In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df=pd.read_csv("laptop.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	laptop	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	
0	1	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	In Gr
1	2	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	In Gr
2	3	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	In Gr
3	4	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	R P
4	5	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	In Gr

In [4]:
```
df.shape
```

Out[4]:
```
(1303, 13)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   laptop            1303 non-null   int64  
 1   Company           1303 non-null   object 
 2   Product           1303 non-null   object 
 3   TypeName          1303 non-null   object 
 4   Inches            1303 non-null   float64
 5   ScreenResolution  1303 non-null   object 
 6   Cpu               1303 non-null   object 
 7   Ram               1303 non-null   object 
 8   Memory            1303 non-null   object 
 9   Gpu               1303 non-null   object 
 10  OpSys             1303 non-null   object 
 11  Weight            1303 non-null   object 
 12  Price             1303 non-null   float64
dtypes: float64(2), int64(1), object(10)
memory usage: 132.5+ KB
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

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

Out[7]:
```
laptop              0
Company             0
Product             0
TypeName            0
Inches              0
ScreenResolution    0
Cpu                 0
Ram                 0
Memory              0
Gpu                 0
OpSys               0
Weight              0
Price               0
dtype: int64
```

In [8]:
```
df.drop(columns=['laptop'],inplace=True)
```

In [9]:
```
df['Ram']=df['Ram'].str.replace('GB','')
```

In [10]:
```
df['Weight']=df['Weight'].str.replace('kg','')
```

In [11]:
```
df.head()
```

Out[11]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	C
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	m
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	m
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	N
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	m
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	m

In [12]:
```
df['Ram']=df['Ram'].astype('int32')
df['Weiight']=df['Weight'].astype('float32')
```

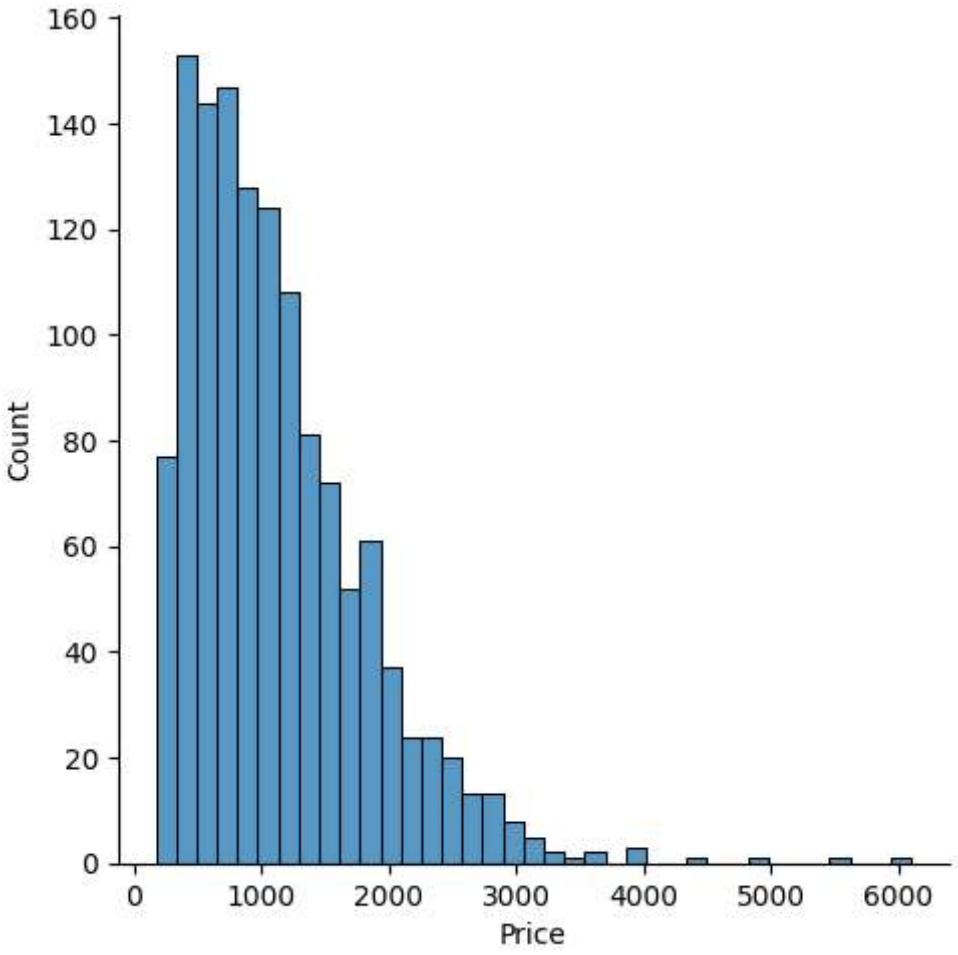
In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Company           1303 non-null   object 
 1   Product           1303 non-null   object 
 2   TypeName          1303 non-null   object 
 3   Inches            1303 non-null   float64
 4   ScreenResolution  1303 non-null   object 
 5   Cpu               1303 non-null   object 
 6   Ram               1303 non-null   int32  
 7   Memory            1303 non-null   object 
 8   Gpu               1303 non-null   object 
 9   OpSys             1303 non-null   object 
 10  Weight            1303 non-null   object 
 11  Price             1303 non-null   float64
 12  Weiight           1303 non-null   float32
dtypes: float32(1), float64(2), int32(1), object(9)
memory usage: 122.3+ KB
```

In [14]:
```
import seaborn as sns
```

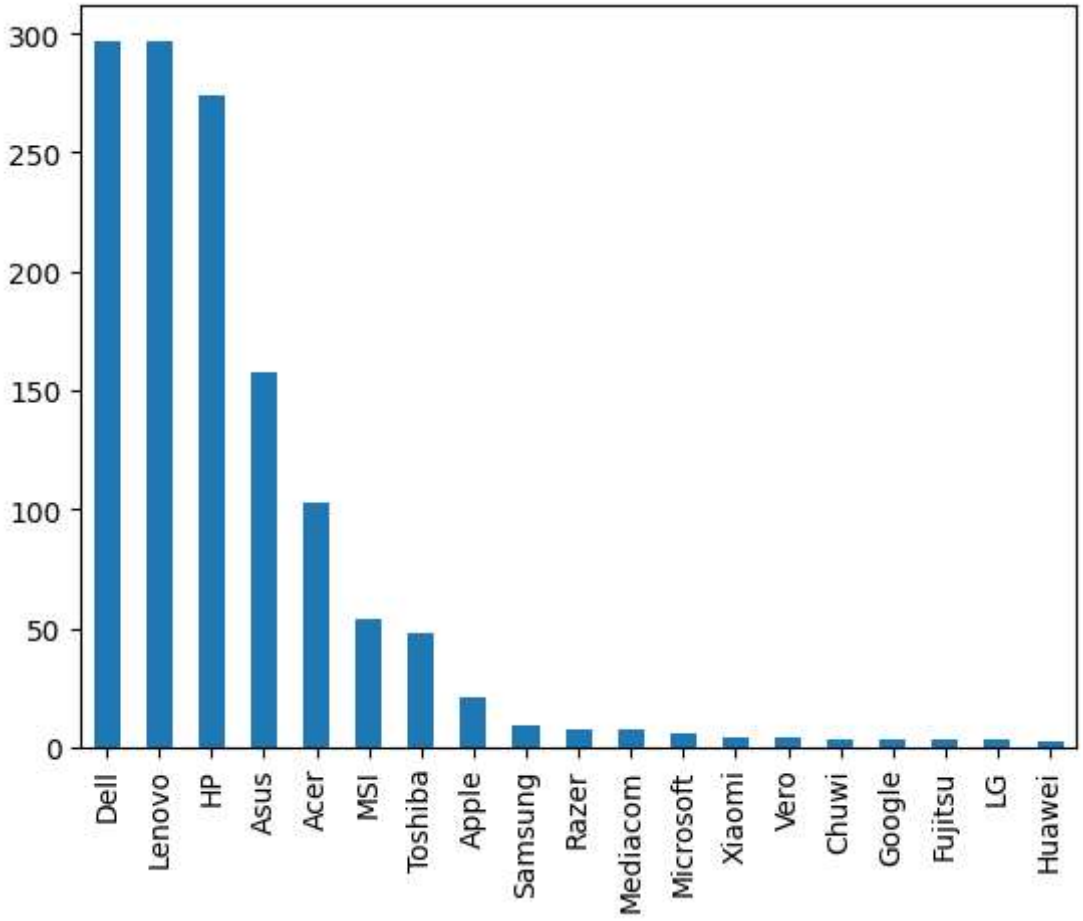
In [15]:
```
sns.displot(df['Price'])
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x2a2b6b810d0>
```



