reducir la repetición de código. Se utiliza cuando los elementos del string aún no se conocen o podrían cambiar. Hay 2 métodos:

-f-strings→ se utiliza añadiendo la letra “f” delante del string y poniendo las variables entre llaves {} EJ: nombre:”Janire”

```
edad=27
```

```
saludo= f“Hola {nombre}, acabas de cumplir {edad} años!”
```

.format→se utiliza añadiendo .format al final de string, lleva entre paréntesis las variables que queremos usar. EJ: nombre:”Janire”

```
edad=27
```

```
saludo= “Hola {0}, acabas de cumplir {1} años!”.format(nombre, edad)
```

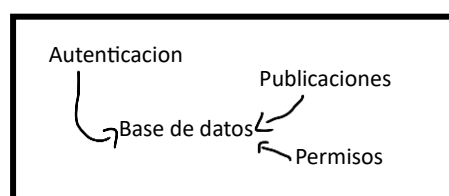
El 0 hace referencia a la primera variable “nombre” y el 1 a “edad”.

¿Cuándo deberíamos usar comentarios en Python?

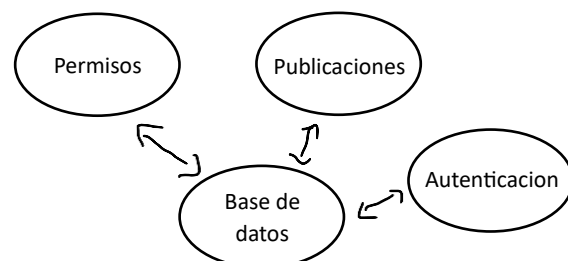
Los comentarios deberíamos usarlos para organizar nuestro código, estos comentarios tienen que ser claros y evitar dar explicaciones del código con ellos, ya que si en algún momento cambia, ese comentario quedaría obsoleto y podría crear confusiones y errores.

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

APLICACIÓN MONOLITICA	APLICACIÓN DE MICROSERVICIOS
La función de todo el servicio se basa en una sola aplicación	Cada característica tiene su propia aplicación
Comunicación interna, dentro de la aplicación monolítica	Comunicación externa, a través de APIS
Se tiene que desplegar toda la aplicación	Se puede desplegar individualmente
Los cambios y que se realicen pueden afectar a toda la aplicación	Los cambios y ampliaciones de un servicio no afectan a otros
Se usa un único lenguaje de programación	Usa diferentes tecnologías y lenguajes.



Esta todo dentro de la misma “maquina



Cada elemento esta en su servidor