

# Unit 1: Data Handling using Pandas

## Introduction

Python Pandas Pandas is a software library written for the Python programming language for data manipulation and analysis. Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. It is used for data analysis in Python and developed by Wes McKinney in 2008.

There are 3 well-established python libraries namely NumPy, Pandas and Matplotlib specially for scientific and analytical use. These libraries allow us to manipulate, transform and visualise data easily and efficiently.

Using the Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data. These steps are- load, prepare, manipulate, model and analyse

## Benefits of Pandas

The benefits of pandas over using the languages are

- Data representation: It can easily represent data in a form naturally suited for data analysis through its DataFrame and Series data structures in a concise manner. Doing the equivalent in C/C++ or Java would require many lines of custom code, as these languages were not built for data analysis but rather networking and kernel development.
- Clear code: The clear API of the Pandas allows you to focus on the core part of the code. So, it provides clear code.

## Matplotlib

It is an amazing visualization library in Python that used for 2D plots of arrays. It is a multi-platform data visualization library which build NumPy arrays. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers and various graphical user interface toolkits. To get matplotlib up and running in our environment, we need to import it. `import matplotlib.pyplot as plt`.

## Data structures in Pandas:

Data structure is defined as the storage and management of the data for its efficient and easy access in the future where the data is collected, modified and the various types of operations are performed on the data respectively. Pandas provides two data structures for processing the data, which are explained below:

**(1) Series:** It is one dimensional object similar to an array, list or column in a table. It will assign a labelled index to each item in the series. By default, each item will receive an index label from 0 to N, where N is the length of the series minus one.

**(2) DataFrame:** It is a tabular data structure comprised of rows and columns. DataFrame is defined as a standard way to store data which has two different indexes i.e., row index and column index.

### **Know the Terms:**

**Pandas:** The Pandas is a high-performance open source library for data analysis in Python.

**Matplotlib:** It is a visualization library in Python that used for 2D plots of arrays.

**Series:** It is a one-dimensional array containing a sequence of values. Each value has a data label associated with it also called its index. DATA HANDLING USING PANDAS-I 7

**Selection:** This function returns data corresponding to axis labels matching criteria.

**Indexing:** This function is used to get or set the index labels of the given series object.

**Slicing:** Slicing is a powerful approach to retrieve subsets of data from a Pandas object.

**Series** is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56, ... Pandas Series is a one-dimensional labelled array capable of holding any data type (integer values, string values, double value and more). A Series represents a single column in memory. Series can only contain single list with index, whereas DataFrames can be made of more than one series or we can say that a DataFrames is a collection of series.

### **Key Points of Series**

- Homogeneous data
- Size Immutable
- Values of Data Mutable

A pandas Series can be created using the following constructor – pandas. Series (data, index, dtype, copy)

### **CREATION OF SERIES**

A series can be created using various inputs like:

- List
- Tuple
- Dictionary
- Array

- Scalar value or constant

### Create an Empty Series

A basic series, which can be created is an Empty Series.

1. Write a program to create an empty series

```
#import the pandas library and aliasing as pd
```

Ans.

```
import pandas as pd
s =pd.Series()
print (s)
```

**Output** is as follows: Series ([], dtype: float64)

### Create a Series from List

2. Write a program to create a series by given list ['red','green','blue'].

Ans.

```
import pandas as pd
c=['red','green','blue']
p=pd.Series(c)
print (p)
```

OR

```
import pandas as pd
c=['red','green','blue']
p=pd.Series(data=c)
print (p)
```

OR

```
import pandas as pd
p=pd.Series(['red','green','blue'])
print (p)
```

#### Output:

```
0  red
1  green
2  blue
```

We did not pass any index, so by default, it assigned the indexes ranging from 0 to len(data)-1, i.e., 0 to 2.

3. Modify the above program and change the index values to r, g, b

Ans.

```
import pandas as pd
c=['red','green','blue']
p=pd.Series(c,index=['r','g','b'])
```

```
print (p)
```

OR

```
import pandas as pd
p=pd.Series(['red','green','blue'],index=[r,g,b])
print (p)
```

OR

```
import pandas as pd
c=['red','green','blue']
p=pd.Series(c,[r,g,b])
print (p)
```

OR

```
import pandas as pd
c=['red','green','blue']
p=pd.Series(data=c,index=[r,g,b])
print (p)
```

Output:

```

r      red
g    green
b     blue
```

### Create a Series from Tuple

4. Write a program to create a series from a given Tuple data  
data=('1','Aman',86.3,'A').

Ans:

```
import pandas as pd
data=('1','Aman',86.3,'A')
p=pd.Series(data)
print (p)
```

Output

```
0    1
1  Aman
2  86.3
3    A
```

### Create a Series from Dictionary

A dictionary can be passed as input and if no index is specified, then keys of the dictionary are used to represent the index of the Series.

