

03_Pythonico_es_mas_bonito_2

January 29, 2021

1 Python de cero a experto

Autor: Luis Miguel de la Cruz Salas

Python de cero a experto by Luis M. de la Cruz Salas is licensed under Attribution-NonCommercial-NoDerivatives 4.0 International

Objetivos. Revisar los conceptos de funciones, *docstring*, excepciones, iterables, mapeo, filtrado, funciones lambda y *comprehensions*.

1.1 Convertidor de unidades

El fútbol americano es un derivado del rugby inglés y nació hace más de cien años en Estados Unidos. Es bastante popular en ese país y se practica en más de 60 países. Para un espectador principiante que vive en un país donde se utiliza el Sistema Internacional de Unidades (SI) no es sencillo entender las dimensiones de la cancha dado que están en yardas y en pies.

1.1.1 Ejemplo 1.

El lanzamiento de balón, de futbol americano, más largo registrado fue de Vinny Testaverde, quien supuestamente lanzó una pelota de 80 yardas en 1988. Randall Cunningham una vez registró 76 yardas y Brett Favre lanzó una bomba de 75 yardas en la competencia del 97. El mariscal de campo Patrick Mahomes asegura que puede lanzar el balón 83 yardas.

1. Realice un código que convierta yardas a metros y ordene los lanzamientos mencionados en el párrafo anterior de mayor a menor. Utilice funciones para realizar lo anterior.

Funciones y documentación

SOLUCIÓN.

```
[1]: def yardaToMeter(l):  
      return l / 1.093  
  
[2]: lanzamientos = {'Vinny Testaverde':80, 'Randall Cunningham':76, 'Brett Favre':  
      ↪75, 'Patrick Mahomes':83}  
      lanzamientos  
  
[2]: {'Vinny Testaverde': 80,  
      'Randall Cunningham': 76,  
      'Brett Favre': 75,
```

```
'Patrick Mahomes': 83}
```

```
[3]: for key, val in lanzamientos.items():  
      print(key, val)
```

```
Vinny Testaverde 80  
Randall Cunningham 76  
Brett Favre 75  
Patrick Mahomes 83
```

```
[4]: lanza_metros = {}  
     for key, val in lanzamientos.items():  
         lanza_metros[key] = yardaToMeter(val)  
  
lanza_metros
```

```
[4]: {'Vinny Testaverde': 73.19304666056725,  
      'Randall Cunningham': 69.53339432753889,  
      'Brett Favre': 68.6184812442818,  
      'Patrick Mahomes': 75.93778591033852}
```

```
[5]: sorted(lanza_metros)
```

```
[5]: ['Brett Favre', 'Patrick Mahomes', 'Randall Cunningham', 'Vinny Testaverde']
```

```
[6]: sorted(lanza_metros.values())
```

```
[6]: [68.6184812442818, 69.53339432753889, 73.19304666056725, 75.93778591033852]
```

```
[7]: import operator  
     sorted(lanza_metros.items(), key=operator.itemgetter(1))
```

```
[7]: [('Brett Favre', 68.6184812442818),  
      ('Randall Cunningham', 69.53339432753889),  
      ('Vinny Testaverde', 73.19304666056725),  
      ('Patrick Mahomes', 75.93778591033852)]
```

```
[8]: sorted(lanza_metros.items(), key=lambda x: x[1])
```

```
[8]: [('Brett Favre', 68.6184812442818),  
      ('Randall Cunningham', 69.53339432753889),  
      ('Vinny Testaverde', 73.19304666056725),  
      ('Patrick Mahomes', 75.93778591033852)]
```

```
[9]: sorted(lanza_metros.items(), key=lambda x: x[1], reverse=True)
```

```
[9]: [('Patrick Mahomes', 75.93778591033852),  
      ('Vinny Testaverde', 73.19304666056725),  
      ('Randall Cunningham', 69.53339432753889),  
      ('Brett Favre', 68.6184812442818)]
```

```
[10]: from collections import OrderedDict  
      OrderedDict(sorted(lanza_metros.items(), key=lambda x: x[1]))
```

```
[10]: OrderedDict([('Brett Favre', 68.6184812442818),  
                  ('Randall Cunningham', 69.53339432753889),  
                  ('Vinny Testaverde', 73.19304666056725),  
                  ('Patrick Mahomes', 75.93778591033852)])
```

Excepciones

Iterables, Mapeo y Filtrado

Lambda Expressions y Reduce

Comprehensions

1.1.2 Miniproyecto 2.

Obtener información de internet sobre los mariscales de campo que tienen el mayor número de yardas de todos los tiempos. Para ello siga las instrucciones que se dan en “Obteniendo datos de la web”. Con esa información hacer lo siguiente: 1. Transformar los datos de DataFrame a un diccionario usando *comprehensions*. 2. Corregir los nombres para que solo la primera letra del nombre y apellido sean mayúscula. 3. Generar un menú que permita al usuario elegir en qué unidades desea ver la información. * Metros. * Pies. * Pulgadas * Centímetros. 4. El usuario podrá decidir el rango de jugadores que desea ver, por ejemplo: 4-8, y solo mostrará la información de los jugadores del 4 al 8. Si el usuario pone 0-0 entonces se mostrará toda la lista. 5. La información se muestra en dos columnas: Jugador, Yardas. 6. Una vez mostrada la información, el usuario podrá elegir entre salir o hacer otra consulta. 7. Para las conversiones de unidades usar funciones lambda. 8. Usar manejo de excepciones para los posibles errores que pueda cometer el usuario.

