

M2C3 Preguntas

1. ¿Cuáles son los tipos de Datos en Python?

Los tipos de datos de python son Boolean, Number, String, None, Binarios (byte, byte arrays...) y Collections (el más utilizado siendo List).

```
boolean: True
number = 5
string = "This is a string."
none = None
list = ('magpie', 'blackbird', 'robin', 'crow')
```

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

Las variables deben escribirse en el llamado “Snake Case”, todo en minúscula y usando barra baja (_) para separar las palabras.

```
this_is_how_you_name_a_variable = "Hello World!"
```

3. ¿Qué es un Heredoc en Python?

Un Heredoc es un string multi-línea. La forma más común de crear una es escribiendo un bloque de texto con varias líneas y añadiendo “""" al principio y al final de esta.

```
heredoc = """
Lorem ipsum dolor sit

Lauren epsum solo hit

Dougus dippus deltoid dump

Crampus krungus forrest gump
"""
```

4. ¿Qué es una interpolación de cadenas?

Una interpolación de cadenas (string interpolation) es una forma de insertar variables o código python (como operaciones aritméticas) dentro de un string. Una de las formas más sencillas de hacerlo es usando un f-string; añades una “f” antes de las comillas de un string y después insertas la variable o el código dentro de unos brackets {}.

```
variable = "Dani"

variable_interpolation = f"My name is {variable}."

print(variable_interpolation)
```

```
My name is Dani.
```

```
operation_interpolation = f"I'm {2025-1999} years old."

print(operation_interpolation)
```

```
I'm 26 years old.
```

En el caso de querer poner algo entre brackets que no quieras que python interprete como algo a insertar, tendrás que poner doble brackets.

```
sentence = f"Please ignore these {{brackets}}."

print(sentence)
```

```
Please ignore these {brackets}.
```

5. ¿Cuándo deberíamos usar comentarios en Python?

Aunque haya diferentes opiniones al respecto, se considera una buena práctica utilizar la menor cantidad de comentarios posible y utilizar nombres lo más descriptivos posibles en su lugar.

Uno de los casos en los que sí es interesante usar comentarios es para añadirle "TODO"s al código. Estos "TODO"s sirven como recordatorio para tareas o cambios que quieres realizar más adelante.

```
name = "Maite" #TODO: change name later
```

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

Las aplicaciones monolíticas se compilan como una unidad unificada, todas las funciones están en el mismo servidor y utilizan el mismo sistema de archivos.

Su ventaja principal es que son más baratos y fáciles de implementar.

En las de microservicios las diferentes funciones están en entidades separadas con su propio servidor, y después utilizan APIs para conectarse entre ellas.

Son más fáciles de mantener, y se pueden actualizar o escalar funciones específicas en vez de la toda aplicación completa. Esto también permite que en el caso de que una de las funciones falle, el resto de la aplicación siga funcionando.