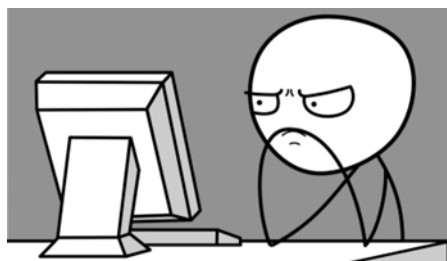
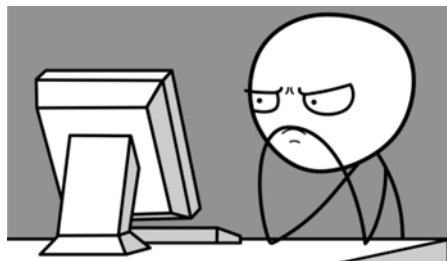


14_Ejercicios_Series

November 28, 2023



0.1 EJERCICIOS: Series

Para ejercitarte y afianzar lo aprendido sobre Series, completa los siguiente ejercicios.

La solución a los mismos se publicarán en el repo de ejercicios a lo largo del curso.

0.1.1 Ejercicio 0:

Importa pandas "como debe ser" y cualquier otra librería que emplees en el notebook

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

0.1.2 Ejercicio 1

Crea una serie con los siguientes valores e índice

- Valores = [45,46,12,10]
- Índices = ["Mum", "Dad", "Jing", "Jang"]

Dale un nombre que creas que refleje el sentido de la serie. Muestra el resultado por pantalla.

```
[2]: valores = [45,46,12,10]
indices = ["Mum","Dad","Jing","Jang"]

serie1 =pd.Series(valores, indices)
print(serie1)
```

```
Mum      45
Dad      46
Jing     12
Jang     10
dtype: int64
```

0.1.3 Ejercicio 2

Para la serie anterior y utilizando atributos de **Series**, muestra sus valores y su índice

```
[3]: print(serie1.values)
print(serie1.index)
```

```
[45 46 12 10]
Index(['Mum', 'Dad', 'Jing', 'Jang'], dtype='object')
```

0.1.4 Ejercicio 3

Crea una series con tus 4 series preferidas, pero como valores no uses los nombres, utiliza el año de su primera temporada y como índices los nombres de las series.

```
[4]: indices= ["Fauda", "Breaking Bad", "Stranger things", "Juego de Tronos"]
valores =[2015, 2008,2016, 2011]
#creo un dictionary, usando la funcion zip
dict_tv = dict(zip(indices, valores))
dict_tv
# creo la serie panda
serie2 = pd.Series(dict_tv)
serie2
```

```
[4]: Fauda                2015
      Breaking Bad       2008
      Stranger things    2016
      Juego de Tronos    2011
dtype: int64
```

0.1.5 Ejercicio 4

¿Cuántos años tiene la más antigua? (Resuelvelo utilizando un método, además de lo que necesites adicionalmente)

```
[5]: años_fin=[2022, 2013,2023,2019]
#creo nuevo serie con años fin de las series
serie3 = pd.Series(años_fin, indices)
serie3
#la serie mas antigua es
#print(serie1.min())
# la serie2 con el año menor correspnde a la serie mas antigua
#print(serie2.min())
# saber los años que ha durado la mas antigua
resta_3_2 = serie3 - serie2
#print(resta_1_2)
print(f" La serie mas antigua se ha retrasmitido durante {resta_3_2.min()}_
↳anyos, desde el año {serie2.min()} hasta el año {serie3.min()}")
```

La serie mas antigua se ha retrasmitido durante 5 años, desde el año 2008 hasta el año 2013

Encuentra cual es la más antigua y la más reciente empleando métodos de Series

```
[6]: print(serie2.min())
print(serie2.max())
```

2008

2016

0.1.6 Ejercicio 6

Accede a la más reciente empleando el índice explícito y el implícito

```
[13]: serie1
#metodo implicito .loc accedemos por etiqueta o condcion(bool)
reciente_i = serie2.loc["Breaking Bad"]
print(f"La serie mas reciente de la serie Panda 1 por el metodo implicito es_
↳del año {reciente_i}")
#metodo explicito, se accede por slicing o posicion en python
reciente_e = serie2.iloc[1:2]
print(f"La serie mas reciente de la serie Panda 1 por el metodo explicito es_
↳{reciente_e}")
```

La serie mas reciente de la serie Panda 1 por el metodo implicito es del anyo 2008

La serie mas reciente de la serie Panda 1 por el metodo explicito es Breaking Bad 2008
dtype: int64

0.1.7 Ejercicio 7

Haz un slicing de la primera serie (ejercicio 1) de forma que no incluya a "Mum" y usando índices explícitos (el índice de la Serie, vaya)

```
[14]: serie1
      serie1.iloc[1:]
```

```
[14]: Dad      46
      Jing     12
      Jang     10
      dtype: int64
```

0.1.8 Ejercicio 8

Crea un objeto Series de 10 elementos todos ellos con el valor "DataScience", cambia el elemento 5 por "TheBridge" y el 8 por "Bootcamp".

```
[23]: lista= ["DataScience"]*10
      #creo serie panda
      serie4= pd.Series(lista)
      serie4
      cambio1 = serie4[4] ="theBridge"
      serie4
      cambio2 = serie4[7] = "Bootcamp"
      serie4
```

```
[23]: 0    DataScience
      1    DataScience
      2    DataScience
      3    DataScience
      4     theBridge
      5    DataScience
      6    DataScience
      7     Bootcamp
      8    DataScience
      9    DataScience
      dtype: object
```

```
[ ]:
```