

Actividad de aprendizaje: Python básico

Gonzalez Soto Oscar Alejandro

2024-11-26

▼ Introducción

Este cuaderno es una plantilla inicial de trabajo, realice una copia en la misma carpeta. Renombre la copia con su nombre, sin tildes ni ñes con el siguiente formato "**Apellido1_Apellido2_Nombre1_Nombre2.ipynb**" (Ejemplo: "**Ibanez_Forero_Luis_Eduardo.ipynb**") para trabajar sobre él.

- Cuando termine de resolver el proyecto, ejecutelo todo (**NO LE DEBE APARECER NINGUN ERROR**), después descargue el proyecto en formato html, ábralo y guárdelo en formato pdf.
- Mis explicaciones van en azul y estas no deben ser modificadas, tampoco deben ser modificadas por ninguna razón las celdas donde diga "**Prueba de funcionamiento**". Su trabajo consiste en plantear las celdas necesarias, ya sean en formato *code* o formato *markdown* que le permita resolver el proyecto.

```
#####
#####
# Espacio para las librerías que va a necesitar
# Ejemplo: import numpy as np
#####
#####
import numpy as np
```

Ejercicio 1: Crear una variable llamada *a* que contenga el texto "**De aquel amor. De música ligera. Nada nos libra.**". Crear otra variable llamada *b* que contenga el texto "**Nada más queda**". Crear una tercera variable llamada *c* donde solo se tome el texto "**Nada más**" de *b*. Crear una cuarta variable llamada *d* donde repita el texto guardado en *b* cuatro veces. Por último, crear una variable llamada *e* que una todo el texto de la siguiente forma.

"De aquel amor. De música ligera. Nada nos libra. Nada más queda. Nada más Nada más queda. Nada más queda. Nada más queda. Nada más queda."

```
#####
## Este espacio debe ser llenado por el estudiante.
#####

a="De aquel amor. De música ligera. Nada nos libra. "
b="Nada más queda. "
c=b[:-7] #Forma alternativa: c = b[:9] --> c = 'Nada más '
d=b*4
e=a+b+c+d
```

```
#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####
import numpy as np
p11=isinstance(a, str) & (len(a)==49)
p12=isinstance(b, str) & (len(b)==16)
p13=isinstance(c, str) & (len(c)==9)
p14=isinstance(d, str) & (len(d)==64)
p15=isinstance(e, str) & (len(e)==138)
p1=p11 & p12 & p13 & p14 & p15
if p1==True:
    n1=1
```

```

else:
    n1=0

print(f"¡¡¡Felicidades!!! Tu nota hasta este momento es {n1} de 5" if p1==True else f"Hay un error, vuelve a revisar tu codigo. Tu nota hast
#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

↵ ¡¡¡Felicidades!!! Tu nota hasta este momento es 1 de 5

```

2) Crear un vector llamado $v = [789, 796, \dots, 922]$ que empiece en 789 y contenga los múltiplos de 7 usando la función `arange()` de la librería `numpy`. Después crear una matriz A de tamaño (4×5) usando la función `reshape()` de la librería `numpy`. Calcular la transpuesta de la matriz A usando el comando $A.T$ y guardar el resultado en una variable llamada B . Por último, realizar el producto matricial de A y B usando el operador `@`.

```

#####
## Este espacio debe ser llenado por el estudiante.
#####

v=np.arange(789,923,7)
A=v.reshape(4,5)
B=A.T
C=A@B
print(C)
print(A)
print(A.dtype)

↵ [[3224535 3365060 3505585 3646110]
   [3365060 3511710 3658360 3805010]
   [3505585 3658360 3811135 3963910]
   [3646110 3805010 3963910 4122810]]
   [[789 796 803 810 817]
   [824 831 838 845 852]
   [859 866 873 880 887]
   [894 901 908 915 922]]
   int64

#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

p21= isinstance(v,np.ndarray) and v.shape==(20,)
p22= isinstance(A,np.ndarray) and A.shape==(4,5)
p23= isinstance(B,np.ndarray) and B.shape==(5,4)
p24= isinstance(C,np.ndarray) and C.shape==(4,4) and np.linalg.det(C)==0

p2=p21 & p22 & p23 & p24
if p2==True:
    n2=1
else:
    n2=0

print(f"¡¡¡Felicidades!!! Tu nota hasta este momento es {n1+n2} de 5" if p2==True else f"Hay un error, vuelve a revisar tu codigo. Tu nota h
#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

↵ ¡¡¡Felicidades!!! Tu nota hasta este momento es 2 de 5