- 5.** A dictionary data = {'a':0., 'b':1., 'c':2} is given.  
Write a program to create series from dictionary data.

Ans.

```
import pandas as pd
data = {'a':0., 'b':1., 'c':2.}
s = pd.Series(data)
print(s)
```

Its output is as follows –

```
a 0.0
b 1.0
c 2.0
dtype: float64
```

If index is passed, the values will be displayed in the same sequence as index values are passed.

```
import pandas as pd
data = {'a':0., 'b':1., 'c':2.}
s = pd.Series(data, index=['b', 'c', 'a'])
print(s)
```

Its output is as follows –

```
b 1.0
c 2.0
a 0.0
```

You can also show the specified values by giving their keys as index values.

- 6.** Modify the above program and display values of 'a' and 'c' only.

Ans.

```
import pandas as pd
data = {'a':0., 'b':1., 'c':2.}
s = pd.Series(data, index=['a', 'c'])
print(s)
```

Its output is as follows –

a 0.0

c 2.0

If index value is passed other than keys value then NaN (Not a Number) as value will be displayed.

**7.** Give the output:

Ans.

```
import pandas as pd
```

```
data = {'a':0,'b':1.,'c':2.}
```

```
s =pd.Series(data,index=['b','c','d','a'])
```

```
print(s)
```

Its output is as follows –

b 1.0

c 2.0

d NaN

a 0.0

dtype: float64

Observe – Index order is persisted and the missing element is filled with NaN (Not a Number).

### **Create a Series from Scalar/Constant value**

If data is a scalar value, an index must be provided. The value will be repeated to match the length of index.

**8.** Write a program to create a series to print scalar value “5” four times. Ans.

```
import pandas as pd
```

```
s =pd.Series(5, index=[0,1,2,3])
```

```
print(s)
```

Its output is as follows –

0     5

1     5

2     5

3     5

As you can see the “5” is printed 4 times because the length of index is 4.

### **Create a Series from ndarray**

**9.** Write a program to create a series from ndarray with elements 'a','b','c','d'

Ans.

```
import pandas as pd
import numpy as np
data =np.array(['a','b','c','d'])
s =pd.Series(data)
print(s)
```

Its output is as follows –

```
0 a
1 b
2 c
3 d
dtype: object
```

**10.** Give the output:

```
import pandas as pd
import numpy as np
data =np.arange(10,15)
s =pd.Series(data)
print(s)
```

Ans.

```
0 10
1 11
2 12
3 13
4 14
```

## REINDEXING

Reindexing means to conform the data to match a given set of labels along a particular axis. Reorder the existing data to match a new set of labels. Reindexing does not mean providing new index values; in fact, it is used to reorder the existing data or print specified data.

**11.** Give the output:

```
import pandas as pd
name=['Raj','Ankur','Harsh']
p=pd.Series(name,index=[2,5,6])
print(p)
# Reindex the series and create a new series variable
p1=p.reindex([6,2,5])
print (p1)
```

Ans.

```
2 Raj
5 Ankur
6 Harsh
6 Harsh
2 Raj
5 Ankur
```

**12.** Give the output:

```
import pandas as pd
name=['Raj','Ankur','Harsh']
p=pd.Series(name,index=[2,5,6])
print(p) p1=p.reindex([2,5])
print (p1)
```

Ans.

```
2 Raj
5 Ankur
6 Harsh
dtype: object

2 Raj
5 Ankur
dtype: object
```

if other than existing index value is provided to reindex then NaN will be displayed.

**13.** Give the output:

```
import pandas as pd
name=['Raj','Ankur','Harsh']
p=pd.Series(name,index=[2,5,6])
print(p) p1=p.reindex([2,4,5])
print (p1)
```

Ans.

```
2 Raj
5 Ankur
6 Harsh
dtype: object
2 Raj
4 NaN
5 Ankur

dtype: object
```



## ALTER INDEX VALUES

The Series index function does not only allow you to display the index items, but you can also alter it as well. This example changes the actual index items and places the integer values as the index.

**14.** Give the output:

```
import pandas as pd
S = pd.Series([10,20,30,40,50], index = ['a', 'e', 'i', 'o', 'u'])
print(S)
# Assigning New Index Values
S.index = [1, 2, 3, 4, 5]
print("Series after new index values")
print(S)
```

Ans.

```
a 10
e 20
i 30
o 40
u 50
dtype: int64
```

```
Series after new index values
1 10
2 20
3 30
4 40
5 50
dtype: int64
```

## SIZE ATTRIBUTE

All pandas data structures are value-mutable (the values they contain can be altered). All pandas data structures are size mutable except Series. The length of a Series cannot be changed, i.e. number of columns and rows can't be altered once defined. Series is size immutable.

**15.** What is Series Size?

Ans. Size attribute returns the number of elements in the underlying data for the given series objects.

**16.** Give the output:

```
import pandas as pd
L=[10,20,30]
S=pd.Series(L)
print(S.size)
```

Ans. 3

## PRACTICE QUESTIONS

1. What is series? Explain with the help of an example.

Ans. Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index.

Example:

```
import pandas as pd # simple array
data =pd.Series([1,2,3,4,5])
print(data)
```

2. Write a suitable Python code to create an empty series.

Ans.

```
import pandas as pd
s=pd.Series()
print (s)
```

3. Write single line Pandas statement to declare a Pandas series named Packets having dataset as: [125, 92, 104, 92, 85, 116, 87, 90]

Ans. Packets = pd.Series([125, 92, 104, 92, 85, 116, 87, 90])

4. Write single line Pandas statement to declare a Pandas series named S having dataset as: (44,65,35,77,87,90)

Ans. S=pd.Series((44,65,35,77,87,90))

5. Write single line Pandas statement to declare a Pandas series named SR having dataset as: {1:'one',2:'two',3:'three'}

Ans. SR=pd.Series({1:'one',2:'two',3:'three'})

6. Write python code to create the Series EMP with following data (using Dictionary)

Ans.

```
import pandas as pd
d={"E1":"Sanya","E2":"Krish","E3":Rishav","E4":"Deepak"}
EMP=pd.Series(d)
EMP.index.name="code"
```

Code	
E1	Sanya
E2	Krish
E3	Rishav
E4	Deepak

```
print(EMP)
```

7. Write python series to print scalar value "100" 5 times with index values(1,2,3,4,5)

Ans.

```
import pandas as pd
s = pd.Series(100, index=[1, 2, 3, 4,5])
OR
s = pd.Series(100, [1, 2, 3, 4,5])
print (s)
```

8. Write python code to create the following series using Dictionary:

```
101 Harsh
102 Arun
103 Ankur
104 Harpahul
105 Divya
106 Jeet
```

Ans:

```
import pandas as pd
D={101:"Harsh",102:"Arun",103:"Ankur",104:"Harpahul",105:"Divya",
,106:"Jeet" }
s=pd.Series(D)
print(s)
```

9. Write a program to create a series by using given an array ['a','b','c','d'] and assign index values 100,101.....

Ans.

```
import pandas as pd
import numpy as np
data =np.array(['a','b','c','d'])
s =pd.Series(data,index=[100,101,102,103])
print(s)
```

Its output is as follows –

```
100 a
101 b
102 c
103 d
dtype: object
```

10. Give the output:

```
import pandas as pd
s =pd.Series(10, index=[5,6,7,9])
print(s)
```

Ans.

```
5 10
6 10
7 10
9 10
```

## ACCESSING DATA

### Accessing using head()

By default Series.head() function display top 5 rows. To print n no of top rows, pass n as parameter i.e. Series. head(n)

17. Write a code to create a series from empno list and show the first five rows empno = [101,102,103,104,105,106,107]

Ans.

```
import pandas as pd
p=pd.Series(empno)
print (p.head())
```

output:

```
0 101
1 102
2 103
3 104
4 105
```

18. Using the above series write a single line statement to show the first rows using head()

Ans. `print(p.head(3))`

output:

0 101

1 102

2 103

### **Accessing using tail()**

By default `Series.tail()` function display last 5 rows. To print n no of last rows, pass n as parameter i.e. `Series.tail(n)`

