Extracting information in Python(using split())

In [1]:
```
s="820 sandcanyon irvine CA,september 12th 2025,10-30-15"
```

In [3]:
```
seperate=s.split(',')
```

In [11]:
```
type(seperate)
```

Out[11]:
```
list
```

In [7]:
```
print(seperate)
len(seperate)
```

```
['820 sandcanyon irvine CA', 'september 12th 2025', '10-30-15']
```

Out[7]:
```
3
```

In [33]:
```
sep=seperate[0].split()+seperate[1].split()+seperate[2].split("-")
```

In [35]:
```
sep
```

Out[35]:
```
['820',
 'sandcanyon',
 'irvine',
 'CA',
 'september',
 '12th',
 '2025',
 '10',
 '30',
 '15']
```

In [133…
```
print("**************Detailed information**********************")
print(f" apartment no: {sep1[0]}")
print(f" street name: {sep1[1]}")
print(f" city: {sep1[2]}")
print(f" State :{sep1[3]}")
print(f" Month: {sep1[4]}")
print(f" day :{sep1[5]}")
print(f" year :{sep1[6]}")
print(f" hours :{sep1[7]}")
print(f" Minutes :{sep1[8]}")
print(f" seconds :{sep1[8]}")
print("pgm execution completed")
```

```
**************Detailed information*********************
 apartment no: 820
 street name: sandcanyon
 city: irvine
 State :CA
 Month: september
 day :12th
 year :2025
 hours :10
 Minutes :30
 seconds :30
pgm execution completed
```

Table format information using-------{:<}formatting instructions and end=" "----------in PYTHON

In [139…
```
s="1000 Sandcanyon irvine CA,september 12th 2025,10-30-15"
st=s.split(',')
```

In [141…
```
st
```

Out[141…
```
['1000 Sandcanyon irvine CA', 'september 12th 2025', '10-30-15']
```

In [143…
```
st=st[0].split()+st[1].split()+st[2].split('-')
st
```

Out[143…
```
['1000',
 'Sandcanyon',
 'irvine',
 'CA',
 'september',
 '12th',
 '2025',
 '10',
 '30',
 '15']
```

In [57]:
```
type(st)
```

Out[57]:
```
list
```

....in f-strings ------f"{labels[0]:<13} ....in format()-------fm={:<Width}".format('YourText')

```
    : (Colon)-- Introduces the formatting instructions in both f-
strings and the format() method.
    It's placed within the curly braces {} to indicate the start of
formatting details
```

In [163…
```
# Print header
labels=["Aptno", "StName","city","State","Month","dayNo","yearNO","hours","Minute",
print(f"{labels[0]:<13} {labels[1]:<15} {labels[2]:<10} {labels[3]:<14} {labels[4]:
```

```
for i in st:
    print(f"{i:<14}",end='')
```

```
Aptno         StName          city       State          Month         dayNo         ye
arNO       hours        Minute       seconds
1000          Sandcanyon    irvine        CA            september     12th
2025          10            30            15
```

Extracting information From 2D DataSet in Pandas(using str.split())

str.split(',', expand=True):

Splits the Address column into multiple parts based on the comma , delimiter.

The expand=True option ensures that the result is returned as separate columns (a DataFrame).

In [175…
```
import pandas as pd
```

In [187…
```
data = {
    "Name": ["John Doe", "Jane Smith"],
    "Address": [
        "1010,Sunset Blvd, Los Angeles, CA, 90028",
        "2020 Main St, Houston, TX, 77002"
    ],
    "Date": ["03/20/2025", "03/21/2025"],
    "Time": ["10:15:40", "12:45:00"]
}
```

In [189…
```
df=pd.DataFrame(data)
df
```

Out[189…

	Name	Address	Date	Time
0	John Doe	1010,Sunset Blvd, Los Angeles, CA, 90028	03/20/2025	10:15:40
1	Jane Smith	2020 Main St, Houston, TX, 77002	03/21/2025	12:45:00

In [191…
```
#splitting the address

df[['Aptno','StreetName','city','State','Zipcode']]=df["Address"].str.split(',', ex
```

In [195…
```
#splitting the Date

df[['Month','Day','Year']]=df['Date'].str.split('/',expand=True)
```

In [197…
```
df
```

Out[197…

	Name	Address	Date	Time	Aptno	StreetName	city	State	Zipcode
0	John Doe	1010,Sunset Blvd, Los Angeles, CA, 90028	03/20/2025	10:15:40	1010	Sunset Blvd	Los Angeles	CA	90028
1	Jane Smith	2020 Main St, Houston, TX, 77002	03/21/2025	12:45:00	2020 Main St	Houston	TX	77002	None

In [199…

```
df[['Hour','Minute','Sec']]=df['Time'].str.split(':',expand=True)
```

In [201…

```
df
```

Out[201…

	Name	Address	Date	Time	Aptno	StreetName	city	State	Zipcode
0	John Doe	1010,Sunset Blvd, Los Angeles, CA, 90028	03/20/2025	10:15:40	1010	Sunset Blvd	Los Angeles	CA	90028
1	Jane Smith	2020 Main St, Houston, TX, 77002	03/21/2025	12:45:00	2020 Main St	Houston	TX	77002	None

--inplace=True: the operation is performed directly on the original DataFrame, and the DataFrame is updated in place.

--If inplace=False (or omitted), the operation returns a new DataFrame with the columns dropped, leaving the original DataFrame unchanged.

In [203…

```
df.drop(columns=['Address','Date','Time'],inplace=True)
```

In [205…

```
df
```

Out[205…

	Name	Aptno	StreetName	city	State	Zipcode	Month	Day	Year	Hour	Minute
0	John Doe	1010	Sunset Blvd	Los Angeles	CA	90028	03	20	2025	10	15
1	Jane Smith	2020 Main St	Houston	TX	77002	None	03	21	2025	12	45

In [ ]:

In [ ]:

In [ ]: