

10. Pandas – DataFrame – Rename column & index

Contents

| | |
|--|----|
| 1. rename(p) method - Changing single column name | 2 |
| 2. rename(p) method - Changing multiple column names | 5 |
| 3. columns attribute - Changing multiple column names..... | 8 |
| 4. rename(p) method - Changing index in DataFrame..... | 10 |
| 5. index attribute - Changing multiple column names..... | 12 |
| 6. Converting column names from lower case to upper case | 15 |

10. Pandas – DataFrame – Rename column & index

- ✓ Sometimes we need to process column names and index.
- ✓ Based on requirement we can rename/change/modify/update column names and index.

1. rename(p) method - Changing single column name

- ✓ rename(p) is a predefined method in DataFrame
- ✓ We can access rename() method by using DataFrame object.
- ✓ By using this method we can change single column name in DataFrame.

Program Name Creating DataFrame and checking column names
demo1.py
Input file sales3.csv

```
import pandas as pd

df = pd.read_csv("sales3.csv")
print(df.head())
print()
print(df.columns)
```

Output

```
ord id  cust name  cust id  prod name  prod cost
0  192837    Veeru      3      LG Mobile    65999
1  192838   Neelima    19  Apple iPad 10.2-inch  63000
2  192839   Balaji    12  34in Ultrawide Monitor  75999
3  192840   Shahid    20    iPhone 11    60000
4  192841    Vinay    10  Bose SoundSport Headphones  69999

Index(['ord id', 'cust name', 'cust id', 'prod name', 'prod cost'], dtype='object')
```

Note

- ✓ We need to use python dictionary to specify old column as key and new column as value.
- ✓ Whenever changing column then key should be match with column name in DataFrame otherwise changes will not reflects.

Program Renaming single column in DataFrame
Name demo2.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")
d = {
    "ord id": "Order Id"
}

df2 = df1.rename(columns = d)

print(df1.head())
print()
print(df2.head())
```

Output

```
ord id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999

Order Id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999
```

Note

- ✓ Whenever changing column then key should be match with column name in DataFrame otherwise changes will not reflects.

Program Renaming single column in DataFrame
Name demo3.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

d = {
    "orddddd id": "Order Id"
}

df2 = df1.rename(columns = d)

print(df1.head())
print()
print(df2.head())
```

Output

```
ord id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999

ord id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999
```

2. rename(p) method - Changing multiple column names

- ✓ rename(p) is a predefined method in DataFrame
- ✓ We can access rename() method by using DataFrame object.
- ✓ We can even change multiple column names by using rename(p) method.
- ✓ We need to use python dictionary to specify old column as key and new column as value.

Program Renaming multiple column in DataFrame
Name demo4.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

d = {
    'ord id': 'Order Id',
    'cust name': 'Customer Name',
    'cust id': 'Customer Id',
    'prod name': 'Product Name',
    'prod cost': 'Product Cost'
}

df2 = df1.rename(columns = d)

print(df1.head())
print()
print(df2.head())
```

Output

```
ord id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999

Order Id Customer Name Customer Id Product Name Product Cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999
```

Program Renaming multiple column in DataFrame
Name demo5.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

d = {
    "ord id": "order_id",
    "cust name": "customer_name",
    "cust id": "customer_id",
    "prod name": "product_name",
    "prod cost": "product_cost"
}

df2 = df1.rename(columns = d)

print(df1.head())
print()
print(df2.head())
```

Output

```
ord id cust name cust id prod name prod cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999

order_id customer_name customer_id product_name product_cost
0 192837 Veeru 3 LG Mobile 65999
1 192838 Neelima 19 Apple iPad 10.2-inch 63000
2 192839 Balaji 12 34in Ultrawide Monitor 75999
3 192840 Shahid 20 iPhone 11 60000
4 192841 Vinay 10 Bose SoundSport Headphones 69999
```

3. columns attribute - Changing multiple column names

- ✓ columns is predefined attribute in DataFrame class.
- ✓ We can access columns attribute by using DataFrame object.
- ✓ By using columns attribute we can even change multiple column names.