```

3) Crear una lista llamada `**meses_mayuscula**` donde convierta todos los elementos de la lista `**meses**` usando la función ``upper()`` por medio del código ``[mes.upper() for mes in meses]``. Crear una variable llamada `**filtro_bre**` donde se aplique un filtro sobre la variable `**meses_mayusculas**` usando la función ``filter()`` y ``lambda``, tal que se filtren los meses que contengan el término "BRE". Después crear una variable llamada `**filtro_ju**` donde se aplique un filtro sobre la variable `**meses_mayusculas**` usando la función ``filter()`` y ``lambda``, tal que se filtren los meses que contengan el término "JU". Por último, crear una variable llamada `**lista_bre_ju**` donde se unan las listas `**lista_bre**` y `**lista_ju**`.

```

#####
## Este espacio debe ser llenado por el estudiante.
#####

meses = ['enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio', 'julio', 'agosto', 'septiembre', 'octubre', 'noviembre', 'diciembre']

```

```

meses_mayusculas = [mes.upper() for mes in meses]

filtro_bre=filter(lambda mes: 'BRE' in mes,meses_mayusculas)
filtro_ju=filter(lambda mes: 'JU' in mes,meses_mayusculas)

lista_bre=list(filtro_bre)
lista_ju=list(filtro_ju)

lista_bre_ju=lista_bre+lista_ju

#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

p31=set([mes.isupper() for mes in meses_mayusculas])==set([True]*12)
p32=len(lista_bre)==5
p33=len(lista_ju)==2
p34=len(lista_bre_ju)==7

p3=p31 & p32 & p33 & p34
if p3==True:
    n3=1
else:
    n3=0

print(f"¡¡¡Felicidades!!! Tu nota hasta este momento es {n1+n2+n3} de 5" if p3==True else f"Hay un error, vuelve a revisar tu código. Tu nota es {n3} de 5")
#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

→ ¡¡¡Felicidades!!! Tu nota hasta este momento es 3 de 5

```

4) Se leyó el dataset **movies** de una url. A partir de una transformación se obtuvieron dos listas llamadas **cabecera** y **lineas_sin_cabecera**. Así pues, se pide: - Crear una lista vacía llamada **nombre_pelicula**, luego usar el método `append()` y el método `split()` para seleccionar el nombre de las películas, tal y como se muestra en el siguiente ejemplo

```

nombre_pelicula=[]
for l in lineas_sin_cabecera:
    nombre_pelicula.append(l.split(',')[0])

```

- Crear listas vacías llamadas **año** y **género** y repetir el procedimiento anterior, seleccionando las variables **year** y **genre** respectivamente del dataset.
- Crear una lista vacía llamada **recaudacion_mundial** y aplicar el siguiente código:

```

for l in lineas_sin_cabecera:
    valor=re.findall(r'\d{1,3}.\d{1,2}\S',l.split(',')[0])
    recaudacion=''.join(valor)
    recaudacion_mundial.append(recaudacion)

```

```

i=recaudacion_mundial.index('')
recaudacion_mundial[i]='0'
recaudacion_mundial=[float(x) for x in recaudacion_mundial]

```

```

#####
## Lectura de datos
## Esta celda no debe ser modificada
#####

import urllib.request
import re

url = 'https://gist.github.com/tianguechen/b68782efa49a16edaf07dc2cdaa855ea/raw/0c794a9717f18b094eabab2cd6a6b9a226903577/movies.csv'

with urllib.request.urlopen(url) as datos:

```

```

lineas=str(datos.readlines()).split('\n')

lineas_sin_cabecera=[]
for linea in lineas:
    if linea==lineas[0]:
        cabecera=lineas[0][5:]
    else:
        if linea==lineas[len(lineas)-1]:
            lineas_sin_cabecera.append(linea[5:-1])
        else:
            lineas_sin_cabecera.append(linea[5:])

print(cabecera)
for l in lineas_sin_cabecera[:5]:
    print(l)
#####
## Lectura de datos
## Esta celda no debe ser modificada
#####

↳ lm,Genre,Lead Studio,Audience score %,Profitability,Rotten Tomatoes %,Worldwide Gross,Year

#####
## Este espacio debe ser llenado por el estudiante.
#####
nombre_pelicula=[]
for l in lineas_sin_cabecera:
    nombre_pelicula.append(l.split(',')[0])

anno = []
for l in lineas_sin_cabecera:
    anno.append(l.split(',')[7])

genero = []
for l in lineas_sin_cabecera:
    genero.append(l.split(',')[1])

recaudacion_mundial = []
for l in lineas_sin_cabecera:
    valor=re.findall(r'\d{1,3}.\d{1,2}\S',l.split(',')[2])
    recaudacion=''.join(valor)
    recaudacion_mundial.append(recaudacion)

i=recaudacion_mundial.index('')
recaudacion_mundial[i]='0'
recaudacion_mundial=[float(x) for x in recaudacion_mundial]

#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

import collections

p41=set(list(collections.Counter(genero).values()))==set([13, 41, 13, 4, 1, 1, 1, 1, 1, 1])
p42=set(list(collections.Counter(anno).values()))==set([20, 20, 14, 11, 11, 1])
p43=np.mean(recaudacion_mundial)==142.3733766233766
p44=len(nombre_pelicula)==77

p4=p41 & p42 & p43 & p44
if p4==True:
    n4=1
else:
    n4=0

print(f"¡¡¡Felicidades!!! Tu nota hasta este momento es {n1+n2+n3+n4} de 5" if p4==True else f"Hay un error, vuelve a revisar tu codigo. Tu

#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####

