```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
```

Load data from multiple sheets of an Excel file

```
In [3]: list_SheetName = ['Oct-2024','Nov-2024','Dec-2024','Jan-2025','Feb-2025']
        da_df_merge = pd.read_excel(r'D:\Python_Code\NIT_Project\DA_Classes.xlsx', sheet_name=list_SheetName)
        da_df = pd.concat(da_df_merge, axis=0, ignore_index=True)
```

```
In [4]: da_df
```

```
Out[4]:
       Sno        Date  Day  Type
  0      1  2024-10-16  Wed     W
  1      2  2024-10-17  Thu     W
  2      3  2024-10-18  Fri     W
  3      4  2024-10-19  Sat    WO
  4      5  2024-10-20  Sun    WO
...    ...         ...  ...   ...
118     11  2025-02-11  Tue    SL
119     12  2025-02-12  Wed     W
120     13  2025-02-13  Thu     W
121     14  2025-02-14  Fri     W
122     15  2025-02-15  Sat     W

123 rows × 4 columns
```

```
In [5]: type(da_df)
```

```
Out[5]: pandas.core.frame.DataFrame
```

```
In [6]: da_df.shape
```

```
Out[6]: (123, 4)
```

```
In [7]: da_df.isnull().sum()
```

```
Out[7]: Sno     0
        Date    0
        Day     0
        Type    0
        dtype: int64
```

```
In [8]: da_df.dtypes
```

```
Out[8]: Sno              int64
        Date    datetime64[ns]
        Day             object
        Type            object
        dtype: object
```

Load the data from a specific sheet in an Excel file

```
In [10]: DayType_DF = pd.read_excel(r'D:\Python_Code\NIT_Project\DA_Classes.xlsx', sheet_name='Day_Type')
         DayType_DF
```

Out[10]:

	Day_Type	Day_Type_Description
0	W	Working
1	WO	Weekly Off
2	PH	Public Holiday
3	CT	Class Test
4	SL	Sick Leave

Merge the data from 2 different dataframe

Simliar to join in Sql (where we specify table names and based on what columns to be joined and type (inner, left,right,outer)

In [12]:

```
daMerge_dF = pd.merge(da_df, DayType_DF, on=None ,left_on='Type', right_on='Day_Type',how='inner')
FinalDA_DF = daMerge_dF[['Sno', 'Date', 'Day','Day_Type','Day_Type_Description']]
FinalDA_DF
```

Out[12]:

	Sno	Date	Day	Day_Type	Day_Type_Description
0	1	2024-10-16	Wed	W	Working
1	2	2024-10-17	Thu	W	Working
2	3	2024-10-18	Fri	W	Working
3	4	2024-10-19	Sat	WO	Weekly Off
4	5	2024-10-20	Sun	WO	Weekly Off
...	...	...	...	...	...
118	11	2025-02-11	Tue	SL	Sick Leave
119	12	2025-02-12	Wed	W	Working
120	13	2025-02-13	Thu	W	Working
121	14	2025-02-14	Fri	W	Working
122	15	2025-02-15	Sat	W	Working

123 rows × 5 columns

In [13]:

```
FinalDA_DF.head(10)
```

Out[13]:

	Sno	Date	Day	Day_Type	Day_Type_Description
0	1	2024-10-16	Wed	W	Working
1	2	2024-10-17	Thu	W	Working
2	3	2024-10-18	Fri	W	Working
3	4	2024-10-19	Sat	WO	Weekly Off
4	5	2024-10-20	Sun	WO	Weekly Off
5	6	2024-10-21	Mon	W	Working
6	7	2024-10-22	Tue	W	Working
7	8	2024-10-23	Wed	W	Working
8	9	2024-10-24	Thu	W	Working
9	10	2024-10-25	Fri	W	Working

In [14]:

```
FinalDA_DF.tail(10)
```

Out[14]:

	Sno	Date	Day	Day_Type	Day_Type_Description
113	6	2025-02-06	Thu	W	Working
114	7	2025-02-07	Fri	W	Working
115	8	2025-02-08	Sat	WO	Weekly Off
116	9	2025-02-09	Sun	WO	Weekly Off
117	10	2025-02-10	Mon	SL	Sick Leave
118	11	2025-02-11	Tue	SL	Sick Leave
119	12	2025-02-12	Wed	W	Working
120	13	2025-02-13	Thu	W	Working
121	14	2025-02-14	Fri	W	Working
122	15	2025-02-15	Sat	W	Working

In [15]:

```
FinalDA_DF['Month_Name'] = FinalDA_DF['Date'].dt.strftime('%B')
```

In [16]:

```
FinalDA_DF['Year'] = FinalDA_DF['Date'].dt.year
```

In [17]:

```
FinalDA_DF
```

Out[17]:

	Sno	Date	Day	Day_Type	Day_Type_Description	Month_Name	Year
0	1	2024-10-16	Wed	W	Working	October	2024
1	2	2024-10-17	Thu	W	Working	October	2024
2	3	2024-10-18	Fri	W	Working	October	2024
3	4	2024-10-19	Sat	WO	Weekly Off	October	2024
4	5	2024-10-20	Sun	WO	Weekly Off	October	2024
...	...	...	...	...	...	...	...
118	11	2025-02-11	Tue	SL	Sick Leave	February	2025
119	12	2025-02-12	Wed	W	Working	February	2025
120	13	2025-02-13	Thu	W	Working	February	2025
121	14	2025-02-14	Fri	W	Working	February	2025
122	15	2025-02-15	Sat	W	Working	February	2025

123 rows × 7 columns

In [18]:
```
FinalDA_DF.shape
```

Out[18]:
```
(123, 7)
```

In [19]:
```
FinalDA_DF.head(10)
```

Out[19]:

	Sno	Date	Day	Day_Type	Day_Type_Description	Month_Name	Year
0	1	2024-10-16	Wed	W	Working	October	2024
1	2	2024-10-17	Thu	W	Working	October	2024
2	3	2024-10-18	Fri	W	Working	October	2024
3	4	2024-10-19	Sat	WO	Weekly Off	October	2024
4	5	2024-10-20	Sun	WO	Weekly Off	October	2024
5	6	2024-10-21	Mon	W	Working	October	2024
6	7	2024-10-22	Tue	W	Working	October	2024
7	8	2024-10-23	Wed	W	Working	October	2024
8	9	2024-10-24	Thu	W	Working	October	2024
9	10	2024-10-25	Fri	W	Working	October	2024

In [20]:

```
FinalDA_DF.tail(10)
```

Out[20]:

	Sno	Date	Day	Day_Type	Day_Type_Description	Month_Name	Year
113	6	2025-02-06	Thu	W	Working	February	2025
114	7	2025-02-07	Fri	W	Working	February	2025
115	8	2025-02-08	Sat	WO	Weekly Off	February	2025
116	9	2025-02-09	Sun	WO	Weekly Off	February	2025
117	10	2025-02-10	Mon	SL	Sick Leave	February	2025
118	11	2025-02-11	Tue	SL	Sick Leave	February	2025
119	12	2025-02-12	Wed	W	Working	February	2025
120	13	2025-02-13	Thu	W	Working	February	2025
121	14	2025-02-14	Fri	W	Working	February	2025
122	15	2025-02-15	Sat	W	Working	February	2025

Get the Count of the total rows

In [22]:

```
FinalDA_DF.shape[0]
```

Out[22]:

```
123
```

Get the breakdown of day types for the entire class duration.

In [24]:

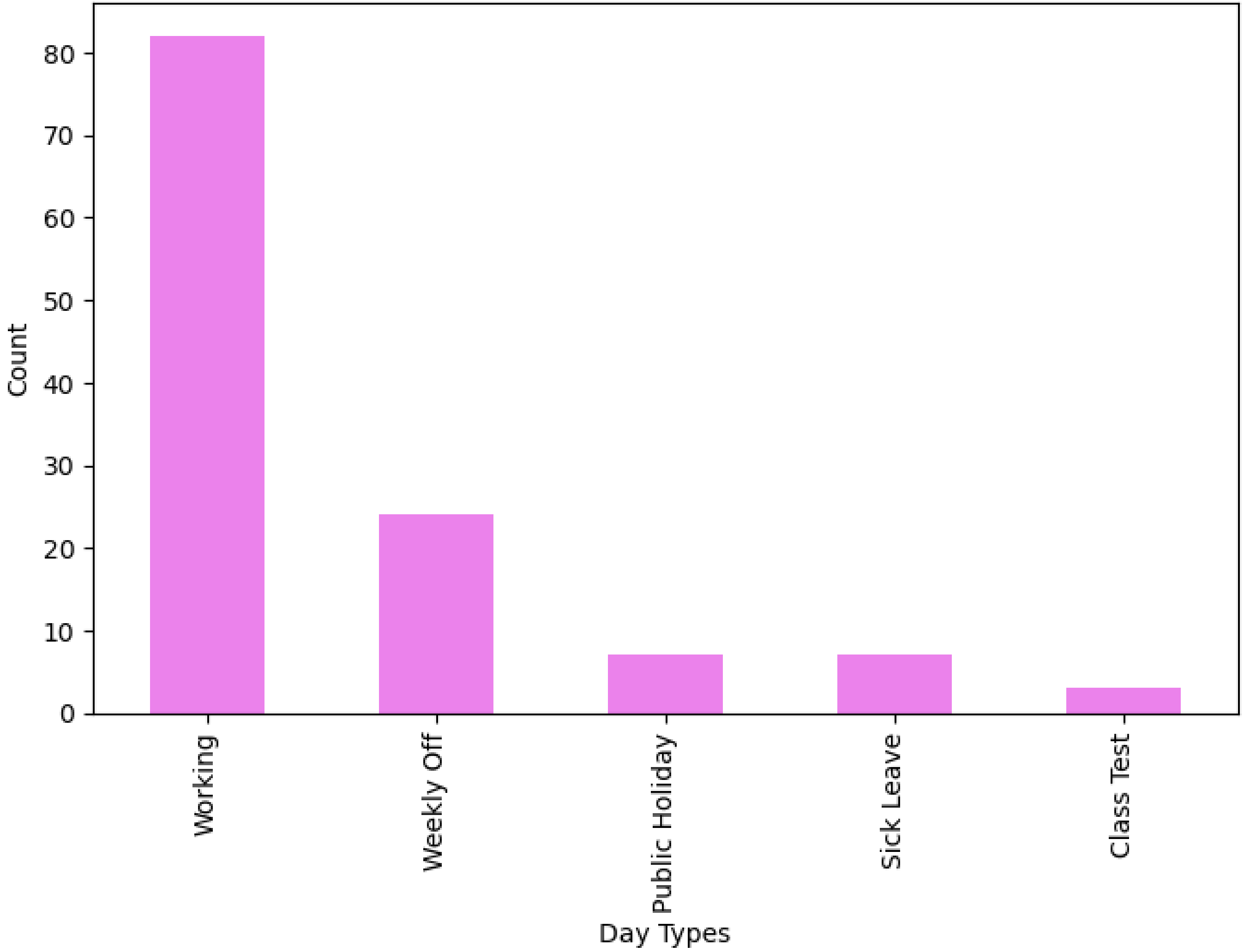
```
FinalDA_DF['Day_Type_Description'].value_counts()
```

Out[24]:

```
Day_Type_Description
Working           82
Weekly Off        24
Public Holiday     7
Sick Leave         7
Class Test         3
Name: count, dtype: int64
```

Get the breakdown of day types for the entire class duration.

In [26]:
```
plt.figure(figsize=(8,5))
FinalDA_DF['Day_Type_Description'].value_counts().plot(kind='bar',color='violet')
plt.xlabel("Day Types")
plt.ylabel('Count')
plt.show()
```



Get the breakdown of Day of week for the entire class duration.