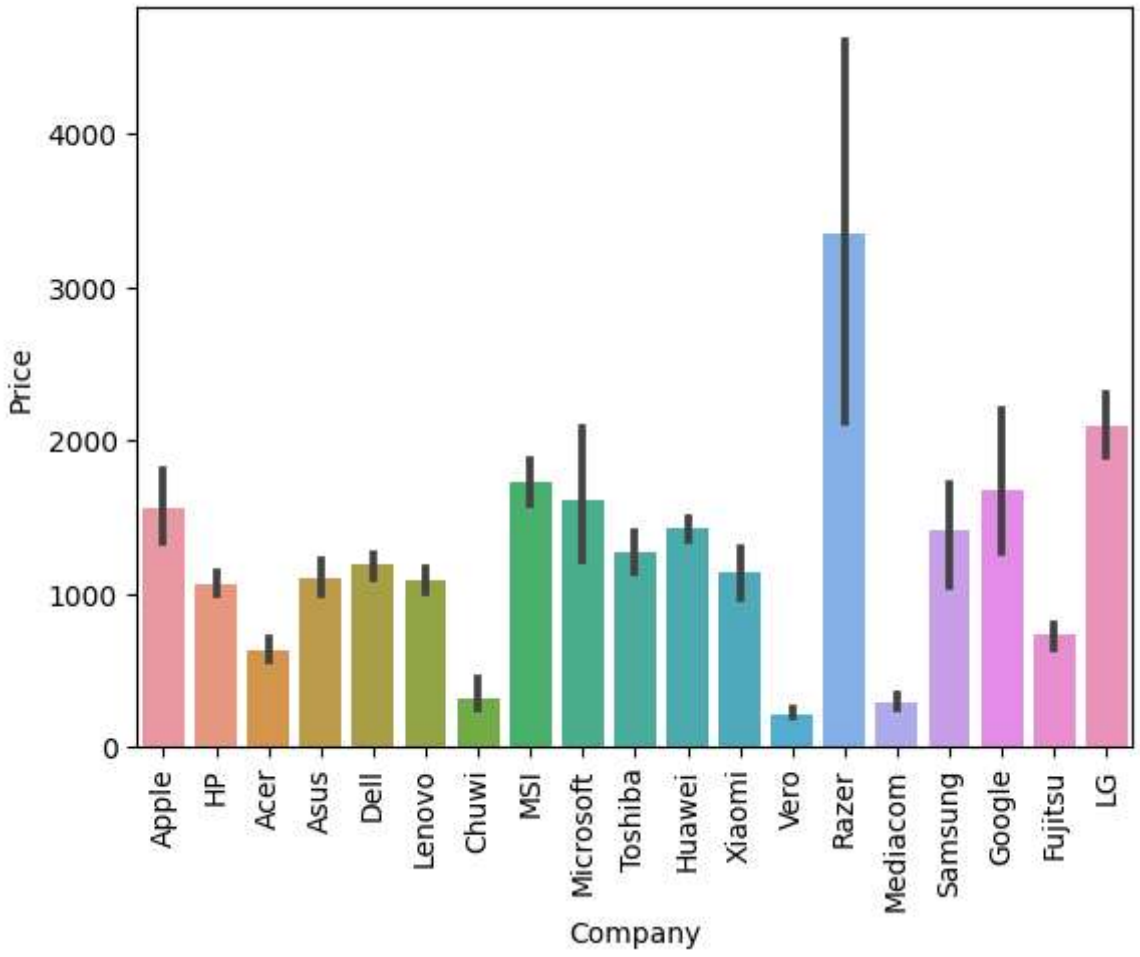
In [16]:
```
df['Company'].value_counts().plot(kind='bar')
```

Out[16]: <Axes: >



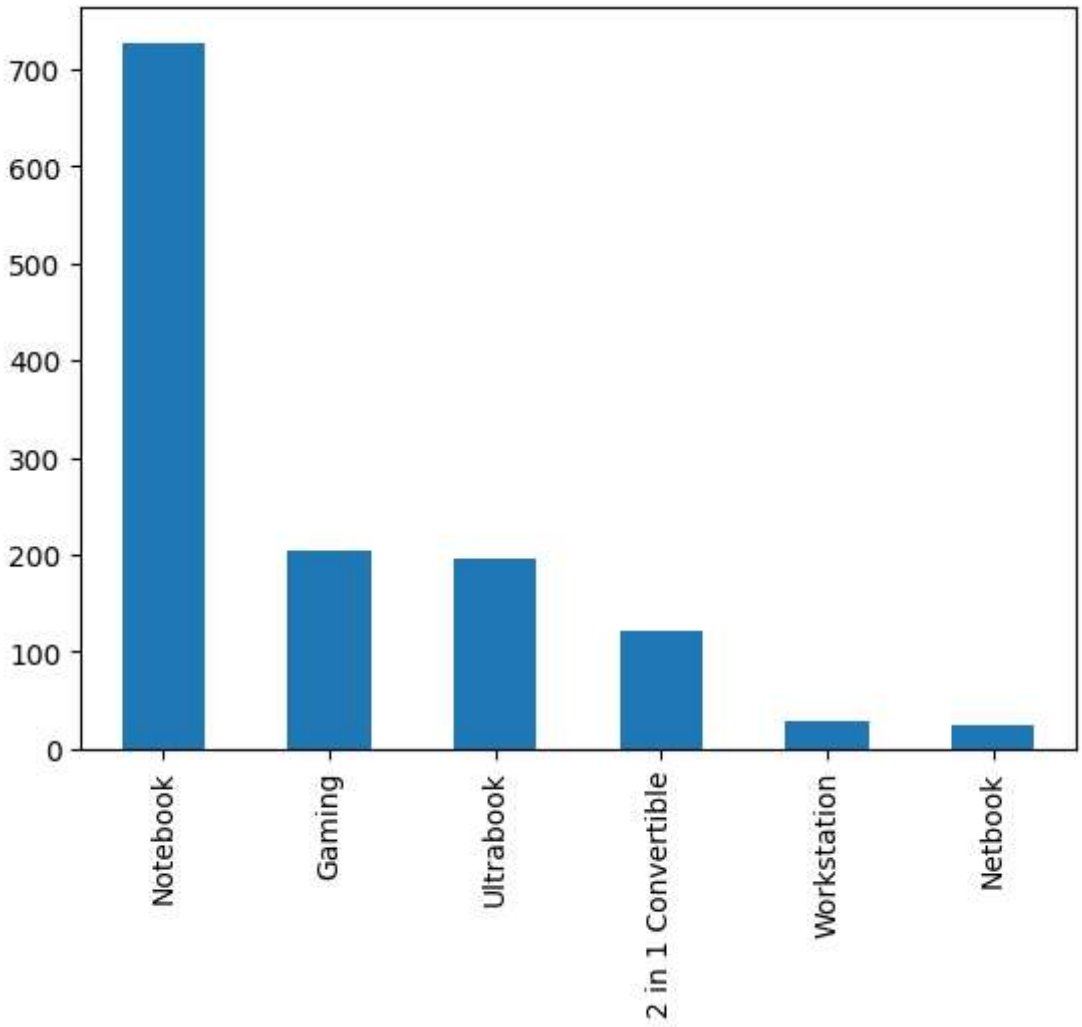
In [17]:

```
import matplotlib.pyplot as plt

sns.barplot(x=df['Company'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```

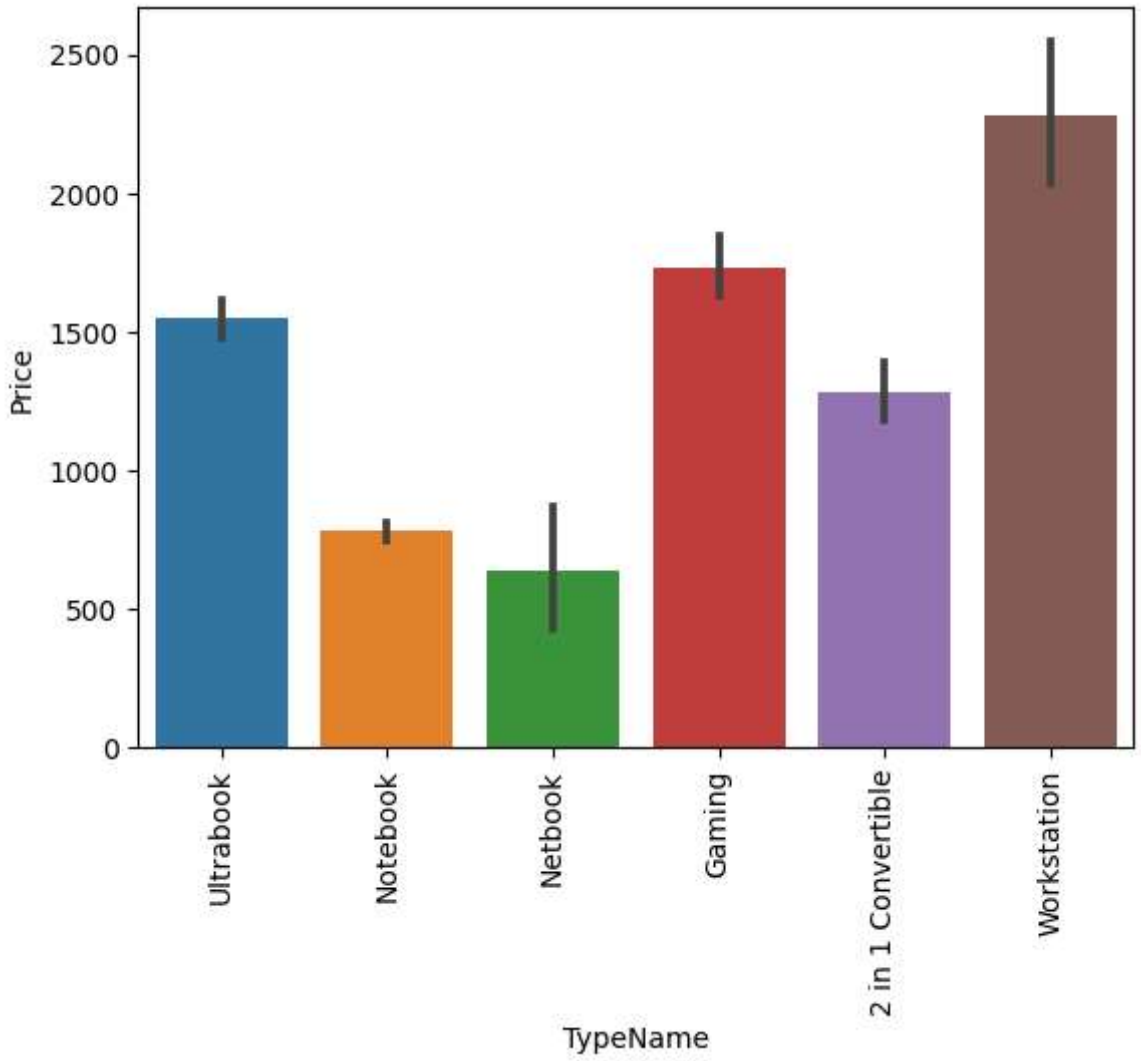


In [18]:
```
df['TypeName'].value_counts().plot(kind='bar')
```

Out[18]: <Axes: >

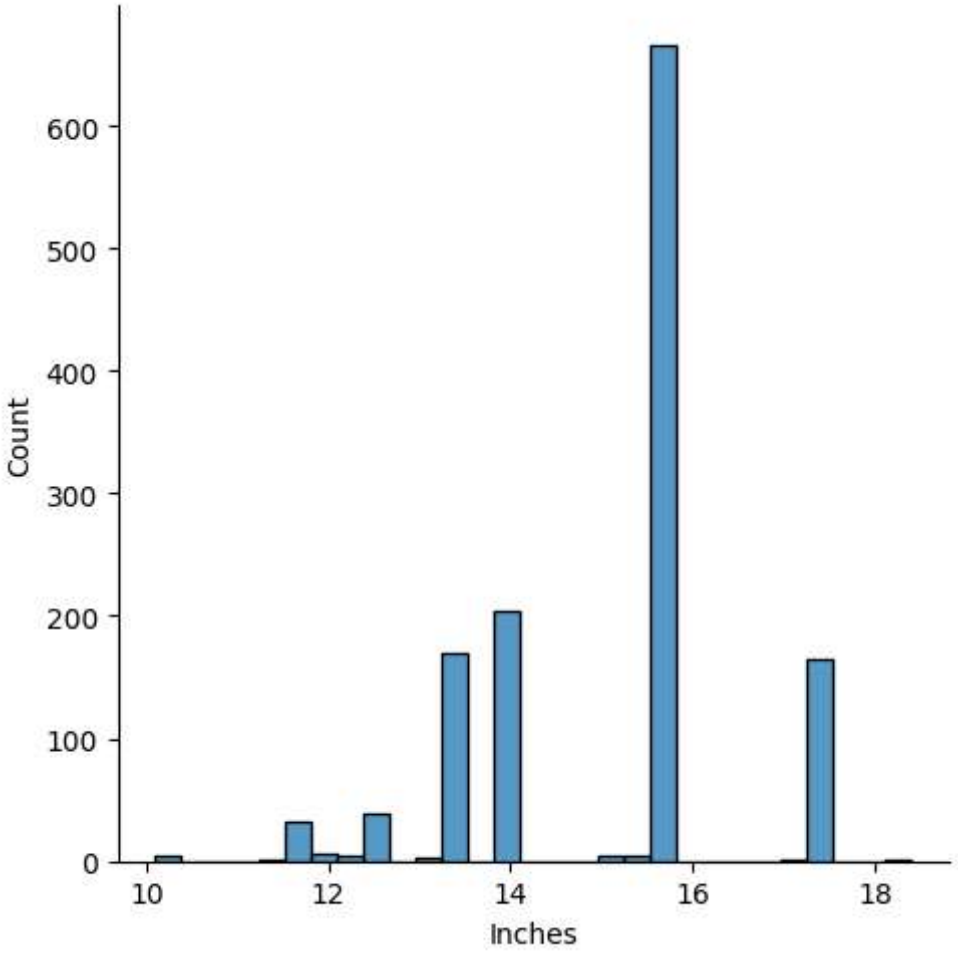


In [19]:
```
#to check the average value
sns.barplot(x=df['TypeName'],y=df['Price'])

