Welcome to the Airbnb Mini Practice Project

We've provided a file called `airbnb_2.csv` that you'll need to import.

Let's do this first before we start our analysis.

Don't forget to import the libraries you need to read .csv files!

Step 1: Import Libraries

Import the pandas library below.

Put your code in the box below

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

Step 2: Ingest the Airbnb CSV file into your Jupyter Notebook

Now that you have the Pandas Libraries imported, it's time to import the airbnb dataset.

i) Please ingest the airbnb dataset using the `.read_csv()` syntax.

ii) Upon completion of this, use .info() to better understand the variables inside your dataset.

Put your code in the box below

In [2]:
```
airbnb_df = pd.read_csv("airbnb_2.csv")
airbnb_df.head()
airbnb_df.describe()
```

Out[2]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	avail
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.000000	48
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.143982	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.952519	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.000000	

Step 3: Exploring your data with Pandas

The rest of these questions will have you focus on using the following Pandas Skills:

- Subsetting a Pandas dataframe using [] and boolean operators
- Summing up Records with value_counts()
- Creating calculated fields
- Group By in Pandas
- Creating Bar Plots with Matplotlib

i) Please count how many airbnb listings are in each of the 5 Neighbourhood Groups (Manhattan, Brooklyn, Queens, Bronx, Staten Island) and identify which Neighbourhood Groups has the largest number of Airbnb Listings

Hint: Think about how you might use the `.value_counts()` methodology!

airbnb_df.head()

In [3]:
```
airbnb_df["neighbourhood_group"].value_counts()
```

Out[3]:
```
neighbourhood_group
Manhattan        21661
Brooklyn         20104
Queens            5666
Bronx             1091
Staten Island      373
Name: count, dtype: int64
```

We want to focus our attention on the Neighbourhood Groups that have the top 3 number of Airbnb Listings.

ii) Calculate the % listings that each Neighbourhood Group contains.

Hint: Take a look at the examples shown here!

Put your code in the box below

In [4]:
```
airbnb_df["neighbourhood_group"].value_counts(normalize=True)
```

Out[4]:
```
neighbourhood_group
Manhattan        0.443011
Brooklyn         0.411167
Queens           0.115881
Bronx            0.022313
Staten Island    0.007629
Name: proportion, dtype: float64
```

iii) Create a new calculated field called Revenue and place this into the Airbnb Dataframe. This is to be calculated by using the Price Column x Number_Of_Reviews Columns

Put your code in the box below

In [5]:
```
airbnb_df["revenue"] = airbnb_df["price"] * airbnb_df["number_of_reviews"]
airbnb_df.head()
```

Out[5]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_rev
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	19/10/2
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	21/05/2
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	N
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	5/07/2
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	19/11/2

iv) Create a Bar Plot that shows which Neighbourhood Group has the highest average revenues. In order to best calculate this, you'd want to consider how you can use the .groupby() syntax to assist you!