```

🔄 |||Felicidades!!! Tu nota hasta este momento es 4 de 5

5) Crear una lista llamada ****tupla_atributos**** donde cada componente de la lista sea una tupla que contenga los elementos de las variables ****genero****, ****anno**** y ****recaudacion_mundial****; por ejemplo

```
[('Romance', '2008', 41.94),
('Comedy', '2010', 19.62),
('Comedy', '2010', 26.66),
('Comedy', '2010', 43.04),
('Comedy', '2008', 219.37),
('Drama', '2011', 117.09),
('Animation', '2008', 521.28),
('Romance', '2007', 22.18),...]
```

Crear una variable llamada ****diccionario_peliculas**** cuyas llaves sean los nombres de las películas y sus valores las tuplas de la lista ****tupla_atributos****, ejemplo

```
{'Zack and Miri Make a Porno': ('Romance', '2008', 41.94),
'Youth in Revolt': ('Comedy', '2010', 19.62),
'You Will Meet a Tall Dark Stranger': ('Comedy', '2010', 26.66),
'When in Rome': ('Comedy', '2010', 43.04),
'What Happens in Vegas': ('Comedy', '2008', 219.37),
'Water For Elephants': ('Drama', '2011', 117.09),
'WALL-E': ('Animation', '2008', 521.28),
'Waitress': ('Romance', '2007', 22.18),
'Waiting For Forever': ('Romance', '2011', 0.03),...}
```

Por último, crear un **DataFrame**, llamado **df**, usando la librería **pandas** y usando el siguiente código

```
import pandas as pd
nom_keys=list(diccionario_peliculas.keys())
df=pd.DataFrame( [ diccionario_peliculas[key] for key in diccionario_peliculas.keys()],index= nom_keys,columns=[['genero','anno','recaudacion(mill
```

```
#####
## Este espacio debe ser llenado por el estudiante.
#####
```

```
# Creating tupla_atributos
tupla_atributos = [(genero[i], anno[i], recaudacion_mundial[i]) for i in range(len(genero))]
```

```
# Creating diccionario_peliculas
diccionario_peliculas = dict(zip(nombre_pelicula, tupla_atributos))
```

```
# Creating the DataFrame (df)
import pandas as pd
nom_keys = list(diccionario_peliculas.keys())
df = pd.DataFrame([diccionario_peliculas[key] for key in diccionario_peliculas.keys()],
                  index=nom_keys,
                  columns=[('genero',), ('anno',), ('recaudacion(millones_dolares)',)])
```

```
#####
## Prueba de funcionamiento
## Esta celda no debe ser modificada
#####
```

```
p51=np.array(tupla_atributos).shape==(77,3) and type(tupla_atributos)==list and type(tupla_atributos[0])==tuple
p52=type(diccionario_peliculas)==dict and set(list(diccionario_peliculas.keys()))==set(nombre_pelicula) and set(list(diccionario_peliculas.va
p53=type(df)==pd.core.frame.DataFrame and df.shape==(75,3) and set(list(df.columns))==set([('genero',), ('anno',), ('recaudacion(millones_dol
```

```
p5=p51 & p52 & p53
if p5==True:
    n5=1
else:
    n5=0
```

```
print(f"¡¡¡Felicidades!!! Tu nota hasta este momento es {n1+n2+n3+n4+n5} de 5" if p5==True else f"Hay un error, vuelve a revisar tu codigo. "
```

```
#####  
## Prueba de funcionamiento  
## Esta celda no debe ser modificada  
#####
```

```
↩ ¡¡¡Felicidades!!! Tu nota hasta este momento es 5 de 5
```

Double-click (or enter) to edit