



# INTRODUCCIÓN A PYTHON

## LISTAS

Miguel Carrasco  
Junio 2019

Background designed by kjpargeter / Freepik

- ▶ Estructuras de control
- ▶ Listas
  - Definición y ejemplos



```
#Read item in dictionary
for key, value in item.FidValue.items():
    typeOfFID = mapFidType[key]
    if(typeOfFID == "DATE"):
        d = datetime.datetime.strptime(value, "%Y-%m-%d")
        dataCal = datetime.date(d.year, d.month, d.day)
        FidAndValue = FidValue + " " + str(dataCal)
    else:FidAndValue = FidValue + " " + value
```

```
try:
    start = date(int(self.start_year.get()),
                 self.months.index(self.start_month.get()),
                 int(self.start_day.get()))
    end = date(int(self.end_year.get()),
               self.months.index(self.end_month.get()),
               int(self.end_day.get()))
```



```
for i in colección:  
    #instrucciones de ciclo for
```

El ciclo **for** recorre los elementos que forman parte de la colección **uno a uno**.

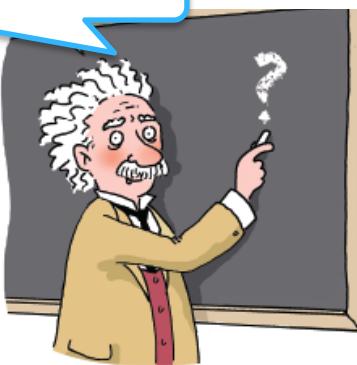
Una colección puede ser:

1. un texto (recorrerá cada letra),
2. un conjunto de elementos (como entrega range) o
3. una lista

```
for i in 'hola':  
    print(i) #imprime cada letra de hola
```

```
for i in range(1,11):  
    print(i) #imprime los nros del 1 al 10
```

¿qué es una lista?





Una **lista** es una variable que puede guardar muchas cosas al mismo tiempo dentro.

Usamos como analogía una cajonera que puede contener variadas cosas.



# ¿Qué es una lista?

Algunas  
Características de  
las listas son:

Tienen un  
nombre



Cada espacio se identifica con un número que comienza en 0. Es decir el primer elemento es el lugar 0 de la lista. A este identificador lo llamaremos Índice.

Puedes guardar lo que quieras en cada espacio. Números, palabras o frases

El número que identifica la posición en la lista lo llamaremos índice



Para ahorrar código teniendo varios valores en una sola variable.

- Relacionar valores a un concepto en particular por ejemplo, el listado de las notas.
- Ordenar valores
- Trabajar con múltiples valores de forma ordenada y dinámica.



¿Cómo se ve?



Lista



```
lista = [45, 5, 3, "hola", "Otra cosa"]
```



Los valores que se asignan a una lista, van separados por comas y entre corchetes.



El formato es el siguiente:

```
nombre_lista = [elementos]
```

```
lista = [45,5,3,"holo","Otra cosa"]  
print(lista[0])
```



45



Para obtener un elemento debo indicar su índice:

```
print(lista[0])
```

```
lista = [45,5,3,"holo","Otra cosa"]  
print(lista[3])
```



holo



Para mostrar todos los valores aprovechamos la estructura del ciclo for, de tal forma de escribir el código necesario y así recorrer toda la lista.

```
lista = [45,5,3,"hola","Otra cosa"]
for elemento in lista:
    print(elemento)
```

Esta forma recorre todos los elementos de la lista y permite realizar acciones con cada uno. El formato es el siguiente:



```
for elemento in lista:
    <<acciones>>
```



- **elemento** es una variable que guardará cada elemento de la lista
- **lista** es el nombre de la lista
- <<acciones>> es código que se ejecuta para cada elemento de la lista



Otra forma de observar cada valor de la lista es usando el índice como identificador.

```
lista = [45,5,3,"hola","Otra cosa"]
for i in range (0,len(lista)):
    print(lista[i])
```

Dado que el índice está ordenado y sabemos el largo de la lista, fabricamos el rango en donde la variable *i* debe moverse. El formato es el siguiente:



```
for i in range (0,len(lista)):
    <<acciones>>
```



- *i* es una variable que avanza de uno en uno
- **len()** es una función que nos entrega el largo de la lista
- <<acciones>> es código que se ejecuta para cada elemento de la lista

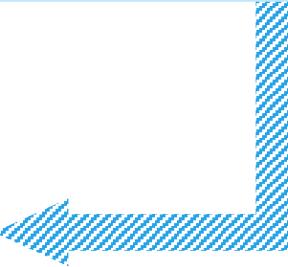
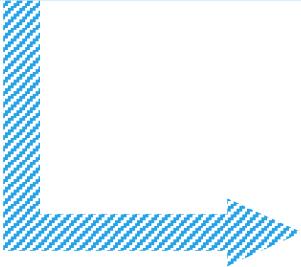
# Ejemplo 1



