

CheckPoint 3 – Preguntas teóricas

¿Cuáles son los tipos de Datos en Python?

En Python tenemos nueve tipos de datos que pueden ser almacenados en variables:

-Strings: son secuencias de bytes alpha numéricos como son los nombres o textos.

```
str = 'Hola cómo estás?'
```

-Numbers: son datos numéricos enteros (1, 2, 3,...), decimales, flotantes, fracciones,...

```
number = 100
```

-Booleans: son datos que representan el valor verdadero o falso. Se utiliza para dar una respuesta de sí o no.

```
boolean = True
```

-Bytes: Son secuencias inmutables de bytes que se utilizan para trabajar con datos binarios como archivos o imágenes.

```
byte = b"Hola"
```

-None: Sirve para definir variables, pero sin darles un valor. Es interesante ya que se puede comprobar si la variable tiene o no un valor.

```
variable = None
```

-Lists: es una estructura de datos que permite gestionar colecciones de datos. Este tipo de datos son secuencias ordenadas de elementos.

```
lista = [1, 2, 3, 4]
```

-Tuplas: también son una estructura de datos, son secuencias ordenadas de elementos inmutables.

```
tupla = (1, 2, 3, 4)
```

-Sets: Son una estructura de datos, son colecciones desordenadas de elementos.

```
set = {1, 2, 3, 4}
```

-Dictionaries: Es una estructura de datos, son colecciones desordenadas de pares clave-valor.

```
diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}
```

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

En Python, la convención estándar para nombrar variables recomienda que se deben nombrar las variables que contengan varias palabras separando las palabras por un guión bajo. (Snake Case)

Por ejemplo:

```
varias_variables = XXX
```

En Python, la nomenclatura 'Camel Case' se reserva para nombrar clases.

¿Qué es un Heredoc en Python?

En Python, un heredoc es un string multilínea. Para asignar un heredoc a una variable se deben escribir, al principio y al final del heredoc, tres comillas, simples o dobles.

Por ejemplo:

```
variable = """  
escribir aquí el texto multilínea  
"""
```

¿Qué es una interpolación de cadenas?

Una interpolación de strings es una forma de procesar código Python dentro de un string. Por ejemplo, si deseamos crear un correo electrónico estándar para enviar a los clientes que han comprado un determinado producto, podemos usar este método para integrar los diferentes nombres de los clientes y los productos que han comprado dentro del texto del email que es común para todos los clientes. El código se escribe señalizando con la f flag antes del string para que Python procese todo lo que esté entre corchetes en el string.

Por ejemplo:

```
name = 'Cecilia'  
day = 'lunes'  
date = '15 de abril'  
hour = '9h00'  
  
email_content = f"""  
  
Buenos días {name}  
  
Te recordamos que tu próxima cita será el {day} {date} a las {hour}.
```

Si no puedes acudir llámanos al 555 555 555.

Saludos
""""

¿Cuándo deberíamos usar comentarios en Python?

Los comentarios dentro del código deben usarse para organizarlo cuando se trata de códigos extensos.

```
# Exercise1
print('Hello')

# Exercise2
name = 'Mary'
print(name)
```

Los comentarios no deben usarse para dar instrucciones o para explicar el comportamiento o funcionalidad del código, ya que podría inducir a error si no se actualizan debidamente, para ello es conveniente dar nombres descriptivos a las variables para evitar tener que explicarlas con comentarios.

```
#La variable var contiene nombres
var = 'Mary'
```

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

Las aplicaciones monolíticas y las aplicaciones microservicios tienen diferentes arquitecturas.

Las monolíticas tienen un sistema basado en una sola aplicación mientras que las microservicios tienen características que constituyen aplicaciones separadas. Así, en las monolíticas, el proceso es uno solo y en las microservicios hay varios procesos.

Las monolíticas son más fáciles de desarrollar y más rápidas ya que no necesitan comunicarse con otras partes del sistema, mientras que en las microservicios los procesos se dividen en diferentes partes o servicios que se comunican a través de APIs. Pero las monolíticas requieren mantenimientos difíciles ya que cualquier cambio afecta a todo el proceso, mientras que las microservicios permiten ampliar servicios y si uno de ellos cae, el sistema no se ve afectado.