

Python 4: Diccionarios, Series y DataFrames (pandas)

Dora Suárez, Juan F. Pérez

Departamento MACC
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario

juanferna.perez@urosario.edu.co

2018

Contenidos

1 Diccionarios

2 Series en Python (pandas)

3 DataFrames

Diccionarios

Diccionarios en Python

- Identificar elementos en una lista
- Key + Value: identificador + valor almacenado

Diccionarios en Python

En un script de Python:

```
# -*- coding: utf-8 -*-  
moneda ={'Colombia': 'COP',  
        'EEUU': 'USD',  
        'Suiza': 'CHF'  
}  
print(moneda)
```

Diccionarios en Python

```
moneda[ 'Reino Unido' ] = 'GBP'
```

```
moneda[ 'Japon' ] = 'JPY'
```

```
print(moneda)
```

```
print(moneda[ 'Suiza' ])
```

```
print("La moneda de Colombia es ", moneda[ 'Colombia' ])
```

Más diccionarios en Python

```
for pais, abrev in moneda.items():  
    print("La moneda de %s es %s" % (pais, abrev))
```

Más diccionarios en Python (cont.)

```
capitales ={'Colombia': 'Bogota',  
            'Japon': 'Tokio',  
            'Reino Unido': 'Londres',  
            'EEUU': 'Washington',  
            'Suiza': 'Zurich'  
}
```


Más diccionarios en Python (cont.)

```
for pais, abbrev in moneda.items():  
    print("""  
        La moneda de %s es %s  
        y su capital es %s  
        """ % (pais, abbrev, capitales[pais]))
```

Datos faltantes en Python

```
# -*- coding: utf-8 -*-  
moneda ={'Colombia': 'COP',  
        'EEUU': 'USD',  
        'Suiza': 'CHF',  
        'Reino Unido': 'GBP',  
        u'Japón': 'JPY'  
}  
capitales ={'Colombia': u'Bogotá',  
            u'Japón': 'Tokio',  
            'Reino Unido': 'Londres',  
            }
```

Datos faltantes en Python (cont.)

```
for pais, abrev in moneda.items():
    capital = capitales.get(pais, None)
    if capital:
        print("La moneda de %s es %s y su capital \
              es %s" % (pais, abrev, capitales[pais]))
    else:
        print(u"No se encontró la capital de %s" \
              % pais)
```

Series en Python (pandas)

Series en Python

```
# -*- coding: utf-8 -*-  
from pandas import Series, DataFrame  
import pandas as pd  
  
obj1 = Series([3, -2, -5, 6])  
print("Serie obj1:\n", obj1)  
print("\nValores:\n", obj1.values)  
print("\nIndices:\n", obj1.index)
```

Series con índices

```
obj2 = Series([3, -2, -5, 6],  
              index = ['uno', 'dos', 'tres', 'cuatro'] )  
print("Serie obj2:\n", obj2)  
  
print(obj1[1])  
print(obj2['tres'])
```

Operando con Series

```
obj3 = obj2[obj2 > 0]
print(obj3)
obj4 = obj2 * 3
print(obj4)

print('uno' in obj2)
print('cinco' in obj2)
```

Operando con varias series

```
países = [ 'Chile', 'Colombia', 'Mexico', 'Peru' ]  
pob = [18.286, 49.067, 122.916, 31.660]  
obj_pob = Series(pob, index = países)  
print(obj_pob)  
print(u"Población de Chile: %f" % obj_pob[ 'Chile' ])
```


Operando con varias series

```
países2 = ['Argentina', 'Chile', 'Colombia']  
obj_pob2 = Series(obj_pob, index=países2)  
print(obj_pob2)
```

Operando con varias series

```
dic_pob = {'Chile': 18.286, 'Colombia': 49.067,  
           'Mexico': 122.916, 'Peru': 31.660}  
obj_pob3 = Series(dic_pob, index=paises2)  
print(obj_pob3)  
print(obj_pob3.isnull())
```

Operando con varias series

```
obj_pob4 = obj_pob2 + obj_pob3  
print(obj_pob4)
```

Operando con varias series

```
obj_pob4.name = 'poblacion'  
obj_pob4.index.name = 'pais'  
print(obj_pob4)
```

DataFrames

Operando con DataFrames

```
países = [ 'Chile', 'Colombia', 'Mexico', 'Peru' ]  
pob2000 = [15.412, 42.120, 100.088, 25.952]  
pob2017 = [18.286, 49.067, 122.916, 31.660]  
ano2000 = [2000, 2000, 2000, 2000]  
ano2017 = [2017, 2017, 2017, 2017]
```

Operando con DataFrames

```
datos = { 'pais': paises+paises ,  
          'año': ano2000+ano2017 ,  
          'pob': pob2000+pob2017 }
```

```
frame = DataFrame(datos)  
print(frame)
```

Operando con DataFrames

```
frame = DataFrame(datos, columns =  
    ['pais', 'año', 'pob'])  
print(frame)
```


Seleccionando Columnas

```
colPob = frame['pob']  
print(colPob)  
colPob2 = frame.pob  
print(colPob2)
```

Seleccionando Filas

```
print(frame.iloc[0])  
print(frame.iloc[5])
```

Agregando Columnas

```
frame['status'] = 'fundador'  
print(frame)
```

Agregando Columnas

```
frame['ranking'] = Series(range(8))  
print(frame)
```