plt.xticks(rotation='vertical')
plt.show()
```



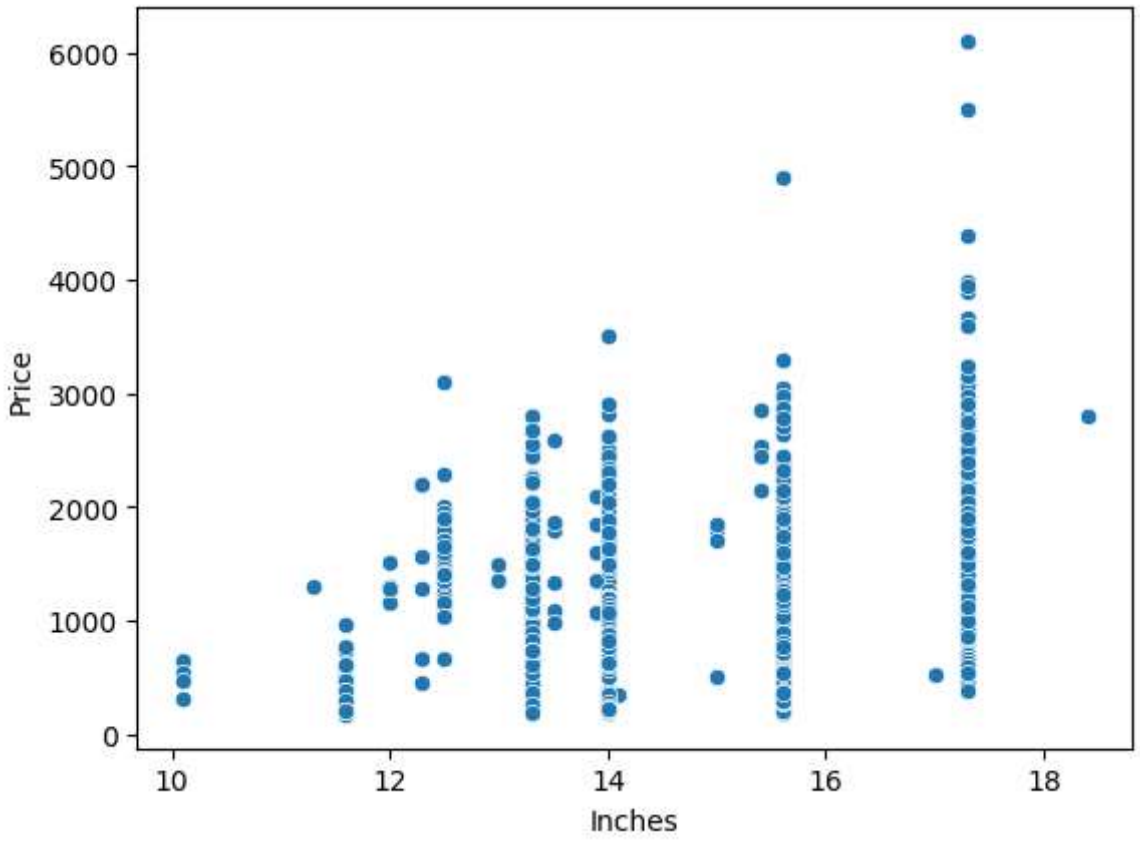
In [20]:
```
#to check the  size
sns.displot(df['Inches'])
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x2a2c29fd550>



In [21]:
```
#to find clear value of size
sns.scatterplot(x=df['Inches'],y=df['Price'])
```

Out[21]: <Axes: xlabel='Inches', ylabel='Price'>



In [22]:
```
#to find screenresolution value
df['ScreenResolution'].value_counts()
```

Out[22]:
```
Full HD 1920x1080                                507
1366x768                                         281
IPS Panel Full HD 1920x1080                      230
IPS Panel Full HD / Touchscreen 1920x1080         53
Full HD / Touchscreen 1920x1080                   47
1600x900                                          23
Touchscreen 1366x768                              16
Quad HD+ / Touchscreen 3200x1800                  15
IPS Panel 4K Ultra HD 3840x2160                   12
IPS Panel 4K Ultra HD / Touchscreen 3840x2160     11
4K Ultra HD / Touchscreen 3840x2160               10
4K Ultra HD 3840x2160                              7
Touchscreen 2560x1440                              7
IPS Panel 1366x768                                 7
IPS Panel Quad HD+ / Touchscreen 3200x1800         6
IPS Panel Retina Display 2560x1600                 6
IPS Panel Retina Display 2304x1440                 6
Touchscreen 2256x1504                              6
IPS Panel Touchscreen 2560x1440                    5
IPS Panel Retina Display 2880x1800                 4
```

Feature Engineering

In [23]:
```
#add a new column to find touch screening
df['Touchscreen']=df['ScreenResolution'].apply(lambda x:1 if 'Touchscreen' in x
```

In [24]:
```
df.head()
```

Out[24]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	C
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	m
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	m
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	I
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	m
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	m

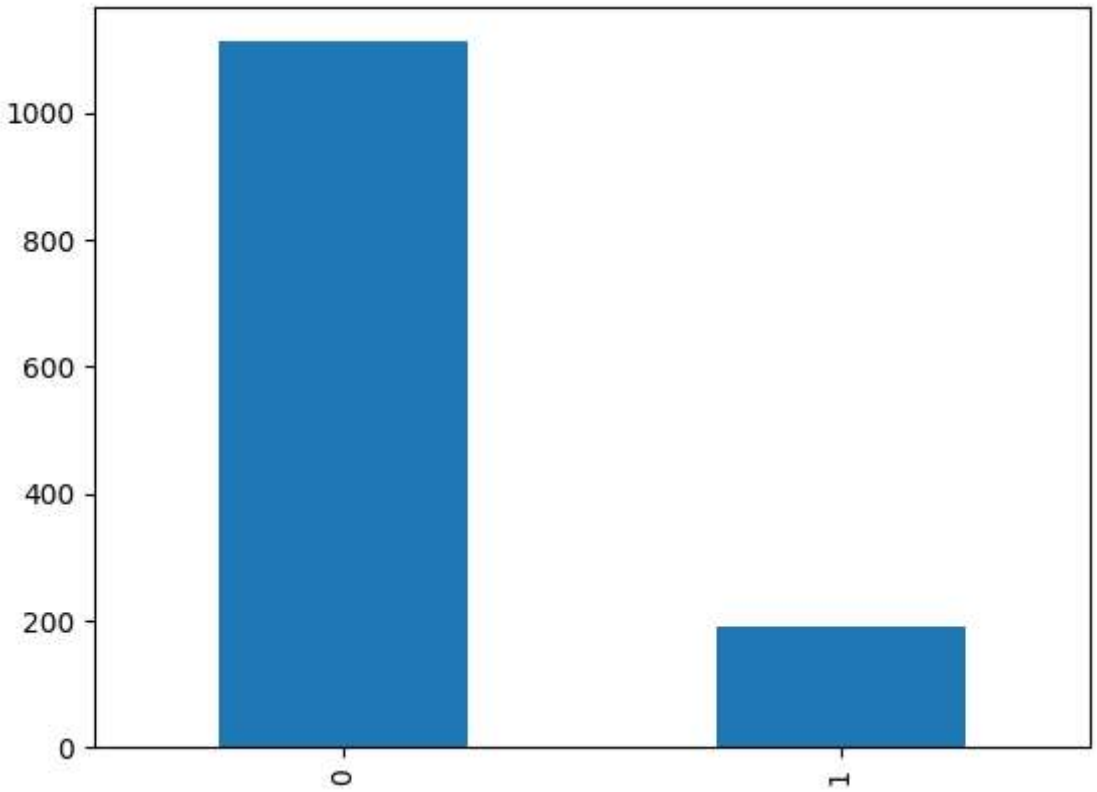
In [25]:
```
df.sample(5)
```

Out[25]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory
944	Lenovo	Yoga 700-11ISK	2 in 1 Convertible	11.3	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core M m7-6Y75 1.2GHz	8	256GB SSD
415	Dell	Inspiron 3567	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	4	256GB SSD
1030	HP	ProBook 440	Notebook	14.0	1366x768	Intel Core i5 7200U 2.5GHz	4	256GB SSD
725	Lenovo	V310-15IKB (i5-7200U/4GB/1TB/No	Notebook	15.6	1366x768	Intel Core i5 7200U 2.5GHz	4	1TB HDD
429	Mediacom	FlexBook Edge	2 in 1 Convertible	11.6	IPS Panel Full HD / Touchscreen 1920x1080	Intel Celeron Dual Core N3350 1.1GHz	4	32GB SSD

