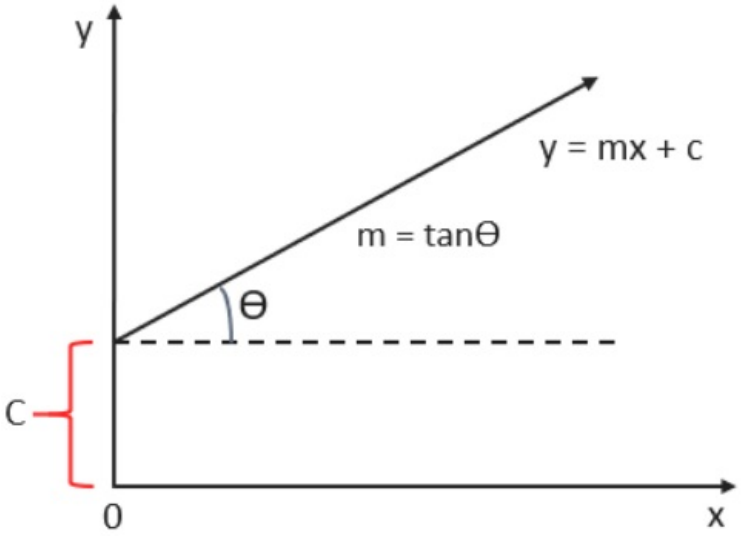
y=mx+c (Linear Regression)

Its believed, assumed y equal to x : y=x, c is constant, and c is nothing , its y when x is equal to zero. m is nothing, its tan which is equal to 1. See below drawing: when x is equal to y, angle will be 45 degree and tan 45 is equal to 1.



When x is 0, what is value of y- That is C.

Linear regression Types

- Simple linear regression
- Multiple Linear regression
- Polynomial linear regression

In simple linear regression, there is one column for input and one column for output.

Suppose we have data, one is CGPA, second is PACKAGE(i.e.How many income student earning after graduation) - like from this data we can know what earning and what is his/her CGPA. Means Based on CGPA, in future we can predict this student can that much because we have both input and output historical data and on the basis of that data we can built model for machine learning.

In [3]:
```
#lets input data
import pandas as pd
```

In [4]:
```
df=pd.read_csv(r"C:\Users\USER\Downloads\placement.csv")
```

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

Out[5]:

	cgpa	package
0	6.89	3.26
1	5.12	1.98
2	7.82	3.25
3	7.42	3.67
4	6.94	3.57

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   cgpa     200 non-null    float64
 1   package  200 non-null    float64
dtypes: float64(2)
memory usage: 3.3 KB
```

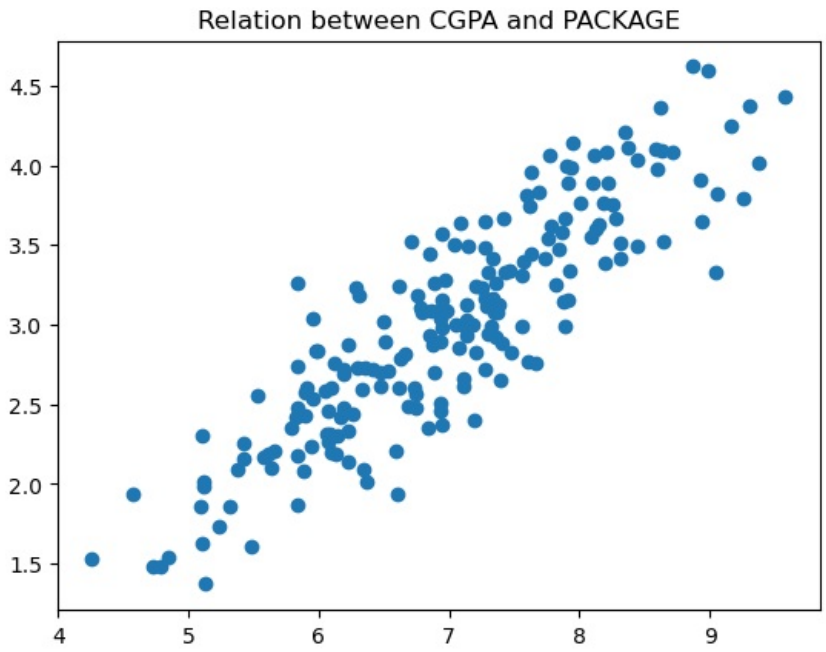
In [7]:
```
#So from info we can know we have 200 data in float data type.
```

if we see trying plotting that plot

In [8]:
```
import matplotlib.pyplot as plt
```

In [9]:
```
plt.scatter(df["cgpa"],df["package"])         #in scatter plot, we keep y first and x
plt.title("Relation between CGPA and PACKAGE")
```

Out[9]:
```
Text(0.5, 1.0, 'Relation between CGPA and PACKAGE')
```



So, is this graph exact linear? No it is sort of linear beacuse it is real world data set and in real world there may be various factor except this two dependent and independent variable like similar CGPA students, but one may get high salary due to external factor - like good interview, he/she has extra internship experience or good public relation with Company. So this type of factor will not give us exact linear, this is called stochastic error.

Building machine learning model:

lets divide data into training and test, training set is to train model about data and its pattern, test is for later, after training to check whether model is working properly or nor

In [10]:
```
#So, before building model, we should always know x and yo should be separate
x = df.iloc[:,0:1]
y = df.iloc[:,-1]
```

In [11]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [13]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,test_size=0.2,random_state=2)
model=LinearRegression()
model.fit(xtrain, ytrain)
```

Out[13]:
```
▾ LinearRegression
LinearRegression()
```

The specific value we use for random_state is arbitrary. The important aspect is that if we use the same random_state value across different runs of our program/data, we should get the same train/test split. This is useful for reproducibility.

In [14]:
```
#Checking accuracy of model
model.score(xtest, ytest)
```

Out[14]:
```
0.780730147510384
```

In [15]:
```
#first of all before predicting lets see our x test and y test data so that we can check and compare
```

In [16]:
```
print(xtest)
```

```
     cgpa
112  8.58
29   7.15
182  5.88
199  6.22
193  4.57
85   4.79
10   5.32
54   6.86
115  8.35
35   6.87
12   8.94
92   7.90
13   6.93
126  5.91
174  7.32
2    7.82
44   5.09
3    7.42
113  6.94
14   7.73
23   6.19
25   7.28
6    6.73
134  7.20
165  8.21
173  6.75
45   7.87
65   7.60
48   8.63
122  5.12
178  8.15
64   7.36
9    8.31
57   6.60
78   6.59
71   7.47
128  7.93
176  6.29
131  6.37
53   6.47
```

In [17]:
```
print(ytest)
```

```
112    4.10
29     3.49
182    2.08
199    2.33
193    1.94
85     1.48
10     1.86
54     3.09
115    4.21
35     2.87
12     3.65
92     4.00
13     2.89
126    2.60
174    2.99
2      3.25
44     1.86
3      3.67
113    2.37
14     3.42
23     2.48
25     3.65
6      2.60
134    2.83
165    4.08
173    2.56
45     3.58
65     3.81
48     4.09
122    2.01
178    3.63
64     2.92
9      3.51
57     1.94
78     2.21
71     3.34
128    3.34
176    3.23
131    2.01
53     2.61
Name: package, dtype: float64
```

In [18]:
```
#lets predict
predicted_y=model.predict([[2.61]])  #if cgpa is 2.61 what will be the package
```

```
C:\Users\USER\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names
, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [19]:
```
print(predicted_y)
```

```
[0.56014273]
```

So, according to this model if CGPA is 2.61 then he/she may earn 56000.