In [ ]:

```
'''
Author:Shruthi Bhat

INTRODUCTION:

This is a Web scraping project which crawls into Yahoo finance website and
programmatically collects one month historical data of AMZN stock using
Python modules beautifulsoup and requests and writes to a CSV file.Based on
the data collected,opening stock price for current day is predicted using
linear regression.Python module scikit learn is used to program linear
regression and Python modules matplotlib is used to plot stock prices and
the corresponding linear regression line in the graph,stock's High values
and Low values over three months period.
url:https://finance.yahoo.com/quote/AMZN/history?period1=1495090800&period2=15030396

'''
```

In [ ]:

```
'''

REQUIREMENTS:

Python version used 2.7

Modules needed:

1.requests
2.beautifulSoup
3.pandas
4.numpy
5.matplotlib
6.scikit-learn
7.datetime
8.time
9.csv

'''
```

In [1]:

```
'''
This makes the Jupyter cells wider

'''
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```

In [2]:

```
'''
The code below uses BeautifulSoup library to scrape July month Amazon stock data fro
Yahoo finance website and writes to AMZN_Stock.csv file

'''
import datetime
import requests
import csv
from bs4 import BeautifulSoup

html_doc=requests.get('https://finance.yahoo.com/quote/AMZN/history?period1=1498892
soup=BeautifulSoup(html_doc.content,'html.parser')
div=soup.find(id="Col1-1-QuoteLeaf-Proxy")
table=div.select_one("table")
#tbody=table.select_one("tbody")

headers = [th.text.encode("utf-8") for th in table.select("tr th")]

with open("AMZN_Stock.csv", "w") as f:
    wr = csv.writer(f)
    wr.writerows([[td.text.encode("utf-8") for td in row.find_all("td")] for row in
```

In [4]:

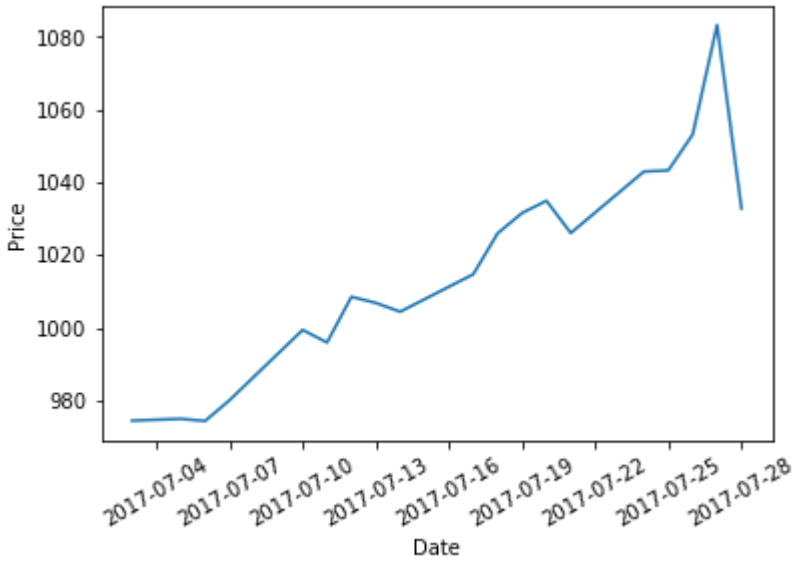
```
'''
Pandas module is used to read the values from csv file and display it
'''
import pandas as pd
import datetime
df = pd.read_csv('AMZN_Stock.csv',names=['Date','Open','High','Low','Close','Adj Clo
df
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-07-28	1012.14	1032.85	1001.00	1020.04	1020.04	7709400
1	2017-07-27	1069.55	1083.31	1040.18	1046.00	1046.00	10991700
2	2017-07-26	1043.20	1053.20	1043.20	1052.80	1052.80	2921300
3	2017-07-25	1038.05	1043.33	1032.48	1039.87	1039.87	2447600
4	2017-07-24	1028.34	1043.01	1027.43	1038.95	1038.95	3288000
5	2017-07-21	1021.28	1026.10	1011.00	1025.67	1025.67	2734600
6	2017-07-20	1031.59	1034.97	1022.52	1028.70	1028.70	3097500
7	2017-07-19	1025.00	1031.59	1022.50	1026.87	1026.87	2964000
8	2017-07-18	1006.00	1026.03	1004.00	1024.45	1024.45	4007600
9	2017-07-17	1004.69	1014.75	1003.81	1010.04	1010.04	3712600
10	2017-07-14	1002.40	1004.45	996.89	1001.81	1001.81	2102500
11	2017-07-13	1004.62	1006.88	995.90	1000.63	1000.63	2880800
12	2017-07-12	1000.65	1008.55	998.10	1006.51	1006.51	3608600
13	2017-07-11	993.00	995.99	983.72	994.13	994.13	2982700
14	2017-07-10	985.00	999.44	983.50	996.47	996.47	3546300
15	2017-07-07	969.55	980.11	969.14	978.76	978.76	2643400
16	2017-07-06	964.66	974.40	959.02	965.14	965.14	3259600
17	2017-07-05	961.53	975.00	955.25	971.40	971.40	3653000
18	2017-07-03	972.79	974.49	951.00	953.66	953.66	2909100

