# Preguntas teóricas M2C4

1. ¿Cuál es la diferencia entre una lista y una tupla en Python?

La principal diferencia es que una lista es mutable y se puede modificar y una tupla es inmutable y no se puede cambiar, solo reasignar. Aunque se utilice el mismo nombre de la variable de la tupla, se trabaja sobre nuevos elementos.

La sintaxis entre ambas también es diferente: en las listas se utilizan los corchetes ( [] ) para crearlas y asignarlas a una variable, mientras que en las tuplas se utilizan los paréntesis ( () ).

1. ¿Cuál es el orden de las operaciones?

El orden de las operaciones es PEMDAS, dicho así para recordarlo mejor. La referencia es tomada por las siglas en inglés:  
Paréntesis – Exponentes – Multiplicación – División – Suma – Resta

Parenthesis – Exponents – Multiplication – Division – Addition – Substract

1. ¿Qué es un diccionario en Python?

Es una estructura de datos en el que se almacena información de forma muy parecida a un diccionario de la vida real, con una clave a modo de índice y un valor correspondiente. A esto se le define como “clave – valor”.

La sintaxis de inicio de un diccionario se reconoce porque se colocan llaves de apertura y cierre ( {} ), además de una clave, seguido de dos puntos y un valor. Por ejemplo:

diccionario\_ejemplo = {

‘nombre’ : ‘Pablo’,

‘apellido’ : ‘Perez’,

‘edad’ : 35

}

En el anterior ejemplo “diccionario\_ejemplo = ” es el nombre dado a la variable junto con el signo = de asignación. La apertura y cierre con las llaves, y después las claves con sus correspondientes valores: clave “nombre” : valor “Pablo”, clave “apellido” : valor “Perez”, clave “edad”: valor 35

1. ¿Cuál es la diferencia entre un método ordenado y la función de ordenación?

La diferencia es que el método ordenado modifica el elemento original (solo se puede utilizar en elementos mutables) y la función de ordenación no modifica el valor original, por lo que se puede utilizar en cualquier tipo de elemento, ya sea mutable, como las listas, o inmutable, como las tuplas, por ejemplo.

El método ordenado se utiliza, por ejemplo en listas, un ejemplo es: dada una lista, se ordena de forma alfabética (o inversa que también es posible) y se imprime por pantalla la lista modificada:

lista\_ejemplo = [

‘lista1’,

‘lista3’,

‘lista5’,

‘lista2’,

‘lista4’

]

lista\_ejemplo.sort() esto ordenaría la lista de manera alfabética modificándola en su inicio.

lista\_ejemplo.sort(reverse=true) esto ordenaría la lista de manera alfabética también, pero a la inversa, también modificándola en su inicio.

Sin embargo, la función de ordenación, sorted() se utiliza reutilizando variables o utilizando nuevas, es decir:

tupla\_ejemplo = (

‘tupla1’,

‘tupla3’,

‘tupla5’,

‘tupla2’,

‘tupla4’

)

tupla\_ejemplo = sorted(tupla\_ejemplo)

print(tupla\_ejemplo)

Pero en este caso, lo que se ha hecho ha sido reasignar valores a la variable, no modificarla.

1. ¿Qué es un operador de asignación?

Es un operador que se utiliza para darle valor a las variables; y se representa con el símbolo “ = “.

El operador de asignación se utiliza también para reasignar valores a las variables, si se quiere hacer operaciones matemáticas con ellas, o añadir elementos a estructuras de datos, etc.

Por ejemplo, para sumar y reasignar un valor a una variable hay dos maneras para hacerlo:

sum = 5

sum = sum + 6

sum += 6

Donde en el ejemplo anterior se ha declarado la variable “sum” con un valor entero de 5, se ha reasignado esa variable utilizando el valor que ya tenía y sumándole 6 más; o de manera simplificada uniendo los operadores “+” y “=” después del nombre de la variable que actúa de la misma manera y evitamos repetir código para que quede más limpio y legible.