M2C3 TEORÍA

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

Los nueve tipos más importantes de datos en Python son:

los números (pueden ser enteros, fracciones, floating-point…).

Ejemplos:

integer= 3

floating= 3,25

complex =2+4z (z la parte imaginaria)

string/cadenas (son una sucesión de caracteres) entre comillas dobles o sencillas.

Ejemplo:

string_var= “Galileo Galilei”

Booleans Se utiliza en declarciones condicionales y las dos únicas opciones son Verdadero/Falso

edited_post = True

None se utliza en los casos queremos crear una variable pero no asignable , se usa para chequear si una variable sigue siendo nula. También para argumentos con valores por defecto.

Ejemplo:

```
def greeting (name):
if name is None:
print("Hi , How are you?")
else:
print("Hi", name)
```

Arrays : son contenedores estructurados que almacenan cualquier tipo de dato (se accede a través de un índice). Puede ser modificada no es un contenedor estático e inmutable. (se escribe entre

corchetes). Deben ser declarados a diferencia de las listas que forman parte de la sintaxis de python.

array= [[4, 2, 0], [6,-4, 0], [6, 4,-2]]

Tuples es un a colección de datos o elmentos pueden ser iguales o distintos, no puede ser modificado y se calasifican por un índice. (se escribe entre paréntesis)

tuple = ("manzana","pera",”fresa")

print(tuple)

Por otra parte los sets no pueden tener elementos duplicados. No poseen un órden, y por tanto, tampoco números de índex. No podemos decidir el orden en el cual aparecerán sus elementos. Los sets se escriben entre llaves.

set= {"manzana","pera",”fresa"}

Las listas: sun una sucesión de elementos son similiares a las arrays, los elementos que las componen no tienen que ser idénticos, otra de de las caracteristicas de ellas es que se pueden modificar es, decir pueden ir aumentado, ampliando, si fuese necesario. Se escriben entre corchetes.

list = ["manzana", "pera", "fresa", "arándano", "frambuesa"]

print(list)

Dictionaries son un conjunto de datos que operan por parejas key y su valor. Se pueden modificar al igual que las listas, se pueden ampliar. Se escribe con llaves

Dict = {“fruta:"manzana", “Calorias”:"52"}

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

La nomblecatura que se utiliza en python es letra minuscula y guión bajo si fuese por ejemplo sin guión y mayúsculas podría entenderse que se refiere una clase en lugar de una variable este tipo nomencaltura se denomina SNAKE.

Ejemplo:

num_var= 100

3. ¿Qué es un Heredoc en Python?

Es una forma de especificar un bloque de texto. Se utiliza las comillas por triplicado para definir una cadena multiple. De este modo se transcribe de un modo idéntico el bloque de texto.

4. ¿Qué es una interpolación de cadenas?

Permite incluir en un texto datos concretos y únicos y especificos que se utilizan por ejemplo en un escrito/carta/informe personalizado dependiendo del dato que queramos mostrar (La dirección de una empresa, el nombre de un cliente, el producto que se ha vendido, el lugar de entrega….). La sintaxis es una f minuscula seguido de tres comillas seguidamente se introduce las llaves que corresponde a la variable que se quiere incluir se finaliza con tres comillas al igual que al principio.

Ejemplo:

name = "Gisela" product = "test sample"

text = f """

Dear {name},

We would be very grateful to send you a {product}

Yours sincerely,

bd Ltd.

“””

5. ¿Cuándo deberíamos usar comentarios en Python?

Los comentarios se limitaran a casos de códigos extenso, y que sea estrictamenet necesarios ya que la definición de la variable debe ser lo suficientemente explicita para evitar los comentarios.

La sintaxis comienza con # por ejemplo # comentario adicional

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

Las aplicaciones monoliticas tienen la ventaja frente a los microservicios que son más rápidas, pero el inconveniente que una modificación en un servicio o un error puede afectar al conjunto completo.

Las aplicaciones de microservicios sus diferentes servicios son indenpendientes tiene la ventaja que pueden ser modificados, cambiados , y no tiene que afectar al resto de los componentes del conjunto. Tiene la ventaja que se puede escalar cada uno de los servicios por separado de un un modo independiente. En contraste con el monolitico es más lento.

Al estar fragmentado los servicios se puede enfocar en cada uno de ellos de un modo independiente siendo muy práctico si se quiere realizar alguna modificación puntual.