Program Renaming multiple column in DataFrame
Name demo6.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

print(df1.head())

df1.columns = [
    "order_id",
    "customer_name",
    "customer_id",
    "product_name",
    "product_cost"
]

print()
print(df1.head())
```

Output

```
ord id  cust name  cust id  prod name  prod cost
0  192837    Veeru      3      LG Mobile    65999
1  192838   Neelima     19  Apple iPad 10.2-inch  63000
2  192839   Balaji     12  34in Ultrawide Monitor  75999
3  192840   Shahid     20    iPhone 11    60000
4  192841    Vinay     10  Bose SoundSport Headphones  69999

order_id  customer_name  customer_id  product_name  product_cost
0    192837         Veeru           3      LG Mobile    65999
1    192838        Neelima          19  Apple iPad 10.2-inch  63000
2    192839        Balaji          12  34in Ultrawide Monitor  75999
3    192840        Shahid          20    iPhone 11    60000
4    192841         Vinay          10  Bose SoundSport Headphones  69999
```


Note

- ✓ While using columns attributes to change column names then Number of DataFrame column names should be match with existing columns otherwise we will get an error.

Program Renaming multiple column in DataFrame
Name demo7.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

print(df1.head())

df1.columns = [
    "order_id",
    "customer_name",
    "customer_id",
    "product_name",
    "product_cost",
    "total"
]

print()
print(df1.head())
```

Output

ValueError: Length mismatch: Expected axis has 5 elements, new values have 6 elements

4. rename(p) method - Changing index in DataFrame

- ✓ rename(p) is a predefined method in DataFrame
- ✓ We can access rename() method by using DataFrame object.
- ✓ Based on requirement we can change the index in DataFrame.
- ✓ By using rename(p) we can change the DataFrame index

Program Name Creating a sample Dataframe
demo8.py

```
import pandas as pd

d = {
    "order_id": [11, 21, 31],
    "customer_name": ["Prasad", "Daniel", "Jeswanth"],
    "product": ["iPhone", "hTC", "macbook"]
}

df1 = pd.DataFrame(d)
print(df1)
```

Output

| | order_id | customer_name | product |
|---|----------|---------------|-----------|
| 0 | 11 | Prasad | iPhone 11 |
| 1 | 21 | Daniel | hTC |
| 2 | 31 | Jeswanth | macbook |

Program Name Changing index in DataFrame
demo9.py

```
import pandas as pd

d = {
    "order_id": [11, 21, 31],
    "customer_name": ["Prasad", "Daniel", "Jeswanth"],
    "product": ["iPhone", "hTC", "macbook"]
}

i = {0: 77, 1: 88, 2: 99}

df1 = pd.DataFrame(d)
df2 = df1.rename(index = i)

print(df1)
print()
print(df2)
```

Output

```
   order_id customer_name product
0         11        Prasad  iPhone
1         21        Daniel    hTC
2         31     Jeswanth  macbook

   order_id customer_name product
77         11        Prasad  iPhone
88         21        Daniel    hTC
99         31     Jeswanth  macbook
```

5. index attribute - Changing multiple column names

- ✓ index is predefined attribute in DataFrame class.
- ✓ We can access index attribute by using DataFrame object.
- ✓ By using index attribute we can even change index in DataFrame

Program Name Changing index in DataFrame, using axis parameter
demo10.py

```
import pandas as pd

d = {
    "order_id": [11, 21, 31],
    "customer_name": ["Prasad", "Daniel", "Jeswanth"],
    "product": ["iPhone", "hTC", "macbook"]
}

df1 = pd.DataFrame(d)
print(df1)

df1.index = [77, 88, 99]

print()
print(df1)
```

Output

```
   order_id customer_name product
0         11        Prasad  iPhone
1         21        Daniel    hTC
2         31       Jeswanth  macbook

   order_id customer_name product
77         11        Prasad  iPhone
88         21        Daniel    hTC
99         31       Jeswanth  macbook
```

Program Changing index in DataFrame, using index attribute.
Name demo11.py
Input file sales31.csv

```
import pandas as pd

df1 = pd.read_csv("sales31.csv")
print(df1)

df1.index = range(10, 20)

print()
print(df1)
```