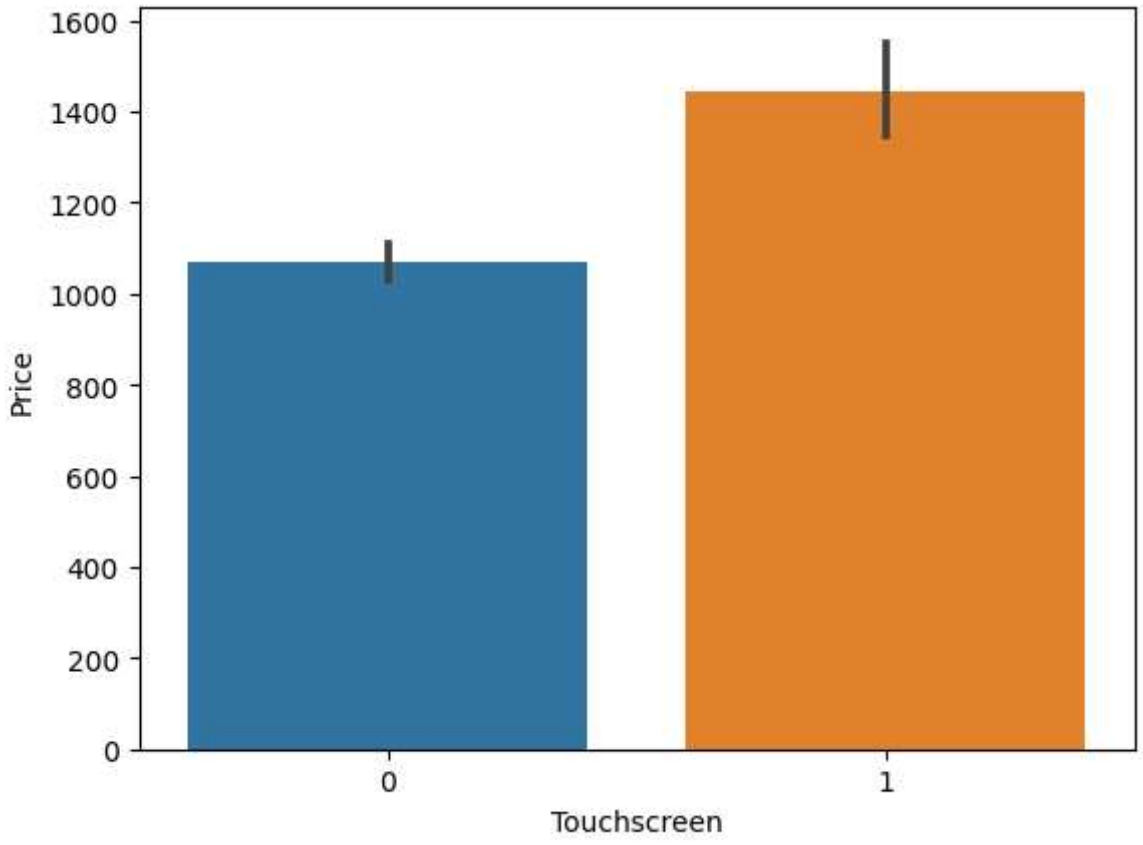
In [26]:
```
#to find the touchscreen laptop
df['Touchscreen'].value_counts().plot(kind='bar')
```

Out[26]: <Axes: >



In [27]:
```
sns.barplot(x=df['Touchscreen'],y=df['Price'])
```

Out[27]:
```
<Axes: xlabel='Touchscreen', ylabel='Price'>
```



In [28]:
```
#to find the IPS panel in  laptop ,add a column
df['Ips'] = df['ScreenResolution'].apply(lambda x:1 if 'IPS' in x else 0)
```

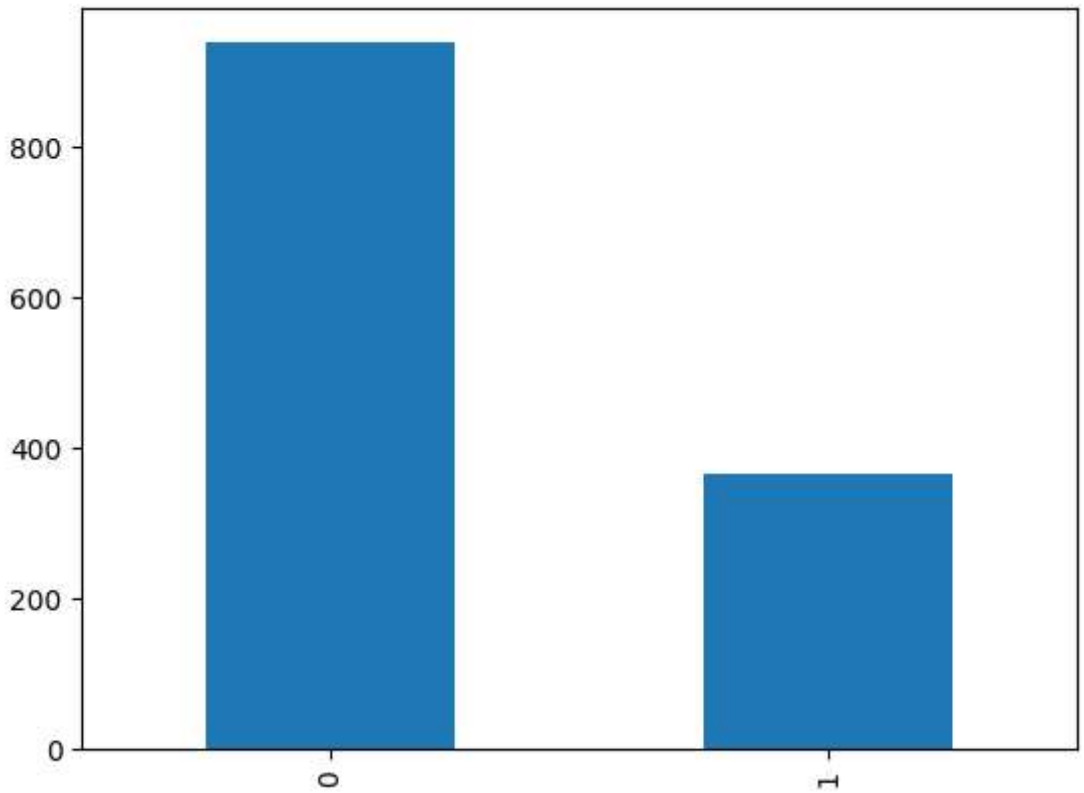
In [29]:
```
df.head()
```

Out[29]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	C
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	m
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	m
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	I
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	m
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	m

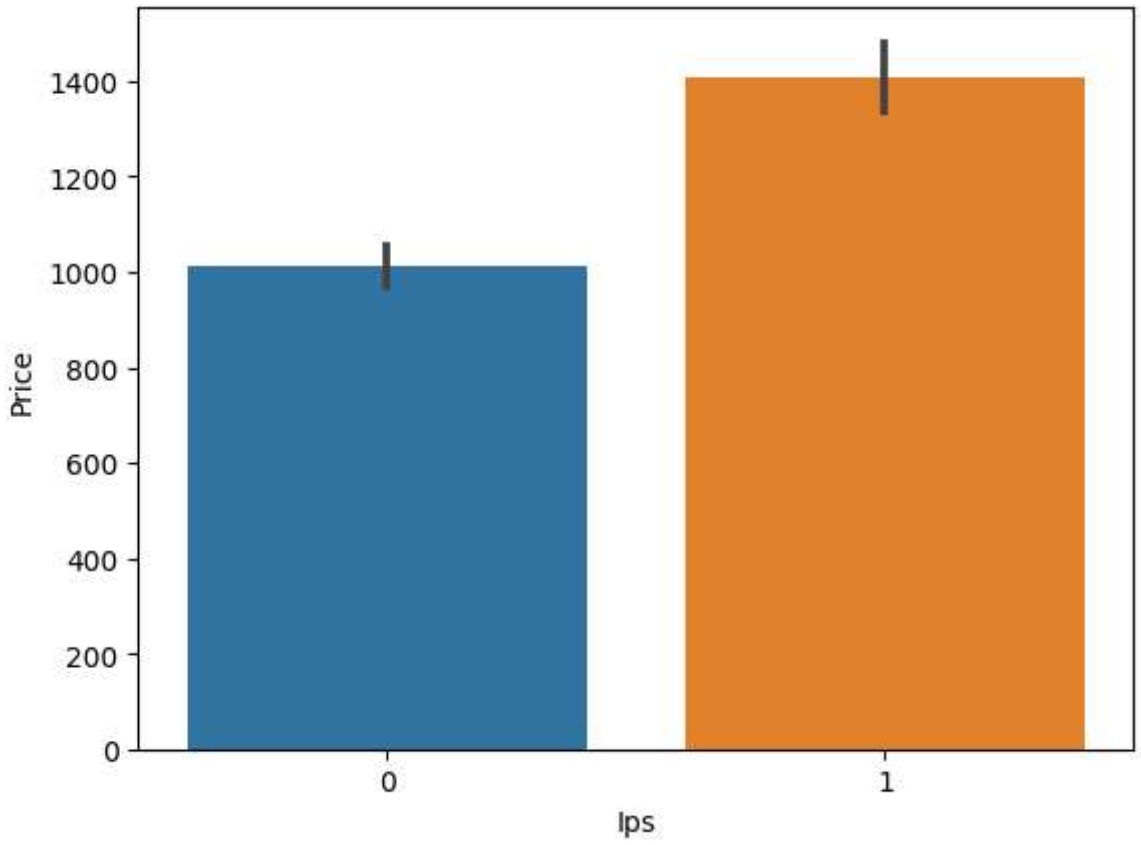
In [30]:
```
df['Ips'].value_counts().plot(kind='bar')
```

