

ITERATION ON DATAFRAME ASSIGNMENT

1. Create following given dataframe

	name	degree	score
0	aparna	MBA	90
1	pankaj	BCA	40
2	sudhir	M.Tech	80
3	Geeku	MBA	98

2. Now apply `iterrows()` function in order to get a each element of rows.

3. Now apply a `itertuples()` function in order to get tuple for each row.

4. Display the following output:

Output:

```
sudhir  
M.Tech  
80
```

5. Now apply a `iteritems()` function in order to retrieve an rows of dataframe.

6. Create the following dataframe and display the following:

Output:

```
Original DataFrame:
   age  name
0   10  Sujeet
1   11  Sameer
2   12   Sumit

Rows iterated using iterrows() :
Sujeet 10
Sameer 11
Sumit 12
```

7. Display the following using itertuple()

Output:

```
Original DataFrame:
   age  name
0   10  Sujeet
1  110  Sameer
2  120   Sumit

Rows iterated using itertuples() :
Sujeet 10
Sameer 110
Sumit 120
```

SOLUTIONS

1.

```
import pandas as pd
# dictionary of lists
dict = {'name':["aparna", "pankaj", "sudhir", "Geeku"],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
        'score':[90, 40, 80, 98]}

# creating a dataframe from a dictionary
df = pd.DataFrame(dict)
print(df)
```

2.

```
import pandas as pd

# dictionary of lists
dict = {'name':["aparna", "pankaj", "sudhir", "Geeku"],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
```

```

        'score':[90, 40, 80, 98]}
# creating a dataframe from a dictionary
df = pd.DataFrame(dict)

# iterating over rows using iterrows() function
for i, j in df.iterrows():
    print(i, j)
    print()

```

3.

```

import pandas as pd

# dictionary of lists
dict = {'name':['aparna', "pankaj", "sudhir", "Geeku"],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
        'score':[90, 40, 80, 98]}

# creating a dataframe from dictionary
df = pd.DataFrame(dict)

# using a itertuples()
for i in df.itertuples():
    print(i)

```

4.

```

columns = list(df)

for i in columns:

    # printing the third element of the column
    print (df[i][2])

```

5.

```

import pandas as pd

# dictionary of lists
dict = {'name':['aparna', "pankaj", "sudhir", "Geeku"],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
        'score':[90, 40, 80, 98]}

# creating a dataframe from a dictionary
df = pd.DataFrame(dict)

# using iteritems() function to retrieve rows
for key, value in df.iteritems():
    print(key, value)
    print()

```

6.

```

import pandas as pd

# list of dicts
input_df = [{'name':'Sujeet', 'age':10},
            {'name':'Sameer', 'age':11},
            {'name':'Sumit', 'age':12}]

```

```
df = pd.DataFrame(input_df)
print('Original DataFrame: \n', df)
```

```
print('\nRows iterated using iterrows() : ')
for index, row in df.iterrows():
    print(row['name'], row['age'])
```

7.

```
import pandas as pd

# list of dicts
input_df = [{'name':'Sujeet', 'age':10},
            {'name':'Sameer', 'age':110},
            {'name':'Sumit', 'age':120}]

df = pd.DataFrame(input_df)
print('Original DataFrame: \n', df)

print('\nRows iterated using itertuples() : ')
for row in df.itertuples():
    print(getattr(row, 'name'), getattr(row, 'age'))
```

EXTRA QUESTIONS

Q1.What do you mean by Iterating over rows and columns in Pandas DataFrame?

Iteration is a general term for taking each item of something, one after another. Pandas DataFrame consists of rows and columns so, in order to iterate over dataframe, we have to iterate a dataframe like a dictionary. In a dictionary, we iterate over the keys of the object in the same way we have to iterate in dataframe.

Q2. Difference between series and dataframe.

A Pandas Series is one dimensioned whereas a DataFrame is two dimensioned. Therefore, a single column DataFrame can have a name for its single column but a Series cannot have a column name. In fact, each column of a DataFrame can be converted to a series.

Series			Series			DataFrame	
	apples			oranges			apples oranges
0	3		0	0		0	3 0
1	2	+	1	3	=	1	2 3
2	0		2	7		2	0 7
3	1		3	2		3	1 2

Q3. What is the difference between series and list in python?

Creating a Pandas Series from Lists. A Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.). It has to be remembered that unlike Python lists, a Series will always contain data of the same type.

Q4. Name two dataStructure in Python Pandas.