Estructura de Datos

Sesion 02

Ing. Gómez Marín, Jaime[1]

[1]Fundamentos de Python para Ciencia de Datos
Departamento de TdG

August 2019

Índice

- Introducción
- Tuples
- List
- Set
- Dictionary
- Conclusiones
- Bibliografía

Introducción

En esta sesión se aprenderá a usar estructura de datos con los siguiente tipos de datos : Tupla, List , Set y Dictionary en Python.

Tupla

Una tupla es una secuencia de objetos de Python inmutables (no se puede modificar). Las tuplas son secuencias, al igual que las listas. Las diferencias entre tuplas y las listas son, las tuplas no se pueden cambiar a diferencia de las listas y tuplas utilizan paréntesis, mientras que las listas utilizan corchetes.

Code - Tuple : Definir

```python
'''
Programa : Ciencia de Datos con Pthon
Modulo 01 : Fundamentos de Python para Ciencia de Da
Sesion 02 : Estructura de datos - Tuplas
Fecha : 04/08/2019
Version : 1
Author : Jaime Gomez
'''

tuple1=("Smartphone",10,1.2)

print("--------------------------------------")
print(tuple1)
print(type(tuple1))
```

Code - Tuple - Tuplede : Acceder a elementos

```python
print("--------------------------------------")
print(tuple1)
print(tuple1[0])
print(tuple1[1])
print(tuple1[2])
print("--------------------------------------")
print(tuple1)
print(type(tuple1[0]))
print(type(tuple1[1]))
print(type(tuple1[2]))
print("--------------------------------------")
print(tuple1)
print(tuple1[-1])
print(tuple1[-2])
print(tuple1[-3])
```

Code - Tuple : Es inmmutable

```
'''
print("----------------------------------------")
tuple1=("Smartphone",10,1.2)
print("tuple1[0]  :  ", tuple1[0])
tuple1[0]  =  "Tablet"
print("tuple1[0]  :  ", tuple1[0])
print("tuple1     :  ", tuple1)
#'''
```

Code - Tuple : Acceder a bloques de datos

```python
print("--------------------------------------")
tuple2 = tuple1 + ("tablet",8)
print(tuple2)
print(tuple2[0:3])
print(tuple2[3:5])
print(len(tuple2))
```

Code - Tuple : Definiciones complejas

```python
###
nestledTuple = (2,4,"Panam Sports", (1,2),
                ("Soccer Game", 4, (5,8)))

print("--------------------------------")
print("Tuple      :", nestledTuple)
print("element 0 :", nestledTuple[0])
print("element 1 :", nestledTuple[1])
print("element 2 :", nestledTuple[2])
print("element 3 :", nestledTuple[3])
print("element 4 :", nestledTuple[4])
```

Code - Tuple : Definiciones complejas 2

```python
print("--------------------------------------")
print("Tuple         :", nestledTuple)
print("element 0     :", nestledTuple[0])
print("element 2     :", nestledTuple[2])
print("element 2,0   :", nestledTuple[2][0])
print("element 3,1   :", nestledTuple[3][1])
print("element 4     :", nestledTuple[4])
print("element 4,0   :", nestledTuple[4][0])
print("element 4,1   :", nestledTuple[4][1])
print("element 4,2   :", nestledTuple[4][2])
print("element 4,2,0 :", nestledTuple[4][2][0])
```

Code - Tuple : Ordenación

```python
print("--------------------------------------")
ratings = (0,3,4,19,5,6,7,8)
ratingsSorted = sorted(ratings)
print(ratings)
print(ratingsSorted)

print("--------------------------------------")
ratings = (0,3,4,19,5,6,7,8)
ratingsClone = ratings
print(ratingsClone)
```

Code - Tuple : Max, Min y Sum

```python
print("--------------------------------------")
ratings = (0,3,4,19,5,6,7,8)
print("ratings         :", ratings)
print("min(ratings)    :", min(ratings))
print("max(ratings)    :", max(ratings))
print("sum(ratings)    :", sum(ratings))
```

