In [2]:
```
import pandas as pd
import numpy as np
```

In [39]:
```
#1
a = {'col1': [1, 4, 3, 4, 5], 'col2': [4, 5, 6, 7, 8], 'col3': [7, 8, 9, 0, 1]}
b = pd.DataFrame(data=a)
print("Original DataFrame")
print(b)
print('Rows for colum1 value == 4')
print(b.loc[b['col1'] == 4])
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     4     5     8
2     3     6     9
3     4     7     0
4     5     8     1
Rows for colum1 value == 4
   col1  col2  col3
1     4     5     8
3     4     7     0
```

In [40]:
```
#2
a = {'col1': [1, 4, 3, 4, 5], 'col2': [4, 5, 6, 7, 8], 'col3': [7, 8, 9, 0, 1]}
b = pd.DataFrame(data=a)
print("Original DataFrame")
print(b)
b = b[b.col2 != 5]
print("New DataFrame")
print(df)
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     4     5     8
2     3     6     9
3     4     7     0
4     5     8     1
New DataFrame
  company_code                   address num_less
0        c0001          72 Surrey Ave.11    72 11
1        c0002         92 N. Bishop Ave.       92
2        c0003      9910 Golden Star St.
3        c0003            102 Dunbar St.
4        c0004  17 West Livingston Court       17
```

In [15]:
```
#3
s = pd.Index([' A', 'B ', ' C ', 'D', ' E '])
print("Original series:")
print(s)
print("\nRemove whitespace")
print(s.str.strip())
print("\nRemove left sided whitespace")
print(s.str.lstrip())
print("\nRemove Right sided whitespace")
print(s.str.rstrip())
```

```
Original series:
Index([' A', 'B ', ' C ', 'D', ' E '], dtype='object')

Remove whitespace
Index(['A', 'B', 'C', 'D', 'E'], dtype='object')

Remove left sided whitespace
Index(['A', 'B ', 'C ', 'D', 'E '], dtype='object')

Remove Right sided whitespace
Index([' A', 'B', ' C', 'D', ' E'], dtype='object')
```

In [18]:
```
#4
df = pd.DataFrame({
    's0': ['abaabab','bbbbba','aaaaaaa', 'ababb', 'abababababbb'],
})
print("Original DataFrame:")
print(df)
print("\nCount occurrence of ab:")
df['count'] = list(map(lambda x: x.count("ab"), df['s0']))
print(df)
```

```
Original DataFrame:
             s
0      abaabab
1       bbbbba
2      aaaaaaa
3        ababb
4  abababababbb

Count occurrence of ab:
             s  count
0      abaabab      3
1       bbbbba      0
2      aaaaaaa      0
3        ababb      2
4  abababababbb      4
```

In [30]:
```
#5
df = pd.DataFrame({
    's1': ['123456','1 001', '2oj055', '123456789o', '123345']})
print("Original DataFrame:")
print(df)
print("\nNumeric values present in s1:")
df['verdict'] = list(map(lambda x: x.isdigit(), df['s1']))
print(df)
```

```
Original DataFrame:
            s
0      123456
1       1 001
2      2oj055
3  123456789o
4      123345

Numeric values present in s:
            s  verdict
0      123456     True
1       1 001    False
2      2oj055    False
3  123456789o    False
4      123345     True
```

In [34]:
```
#6
df = pd.DataFrame({
```

```
    's2': ['Abcd','EFGF', 'Hhhh', 'abcd', 'EAWQaaa']})
print("Original DataFrame:")
print(df)
print("\nIs proper case or title case?")
df['verdict1'] = list(map(lambda x: x.istitle(), df['s2']))
print(df)
```

```
Original DataFrame:
        s2
0     Abcd
1     EFGF
2     Hhhh
3     abcd
4  EAWQaaa

Is proper case or title case?
        s2  verdict1
0     Abcd      True
1     EFGF     False
2     Hhhh      True
3     abcd     False
4  EAWQaaa     False
```

In [33]:

```
#7
df = pd.DataFrame({
    's3': ['A','B', 'C', 'D', 'A']
})

print("Original DataFrame:")
print(df)

print("\nReplace A with c:")
df = df.replace("A", "C")
print(df)
```

```
Original DataFrame:
  s3
0  A
1  B
2  C
3  D
4  A

Replace A with c:
  s3
0  C
1  B
2  C
3  D
4  C
```