19. Write a code to create a series from empno list and show the last five rows `empno=[101,102,103,104,105,106,107]`

Ans.

```
import pandas as pd
```

```
p=pd.Series(empno)
```

```
print (p.tail())
```

output:

2 103

3 104

4 105

5 106

6 107

20. Fill the missing statements

```
import pandas as pd L=[101,102,103,104,105,106,107]
```

```
_ = pd.Series(L) #statement 1
```

```
print (p.__(3)) #statement 2
```

output:

4 105

5 106

6 107

Ans.

```
p=pd.Series(L) #statement 1
```

```
print (p.tail(3)) #statement 2
```

## Indexing

Pandas now supports three types of indexing.

(i) loc: is label based indexing.

(a) A single label

21. Give the output:

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
print(s.loc['a'])
print(s.loc[0])
```

Ans.

10

40

# in loc[0], 0 is interpreted as a label of the index. This is not an integer position along the index.

(b) A list of labels

22. Give the output:

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
print(s.loc[['b','c',1]])
```

Ans.

b 20

c 30

1 50

(c) A slice object with labels

A slice is a subset of series elements.

my\_series[start:stop:step] where start is the index of the first element to include, stop is the index of the item to stop and step sets the interval

23. Give the output:

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
print(s.loc['a':'c'])
```

Ans.

a 10

b 20

c 30

Note: Both the start and the stop are included, when present in the label index

(ii) iloc: integer position based.

This series can also be indexed by position (using integers) even though it has string index entries! The first item is at key 0, and the last item is at key -1

(a) input an integer

24. Give the output:

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
print(s.iloc[0])
```

Ans. 10

(b) input a list of integers

25. Write a program to print the values of 0,2,4 positions from Series s[10,20,30,40,50] using .iloc.

Ans.

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
print(s.iloc[[0,2,4]])
```

output

a 10

c 30

1 50

26. Write a program to print the values of 0,2,4 positions from Series s[10,20,30,40,50] using .iloc.

Ans. import pandas as pd

```
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
```

```
print(s.iloc[[0,2,4]])
```

output

a 10

c 30

1 50

(c) input a slice object with ints

A slice is a subset of series elements. my\_series[start:stop:step] where start is the index of the first element to include, stop is the index of the item to stop without including the stop value and step sets the interval.

27. Show the first 3 values from Series using iloc.

Ans.

```
import pandas as pd
```

```
s=pd.Series([10,20,30,40,50],index=['a','b','c',0,1])
```

```
print(s.iloc[0:3])
```

a 10

b 20

c 30

(iii) with [ ]: Accessing Data from Series with Position and Using Label (index). By using [ ] you can take advantage of both .loc and .iloc. You can access the records using [ ] directly.

If index value is not defined you can pass default index value 0,1,2... to access the elements.

If string index values are defined then elements can be accessed by both ways: passing default index values (0,1,2...) or passing defined string index values.

If integer index values are defined, [ ] will work only as .loc i.e. elements can be accessed by passing defined index values only.



28. Give the output:

```
import pandas as pd
M=[15,-10,56,39,-90,15]
p=pd.Series(M)
print(p[0])
print(p[[0,3,4]])
```

Ans.

```
15
0 15
3 39
4 -90
```

29. Give the output of the following program:

```
import pandas as pd
s=pd.Series([10,20,30,40,50],index=['a','b','c','d','e'])
print(s[0]) #or print(s.iloc[0])
print(s['a':'c']) #or print(s.loc['a':'c'])
print(s[2]) #or print(s.iloc[s[2]])
print(s['a']) #or print(s.loc['a'])
```

Its output is as follows –

Ans.

```
10
a 10
b 20
c 30
30
10
```

### 30. PRACTICE QUESTIONS

(i) Write python code to create the following series

101 Harsh

102 Arun

103 Ankur

104 Harpahul

105 Divya

106 Jeet

(ii) Show details of 1st 3 employees using head function

(iii) Show details of last 3 employees using tail function

(iv) Show details of 1st 3 employees without using head function

(v) Show details of last 3 employee without using tail function

(vi) Show value of index no 102

(vii) Show 2nd to 4th records

(viii) Show values of index no=101,103,105

(ix) Show details of "Arun"

**Ans.**