Code - Tuple : Búsqueda

```python
print("--------------------------------------")
sports = ("Soccer","Tennis","Baseball","Squash")
print("sports                   :", sports)
#print("sports.index(\"Soccer\")   :",
#     sports.index("Soc"))
print("sports.index(\"Soccer\")   :",
      sports.index("Soccer"))
print("sports.index(\"Baseball\") :",
      sports.index("Baseball"))
```

Code - Tuple : Consultas

```python
print("--------------------------------------")
sports = ("Soccer","Tennis","Baseball","Squash")
existShotting = "Shotting" in sports
print("sports                :", sports)
print("existShotting         :", existShotting)
existSoccer = "Soccer" in sports
print("existSoccer           :", existSoccer)

print("--------------------------------------")
sports = ("Soccer","Tennis","Baseball","Squash")
notExistShotting = "Shotting" not in sports
print("sports             :", sports)
print("notExistShotting   :", notExistShotting)
notExistSoccer = "Soccer" not in sports
print("notExistSoccer     :", notExistSoccer)
```

Code - Tuple : Contar elementos

```python
print("--------------------------------------")
samples = (0,3,4,3,5,5,5,5,1,1)
print("samples                :", samples)
print("samples.count(1)       :", samples.count(1))
print("samples.count(3)       :", samples.count(3))
print("samples.count(5)       :", samples.count(5))
```

Code - Tuple : Recorrer elementos

```python
print("----------------------------------------")
tuple1=("Smartphone",10,1.2)
for value in tuple1 :
    print(value)
```

List

La lista es un tipo de datos más versátil disponible en Python que puede escribirse como una lista de valores separados por comas (items) entre corchetes. Lo importante de una lista es que los elementos de una lista no tienen por qué ser del mismo tipo.

Code - List : Definir

```
'''
Programa : Ciencia de Datos con Pthon
Modulo 01 : Fundamentos de Python para Ciencia de Da
Sesion 02 : Estructura de datos - List
Fecha : 04/08/2019
Version : 1
Author : Jaime Gomez
'''

print("--------------------------------------")
list1 = ["Panam Sports", 28.07, 2019, 199]
print("list1        : ", list1)
```

Code - List : Acceder a elementos

```
print("len(list1) : ", len(list1))
print("lists[0]    : ", list1[0])
print("lists[1]    : ", list1[1])
print("lists[2]    : ", list1[2])
print("lists[3]    : ", list1[3])
print("lists[0:2] : ", list1[0:2])  # No incluye lis
print("lists[1:3] : ", list1[1:3])  # No incluye lis

print("--------------------------------------")
list1 = ["Panam Sports", 28.07, 2019, 199]
print("list1       : ", list1)
print("len(list1) : ", len(list1))
print("lists[-4]    : ", list1[-4])
print("lists[-3]    : ", list1[-3])
print("lists[-2]    : ", list1[-2])
print("lists[-1]    : ", list1[-1])
```

Code - List : Modificar

```python
print("--------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
print("list1[0] : ", list1[0])
list1[0] = "Olympics"
print("list1[0] : ", list1[0])
print("list1    : ", list1)
```

Code - List : Eliminar

```python
print("------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
print("list1      : ", list1)
print("list1[0]   : ", list1[0])
del(list1[0])
print("del(list1[0]) ")
print("list1      : ", list1)
```

Code - List : append() y extend()

```python
print("----------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
print("list1            : ", list1)
list1.append(["Rugby", 11])
print("list1.append()   : ", list1)

print("----------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
print("list1            : ", list1)
list1.extend(["Rugby", 11])
print("list1.extend()   : ", list1)
```

Code - List : Unir listas

```python
print("---------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
list2 = [("Rugby", 11), 4, 5]
print("list1                 : ", list1)
print("list2                 : ", list2)
list1.append(list2)
print("list1.append(list2)   : ", list1)

print("---------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
list2 = [("Rugby", 11), 4, 5]
print("list1                 : ", list1)
print("list2                 : ", list2)
list1.extend(list2)
print("list1.extend(list2)   : ", list1)
```

Code - List : Fragmentar listas

```python
print("--------------------------------------")
msg = "Panam Sports - Lima Peru"
print("msg         :", msg)
print("msg.split() :", msg.split())

print("--------------------------------------")
abc = "A,B,C,D,E,F,G"
abcList = abc.split(",")
print("abc             :", abc)
print("abc.split(\",\")   :", abcList)
```