Out[30]:
```
<Axes: >
```



In [31]:
```
sns.barplot(x=df['Ips'],y=df['Price'])
```

Out[31]: <Axes: xlabel='Ips', ylabel='Price'>



In [32]:
```
new = df['ScreenResolution'].str.split('x',n=1,expand=True)
```

In [33]:
```
df['x_res'] = new[0]
df['y_res'] = new[1]
```

In [34]:
```
df.head()
```

Out[34]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	C
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	m
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	m
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	I
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	m
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	m

In [35]:
```
df['x_res'].str.replace(',','').str.findall('r(\d+\.?\d+)')
```

Out[35]:
```
0       []
1       []
2       []
3       []
4       []
       ..
1298    []
1299    []
1300    []
1301    []
1302    []
Name: x_res, Length: 1303, dtype: object
```

In [36]:
```
df.head()
```

Out[36]:

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	C
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	m
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	m
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	I
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	m
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	m

In [37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Company           1303 non-null   object 
 1   Product           1303 non-null   object 
 2   TypeName          1303 non-null   object 
 3   Inches            1303 non-null   float64
 4   ScreenResolution  1303 non-null   object 
 5   Cpu               1303 non-null   object 
 6   Ram               1303 non-null   int32  
 7   Memory            1303 non-null   object 
 8   Gpu               1303 non-null   object 
 9   OpSys             1303 non-null   object 
 10  Weight            1303 non-null   object 
 11  Price             1303 non-null   float64
 12  Weiight           1303 non-null   float32
 13  Touchscreen       1303 non-null   int64  
 14  Ips               1303 non-null   int64  
 15  x_res             1303 non-null   object 
 16  y_res             1303 non-null   object 
dtypes: float32(1), float64(2), int32(1), int64(2), object(11)
memory usage: 163.0+ KB
```

In [38]:
```
df.corr()['Price']
```

```
C:\Users\Ankit Goyal\AppData\Local\Temp\ipykernel_4376\1883561535.py:1: Futur
eWarning: The default value of numeric_only in DataFrame.corr is deprecated.
In a future version, it will default to False. Select only valid columns or s
pecify the value of numeric_only to silence this warning.
  df.corr()['Price']
```

Out[38]:
```
Inches         0.068197
Ram            0.743007
Price          1.000000
Weiight        0.210370
Touchscreen    0.191226
Ips            0.252208
Name: Price, dtype: float64
```

In [39]:
```
df.drop(columns=['Inches'],inplace=True)
```

In [40]:
```
df.drop (columns=['ScreenResolution'],inplace=True)
```

In [41]:
```
df.head()
```

Out[41]:

	Company	Product	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	W
0	Apple	MacBook Pro	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	
1	Apple	Macbook Air	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	
2	HP	250 G6	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	
3	Apple	MacBook Pro	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	
4	Apple	MacBook Pro	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	

In [42]:
```
#no  need to add this column#
df.drop(columns=['x_res','y_res'],inplace=True)
```

In [43]:
```
df.head()
```

Out[43]:

	Company	Product	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	W
0	Apple	MacBook Pro	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	
1	Apple	Macbook Air	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	
2	HP	250 G6	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	
3	Apple	MacBook Pro	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	
4	Apple	MacBook Pro	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	

In [44]:
```
df['Cpu'].value_counts()
```

Out[44]:
```
Intel Core i5 7200U 2.5GHz       190
Intel Core i7 7700HQ 2.8GHz      146
Intel Core i7 7500U 2.7GHz       134
Intel Core i7 8550U 1.8GHz        73
Intel Core i5 8250U 1.6GHz        72
                                 ...
Intel Core M M3-6Y30 0.9GHz        1
AMD A9-Series 9420 2.9GHz          1
Intel Core i3 6006U 2.2GHz         1
AMD A6-Series 7310 2GHz            1
Intel Xeon E3-1535M v6 3.1GHz      1
Name: Cpu, Length: 118, dtype: int64
```

In [45]:
```
#feature Engineering and add five columns#
```

In [46]:
```
df['Cpu Name']=df['Cpu'].apply(lambda x:x.split()[0:3])
```

In [47]:
```
df.head()
```

Out[47]:

	Company	Product	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	W
0	Apple	MacBook Pro	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	
1	Apple	Macbook Air	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	
2	HP	250 G6	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	
3	Apple	MacBook Pro	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	
4	Apple	MacBook Pro	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	

In [48]:
```
def fetch_processor(text):
    if text == 'intel core i7' or text == 'Intel core  i5' or text == 'Intel  c
        return textmm
    else:
        if text.split()[0] == 'Intel' :
            return 'Other Intel Processor'
        else:
            return 'AMD Processor'
```

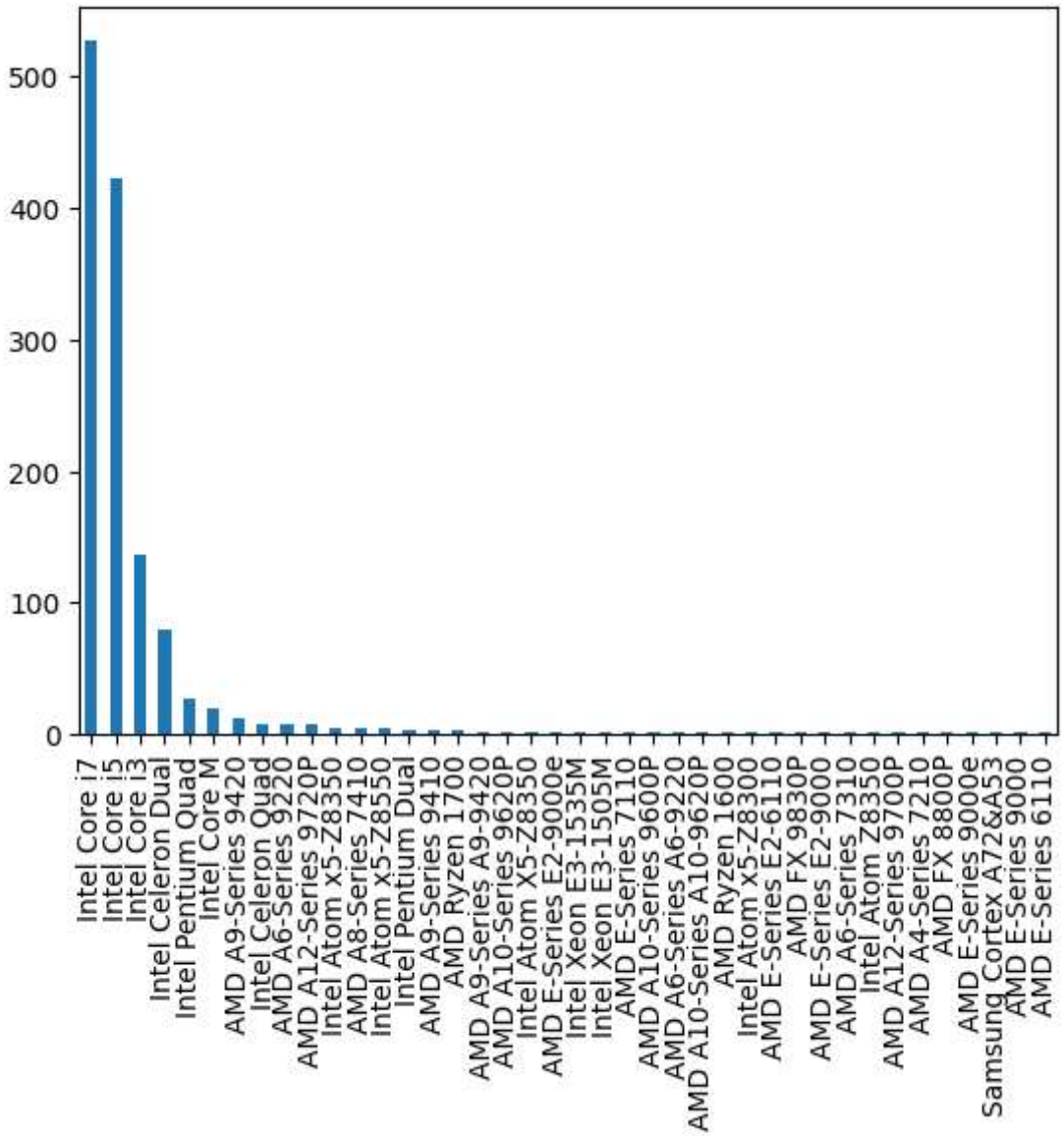
In [49]:
```
df['Cpu brand'] = df['Cpu'].apply(lambda x:" ".join(x.split()[0:3]))
```

In [50]:
```
df.head()
```

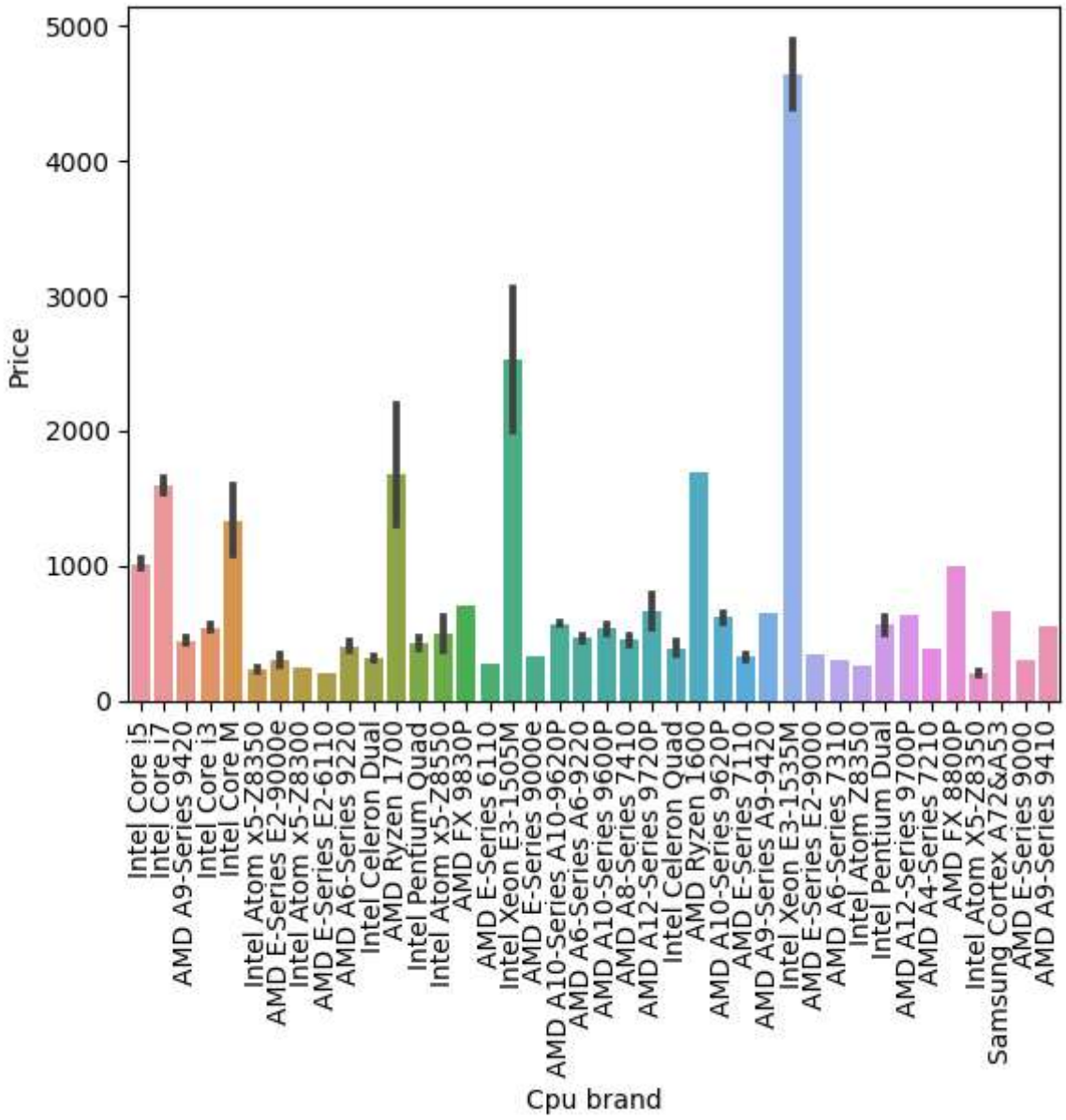
Out[50]:

	Company	Product	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	W
0	Apple	MacBook Pro	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	
1	Apple	Macbook Air	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	
2	HP	250 G6	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	
3	Apple	MacBook Pro	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	
4	Apple	MacBook Pro	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	

In [51]:
```
df['Cpu brand'].value_counts().plot(kind='bar')
plt.xticks(rotation='vertical')
plt.show()
```



In [52]:
```
sns.barplot(x=df['Cpu brand'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [53]:
```
df.drop(columns=['Cpu','Cpu Name'],inplace = True)
```