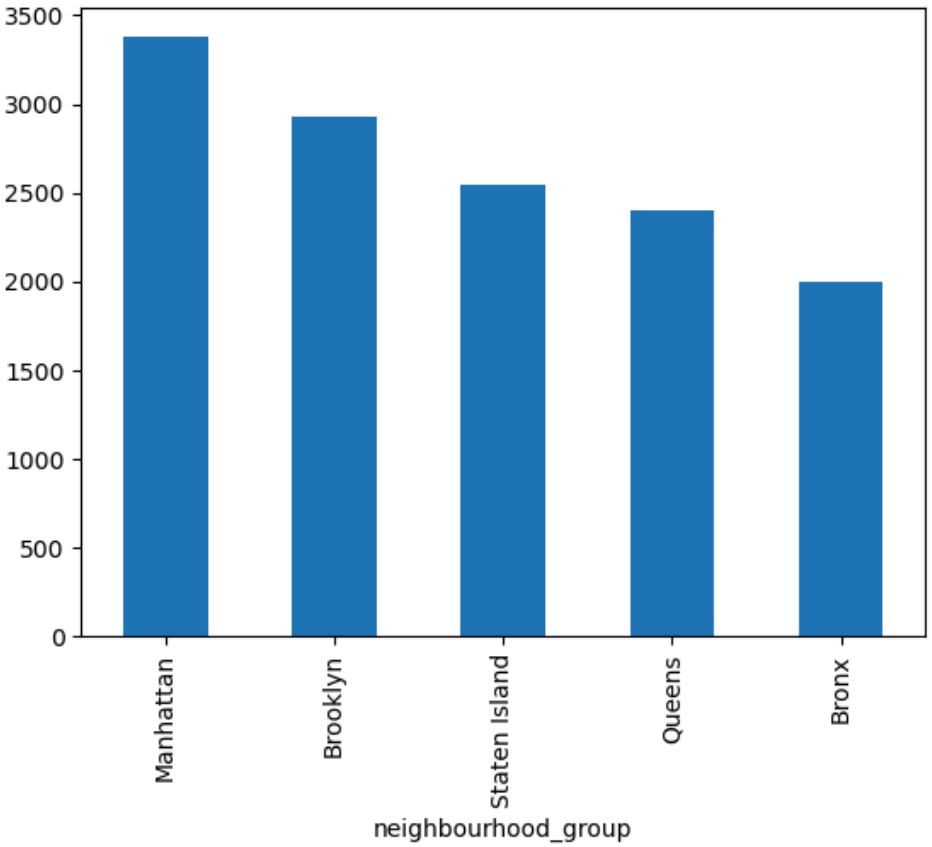
Hint: If you're stuck, we recommend you go back to this datacamp link. Specifically Chapter 4 which covers how GROUP BY is used in Pandas.

Remember, the syntax for GROUP BY is below:

```
dataframe.groupby(['SomeColumn']).someAggregation()
```

Put your code in the box below

In [6]:
```
airbnb_df.groupby(["neighbourhood_group"])['revenue'].mean().sort_values(ascending=False).plot(kind='bar')
plt.show()
```



Challenge Questions

V) Filter the Airbnb Dataframe to include only the Neighbourhood Groups `Manhattan` , `Brooklyn` and `Queens` .

Upon completion of this, identify the `top 3 Revenue Generating Neighborhoods` within each of the `three Neighbourhood_Groups` . This should give us 9 Overall Rows: 3 of the top generating neighbourhoods within each of the 3 Neighbourhood_Groups

This is a tricky question that will *test* your group-by skills.

We recommend you consider breaking down the query into a number of steps.

```
condition1 = someDataFrame['someColumn']=='someCondition'
condition2 = someDataFrame['someColumn']=='someCondition'
```

Step One - Filter the Dataframe using the Conditions

```
filtered_dataframe = someDataFrame[condition1 OR condition 2]
```

You can also make use of the `.isin()` syntax to help filter on multiple conditions in a cleaner manner!

```
dataframe['SomeColumn'].isin(['A','B','C'])
```

Step Two - Group the Data by Neighbourhood_Group and Neighbourhood.

Remember the dataframe syntax for grouping by is:

`dataframe.groupby(['SomeColumn']).someAggregation()`

Once you've now grouped your results - how can you ensure you only return the top 3 for each neighbourhood group?

This is where you'll need to make use of the following functions: `dataframe.reset_index()` `dataframe.groupby()` `dataframe.head()`

You will want to make use of the .reset_index(inplace=True) function to help reset the indexes in your Grouped Up Dataframe...!

