

Python para Análisis de Datos

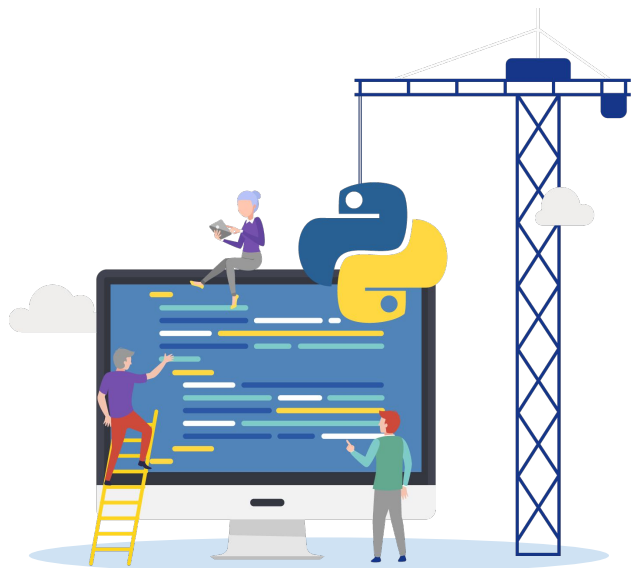
Módulo 01

Objetos

Objetos y referencia

En Python todo son objetos, en el sentido de la programación orientada a objetos. Incluso las funciones y las clases son objetos también. Eso permite que cualquier cosa pueda ser referenciada por una variable. Y decimos *referenciada* porque, en Python, crear una variable es crear un nombre con una referencia (*puntero*) a la dirección de memoria en donde el objeto se encuentra.

Esta es la razón por la que una variable no está limitada a un tipo de dato en particular (*tipado dinámico*) y crear una nueva asignación a otro tipo de dato es simplemente cambiar la referencia de la variable.



Referencia

```
1 var1 = 5
2 var2 = 10
3 var2 = var1
```

var1 ← 5

var2 ← 10

var1 ← 5

var2 ← 10



Borrado por
el garbage
collector

Esto implica que **muchas variables pueden apuntar al mismo objeto**. De hecho, cuando se ejecuta una expresión del tipo `var2 = var1` no se crea una copia del dato sino que se copia la referencia, por lo que las dos variables apuntan al mismo objeto en la memoria. Si el tipo de dato es mutable, cualquier cambio que se haga a través de una de las variables va a ser visto por la otra variable también, como sucede con las listas.

El manejo de la memoria se hace automáticamente. Python tiene lo que se llama ***garbage collector***, que borra todos los objetos que quedan sin referencia.



En este ejemplo se ve como ambas variables apuntan al mismo objeto en la memoria y las modificaciones hechas a través de `lista1` son visibles para `lista2`.

```
1 lista1 = [1,2,3,4]
2 lista2 = lista1
3 lista1[0] = 9999
4 print(lista1)
5 print(lista2)
```

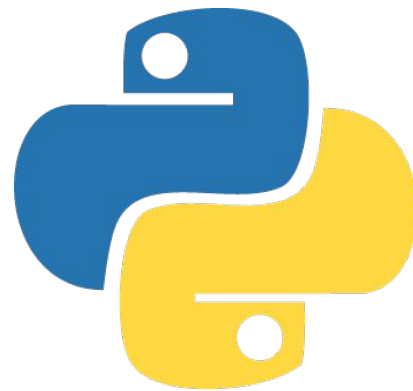
```
[9999, 2, 3, 4]
[9999, 2, 3, 4]
```

Colecciones

Python viene con una variedad de colecciones para agrupar datos en estructuras diferentes. Tenemos colecciones ordenadas o no ordenadas y colecciones mutables o inmutables.

Las **colecciones ordenadas** son aquellas que pueden ser indexadas, es decir, que se pueden usar números enteros para acceder a sus elementos. Son colecciones ordenadas (o secuencias) las **listas**, **tuplas** y **strings**.

Como **colecciones no ordenadas** tenemos los **diccionarios**, **sets** y **frozensets**.



Indexación

La indexación se puede realizar con uno, dos o tres parámetros separados por dos puntos. Con un parámetro se obtiene un elemento en particular. Se empieza a contar desde cero en adelante para seleccionar elementos contando desde la izquierda y números negativos para seleccionar empezando desde la derecha.

Con dos parámetros se puede seleccionar porciones (*slices*) de la secuencia y el tercer parámetro indica el paso con el que los elementos deben tomarse.

```
1 lista = [10,20,30,40,50,60]
2 print(lista[0])
3 print(lista[-1])
4 print(lista[1:5])
5 print(lista[::-2])
```

```
10
60
[20, 30, 40, 50]
[10, 30, 50]
```


Las **listas** son secuencias mutables, se pueden cambiar, agregar o borrar sus elementos. Sin embargo, las **tuplas** y los **strings** son tipos de datos inmutables y una vez creados no se pueden modificar. A lo sumo, es posible crear nuevas tuplas o strings a partir de los originales.

Los elementos de una lista se pueden cambiar mediante simples asignaciones, se puede agregar elementos con los métodos `append` e `insert` y se puede borrar elementos con los métodos `remove`, `pop`, `clear` y la palabra reservada `del`.



Colecciones no ordenadas

Las colecciones no ordenadas no soportan la indexación por números enteros ya que no se puede decir cual de sus elementos es el primero, segundo, etc.

Los **diccionarios** son colecciones en donde cada elemento consta de dos partes: una clave y un valor. Las claves hacen las veces de índice y permiten seleccionar los respectivos valores del diccionario. Los valores pueden ser cualquier tipo de dato, mientras las claves tienen que ser obligadamente un tipo de dato inmutable. Además, las claves no se pueden repetir.

Los valores correspondientes a una clave se pueden modificar y se puede agregar o quitar elementos.

Conjuntos

Los conjuntos son otro tipo de colección no ordenada. Sus elementos, al igual que las claves de un diccionario, deben ser tipos de dato inmutables. Sus elementos no se pueden repetir. Un objeto puede pertenecer o no a un conjunto, pero no puede estar dos veces.

Implementa operaciones matemáticas de conjuntos como intersección, unión y diferencia.

El **set** es una colección mutable, por lo que se puede agregar o remover elementos. Por su parte, el **frozenset** es inmutable, por lo que no se puede modificar, pero puede ser elemento de otros conjuntos o clave de un diccionario.

¡Muchas gracias!

¡Sigamos trabajando!