

▼ MODULE 3

```
import pandas as pd
# make an array
array = [2, 4, 6, 8, 10, 12]
# create a series
series_obj = pd.Series(array)
# convert series object into array
arr = series_obj.values
arr

array([ 2,  4,  6,  8, 10, 12])
```

```
# reshaping series
reshaped_arr = arr.reshape((3, 2))
# show
reshaped_arr
```

```
↳ array([[ 2,  4],
        [ 6,  8],
        [10, 12]])
```

```
# import pandas library
import pandas as pd
# make an array
array = ["ankit", "shaurya",
"shivangi", "priya",
"jeet", "ananya"]
# create a series
series_obj = pd.Series(array)
print("Given Series:\n", series_obj)
# convert series object into array
arr = series_obj.values
arr
```

```
Given Series:
0    ankit
1  shaurya
2  shivangi
3    priya
4     jeet
5   ananya
dtype: object
array(['ankit', 'shaurya', 'shivangi', 'priya', 'jeet', 'ananya'],
      dtype=object)
```

```
# reshaping series
reshaped_arr = arr.reshape((2, 3))
# show
print("After Reshaping: \n", reshaped_arr)
```

```
After Reshaping:
[['ankit' 'shaurya' 'shivangi']
 ['priya' 'jeet' 'ananya']]
```

PIVOTING

```
import pandas as pd
# creating a dataframe
df = pd.DataFrame({'A': ['John', 'Boby', 'Mina'],
'B': ['Masters', 'Graduate', 'Graduate'],
'C': [27, 23, 21]})
df
```

```

      A      B      C
df.pivot('A', 'B', 'C')

```

<ipython-input-8-d3da59eff764>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.pivot will be keyword-only.
df.pivot('A', 'B', 'C')

B Graduate Masters

A

Boby	23.0	NaN
John	NaN	27.0
Mina	21.0	NaN

value is a list

```
df.pivot(index='A', columns='B', values=['C', 'A'])
```

	C		A	
B	Graduate	Masters	Graduate	Masters
A				
Boby	23	NaN	Boby	NaN
John	NaN	27	NaN	John
Mina	21	NaN	Mina	NaN

```

df = pd.DataFrame({'Name': ['John', 'Sammy', 'Stephan', 'Joe', 'Emily', 'Tom'],
'Gender': ['Male', 'Female', 'Male',
'Female', 'Female', 'Male'],
'Age': [45, 6, 4, 36, 12, 43]})
print("Dataset")
print(df)
print("-"*40)
# categorizing in age groups
def age_bucket(age):
    if age <= 18:
        return "<18"
    else:
        return ">18"
df['Age Group'] = df['Age'].apply(age_bucket)
# calculating gender percentage
gender = pd.DataFrame(df.Gender.value_counts(normalize=True)*100).reset_index()
gender.columns = ['Gender', '%Gender']
df = pd.merge(left=df, right=gender, how='inner', on=['Gender'])
# creating pivot table
table = pd.pivot_table(df, index=['Gender', '%Gender', 'Age Group'],
values=['Name'], aggfunc={'Name': 'count'},)
# display table
print("Table")
print(table)

```

Dataset

	Name	Gender	Age
0	John	Male	45
1	Sammy	Female	6
2	Stephan	Male	4
3	Joe	Female	36
4	Emily	Female	12
5	Tom	Male	43

Table

			Name
Gender	%Gender	Age Group	
Female	50.0	<18	2
		>18	1
Male	50.0	<18	1
		>18	2

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge. 