

M2C3 – Preguntas teóricas Python

-¿Cuáles son los tipos de Datos en Python?

Los tipos de datos en Python son string (cualquier tipo de secuencia alfanúmerica representada entre comillas en el código), números, booleans (valores binarios de true or false), “none” como concepto de valor aún sin establecer, y tipos que se explicarán más adelante en el curso como son los bytes y byte arrays, para controlar secuencias de bytes, y lists, tuples, sets y diccionarios, que nos permiten manejar “colecciones” en Python.

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

Deberíamos utilizar nombres claros, descriptivos y concisos, utilizando minúsculas y separando las palabras con guiones bajos (Snake case), por ejemplo “my_variable_name” en lugar de “MyVariableName”. Podemos encontrar todas las convenciones en el manual de estilo PEP 8 de Python.

-¿Qué es un Heredoc en Python?

Un heredoc es un string que ocupa varias líneas. Para usarlo, ponemos tres comillas, tal que así, por ejemplo:

```
"""
```

```
Esto
```

```
Es
```

```
un
```

```
heredoc
```

```
"""
```

-¿Qué es una interpolación de cadenas?

Una interpolación de cadenas nos permite añadir a una cadena/string otros tipos de datos que normalmente no serían compatibles, como son por ejemplo los números.

-¿Cuándo deberíamos usar comentarios en Python?

Como normal general, deberíamos utilizar comentarios cuanto menos posible, utilizando en su lugar nomenclatura clara y descriptiva para el código que estemos escribiendo, dejando los comentarios para instancias en las que realmente sean absolutamente necesarios.

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

Las aplicaciones monolíticas incluyen todos los elementos o servicios que forman la aplicación en un único programa, mientras que las de microservicios separan dichos diferentes elementos en diferentes programas. Ambas formas tienen sus ventajas y desventajas; Las aplicaciones monolíticas son en general más rápidas de desarrollar mientras que las de microservicios son más exigentes, y mientras las aplicaciones de microservicios permiten, por su naturaleza modular, aislar cada servicio para trabajar sobre ellos individualmente sin que los cambios realizados repercutan en el código del resto de servicios, en las aplicaciones monolíticas los cambios realizados a parte del código tienen más facilidades de repercutir en el resto, dando lugar a posibles bugs serios, entre otras.