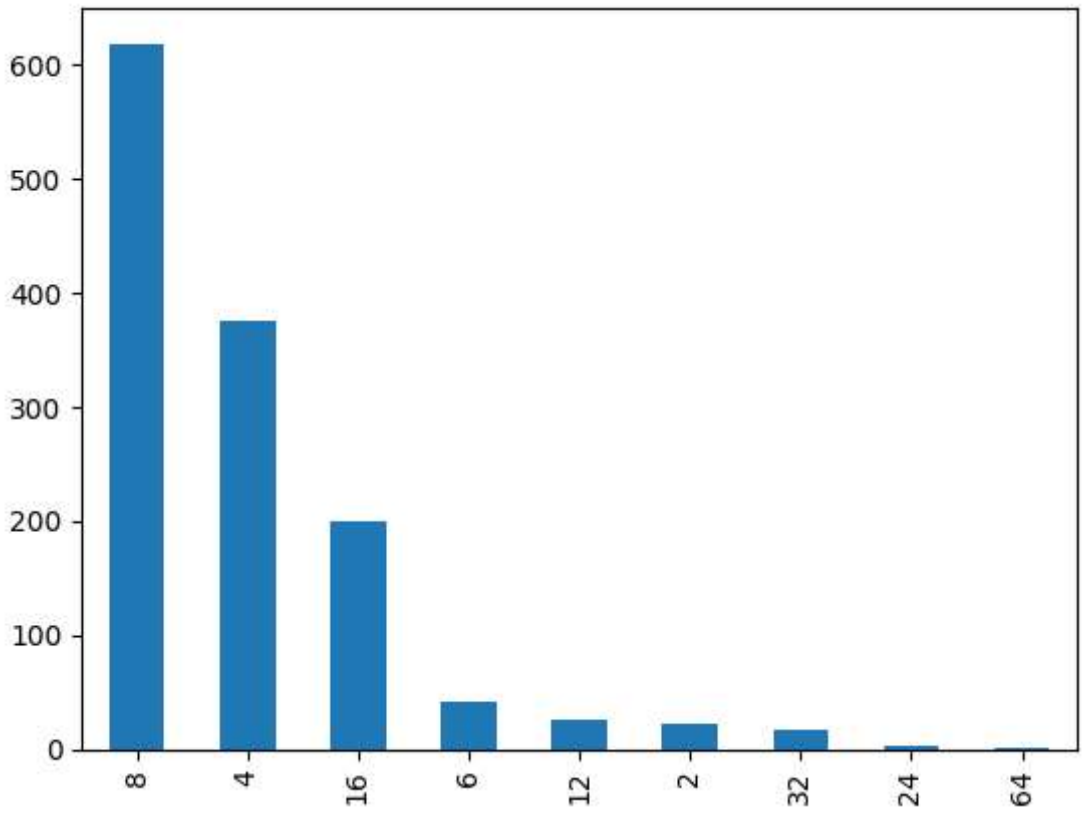
In [54]:
```
df.head()
```

Out[54]:

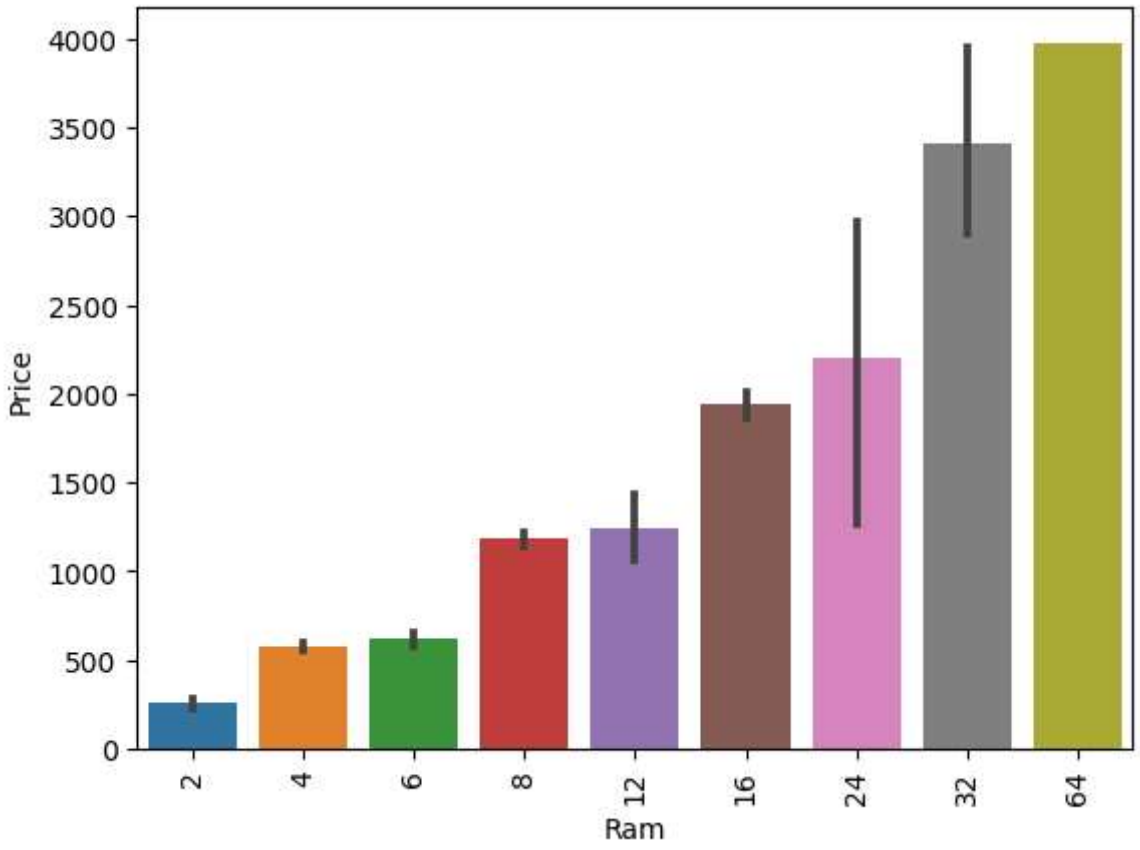
	Company	Product	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Weiight	Tc
0	Apple	MacBook Pro	Ultrabook	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	1.37	
1	Apple	Macbook Air	Ultrabook	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	1.34	
2	HP	250 G6	Notebook	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	1.86	
3	Apple	MacBook Pro	Ultrabook	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	1.83	
4	Apple	MacBook Pro	Ultrabook	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	1.37	