In [28]:

```
FinalDA_DF['Day'].value_counts()
```

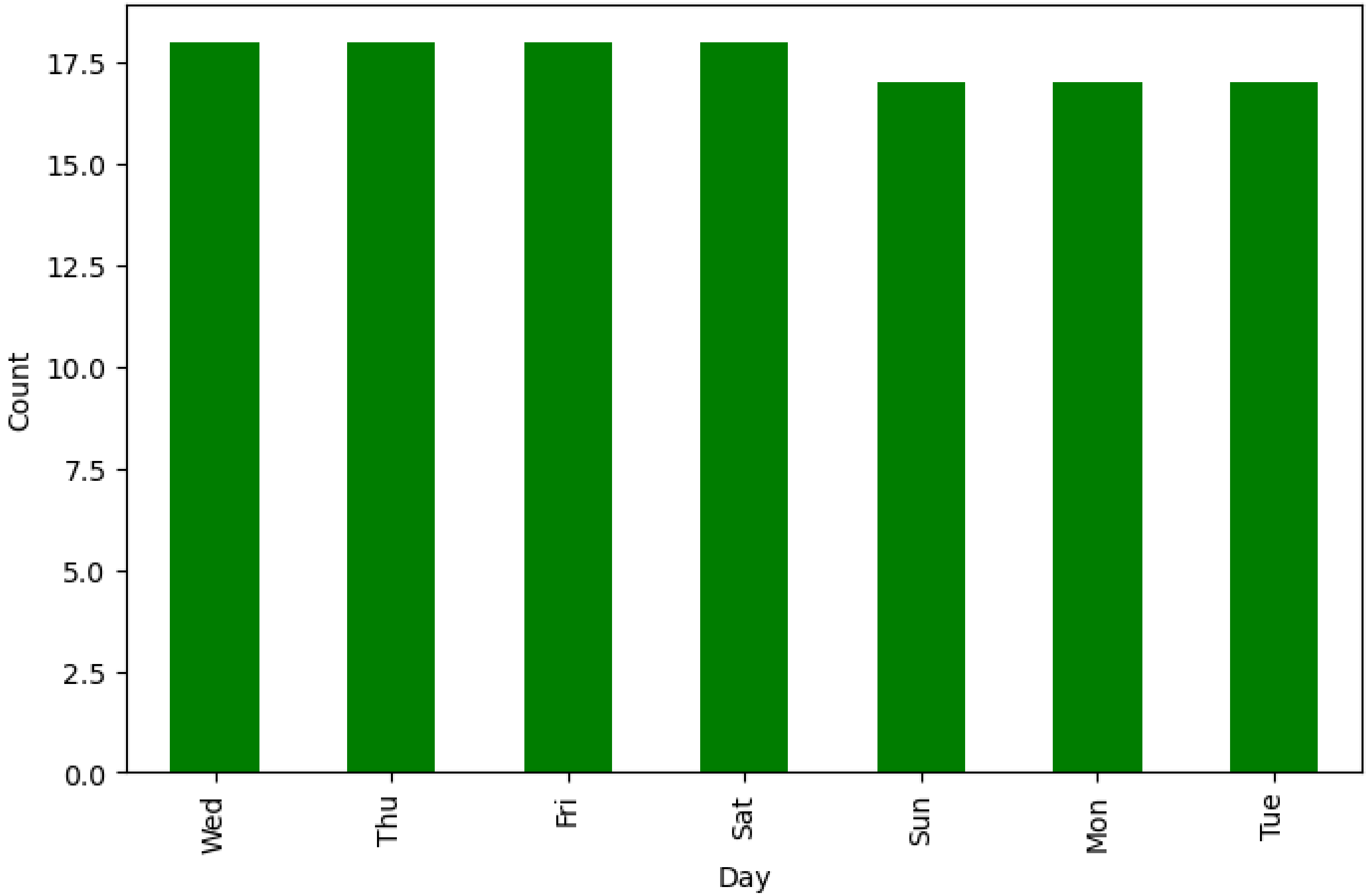
Out[28]:

```
Day
Wed    18
Thu    18
Fri    18
Sat    18
Sun    17
Mon    17
Tue    17
Name: count, dtype: int64
```

Plot the same in the form of graph

In [30]:

```
plt.figure(figsize=(8,5))
FinalDA_DF['Day'].value_counts().plot(kind='bar',color='green')
plt.xlabel("Day")
plt.ylabel('Count')
plt.show()
```



```
In [ ]:
```