Code - List : Referencia de listas

```python
print("---------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
list2 = list1
print("list1 :", list1)
print("list2 :", list2)

list1[0] = "Soccer"
print("list1 :", list1)
print("list2 :", list2)

del(list2[2])
print("list1 :", list1)
print("list2 :", list2)
```

Code - List : Acceder a bloques de datos

```
print("-----------------------------------------")
list1 = ["Panam Sports", 28.07, 2019, 2020, 2021]

print(list1[0:3])
print(list1[2:4])
```

Code - List : Ordenación

```python
print("--------------------------------------")
ratings = [0,3,4,19,5,6,7,8]
print("ratings       :", ratings)
ratings.sort()
print("ratings       :", ratings)
ratings = [0,3,4,19,5,6,7,8]
print("ratings       :", ratings)
ratings.sort(reverse=True)
print("ratings       :", ratings)

print("--------------------------------------")
ratings = [0,3,4,19,5,6,7,8]
ratingsSorted = sorted(ratings)
print("ratings       :", ratings)
print("ratingsSorted :", ratingsSorted)
ratingsSorted = sorted(ratings, reverse=True)
print("ratingsSorted :", ratingsSorted)
```

Code - List : Max, Min y Sum

```python
print("----------------------------------------")
ratings = [0,3,4,19,5,6,7,8]
print("ratings        :", ratings)
print("min(ratings)   :", min(ratings))
print("max(ratings)   :", max(ratings))
print("sum(ratings)   :", sum(ratings))
```

Code - List : Búsqueda

```
print("----------------------------------------")
sports = ["Soccer","Tennis","Baseball","Squash"]
print("sports                   :", sports)
#print("sports.index(\"Soccer\")    :",
#      sports.index("Soc"))
print("sports.index(\"Soccer\")    :",
      sports.index("Soccer"))
print("sports.index(\"Baseball\") :",
      sports.index("Baseball"))
```

Code - List : Otra forma de borrar elementos

```python
print("--------------------------------------")
sports = ["Soccer","Tennis","Baseball","Squash"]
print("sports                   :", sports)
sports.remove("Tennis")
print("sports                   :", sports)
```

Code - List : Consultas

```python
print("--------------------------------------")
sports = ["Soccer","Tennis","Baseball","Squash"]
existShotting = "Shotting" in sports
print("sports                   :", sports)
print("existShotting            :", existShotting)
existSoccer = "Soccer" in sports
print("existSoccer              :", existSoccer)
```

Code - List : Contar elementos

```python
print("---------------------------------------")
samples = [0,3,4,3,5,5,5,5,1,1]
print("samples                :", samples)
print("samples.count(1)       :", samples.count(1))
print("samples.count(3)       :", samples.count(3))
print("samples.count(5)       :", samples.count(5))
```

Code - List : Recorrer elementos

```python
print("------------------------------------")
list1 = ["Panam Sports", 28.07, 2019]
for value in list1:
    print(value)
```

Code - List : List to Tuple

```python
print("--------------------------------------")
sports = ["Soccer","Tennis","Baseball","Squash"]
print("sports                   :", sports)
sports = tuple(sports)
print("tuple(sports)            :", sports)
sports = list(sports)
print("list(sports)             :", sports)
```

Set

Los conjuntos es un tipo de datos disponible en Python que no permite tener registros duplicados y no permanecen ordenados. Puede escribirse como una lista de valores separados por comas (items) entre paréntesis. Lo importante de una lista es que los elementos de una lista no tienen por qué ser del mismo tipo.