In [55]:
```
df['Ram'].value_counts().plot(kind='bar')
```

Out[55]: <Axes: >



In [56]:
```
sns.barplot(x=df['Ram'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [57]:
```
#to focus on memory ,transform the flash storage
df['Memory'].value_counts()
```

Out[57]:
```
256GB SSD                        412
1TB HDD                          223
500GB HDD                        132
512GB SSD                        118
128GB SSD +  1TB HDD              94
128GB SSD                         76
256GB SSD +  1TB HDD              73
32GB Flash Storage                38
2TB HDD                           16
64GB Flash Storage                15
512GB SSD +  1TB HDD              14
1TB SSD                           14
256GB SSD +  2TB HDD              10
1.0TB Hybrid                       9
256GB Flash Storage                8
16GB Flash Storage                 7
32GB SSD                           6
180GB SSD                          5
128GB Flash Storage                4
512GB SSD +  2TB HDD               3
16GB SSD                           3
512GB Flash Storage                2
1TB SSD +  1TB HDD                 2
256GB SSD +  500GB HDD             2
128GB SSD +  2TB HDD               2
256GB SSD +  256GB SSD             2
512GB SSD +  256GB SSD             1
512GB SSD +  512GB SSD             1
64GB Flash Storage +  1TB HDD      1
1TB HDD +  1TB HDD                 1
32GB HDD                           1
64GB SSD                           1
128GB HDD                          1
240GB SSD                          1
8GB SSD                            1
508GB Hybrid                       1
1.0TB HDD                          1
512GB SSD +  1.0TB Hybrid          1
256GB SSD +  1.0TB Hybrid          1
Name: Memory, dtype: int64
```

In [58]:
```
df['Gpu'].value_counts()
```

Out[58]:
```
Intel HD Graphics 620      281
Intel HD Graphics 520      185
Intel UHD Graphics 620      68
Nvidia GeForce GTX 1050     66
Nvidia GeForce GTX 1060     48
                          ...
AMD Radeon R5 520            1
AMD Radeon R7                1
Intel HD Graphics 540        1
AMD Radeon 540               1
ARM Mali T860 MP4            1
Name: Gpu, Length: 110, dtype: int64
```

In [59]:
```
df['Gpu brand']=df['Gpu'].apply(lambda x:x.split()[0])
```

In [60]:
```
df.head()
```

Out[60]:

	Company	Product	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Weiight	To
0	Apple	MacBook Pro	Ultrabook	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69	1.37	
1	Apple	Macbook Air	Ultrabook	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94	1.34	
2	HP	250 G6	Notebook	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00	1.86	
3	Apple	MacBook Pro	Ultrabook	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45	1.83	
4	Apple	MacBook Pro	Ultrabook	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60	1.37	

In [61]:
```
df['Gpu brand'].value_counts()
```

Out[61]:
```
Intel     722
Nvidia    400
AMD       180
ARM         1
Name: Gpu brand, dtype: int64
```

In [62]:
```
df = df[df['Gpu brand'] != 'ARM']
```

In [63]:
```
df['Gpu brand'].value_counts()
```

Out[63]:
```
Intel     722
Nvidia    400
AMD       180
Name: Gpu brand, dtype: int64
```

In [64]:
```
df=df['sns.barplot(x=df['Gpu branrice'],estimator=np.median)
plt.xticks(rotation='vertical')
plt.show()
```

```
  Cell In[64], line 1
    df=df['sns.barplot(x=df['Gpu branrice'],estimator=np.median)
                                            ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [65]:
```
df.drop(columns=['Gpu'],inplace=True)
```

In [66]:
```
df.head()
```

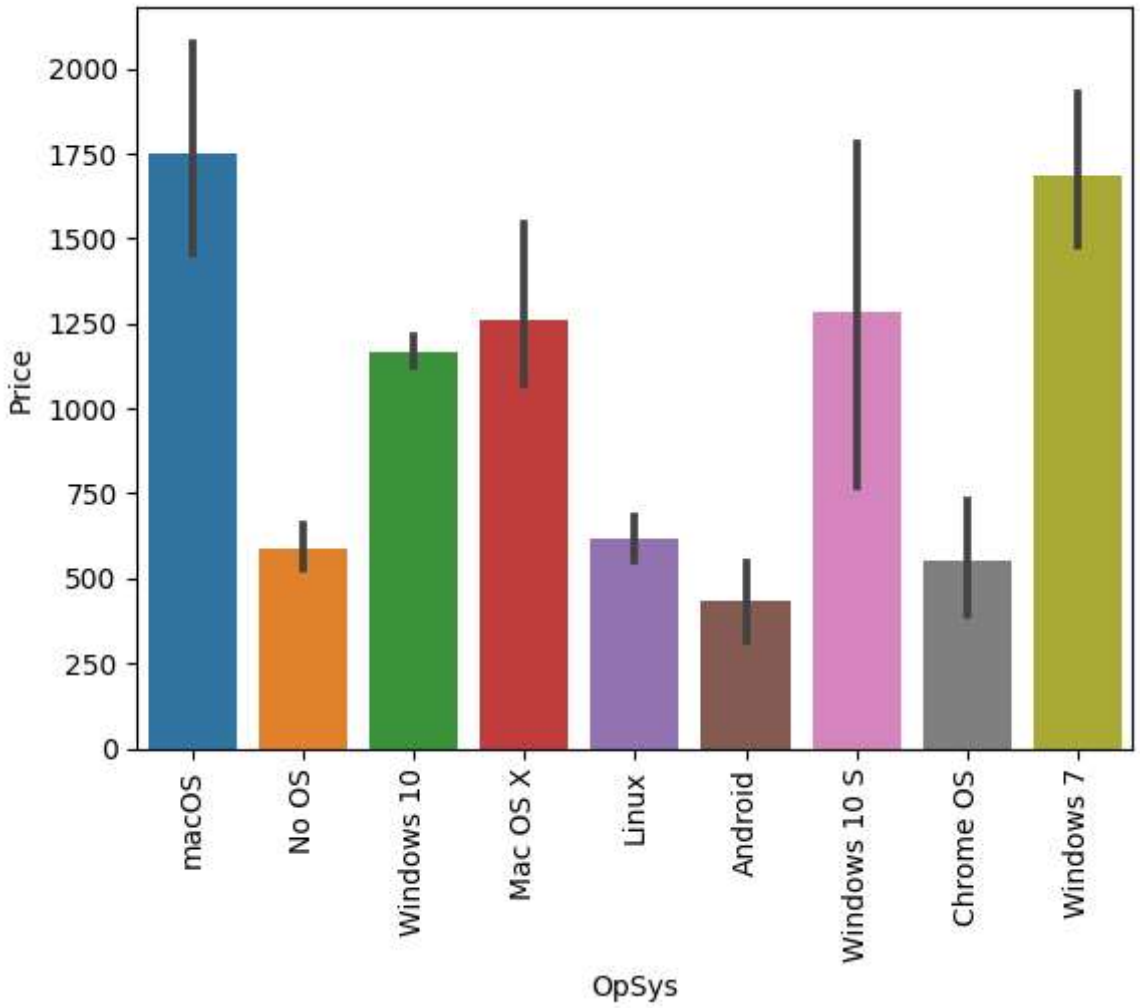
Out[66]:

	Company	Product	TypeName	Ram	Memory	OpSys	Weight	Price	Weiight	Touchscreer
0	Apple	MacBook Pro	Ultrabook	8	128GB SSD	macOS	1.37	1339.69	1.37	(
1	Apple	Macbook Air	Ultrabook	8	128GB Flash Storage	macOS	1.34	898.94	1.34	(
2	HP	250 G6	Notebook	8	256GB SSD	No OS	1.86	575.00	1.86	(
3	Apple	MacBook Pro	Ultrabook	16	512GB SSD	macOS	1.83	2537.45	1.83	(
4	Apple	MacBook Pro	Ultrabook	8	256GB SSD	macOS	1.37	1803.60	1.37	(

In [67]:
```
df['OpSys'].value_counts()
```

Out[67]:
```
Windows 10      1072
No OS             66
Linux             62
Windows 7         45
Chrome OS         26
macOS             13
Mac OS X           8
Windows 10 S       8
Android            2
Name: OpSys, dtype: int64
```

In [68]:
```
sns.barplot(x=df['OpSys'],y=df['Price'])
plt.xticks(rotation ='vertical')
plt.show()
```



In [69]:
```
def cat_os(inp):
    if inp =='Windows 10' or inp == 'Windows 7' or inp =='Windows 10 S':
        return 'Windows'
    elif inp == 'macOS' or inp == 'Mac OS x':
        return 'Mac'
    else:
        return 'Other/No OS/Linux'
```

