

Python 2: Listas, Ejecución Iterativa, 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

Primer Semestre de 2018

Contenidos

- 1 Listas
- 2 Ejecución Iterativa en Python
- 3 Ejecución Iterativa: Número de iteraciones
- 4 Iteración condicional
- 5 Diccionarios
- 6 Series en Python (pandas)
- 7 DataFrames
- 8 Leyendo archivos en dataframes

Listas

Listas en Python

Un tipo de variable para almacenar varios valores, no solo uno

Podemos acceder a la lista recorriendo todos los elementos uno por uno

Podemos acceder a elementos específicos de la lista

Listas en Python

En un script de Python:

```
# -*- coding: utf-8 -*-  
lista_de_numeros = [1, 2, 3, 4, 5]  
print lista_de_numeros  
lista_de_strings = ['hola', 'esta', 'es', 'una', 'lista']  
print lista_de_strings  
print lista_de_numeros[1]  
print lista_de_numeros[3]  
print lista_de_strings[0]  
print lista_de_strings[2]  
print lista_de_strings[4]
```

Listas en Python: range()

```
# -*- coding: utf-8 -*-  
otra_lista = range(5)  
print otra_lista  
otra_lista_mas = range(2,5)  
print otra_lista_mas
```

Listas en Python: `append`

```
# -*- coding: utf-8 -*-  
mi_lista = range(5)  
print mi_lista  
mi_lista.append(10)  
print mi_lista  
mi_lista.append(50)  
print mi_lista
```

Ejecución Iterativa en Python

Recorriendo listas (para una sola lista)

```
# -*- coding: utf-8 -*-  
frutas_disp = ['fresa', 'uva', 'naranja', 'banano']  
for i in frutas_disp:  
    print u"La fruta %s está disponible" % i
```

Recorriendo listas (con contador)

```
# -*- coding: utf-8 -*-  
frutas_disp = ['fresa', 'uva', 'naranja', 'banano']  
frutas_cant = [40, 56, 25, 102]  
for i in range(4):  
    print "i=%d" % i  
    print u"La fruta %s está disponible" \  
        % frutas_disp[i]  
    print u"Hay %d frutas disponibles de \  
este tipo" % frutas_cant[i]
```

Utilizando listas

```
# -*- coding: utf-8 -*-
frutas_inventario = [40, 56, 25, 102]
frutas_pedido = [100, 200, 350, 500]
frutas_nuevo_inventario = []
for i in range(4):
    nuevo_inventario_temp = \
        frutas_inventario[i] + \
        frutas_pedido[i]
    frutas_nuevo_inventario.append(\
        nuevo_inventario_temp)
print u"El nuevo inventario es", \
    frutas_nuevo_inventario
```

Ejecución Iterativa: Número de iteraciones

Ejecución iterativa - Número de iteraciones

```
# -*- coding: utf-8 -*-  
for i in range(10):  
    print " %f:  %f+ %f = %f" \  
        % (i, 2**i, 3**i, 2**i + 3**i)
```

Iteración condicional

Iteración condicional

```
# -*- coding: utf-8 -*-  
remanente = 10  
print "Remanente inicial: %d" % remanente  
while remanente > 0:  
    remanente = remanente - 2  
    print "Remanente actual: %d" % remanente
```

Iteración condicional - Anidadas

```
# -*- coding: utf-8 -*-  
limite = 10  
i = 1  
while i < limite:  
    j = 1  
    while j < 10:  
        suma_cuadrados = i**2 + j**2  
        print " %d^2 + %d^2: %d" \  
            %(i,j,suma_cuadrados)  
        j = j + 1  
        suma_cuadrados = i**2 + j**2  
    i = i + 1
```


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

```
# -*- coding: utf-8 -*-  
moneda ={'Colombia': 'COP',  
        'EEUU': 'USD',  
        'Suiza': 'CHF'  
        }  
  
print moneda  
  
moneda['Reino Unido'] = 'GBP'  
moneda['Japon'] = 'JPY'  
print moneda  
  
print moneda['Suiza']  
print "La moneda de Colombia es ", moneda['Colombia']
```

Más diccionarios en Python

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

moneda['Reino Unido'] = 'GBP'
moneda['Japon'] = 'JPY'
capitales ={'Colombia': 'Bogota',
            'Japon': 'Tokio',
            'Reino Unido': 'Londres',
            'EEUU': 'Washington',
            'Suiza': 'Zurich'
            }

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

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'  
}  
moneda['Reino Unido'] = 'GBP'  
moneda['Japon'] = 'JPY'  
capitales ={'Colombia': 'Bogota',  
            'Japon': 'Tokio',  
            'Reino Unido': 'Londres',  
            }
```

Datos faltantes en Python (cont.)

```
for pais, abbrev in moneda.items():
    capital = capitales.get(pais, None)
    if capital:
        print "La moneda de %s es %s y su capital \
es %s" %(pais, abbrev, 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 "Valores:\n", obj1.values  
print "Indices:\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
```

```
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=países2)  
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 ,  
          'pob': pob2000+pob2017 ,  
          'año': ano2000+ano2017 }
```

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

Operando con DataFrames

```
print "-"*10  
frame = DataFrame(datos, columns =  
    ['pais', 'año', 'pob'])  
print frame
```

Seleccionando Columnas

```
print "-"*10  
colPob = frame['pob']  
print colPob  
colPob2 = frame.pob  
print colPob2
```

Seleccionando Filas

```
print "-" * 10  
print frame.ix[0]  
print frame.ix[5]
```

Agregando Columnas

```
print "-"*10  
frame['status'] = 'fundador'  
print frame
```

Agregando Columnas

```
print "-"*10  
frame['ranking'] = Series(range(8))  
print frame
```


Leyendo archivos en dataframes

Leyendo archivos

```
from pandas import Series , DataFrame  
import pandas as pd
```

```
filename = "data_blood.txt"  
df = pd.read_csv(filename , header=None , sep="\s+" ,  
    names = [u'Indice' , u'Uno' , u'Edad' ,  
        u'Presion Sangre'])  
print df
```

```
df2 = pd.read_table(filename , header=None , sep="\s+" ,  
    names = [u'Indice' , u'Uno' , u'Edad' ,  
        u'Presion Sangre'])  
print df2
```

Leyendo archivos

```
df3 = DataFrame(df, columns=[u'Edad',  
    u'Presion  Sangre'])  
print df3  
  
s1 = df[u'Presion  Sangre'].values  
print s1
```