1.1.3 Obteniendo datos de la web.

```
[11]: import webbrowser  
      website = 'http://www.espn.com/nfl/history/leaders/_/stat/passyards'  
      webbrowser.open(website)
```

```
[11]: True
```

```
[12]: import pandas as pd
```

```
[13]: # Antes de ejecutar esta celda, seleccione todos los datos de la tabla,  
      # incluyendo los títulos y copiarlos al clipboard, es decir teclear [Ctrl + c]
```

```
#-----
# Es probable que en Linux se obtenga un error. La forma de resolverlo es como
↳ sigue
#-----
# You may get an error message that says: "Pyperclip could not find a copy/paste
↳ mechanism for your system. Please see https://pyperclip.readthedocs.io/en/
↳ latest/introduction.html#not-implemented-error for how to fix this."
#
# In order to work equally well on Windows, Mac, and Linux, Pyperclip uses
↳ various mechanisms to do this. Currently, this error should only appear on
↳ Linux (not Windows or Mac). You can fix this by installing one of the copy/
↳ paste mechanisms:
#
# sudo apt-get install xsel to install the xsel utility.
# sudo apt-get install xclip to install the xclip utility.
# pip install gtk to install the gtk Python module.
# pip install PyQt4 to install the PyQt4 Python module.
#-----
nfl_frame = pd.read_clipboard(engine='python')
nfl_frame
```

```
[13]:
```

	RK	PLAYER	YDS
0	1	DREW BREES	80,358
1	2	TOM BRADY	79,204
2	3	Peyton Manning	71,940
3	4	Brett Favre	71,838
4	5	Philip Rivers	63,440
5	6	Dan Marino	61,361
6	7	BEN ROETHLISBERGER	60,348
7	8	Eli Manning	57,023
8	9	MATT RYAN	55,767
9	10	John Elway	51,475
10	11	AARON RODGERS	51,245
11	12	Warren Moon	49,325
12	13	Fran Tarkenton	47,003
13	14	Carson Palmer	46,247
14	15	Vinny Testaverde	46,233
15	16	MATTHEW STAFFORD	45,109
16	17	Drew Bledsoe	44,611
17	18	Dan Fouts	43,040
18	19	JOE FLACCO	40,931
19	20	Kerry Collins	40,922

```
[ ]: # Si no logró obtener los datos del sitio web, puede abrir el archivo local
# 'all_time_pass_yards.csv' con el siguiente comando:
# nfl_frame = pd.read_csv('all_time_pass_yards.csv')
# nfl_frame
```

```
[14]: # Convertir el Dataframe en un diccionario
dummy = nfl_frame.to_dict('split')
```

```
[15]: # Usar los datos de la entrada 'datos'
dummy['data']
```

```
[15]: [[1, 'DREW BREES', '80,358'],
       [2, 'TOM BRADY', '79,204'],
       [3, 'Peyton Manning', '71,940'],
       [4, 'Brett Favre', '71,838'],
       [5, 'Philip Rivers', '63,440'],
       [6, 'Dan Marino', '61,361'],
       [7, 'BEN ROETHLISBERGER', '60,348'],
       [8, 'Eli Manning', '57,023'],
       [9, 'MATT RYAN', '55,767'],
       [10, 'John Elway', '51,475'],
       [11, 'AARON RODGERS', '51,245'],
       [12, 'Warren Moon', '49,325'],
       [13, 'Fran Tarkenton', '47,003'],
       [14, 'Carson Palmer', '46,247'],
       [15, 'Vinny Testaverde', '46,233'],
       [16, 'MATTHEW STAFFORD', '45,109'],
       [17, 'Drew Bledsoe', '44,611'],
       [18, 'Dan Fouts', '43,040'],
       [19, 'JOE FLACCO', '40,931'],
       [20, 'Kerry Collins', '40,922']]
```

```
[16]: # Transformar estos datos en un diccionario como sigue:
datos_yardas = { qb[1]:qb[2] for qb in dummy['data']}
datos_yardas
```

```
[16]: {'DREW BREES': '80,358',
       'TOM BRADY': '79,204',
       'Peyton Manning': '71,940',
       'Brett Favre': '71,838',
       'Philip Rivers': '63,440',
       'Dan Marino': '61,361',
       'BEN ROETHLISBERGER': '60,348',
       'Eli Manning': '57,023',
       'MATT RYAN': '55,767',
       'John Elway': '51,475',
       'AARON RODGERS': '51,245',
       'Warren Moon': '49,325',
       'Fran Tarkenton': '47,003',
       'Carson Palmer': '46,247',
       'Vinny Testaverde': '46,233',
       'MATTHEW STAFFORD': '45,109',
```

```
'Drew Bledsoe': '44,611',  
'Dan Fouts': '43,040',  
'JOE FLACCO': '40,931',  
'Kerry Collins': '40,922'}
```

[]:

Iteradores y generadores

Decoradores

Biblioteca estándar