In [70]:
```
df['os'] = df['OpSys'].apply(cat_os)
```

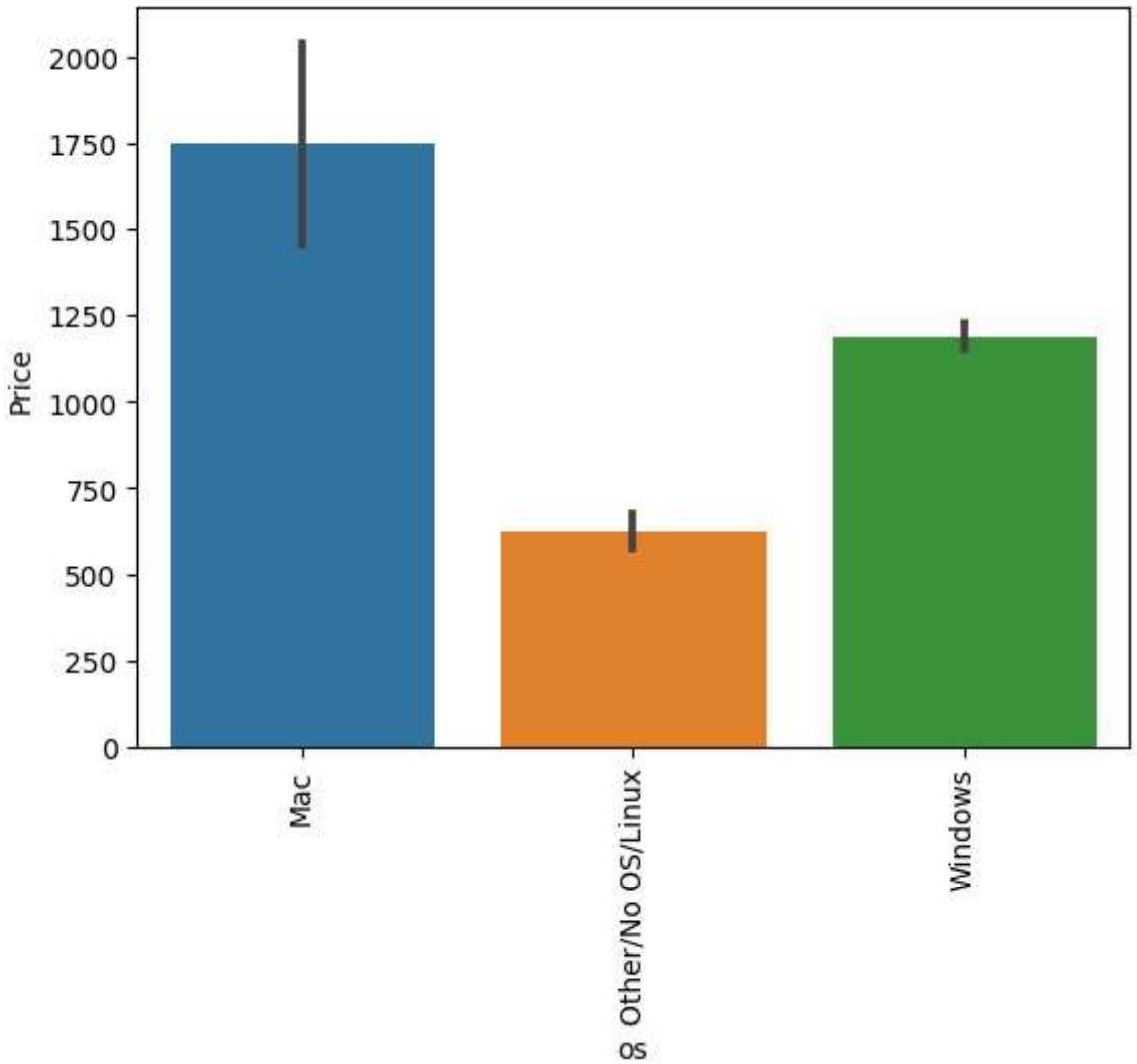
In [71]:
```
df.head()
```

Out[71]:

	Company	Product	TypeName	Ram	Memory	OpSys	Weight	Price	Weiight	Touchscreer
0	Apple	MacBook Pro	Ultrabook	8	128GB SSD	macOS	1.37	1339.69	1.37	(
1	Apple	Macbook Air	Ultrabook	8	128GB Flash Storage	macOS	1.34	898.94	1.34	(
2	HP	250 G6	Notebook	8	256GB SSD	No OS	1.86	575.00	1.86	(
3	Apple	MacBook Pro	Ultrabook	16	512GB SSD	macOS	1.83	2537.45	1.83	(
4	Apple	MacBook Pro	Ultrabook	8	256GB SSD	macOS	1.37	1803.60	1.37	(

In [72]:
```
df.drop(columns=['OpSys'],inplace=True)
```

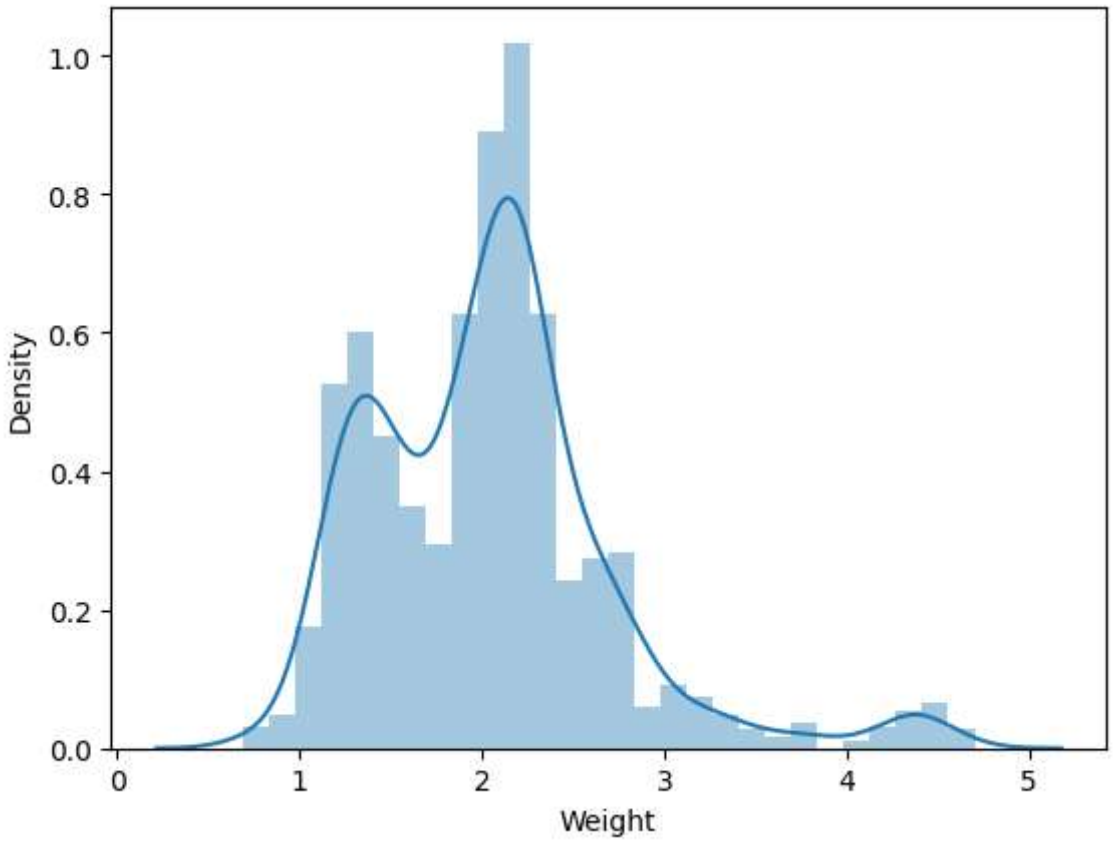
In [73]:
```
sns.barplot(x=df['os'],y=df['Price'])
plt.xticks(rotation ='vertical')
plt.show()
```



In [74]:
```
sns.distplot(df['Weight'])
```

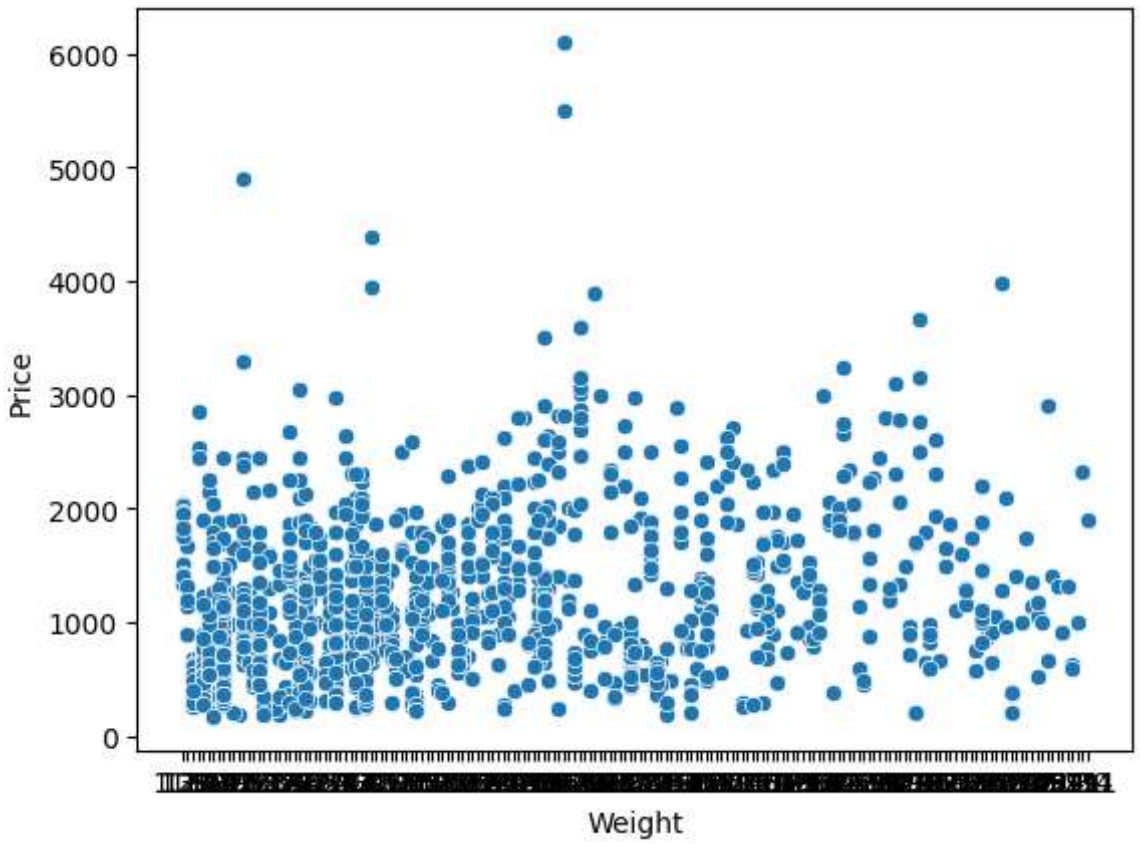
```
C:\Users\Ankit Goyal\AppData\Local\Temp\ipykernel_4376\1125578356.py:1: UserW
arning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['Weight'])
```

Out[74]: <Axes: xlabel='Weight', ylabel='Density'>



In [77]:
```
#we can also add scatterplot
sns.scatterplot(x=df['Weight'],y=df['Price'])
```

Out[77]: <Axes: xlabel='Weight', ylabel='Price'>



In [78]:
```
df.corr()['Price']
```

```
C:\Users\Ankit Goyal\AppData\Local\Temp\ipykernel_4376\815546952.py:1: Future
Warning: The default value of numeric_only in DataFrame.corr is deprecated. I
n a future version, it will default to False. Select only valid columns or sp
ecify the value of numeric_only to silence this warning.
  df.corr()['Price']
```

Out[78]:
```
Ram            0.742905
Price          1.000000
Weiight        0.209867
Touchscreen    0.192917
Ips            0.253320
Name: Price, dtype: float64
```

In [79]:
```
#to match relation to all column
sns.heatmap(df.corr())
```

```
C:\Users\Ankit Goyal\AppData\Local\Temp\ipykernel_4376\1256811914.py:2: Futur
eWarning: The default value of numeric_only in DataFrame.corr is deprecated.
In a future version, it will default to False. Select only valid columns or s
pecify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

Out[79]: <Axes: >