# **Variables 2.0.**

El desempaquetado en Python es una estrategia en la cual le asignamos valores a variables de una variable ya creada previamente, como tomar datos de una tupla y almacenarlos individualmente y almacenarlos en nuevas variables.

datos = ("Alexis", "Emanuel")

nombre, apellido = datos

Como aquí, la variable nombre va a ser igual al primer dato de la tupla o sea “Alexis” y la segunda variable datos al segundo dato de la tupla, o sea “Emanuel”.

datos = ("Alexis", "Emanuel", 10000)

nombre, apellido, dinero = datos

print(nombre)



Entonces des encapsulamos las variables de esta tupla.

Esto tambien funciona con las listas.

datos\_lista = ["Rex", "Raptor", "Trike"]

dino1, dino2, dino3 = datos\_lista

print(dino1 + " " + dino3)



Tambien en conjuntos.

## **Tuple.**

Tenemos la función “tuple” que nos permite crear tuplas:

tupla = tuple(["dato 1", "daot 2", "dato 3"])

print(tupla)

Pero debemos de pasarle una lista para poder usar esta función y convertir nuestra lista a una tupla.

Otra manera de crear tupla es solo añadiendo la coma a los datos de esta manera:

tupla = "dato 1", "dato 2"

print(tupla)

Y para crear tuplas de un solo dato seria haciendo esto:

tupla = "dato 1",

print(tupla)

Solo agregando una coma al final de nuestra tupla.

Las tuplas por lo general se crean para guardar datos de sola lectura.

## **Set (conjuntos).**

Los conjuntos los podemos crear con la función “set()”

conjunto = set(["dato 1", "dato 2"])

print(conjunto)



No podemos guardar una lista dentro de un conjunto ya que las listas se pueden modificar y los conjuntos no, tampoco podemos guardar diccionarios, pero si podemos guardar una tupla dentro de un conjunto:

conjunto = set(["dato 1", "dato 2", ("dato en tupla 1", "dato en tupla 2")])

print(conjunto)

Y tampoco dejare meter un cojunto dentro de otro conjunto, pero si hay una manera de como hacer, usando la función “frozenset”:

conjunto1 = frozenset(["dato 1", "dato 2"])

conjunto2 = {conjunto1, "dato 3"}

print(conjunto2)

Ya que con la función frozenset creamos un conjunto inmutable, no lo podremos modificar.

### **Teoría de conjuntos.**

En teoría de conjunto tenemos por un lado un conjunto y por otro lado un subconjunto que es agarrar datos del conjunto y crear un conjunto aparte.

Osea digamos, teneos un conjunto B con los datos “2, 4, 6, 8, 10” y creamos un subconjunto llamado conjunto A con los datos “2, 4, 6” del conjunto B, entonces A es un subconjunto de B.

Y un conjunto que incluye a otro y tiene mas datos es un superconjunto de otro conjunto.

Entonces siguiendo ele ejemplo, A es un subconjunto de B, mientras que B es un superconjunto de A mientras que A es un conjunto, depende de la perspectiva.

Para manejarlo en código lo haríamos de esta manera:

conjunto\_1 = {1, 3, 5, 7}

conjunto\_2 = { 1, 3, 7}

resultado = conjunto\_2.issubset(conjunto\_1)

Aquí creamos el conjunto 1 y 2, y en resultado almacenamos lo siguiente: ¿conjunto\_2 es un subconjunto de conjunto\_1? Y si lo imprimimos nos dará True ya que es verdad.

conjunto\_1 = {1, 3, 5, 7}

conjunto\_2 = {1, 3, 7}

resultado = conjunto\_2.issubset(conjunto\_1)

resultado = conjunto\_2 <= conjunto\_1

print(resultado)

Tambien lo pode,mos hacer con menor igual que.

Pero tambien tenem,os la forma de verificar si es un superconjunto.

conjunto\_1 = {1, 3, 5, 7}

conjunto\_2 = {1, 3, 7}

resultado = conjunto\_2.issuperset(conjunto\_1)

print(resultado)

Y esto nos devolverá False, porque conjunto 2 no es un superconjunto de 1, es al reves. Y para usarlo con signos usaríamos mayor que.

resultado = conjunto\_2 > conjunto\_1

Y si queremos verificar si hay un numero en común usamos “isdisjoint”

conjunto\_1 = {1, 3, 5, 7}

conjunto\_2 = {1, 3, 7}

resultado = conjunto\_2.isdisjoint(conjunto\_1)

print(resultado)

Esto nos devolverá False ya que tienen elementos en común, y solo dará True cuando los conjuntos que se comparan no tienen ningún elemento en común.

conjunto\_1 = {1, 3, 5, 7}

conjunto\_2 = {2, 4, 6, 8}

resultado = conjunto\_2.isdisjoint(conjunto\_1)

print(resultado)

Pero si solo hay un número que sea igual, se aplicara la función y tirara false.