Put your code in the box below

In [7]:
```
neighbourhood_group_list = ['Manhattan','Brooklyn','Queens']

top9 = airbnb_df[airbnb_df['neighbourhood_group'].isin(neighbourhood_group_list)]\
.groupby(['neighbourhood_group','neighbourhood'])['revenue'].sum().sort_values(ascending=False).reset_index()\
.groupby('neighbourhood_group').head(3)

top9
```

Out[7]:

	neighbourhood_group	neighbourhood	revenue
0	Brooklyn	Williamsburg	12389011
1	Brooklyn	Bedford-Stuyvesant	12352457
2	Manhattan	Harlem	8598692
3	Manhattan	Hell's Kitchen	8238991
4	Manhattan	East Village	7574535
8	Brooklyn	Bushwick	4762224
17	Queens	Astoria	1880840
28	Queens	Long Island City	1374945
33	Queens	Flushing	1140450

In [8]:
```
airbnb_df["revenue"] = airbnb_df["price"] * airbnb_df["number_of_reviews"]

filtered_df = airbnb_df[airbnb_df['neighbourhood_group'].isin(['Manhattan', 'Brooklyn', 'Queens'])]
```

```
grouped_revenue = filtered_df.groupby(['neighbourhood_group', 'neighbourhood'])['revenue'].sum().reset_index()

top_3 = grouped_revenue.sort_values(['neighbourhood_group', 'revenue'], ascending=[True, False])
top_3 = top_3.groupby('neighbourhood_group').head(3).reset_index(drop=True)

print(top_3)
```

```
  neighbourhood_group       neighbourhood   revenue
0            Brooklyn        Williamsburg  12389011
1            Brooklyn  Bedford-Stuyvesant  12352457
2            Brooklyn           Bushwick   4762224
3           Manhattan             Harlem   8598692
4           Manhattan      Hell's Kitchen   8238991
5           Manhattan        East Village   7574535
6              Queens             Astoria   1880840
7              Queens   Long Island City    1374945
8              Queens            Flushing   1140450
```

VI) Building on the previous question where you identified the top 3 Neighbourhoods within each of the three neighbourhood_groups based off Revenues, please filter the Airbnb Dataframe to include only these neighbourhoods.

Upon completion of this, identify the `top average revenue generating room type` for each of the nine neighbourhoods and plot this out in a Bar Chart.

Step One. Think carefully regarding how you can make use of the list of 9 neighbourhoods you've previously analyzed.

Step Two. Filter the original `airbnb` dataframe you created, to include only these top 9 neighbourhoods.

Step Three: Apply your standard aggregation syntax you've previously learned when using the .groupby() function

Step Four. Just as you previously made use of `.head()` and `.reset_index()` to get the top neighbourhoods - how might you use a similar approach to get the top `room_type` for each `neighbourhood` ?

Step Five. Create a bar plot from your dataframe using the `matplotlib` plotting library syntax.

We've included an example of the syntax below:

