

Preguntas teoricas de checkpoint 3

¿Cuáles son los tipos de Datos en Python?

Hay nueve tipos de datos principales que se pueden usar para escribir una variable:

- Cadenas (strings)
- Nùmericos (numbers)
- Listas (lists)
- Conjuntos (sets)
- Booleanos (boolean)
- Byte y byte arrays
- Tuplas (tuples)
- Diccionarios (dictionaries)
- None

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

- Poner barra baja entre palabras “tal_que_así” ya que las variables pueden alargarse y juntando las palabras puede hacer la lectura mas difícil junto con poder confundirlo con una clase en vez de una variable. Esto puede llevar a molestos bugs.
- Evitar usar “L” en minúscula, “O” en mayúscula o “I” en mayúscula. Puesto que estos caracteres son fáciles de confundirse con otros caracteres.

¿Qué es un Heredoc en Python?

Un heredoc es un variable en cadena cuya largura es superior a una sentencia. Predeterminadamente Python solo escanea una sentencia en cadena al ser insertada como variable. Por eso hay que dar la instrucción de crear un Heredoc mediante el uso de tres sets de comillas acompañado de un .strip()

```
1 content = """Mi campeón favorito, sin duda alguna Anivia, la criofenix.  
2 No tendrá mucho ataque, no tendrá mucha defensa... pero me encanta.  
3 Es muy difícil de controlar, sobre todo la Q, porque tienes que petarla/controlar la distancia muy bien para stunear.  
4 Y la W, porque si no es en vez de ayuda.""".strip()  
5 print(content)  
6
```

¿Qué es una interpolación de cadenas?

Es una manera sencilla de insertar variables dentro de una cadena. Requiere algo de preparación de antemano para crear las variables relevantes. Para que python reconozca que

esto es una interpolacion se incluye una “f” antes de que empiece la cadena interpolada y se incluyen las variables dentro de parentesis dobladas .

```
nombre = "Alexia"
dieta = "vegetariana"
pedido = f"El cliente {nombre} tiene una dieta {dieta}."
print(pedido)
```

```
El cliente Alexia tiene una dieta vegetariana.
```

```
[Done] exited with code=0 in 0.229 seconds
```

¿Cuándo deberíamos usar comentarios en Python?

En si se pueden usar cuando se quieran. Pero se recomienda usarlos para auto documentarse, mantener un seguimiento de los segmentos importantes y anotar modificaciones para hacer más tarde en el desarrollo (usando TODO para dejar claro que el comentario menciona algo pendiente). Los comentarios pueden ser de una sola línea, insertadas tras una variable bien sellada o multi línea. En el caso de una sola línea se usa “#” para empezar el comentario, Multiple se usan las 3 comillas al principio y final.

```
#Esta sola linea no crea codigo!

receta = 'Moussaka' # TODO: Poner etiquetas

print (receta)

"""
Esto tendra multiples lineas.
Hay que dejar a las letras respirar.
¿No crees?
"""
```

```
Moussaka
```

```
[Done] exited with code=0 in 0.166 seconds
```

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

Las *aplicaciones monolíticas* estan basadas en tener un sistema cuya función se basa en una sola aplicación. Esto tiene como ventaja que la aplicación se puede desarrollar y

funciona más rápido que una aplicación de microservicios. Sin embargo hay que ser más cuidadoso al hacer mantenimiento ya que si algo no está bien diseñado es posible que un solo cambio cause un efecto tipo domino que puede romper la aplicación.

Las *aplicaciones de microservicios* están basadas en tener múltiples servicios distintos dentro de él que se pueden desarrollar y que se ejecutan en procesos separados. A la hora de escalar cuentas con la libertad de escalar servicios concretos y testarlos independientemente. Esta manera de funcionar le da la ventaja de que si un servicio “cae” el resto de la aplicación puede funcionar. Pero, tiene varias desventajas entre ellas el tiempo de desarrollo siendo más largo debido a tener que configurar cada componente independientemente y la posibilidad de poder tener problemas de compatibilidad entre los componentes.