```
(i) import pandas as pd
    name=['Harsh','Arun','Ankur','Harpahul','Divya','Jeet']
    p=pd.Series(name,index=[101,102,103,104,105,106])
    print (p)
(ii) print (p.head(3))
(iii) print (p.tail(3))
(iv) print(p[:3]) or print(p.loc[101:103]) or print(p.iloc[0:3]) or
    print(p[[101,102,103]])
(v) print (p[-3:]) or print(p[3:]) or print(p[[104,105,106]])
(vi) print(p[102]) or print(p.loc[102])
(vii) print(p[1:4])
(viii) print(p[[101,103,105]])
(ix) print(p[p== 'Arun'])
```

	1 mark MCQ Questions	
1.	The name “Pandas” is derived from the term: a. Panel Data b. Panel Series c. Python Document d. Panel Data Fram	
2.	Given a Pandas series called HEAD, the command which will display the first 3 rows is _____. a. print(HEAD.head(3)) b. print(HEAD.Heads(3)) c. print(HEAD.heads(3)) d. print(head.HEAD(3))	
3.	To display First 15 rows of a series object ‘Ser1’, you may write: a. Ser1.tail(15) b. Ser1.head() c. Ser1.head(15) d. Ser1.tail()	
4.	Which of the following statement will install pandas library in python? a. Pip install pandas pd b. pip install pandas as pd c. pip install pandas.p	
1.	Which of the following statements is wrong? (a) Can't change the index of the Series. (b) We can easily convert the list, tuple, and dictionary into a series, (c) A Series represents a single column in memory. (d) We can create an empty Series.	
2.	What type of error is returned by the following statement ? import pandas as pa pa.Series ([1,2,3,4), index = ['a','b', 'c']) (a) Value Error (b) Syntax Error (c) Name Error (d) Logical error	
3.	Python pandas was developed by: a. Guido van Rossum b. Travis Oliphant c. Wes McKinney d. Brendan Eich	
4.	Pandas Series is: a. 2 Dimensional b. 3 Dimensional c. 1 Dimensional d. Multidimensional	
5.	Pandas is a: a. Package b. Language	

	c. Library d. Software	
6.	We can analyse the data in pandas with a. Series b. Data Frame c. Both of the above d. None of the above	
7.	What will be the output of the given code? import pandas as pd s = pd.Series([1,2,3,4,5], index=['akram','brijesh','charu','deepika','era']) print(s['charu']) a. 1 b. 2 c. 3 d. 4	
8.	27. Assuming the given series, named stud, which command will be used to print 5 as output? Amit 90 Ramesh 100 Mahesh 50 john 67 Abdul 89 Name: Student, dtype: int64 a. stud.index b. stud.length c. stud.values d. stud.size	
9.	A social science teacher wants to use a pandas series to teach about Indian historical monuments and its states. The series should have the monument names as values and state names as indexes which are stored in the given lists, as shown in the code. Choose the statement which will create the series: import pandas as pd Monument=['Qutub Minar','Gateway of India','Red Fort','Taj Mahal'] State=['Delhi','Maharashtra','Delhi','Uttar Pradesh'] a. S=df.Series(Monument,index=State) b. S=pd.Series(State,Monument) c. S=pd.Series(Monument,index=State) d. S=pd.series(Monument,index=State)	
10.	In Python Pandas, while performing mathematical operations on series, index matching is implemented and all missing values are filled in with ____ by default. i. Null ii. Blank iii. NaN iv. Zero	
11.	The data label associated with a particular value of series is called its _____ a. Data Value	

	b. Index c. Value d. None of the above	
12.	To display First 15 rows of a series object 'Ser1', you may write: i. Ser1.tail(15) ii. Ser1.head() iii. Ser1.head(15) iv. Ser1.tail()	
13.	Read the statements given below. Identify the right option from the following for Attribute and method/function. Statement A: Attribute always ends without parenthesis. Statement B: Function/Method cannot work without arguments. a. Both statements are correct. b. Both statements are incorrect. c. Statement A is correct, but Statement B is incorrect d. Statement A is incorrect, but Statement B is correct	
14.	<b>Assertion (A):</b> The output of addition of two series will be NaN, if one of the elements or both the elements have no value(s). <b>Reason (R) :</b> While performing mathematical operations on a series, by default all missing values are filled in with 0. a. Both statements are correct. b. Both statements are incorrect. c. Statement A is correct, but Statement B is incorrect d. Statement A is incorrect, but Statement B is correct	