Output

| | ord id | cust name | cust id | prod name | prod cost |
|---|--------|------------|---------|----------------------------|-----------|
| 0 | 192837 | Veeru | 3 | LG Mobile | 65999 |
| 1 | 192838 | Neelima | 19 | Apple iPad 10.2-inch | 63000 |
| 2 | 192839 | Balaji | 12 | 34in Ultrawide Monitor | 75999 |
| 3 | 192840 | Shahid | 20 | iPhone 11 | 60000 |
| 4 | 192841 | Vinay | 10 | Bose SoundSport Headphones | 69999 |
| 5 | 192842 | Vinay | 10 | 20in Monitor | 65999 |
| 6 | 192843 | Balaji | 12 | iPhone 8 | 55999 |
| 7 | 192844 | Nireekshan | 1 | 27in 4K Gaming Monitor | 55999 |
| 8 | 192845 | Venu | 23 | iPhone 9 | 55999 |
| 9 | 192846 | Veeru | 3 | 20in Monitor | 63000 |

| | ord id | cust name | cust id | prod name | prod cost |
|----|--------|------------|---------|----------------------------|-----------|
| 10 | 192837 | Veeru | 3 | LG Mobile | 65999 |
| 11 | 192838 | Neelima | 19 | Apple iPad 10.2-inch | 63000 |
| 12 | 192839 | Balaji | 12 | 34in Ultrawide Monitor | 75999 |
| 13 | 192840 | Shahid | 20 | iPhone 11 | 60000 |
| 14 | 192841 | Vinay | 10 | Bose SoundSport Headphones | 69999 |
| 15 | 192842 | Vinay | 10 | 20in Monitor | 65999 |
| 16 | 192843 | Balaji | 12 | iPhone 8 | 55999 |
| 17 | 192844 | Nireekshan | 1 | 27in 4K Gaming Monitor | 55999 |
| 18 | 192845 | Venu | 23 | iPhone 9 | 55999 |
| 19 | 192846 | Veeru | 3 | 20in Monitor | 63000 |

Program Name Changing columns and index in a Dataframe
demo12.py

```
import pandas as pd

d = {
    "Ord Id": [11, 21, 31],
    "Customer Name": ["Prasad", "Daniel", "Jeswanth"],
    "Product": ["iPhone", "hTC", "macbook"]
}

df1 = pd.DataFrame(d)
print(df1)

df1.index = [333, 444, 555]
df1.columns = ["order_id", "customer_name", "product"]

print()
print(df1)
```

Output

```
   Ord Id Customer Name Product
0      11      Prasad  iPhone
1      21      Daniel   hTC
2      31    Jeswanth  macbook

   order_id customer_name product
333        11      Prasad  iPhone
444        21      Daniel   hTC
555        31    Jeswanth  macbook
```

6. Converting column names from lower case to upper case

- ✓ Based on requirement we can convert DataFrame column names from lower case to upper case.

Program Converting all columns names into upper case
Name demo13.py
Input file sales3.csv

```
import pandas as pd

df1 = pd.read_csv("sales3.csv")

print(df1.head())

df1.columns = df1.columns.str.upper()

print()
print(df1.head())
```

Output

```
ord id  cust name  cust id  prod name  prod cost
0  192837    Veeru      3      LG Mobile    65999
1  192838   Neelima    19  Apple iPad 10.2-inch  63000
2  192839   Balaji    12  34in Ultrawide Monitor  75999
3  192840   Shahid    20    iPhone 11    60000
4  192841    Vinay    10  Bose SoundSport Headphones  69999

ORD ID  CUST NAME  CUST ID  PROD NAME  PROD COST
0  192837    Veeru      3      LG Mobile    65999
1  192838   Neelima    19  Apple iPad 10.2-inch  63000
2  192839   Balaji    12  34in Ultrawide Monitor  75999
3  192840   Shahid    20    iPhone 11    60000
4  192841    Vinay    10  Bose SoundSport Headphones  69999
```