In [36]:

```
#8
import re as re
pd.set_option('display.max_columns', 10)
df = pd.DataFrame({
    'text_code': ['t0001.','t0002','t0003', 't0004'],
    'text_lang': ['She livedd a long life.', 'How oold is your father?', 'What is t
    })
print("Original DataFrame:")
print(df)
def rep_char(str1):
    tchr = str1.group(0)
    if len(tchr) > 1:
        return tchr[0:1] # can change the value here on repetition
def unique_char(rep, sent_text):
```

```
    convert = re.sub(r'(\w)\1+', rep, sent_text)
    return convert
df['normal_text']=df['text_lang'].apply(lambda x : unique_char(rep_char,x))
print("\nRemove repetitive characters:")
print(df)
```

```
Original DataFrame:
  text_code                    text_lang
0    t0001.      She livedd a long life.
1     t0002     How oold is your father?
2     t0003        What is tthe problem?
3     t0004  TThhis desk is used by Tom.

Remove repetitive characters:
  text_code                    text_lang                normal_text
0    t0001.      She livedd a long life.     She lived a long life.
1     t0002     How oold is your father?    How old is your father?
2     t0003        What is tthe problem?       What is the problem?
3     t0004  TThhis desk is used by Tom.  This desk is used by Tom.
```

In [37]:
```
#9
df = pd.DataFrame({
    'company_code': ['Abcd','EFGF', 'zefsalf', 'sdfslew', 'zekfsdf'],
    'date_of_sale': ['12/05/2002','16/02/1999','05/09/1998','12/02/2022','15/09/199
    'sale_amount': [12348.5, 233331.2, 22.5, 2566552.0, 23.0]
})
print("Original DataFrame:")
print(df)
def find_valid_dates(dt):
    #format: mm-dd-yyyy
    result = re.findall(r'\b(1[0-2]|0[1-9])/(3[01]|[12][0-9]|0[1-9])/([0-9]{4})\b',
    return result
df['valid_dates']=df['date_of_sale'].apply(lambda dt : find_valid_dates(dt))
print("\nValid dates (format: mm-dd-yyyy):")
print(df)
```

```
Original DataFrame:
  company_code date_of_sale  sale_amount
0         Abcd   12/05/2002      12348.5
1         EFGF   16/02/1999     233331.2
2      zefsalf   05/09/1998         22.5
3      sdfslew   12/02/2022    2566552.0
4      zekfsdf   15/09/1997         23.0

Valid dates (format: mm-dd-yyyy):
  company_code date_of_sale  sale_amount       valid_dates
0         Abcd   12/05/2002      12348.5  [(12, 05, 2002)]
1         EFGF   16/02/1999     233331.2                []
2      zefsalf   05/09/1998         22.5  [(05, 09, 1998)]
3      sdfslew   12/02/2022    2566552.0  [(12, 02, 2022)]
4      zekfsdf   15/09/1997         23.0                []
```

In [41]:
```
#10
pd.set_option('display.max_columns', 10)
dfc = pd.DataFrame({
    'company_code': ['c0001','c0002','c0003', 'c0003', 'c0004'],
    'address': ['72 Surrey Ave.11','92 N. Bishop Ave.','9910 Golden Star St.', '102
    })
print("Original DataFrame:")
print(dfc)

def test_num_less(n):
    nums = []
    for i in n.split():
        result = re.findall(r'\b(0*(?:[1-9][0-9]?|100))\b',i)
```

```
        nums.append(result)
        all_num=[",".join(x) for x in nums if x != []]
    return " ".join(all_num)

dfc['num_less'] = dfc['address'].apply(lambda x : test_num_less(x))
print("\nNumber less than 100:")
print(dfc)
```

```
Original DataFrame:
  company_code                   address
0        c0001          72 Surrey Ave.11
1        c0002         92 N. Bishop Ave.
2        c0003      9910 Golden Star St.
3        c0003            102 Dunbar St.
4        c0004  17 West Livingston Court

Number less than 100:
  company_code                   address num_less
0        c0001          72 Surrey Ave.11    72 11
1        c0002         92 N. Bishop Ave.       92
2        c0003      9910 Golden Star St.
3        c0003            102 Dunbar St.
4        c0004  17 West Livingston Court       17
```

In [ ]: