

¿Cuáles son los tipos de Datos en Python?

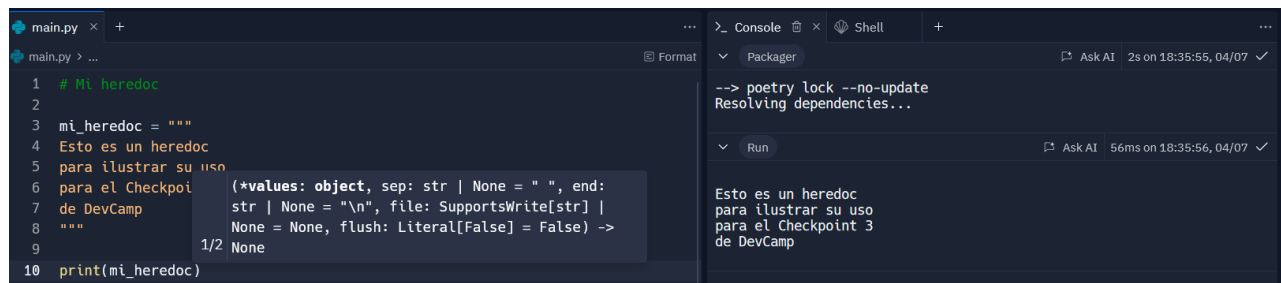
- String, Integer, Float, List, Dictionary, Tuple, Boolean

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

- Intentar que los nombres de las variables sean autodescriptivas o, en su defecto, lo más descriptivas posibles. En caso de que el nombre de la variable contenga más de una palabra, estas dos deberán estar separadas por una barra baja. Esto último es una convención del lenguaje de programación “Python”, ya que en otros lenguajes de programación se pueden (y se recomienda) usar otras formas.

¿Qué es un Heredoc en Python?

- El Heredoc significa un string que ocupa más de una línea de código y que por lo tanto debe tener otra forma de ser declarada que una cadena de texto normal. En Python se usan las triples comillas para este proposito `"""Heredoc"""`.



```
1 # Mi heredoc
2
3 mi_heredoc = """
4 Esto es un heredoc
5 para ilustrar su uso
6 para el Checkpoint
7 de DevCamp
8 """
9
10 print(mi_heredoc)
```

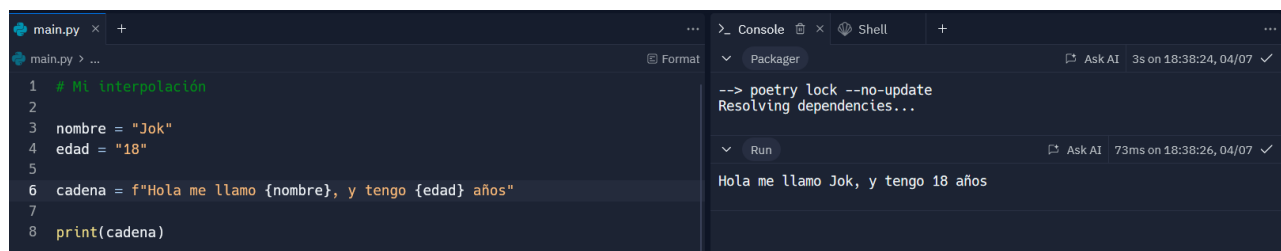
```
--> poetry lock --no-update
Resolving dependencies...

Run

Esto es un heredoc
para ilustrar su uso
para el Checkpoint 3
de DevCamp
```

¿Qué es una interpolación de cadenas?

- Cuando escribimos en código son numerosas la veces en las que deberemos incluir variables o código en cadenas de texto. El propósito de las interpolaciones de cadenas es poder llevar a cabo esta tarea. La denominación en python es la siguiente `f'Hola {nombre}'`. Esto nos permite poder crear cadenas de texto dinámicas, con utilidad por ejemplo para la creación de cadenas de texto para petición de APIs



```
1 # MI Interpolación
2
3 nombre = "Jok"
4 edad = "18"
5
6 cadena = f"Hola me llamo {nombre}, y tengo {edad} años"
7
8 print(cadena)
```

```
--> poetry lock --no-update
Resolving dependencies...

Run

Hola me llamo Jok, y tengo 18 años
```

¿Cuándo deberíamos usar comentarios en Python?

- Deberíamos recurrir a los menos comentarios posibles y solamente hacer uso de ellos cuando, a través del código, no podemos describir los procesos y necesitamos de ayuda complementaria.

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

- La principal diferencia es que una aplicación monolítica reside íntegramente en un solo servidor y todos los servicios dependen del servidor sin que puedan funcionar de forma independiente, teniendo la principal desventaja de que la aplicación es probable que se pare completamente cuando existe un bug, aunque no sea muy grande. Sin embargo, los microservicios están alojados cada uno en un servidor exclusivo y, aunque los servicios están conectados y son interdependientes, si uno de los servicios falla otras partes de la aplicación pueden seguir funcionando. La principal desventaja es el mayor costes de los microservicios frente a una aplicación monolítica, tanto en tiempo como en dinero.