	<b>2 Mark Qns</b>	
1.	What is series? Explain with the help of an example.	
2.	Consider a given Series , M1. Write a program in Python Pandas to create the series. <div style="text-align: center;">Marks</div> Term1    67 Term2    56 Term3    77 Term4    89 Display the even numbers from the series <pre>print(m1[m1%2==0])</pre>	
3.	What will be the output of the following code: <pre>import pandas as pd mydata=pd.Series( ['rajesh', 'amit', 'tarun', 'Radhika'] ) print(mydata &lt; 'rajesh' )</pre>	
4.	Given two series S1 and S2 <div style="display: flex; justify-content: space-between;"> <div>S1</div> <div>S2</div> </div> A 39    A 10 B 41    B 10 C 42    D 10 D 44    F 10 Find the output for following python pandas statements?	

	a. <code>S1[:2]*100</code> b. <code>S1 * S2</code>	
5.	<b>Write a Python program to create a series object, country using a list that stores the capital of each country.</b> <b>Note:</b> Assume four countries to be used as index of the series object are India, UK, Denmark, and Thailand having their capitals as New Delhi, London, Copenhagen, and Bangkok respectively.	
6.	What will be the output of the following code: <pre>import pandas as pd s1=pd.Series(data=2*[3,10]) print(s1)</pre>	
7.	Consider the following series named animal: Write the output of the command: <pre>print(animal[::-3])</pre> <pre> L      Lion B      Bear E      Elephant T      Tiger W      Wolf dtype: object </pre>	
8.	<b>Write the output of the given program:</b> <pre>import pandas as pd S1=pd.Series([5,6,7,8,10],index=['v','w','x','y','z']) l=[2,6,1,4,6] S2=pd.Series(l,index=['z','y','a','w','v']) print(S1-S2)</pre>	
9.	<b>Write the output of the given command:</b> <pre>import pandas as pd s=pd.Series([1,2,3,4,5,6],index=['A','B','C','D','E','F']) print(s[s%2==0])</pre>	
10.	<b>Complete the given Python code to get the required output as: Rajasthan</b> <pre>import _____ as pd di = {'Corbett': 'Uttarakhand', 'Sariska': 'Rajasthan', 'Kanha': 'Madhya Pradesh', 'Gir': 'Gujarat'} NP = _____. Series( _____ ) print(NP[_____])</pre>	
11.	<b>what will be the output of the following code?</b> <pre>import pandas as pd myser pd. Series ( [0, 0,0]) print (myser)</pre>	
12.	<b>Write a Python program to create a series object, country using a list that stores the capital of each country.</b> <b>Note:</b> Assume four countries to be used as index of the series object are India, UK, Denmark, and Thailand having their capitals as New Delhi, London, Copenhagen, and Bangkok respectively.	
13.	<b>Question:</b> Write the code to create a Series with names of 5 students and their marks out of 100 (Names will become index and Marks will become Data). Finally print the details of those students who scored more than 70 Marks.	

14.	Using the Series created, write commands for the following: a) Display the names of the months 3 through 7 from the Series MonthDays. b) Display the Series MonthDays in reverse order.	(4)
15.	Create the following Series and do the specified operations: a) EngAlph, having 26 elements with the alphabets as values and default index values. b) Vowels, having 5 elements with index labels 'a', 'e', 'i', 'o' and 'u' and all the five values set to zero. Check if it is an empty series. c) Friends, from a dictionary having roll numbers of five of your friends as data and their first name as keys. d) MTseries, an empty Series. Check if it is an empty series. e) MonthDays, from a numpy array having the number of days in the 12 months of a year. The labels should be the month numbers from 1 to 12.	(5)
16.	Using the Series created in Question 15, write commands for the following: a) Set all the values of Vowels to 10 and display the Series. b) Divide all values of Vowels by 2 and display the Series. c) Create another series Vowels1 having 5 elements with index labels 'a', 'e', 'i', 'o' and 'u' having values [2,5,6,3,8] respectively. d) Add Vowels and Vowels1 and assign the result to Vowels3. e) Subtract, Multiply and Divide Vowels by Vowels1. f) Alter the labels of Vowels1 to ['A', 'E', 'I', 'O', 'U'].	(5)

\*\*\*\*\*