

¿Cuáles son los tipos de Datos en Python?

Estos son los tipos de datos más comunes:

- Booleans: los valores son True o False. Se suelen utilizar mucho para funciones de condición.

```
jubilado = False
```

- Números:

```
total = 100
```

- Cadenas (string): se pueden representar con (') o (").

```
nombre = 'Beñat'
```

- None: cuando es nulo o sin valor.

```
resultado = None
```

- Listas: conjunto de elementos ordenados. Permiten mezclar diferentes tipos de datos (cadenas, números...)

```
lista = ['a', 'b', 3, 'd']
```

También hay otros tipos de datos como "bytes", "tuples", "sets", y diccionarios.

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

Deberíamos utilizar la sugerida en PEP 8, que es una guía de estilos para el código Python, donde se recomiendan estándares de nomenclatura y también recomiendan nomenclaturas a evitar.

¿Qué es un Heredoc en Python?

Se utilizan para definir cadenas multilíneas de una manera estructurada. Para ello, se colocan tres comillas (""" o ''') al principio y al final de las cadenas definidas.

```
content = """
Nullam id dolor id nibh ultricies vehicula ut id elit. Nullam quis risus eget urna mollis ornare
vel eu leo.

Vestibulum id ligula porta felis euismod semper. Cum sociis natoque penatibus et magnis dis
parturient montes, nascetur ridiculus mus. Cras justo odio, dapibus ac facilisis in.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
"""
```

¿Qué es una interpolación de cadenas?

Es una forma de insertar valores previamente definidos dentro de una cadena.

Ejemplo:

```
nombre = 'Beñat'
plazo = 4

plazo_ejercicio = f"Hola {nombre}, el plazo de entrega es de {plazo} días"

print(plazo_ejercicio)
```

¿Cuándo deberíamos usar comentarios en Python?

Se deben utilizar para dar una breve explicación del código utilizado, para entender mejor lo que se hace o para qué. Debe ser un comentario breve y claro.

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

Ambas muestran cómo se ha desarrollado una aplicación. Las monolíticas funcionan como una única unidad, por lo que todos los elementos están interconectados dentro de la misma estructura. Esto hace que si hay un error en una parte del código, pueda afectar a toda la aplicación. Las aplicaciones microservicios, en cambio, se dividen en elementos independientes que se comunican entre sí. En este caso, si hubiera algún error en una parte del código, afectaría únicamente al elemento afectado.