Code - Set : Definir

```
'''
Programa : Ciencia de Datos con Pthon
Modulo 01 : Fundamentos de Python para Ciencia de Da
Sesion 02 : Estructura de datos - Sets
Fecha : 04/08/2019
Version : 1
Author : Jaime Gomez
'''

print("-----------------------------------")
colourSet  = {"red", "green", "yellow", "blue",
              "blue", "blue"}
colourList = ["red", "green", "yellow", "blue",
              "blue", "blue"]
print("colourSet              :",  colourSet)
print("colourList             :",  colourList)
```

Code - Set : Agregar y eliminar elementos

```
print("------------------------------------")
colours = {"red", "green", "yellow", "blue"}
print("colours                    :", colours)
print("len(colours)               :", len(colours))
print("type(colours)              :", type(colours))
colours.add("white")
print("colours                    :", colours)
colours.add("white")
print("colours                    :", colours)
colours.remove("green")
print("colours                    :", colours)
```

Code - Set : Ordenación

```
print("----------------------------------")
colours = {"red", "green", "yellow", "blue"}
print("colours                          :",
      colours)
print("sorted(colours)                  :",
      sorted(colours))
print("sorted(colours, reverse = True) :",
      sorted(colours, reverse = True))
```

Code - Set : Max, Min y Sum

```python
print("----------------------------------------")
ratings = {0,3,4,19,5,6,7,8}
print("ratings        :", ratings)
print("min(ratings)  :", min(ratings))
print("max(ratings)  :", max(ratings))
print("sum(ratings)  :", sum(ratings))
```

Code - Set : Clonar

```python
print("-------------------------------------")
list1 = {"Panam Sports", 28.07, 2019}
list2 = list1.copy()
print("list1 :", list1)
print("list2 :", list2)
```

Code - Set : Unir Set

```python
print("--------------------------------------")
colours1 = {"red", "green", "yellow", "blue"}
colours2 = {"white", "black", "purple"}
print("colours1 :", colours1)
print("colours2 :", colours2)
#colours1 = colours1 + colours2
colours1.update(colours2)
print("colours1 :", colours1)
```

Code - Set : Consultas

```python
print("--------------------------------------")
sports = {"Soccer","Tennis","Baseball","Squash"}
existShotting = "Shotting" in sports
print("sports                  :", sports)
print("existShotting    :", existShotting)
existSoccer = "Soccer" in sports
print("existSoccer         :", existSoccer)
```

Code - Set : Operaciones entre Set - 1

```
print("--------------------------------------")
sportsA = {"Soccer","Tennis","Baseball","Squash"}
sportsB = {"Soccer","Tennis","Rugby","Judo"}
print("sportsA                          :",
      sportsA)
print("sportsB                          :",
      sportsB)
print("sportsA & sportsB                :",
      sportsA & sportsB)
print("sportsA.intersection(sportsB) :",
      sportsA.intersection(sportsB))
```

Code - Set : Operaciones entre Set - 2

```
print("sportsA | sportsB                  :",
      sportsA | sportsB)
print("sportsA.union(sportsB)             :",
      sportsA.union(sportsB))
print("sportsA.difference(sportsB)        :",
      sportsA.difference(sportsB))
print("sportsB.difference(sportsA)        :",
      sportsB.difference(sportsA))
print("sportsA ^ sportsB                  :",
      sportsA ^ sportsB)
```

Code - Set : Recorrer elementos

```python

print("--------------------------------------")
sports = {"Soccer","Tennis","Baseball","Squash"}
for value in sports:
    print(value)
```

Code - Set : Set - List - Tuple

```
#'''
print("---------------------------------------")
sports = {"Soccer","Tennis","Baseball","Squash"}
print("sports                  :", sports)
sports = tuple(sports)
print("tuple(sports)           :", sports)
sports = list(sports)
print("list(sports)            :", sports)
sports = set(sports)
print("set(sports)             :", sports)
#'''
```

Dictionary

Los diccionarios en Python son un tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor, siendo las claves únicas dentro de un mismo diccionario (es decir que no pueden existir dos elementos con una misma clave).

Code - Dictionary : Definir

```
'''
Programa : Ciencia de Datos con Pthon
Modulo 01 : Fundamentos de Python para Ciencia de Da
Sesion 02 : Estructura de datos - Dictionary
Fecha : 04/08/2019
Version : 1
Author : Jaime Gomez
'''

print("---------------------------------")
dict1 = {"key1":"value1", "key2":"value2",
         "key3":"value3"}
print("dict1                :", dict1)
```

Code - Dictionary : Acceso a elementos

```python
print("len(dict1)        :",   len(dict1))
print("dict1[\"key1\"]   :",   dict1["key1"])
print("dict1[\"key2\"]   :",   dict1["key2"])
print("dict1[\"key3\"]   :",   dict1["key3"])
print("dict1.keys()      :",   dict1.keys())
print("dict1.values()    :",   dict1.values())
```

