```
#%% md
# Task 1 - Python, NumPy and Pandas
#%% md
## Part 1 - Base Python
#%%
import numpy as np
import pandas as pd
#%% md
A drunk man leaves a bar. He can take a step north or a step south but since he is drunk the direction of each step is random. Write a function that takes an array of n random numbers  from zero to one and returns his position after n steps. (His position is the number of steps he ends up to the north or south of the bar.) <br>
Hint: You can define a step north if the random number is greater  than or equal to 0.5 and step south otherwise.
#%%
n = 100
randomArray = np.random.random(n)
def random_walk(list):
 stepSouth = 0
 stepNorth = 0
 totelWay = 0
 for a in list:
  if a >= 0.5:
   stepNorth = stepNorth + 1
  if a < 0.5:
   stepSouth = stepSouth + 1
 if stepNorth > stepSouth:
  totelWay = stepNorth - stepSouth
 if stepNorth < stepSouth:
  totelWay = stepSouth - stepNorth
 return totelWay
#%%
print(random_walk(randomArray))
#%% md
Now test your function using array x as your input:
#%%
x = np.array([0.04213258 ,0.52710927 ,0.77192904 ,0.31142186, 0.14499949 ,0.6811671, 0.03166511, 0.21591565, 0.03623045,
 0.33980672 ,0.14749022 ,0.67885345, 0.71780702 ,0.8916977 , 0.99759387 ,0.10180522, 0.46661761, 0.40967586, 0.94219329,
 0.28663323, 0.06680141, 0.95422541, 0.24546625, 0.88318858, 0.65294073, 0.22424247, 0.68488825, 0.83470679, 0.53413563,
 0.29258516, 0.5885011,  0.13184497, 0.63696717, 0.05045391, 0.36557114, 0.41776146, 0.234408,   0.18254025, 0.19775767,
 0.00195501, 0.08346885, 0.65614844, 0.24971543, 0.64096734, 0.60999048, 0.12395638, 0.30221827, 0.95643157, 0.88786033,
 0.44658592])

print(random_walk(x))
#%% md
Run your function 10000 times, each time with a new random array of length 50. (Each time the function runs he starts off at the bar, not in his previous position.) Calculate his average position over 10000 random walks.
#%%
totalNums = 50
randArr = np.random.random(n)
timesRun = 10000
def newWalk(list1):
 lp = 0
 for i in range(10000):
  if i == 0:
   lp = (lp + random_walk(list1))
  if i > 0:
   lp = (lp + random_walk(list1))/i
  return  lp

#%%
print(newWalk(randArr))
#%% md
Would the result change if a step north is taken when the random number is greater or equal to 0.2? Rewrite your function so it takes a second argument p  with default value 0.5 that defines the probability of a step north and run it on array x with p=0.2.
#%%
p = 0.2
def random_walk2(list):
 stepSouth = 0
 stepNorth = 0
 totelWay = 0
 for a in list:
  if a >= p:
   stepNorth = stepNorth + 1
  if a < p:
   stepSouth = stepSouth + 1
 if stepNorth > stepSouth:
  totelWay = stepNorth - stepSouth
 if stepNorth < stepSouth:
  totelWay = stepSouth - stepNorth
 return totelWay
print(random_walk2(randomArray))
#%% md
## Part 2 - NUMPY
#%%
# What code produces the following array? Save the array as a variable.
'''
[[1 2 3]
 [4 5 6]]
'''
array = np.array([[1,2,3],[4,5,6]])
newAr = np.array2string(array)
print(newAr)
#%%
# What code prints the number 5 from the array?
array[1][1]
#%%
# Reshape your array with one dimension instead of 2 and multiply all the numbers by 10
newArr = array.reshape((1,6))*10
print(newArr)
#%%
#What code prints the numbers 20, 30, 40 ?
print(newArr[0,1:4])
#%%
# What code prints the numbers 20,40,60 ?
print(newArr[0, 1::2])
#%%
# What code prints the data type of the array?
type(newArr)
#%%
# What code changes the data type to string? And then to float? (Hint you can use astype)
#change to string
stringArr = np.array2string(newArr)
print(type(stringArr), stringArr)
#change to float
floatArr = np.asarray(newArr, dtype=float)
print(floatArr)
#%% md
## Part 3 - PANDAS
#%% md
Read the file 'WorldCups.csv' into a data frame. <br>
Print out the first three rows.
#%%
df = pd.read_csv('WorldCups.csv')
#%%
#You might want to re read the data using one of the columns as  the index.
df
#%% md
What is the shape of the data? <br>
What are the data types of the columns?
#%%
df.shape
#%%
#Take a look at 'Attendance'. Why is its data type 'object'(string)?
df['Attendance']
#its a string beacose there are a lot of '.' in some of the numbers
#%%
#How can you change the data type of  'Attendance' to float?
df['Attendance'] = df['Attendance'].str.replace('.','')
df['Attendance'] = df['Attendance'].astype(float)
df['Attendance']
#%%
# Print out the rows of the data frame which have 'Country' as 'France'
df[df['Country'].str.contains('France')]
#%%
# Print out rows 10 to 15 (including 15)
df.iloc[10:16]
#%%
#Make a line plot of'GoalsScored'.
df['GoalsScored'].plot()
#%%
# Make a bar plot of the winners.
ls = df["Winner"].value_counts()
ls.plot.bar()
```