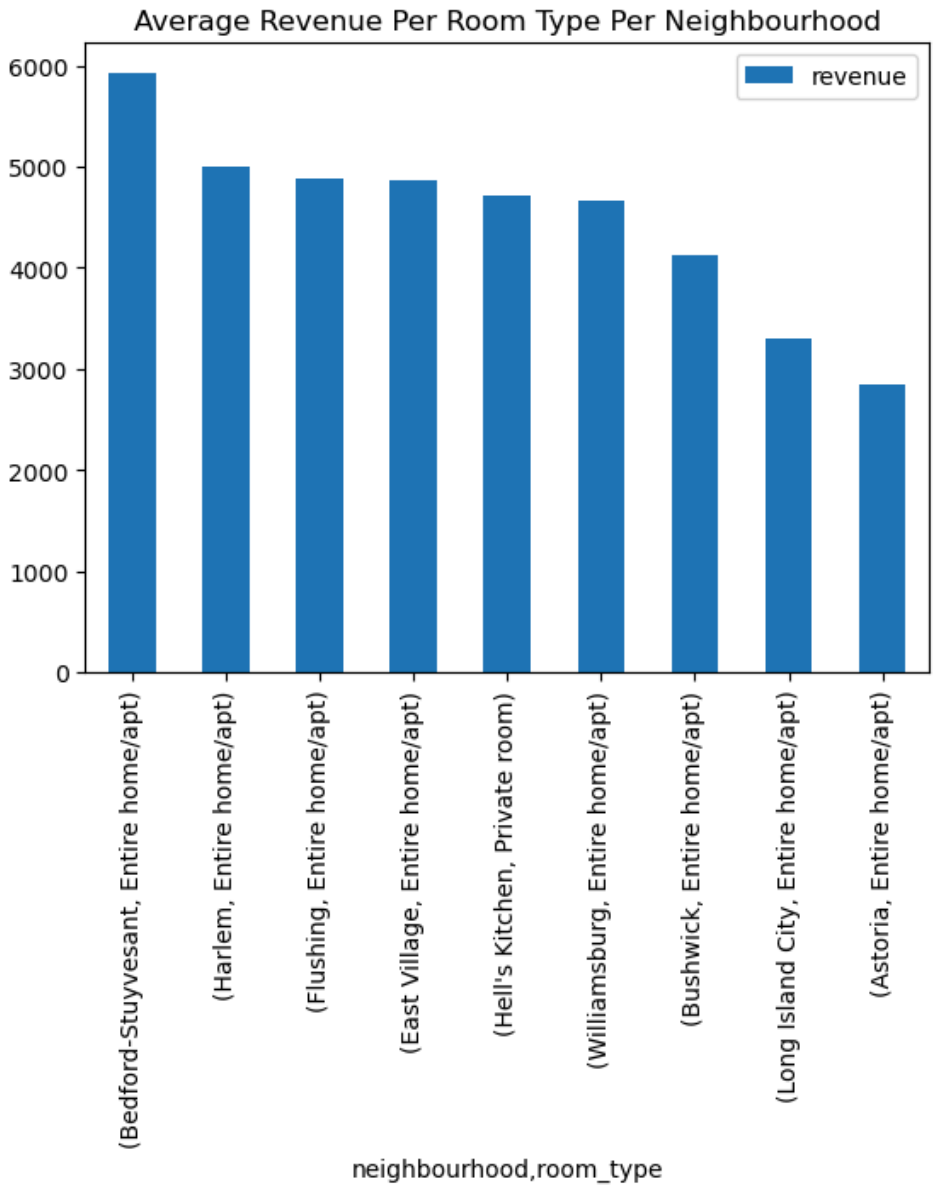
`plt.bar(x=dataframe['x-axis'], height=dataframe['y-axis']`

This is a tricky question that will *test* your group-by skills. Think back to the previous question and how you approached this; you can approach this in a similar manner.

Put your code in the box below

In [9]:
```
airbnb_df[airbnb_df['neighbourhood'].isin(top9['neighbourhood'])]\
.groupby(['neighbourhood','room_type'])['revenue'].mean().sort_values(ascending=False).reset_index()\
.groupby(['neighbourhood']).head(1).set_index(['neighbourhood','room_type'])\
.plot(kind='bar')

plt.title("Average Revenue Per Room Type Per Neighbourhood")
plt.show()
```



In [10]:
```
filtered_df2 = airbnb_df[airbnb_df['neighbourhood'].isin(
    ['Williamsburg','Bedford-Stuyvesant','Bushwick','Harlem',"Hell's Kitchen",'East Village','Astoria','Long Island City', 'Flushing'])]
grouped_rev2 = filtered_df2.groupby(['neighbourhood','room_type'])['revenue'].sum().reset_index()
top_rt = grouped_rev2.sort_values(['neighbourhood', 'revenue'], ascending=[True, False])
top_rt = top_rt.groupby('neighbourhood').head(1).reset_index(drop=True)

top_rt_sorted = top_rt.sort_values(by='revenue', ascending=False)

bars = plt.bar(x=top_rt_sorted['neighbourhood'], height=top_rt_sorted['revenue'], color='b')

custom_labels = [f"{neigh} ({room})" for neigh, room in zip(top_rt_sorted['neighbourhood'], top_rt_sorted['room_type'])]

plt.xticks(ticks=range(len(top_rt['neighbourhood'])), labels=custom_labels, rotation=90)
```

```
plt.xlabel('Neighbourhood')
plt.ylabel('Avg Revenue')
plt.title('Popular Airbnb Neighbourhoods by Room Type')

plt.show()
```