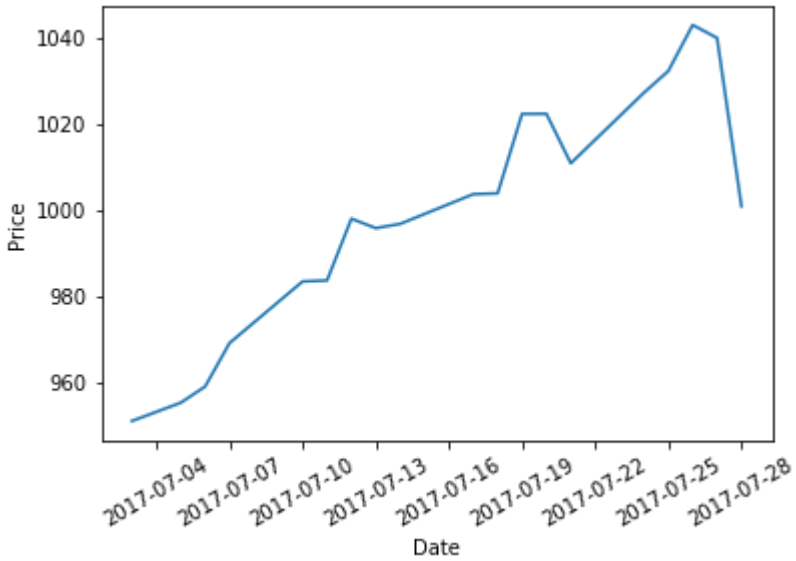
In [5]:

```
'''
The graph below shows the stock's High value for each day in one month period

'''

import numpy as np
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt

high_list=list(df['High'])
date_list=list(df['Date'])
plt.plot(date_list,high_list)
plt.xticks(rotation=30)
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```



In [6]:

```
'''
The graph below shows the stock's Low value for each day in one month period

'''

import numpy as np
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt

low_list=list(df['Low'])
date_list=list(df['Date'])
plt.plot(date_list,low_list)
plt.xticks(rotation=30)
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```