Code - Dictionary : Eliminar elementos

```python
print("--------------------------------------")
dict1 = {"key1":"value1", "key2":"value2",
         "key3":"value3"}
del(dict1["key2"])
print("dict1              :",  dict1)
print("dict1.keys()       :",  dict1.keys())
print("dict1.values()     :",  dict1.values())
```

Code - Dictionary : Actualizar elementos

```
print("------------------------------------")
dict1 = {"key1":"value1", "key2":"value2",
         "key3":"value3"}
dict1["key3"] = "value4"
print("dict1           :", dict1)
dict1.update({"key3":'new value3'})
print("dict1           :", dict1)
dict1.update({"key4":'value4'})
print("dict1           :", dict1)
```

Code - Dictionary : Unir Diccionarios

```python
print("--------------------------------------")
dict1 = {"key1":"value1", "key2":"value2",
         "key3":"value3"}
dict2 = {"key4":"value4", "key5":"value5"}
print("dict1          :", dict1)
print("dict2          :", dict2)
#dict1 = dict1 + dict2
dict1.update(dict2)
print("dict1          :", dict1)
```

Code - Dictionary : Keys , Values e Items

```python
print("--------------------------------------")
eng2spa = {"red":"rojo", "green":"verde",
           "yellow":"amarillo"}
print("eng2spa                :",   eng2spa)
print("eng2spa.keys()         :",   eng2spa.keys())
print("list(eng2spa.keys())   :",   list(eng2spa.keys
print("eng2spa.values()       :",   eng2spa.values())
print("list(eng2spa.values()) :",   list(eng2spa.valu
print("eng2spa.items()        :",   eng2spa.items())
print("list(eng2spa.items())  :",   list(eng2spa.item
```

Code - Dictionary : Consultas

```python
print("--------------------------------------")
eng2spa = {"red":"rojo", "green":"verde",
           "yellow":"amarillo"}
existred = "red" in eng2spa
print("eng2spa                :", eng2spa)
print("existred               :", existred)
notExistred = "red" not in eng2spa
print("notExistred            :", notExistred)
```

Code - Dictionary : Otra forma de acceso a elementos

```
print("---------------------------------------")
eng2spa = {"red":"rojo", "green":"verde",
           "yellow":"amarillo"}
existred = "red" in eng2spa
print("eng2spa                :", eng2spa)
print("eng2spa[\"red\"])         :", eng2spa["red"])
print("eng2spa.get(\"red\")      :", eng2spa.get("red'
#print("eng2spa[\"blue\"])        :", eng2spa["blue"]
print("eng2spa.get(\"blue\")      :",
      eng2spa.get("blue","Not exist"))
```

Code - Dictionary : Clonar

```python
print("--------------------------------------")
eng2spa = {}
cpEng2spa = eng2spa.copy()
cpEng2spa.update({"blue":"azul"})
print("eng2spa                :", eng2spa)
print("cpEng2spa              :", cpEng2spa)
```

Code - Dictionary : Recorrer elementos

```python
print("--------------------------------")
theAcademyAward = {2000:"Gladiator",
                   2001:"A Beautiful Mind",
                   2002:"Chicago",
                   2003:"The Lord of the Rings: The
                   2004:"Million Dollar Baby",
                   2005:"Crash",
                   2006:"The Departed",
                   2007:"No Country for Old Men",
                   2008:"Slumdog Millionaire",
                   2009:"The Hurt Locker",
                   2010:"The King's Speech"}

for (key,value) in theAcademyAward.items():
    print(key,"->", value)
```

Conclusion

- Tuple: Es una secuencia de objetos de Python inmutables. Las tuplas son secuencias, al igual que las listas.
- List: Es un tipo de datos más versátil disponible en Python que puede escribirse como una lista de valores separados por comas (items) entre corchetes.
- Set: Es un tipo de datos sin orden y que no permite datos repetidos
- Dictionary: Son un tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor, siendo las claves únicas dentro de un mismo diccionario.

Bibliography

Naomi Ceder. The Quick Python Book - Manning Publications, 2018.