Llene una lista con números enteros y calcule la suma de ellos

```
lista = [3,56,5,38,2,3,765]
suma = 0
for elemento in lista:
    suma = suma + elemento
print(suma)
```

```
lista = [3,56,5,38,2,3,765]
suma = 0
for i in range (0,len(lista)):
    suma = suma + lista[i]
print(suma)
```



872



**Importante:**

Si la lista tiene valores con texto, no podrá sumarlos. Ejemplo: lista = [1,2,4, "Hola"]

## Ejemplo 2



Dada una lista con números enteros calcule, cuántos números pares hay en la misma

```
lista = [3,56,5,38,2,3,765]
cont=0
for element in lista:
    if (element % 2 == 0):
        cont = cont + 1
print("no. de pares es", cont)
```

```
lista = [3,56,5,38,2,3,765]
cont=0
for i in range (0,len(lista)):
    if (lista [i] % 2 == 0):
        cont = cont + 1
print("no. de pares es", cont)
```

no. de pares es 3

# Ejemplo 3



Dada una lista con números enteros, encuentre el número menor de esta

```
lista = [3,56,5,38,2,3,765]
menor = lista[0]
for element in lista:
    if (element < menor):
        menor = element
print("menor elemento: ", menor)
```

```
lista = [3,56,5,38,2,3,765]
menor = lista[0]
for i in range (0,len(lista)):
    if (lista[i] < menor):
        menor = lista[i]
print("menor elemento: ", menor)
```

menor elemento: 2

- ▶ Estructuras de control
- ▶ Listas
  - Definición
  - Ingreso de datos



```
#Read item in dictionary
for key, value in item.FidValue.items():
    typeOfFID = mapFidType[key]
    if(typeOfFID == "DATE"):
        d = datetime.datetime.strptime(value, "%Y-%m-%d")
        dataCal = datetime.date(d.year, d.month, d.day)
        FidAndValue = FidAndValue + dataCal
    else:FidAndValue = FidAndValue + value
```

```
try:
    start = date(int(self.start_year.get()),
                 self.months.index(self.start_month.get()),
                 int(self.start_day.get()))
    end = date(int(self.end_year.get()),
               self.months.index(self.end_month.get()),
               int(self.end_day.get()))
```





Esta bien ingresar datos por código, pero ¿cómo podemos hacer para que el usuario ingrese los datos?

Usamos los ciclos para hacerlo de forma dinámica porque no sabemos cuántos datos ingresarán.



**lista vacía**

nos sirve para decirle al computador que usaremos una lista, esta queda vacía esperando a que ingresemos información

```
print("ingrese el número de elementos: ")
num = int(input())

lista = []
for i in range (0,num):
    print("ingrese es elemento",i)
    elem = int(input())
    lista.append(elem)
```



**.append(<contenido>)** es una función que nos permite agregar un valor al final de la lista.



Esta bien ingresar datos por código, pero ¿cómo podemos hacer para que el usuario ingrese los datos?

Usamos los ciclos para hacerlo de forma dinámica porque no sabemos cuántos datos ingresarán.

```
print("ingrese el número de elementos: ")
num = int(input())

lista = []
for i in range (0,num):
    print("ingrese es elemento",i)
    elem = int(input())
    lista.append(input())

print ("la lista es:")
for i in range (0,len(lista)):
    print(lista[i])
```



recorremos el  
listado posición  
por posición

```
print("ingrese el num. de elementos: ")
num = int(input())

lista = []
for i in range (0,num):
    print("ingrese numero",i)
    lista.append(input())

print ("la lista es:")
for i in range (0,len(lista)):
    print(lista[i])
```



```
ingrese el num. de elementos
5
ingrese numero 0
4
ingrese numero 1
56
ingrese numero 2
786
ingrese numero 3
54
ingrese numero 4
5678
la lista es:
4
56
786
54
5678
```

# Ejemplo 1



Tiempo : 10 minutos

Genere una lista con 10 números enteros ingresados por el usuario y calcule su promedio

```
print("ingrese numero de elementos ")
num = int(input())
lista = []

for i in range (0,num):
    print("ingrese numero",i)
    lista.append(input())

print ("la lista es:")
for i in range (0,len(lista)):
    print(lista[i])

suma=0
for elemento in lista:
    suma = suma + elemento

promedio = suma/len(lista)
print ("promedio es:", promedio)
```

```
ingrese número de elementos:
4
ingrese numero 0
3
ingrese numero 1
4
ingrese numero 2
3
ingrese numero 3
4
la lista es:
3
4
3
4
promedio es: 3.5
```