

prog_datasci_3_apython

August 7, 2019

1 Programación para *Data Science*

1.1 Unidad 3: Estructuras de control y funciones en Python

1.1.1 Instrucciones de uso

En el módulo anterior introducimos conceptos básicos sobre variables y su uso en Python. En este módulo estudiaremos conceptos más avanzados como las instrucciones de flujo de ejecución (for, while, if), cómo definir y utilizar funciones, cómo leer y escribir ficheros y cómo organizar el código.

1.2 Iteración y operaciones lógicas

En la gran mayoría de ocasiones, tendremos que manipular nuestros datos y para ello utilizaremos los conceptos de iteración y las operaciones lógicas. Las operaciones lógicas nos permiten comparar valores entre variables (mayor, menor, igualdad) y la iteración ir visitando uno a uno los elementos de una lista, tupla, diccionario o cualquier estructura de datos que sea susceptible de secuenciarse.

```
[1]: # Las operaciones lógicas tendrán como resultado un valor cierto (True) o falso
      ↪ (False):

a = 5
b = 1

# ¿Es el valor de a mayor que b?
print(a > b)
```

True

```
[2]: # ¿Es el valor de a menor que b?
print(a < b)
```

False

```
[3]: b = 5
      # ¿Es el valor de b igual al de a?
      print(b == a)
```

True

```
[4]: # Otros operadores lógicos disponibles son menor o igual '<=', mayor o igual '≥',  
      ↳ '>=' o la negación 'not'.  
print(a <= b)  
print(a >= b)  
  
a = False  
print(not a)
```

True

True

True

También podemos alterar el flujo de ejecución de nuestro programa utilizando las estructuras *if...else* o *if...elif...else*. Veamos unos cuantos ejemplos:

```
[5]: a = 5  
b = 6  
if a > b:  
    print('a es mayor que b')  
else:  
    print('a es menor o igual que b')
```

a es menor o igual que b

```
[6]: a = 5  
b = 5  
if a > b:  
    print('a es mayor que b')  
elif a < b:  
    print('a es menor que b')  
else:  
    print('a es igual a b')
```

a es igual a b

En Python existen solo dos formas de iterar una secuencia: mediante **for** o mediante **while**. La primera de las opciones, *for*, iterará uno por uno los elementos contenidos en una lista. En el caso de *while*, iteraremos mientras la condición de permanencia en el bucle se cumpla. Veamos unos ejemplos:

```
[7]: monsters = ['Kraken', 'Leviathan', 'Uroborus', 'Hydra']  
  
# Primer método iterando mediante for:  
for monster in monsters:  
    print(monster)
```