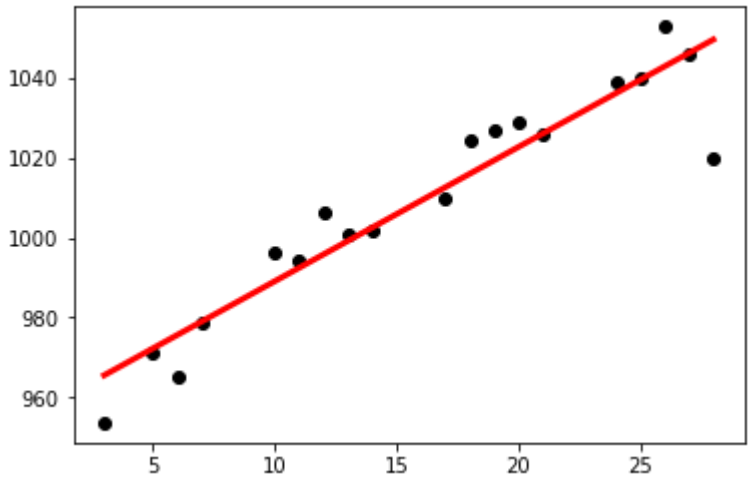


In [7]:

```
'''
matplotlib and scikit learn module are used to plot graph and
calculate simple linear regression respectively

'''
import datetime
import time
import csv
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt


dates = []
prices = []

def get_data(filename):
    '''
        This method is used to read the values from the columns corresponding to dat
        opening price and append the dates and prices list respectively
    '''
    with open(filename,'Ur') as csvfile:
        csvFileReader = csv.reader(csvfile)
        for row in csvFileReader:
            dates.append(int(row[0].split(",")[0].split(" ")[1]))
            prices.append(float(row[4].replace(',','')))
    return

def show_plot(dates,prices):
    '''
        This method is used to calculate the simple linear regression
    '''
    linear_mod = linear_model.LinearRegression()
    dates = np.reshape(dates,(len(dates),1)) # converting to matrix of n X 1
    prices = np.reshape(prices,(len(prices),1))
    linear_mod.fit(dates,prices) #fitting the data points in the model
    plt.scatter(dates,prices,color='black') #plotting the initial datapoints
    plt.plot(dates,linear_mod.predict(dates),color='red',linewidth=3) #plotting the
    plt.show()
    return

def predict_price(dates,prices,x):
    linear_mod = linear_model.LinearRegression() #defining the linear regression mod
    dates = np.reshape(dates,(len(dates),1)) # converting to matrix of n X 1
    prices = np.reshape(prices,(len(prices),1))
    linear_mod.fit(dates,prices) #fitting the data points in the model
    predicted_price =linear_mod.predict(x)
    return predicted_price[0][0],linear_mod.coef_[0][0] ,linear_mod.intercept_[0]

get_data('AMZN_Stock.csv') # calling get_data method by passing the csv file to it
print dates
print prices
print "\n"

show_plot(dates,prices)



today=datetime.datetime.now()
```

```
#passing today's date to the model

predicted_price, coefficient, constant = predict_price(dates,prices,today.day)
print "CONCLUSION:"
print "Based on July's Data, predicted price for AMZN stock for today({}) is:   ${}
print "The regression coefficient is ",str(coefficient),", and the constant is ", st
print "the relationship equation between dates and prices is: price = ",str(coeffici
```

```
[28, 27, 26, 25, 24, 21, 20, 19, 18, 17, 14, 13, 12, 11, 10, 7, 6, 5,
 3]
[1020.04, 1046.0, 1052.8, 1039.87, 1038.95, 1025.67, 1028.7, 1026.87,
 1024.45, 1010.04, 1001.81, 1000.63, 1006.51, 994.13, 996.47, 978.76,
 965.14, 971.4, 953.66]
```




```
CONCLUSION:
Based on July's Data, predicted price for AMZN stock for today(08/28/2
017) is:   $1049.55658813
The regression coefficient is  3.36139457691 , and the constant is  95
5.437539972
the relationship equation between dates and prices is: price =  3.3613
9457691 * date +  955.437539972
```

In [ ]:

```
'''
References:
1.https://www.analyticsvidhya.com/blog/2015/10/beginner-guide-web-scraping-beautiful
2.https://medium.freecodecamp.org/how-to-scrape-websites-with-python-and-beautifulso
3.https://www.dataquest.io/blog/web-scraping-tutorial-python/
4.https://automatetheboringstuff.com/chapter11/
5.https://www.cyberciti.biz/faq/howto-get-current-date-time-in-python/
6.http://beancoder.com/linear-regression-stock-prediction/
7.https://medium.com/towards-data-science/simple-and-multiple-linear-regression-in-p
'''
```