In [20]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
```

In [21]:

```
from sklearn.datasets import load_boston
boston = load_boston()
```

In [22]:

```
df = pd.DataFrame(boston.data)
```

In [23]:

```
df.head()
```

Out[23]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [24]:

```
df.tail()
```

Out[24]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99	9.67
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90	9.08
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90	5.64
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45	6.48
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90	7.88

In [25]:
```
df.columns = boston.feature_names
df
```

Out[25]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99	9.67
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90	9.08
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90	5.64
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45	6.48
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90	7.88

506 rows × 13 columns

In [26]:
```
df['Price'] = boston.target
```

```
In [27]: df
```

```
Out[27]:
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  PTRATIO       B  LSTAT  Price
  0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0     15.3  396.90   4.98   24.0
  1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0     17.8  396.90   9.14   21.6
  2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0     17.8  392.83   4.03   34.7
  3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0     18.7  394.63   2.94   33.4
  4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0     18.7  396.90   5.33   36.2
...      ...   ...    ...   ...    ...    ...   ...     ...  ...    ...      ...     ...    ...    ...
501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786  1.0  273.0     21.0  391.99   9.67   22.4
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875  1.0  273.0     21.0  396.90   9.08   20.6
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675  1.0  273.0     21.0  396.90   5.64   23.9
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889  1.0  273.0     21.0  393.45   6.48   22.0
505  0.04741   0.0  11.93   0.0  0.573  6.030  80.8  2.5050  1.0  273.0     21.0  396.90   7.88   11.9

506 rows × 14 columns
```

```
In [28]: df.shape
```

```
Out[28]: (506, 14)
```

In [29]:

```
df.isnull().sum()
```

Out[29]:

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
Price      0
dtype: int64
```

In [30]:

```
X = df.drop(['Price'],axis=1)
y = df['Price']
```

In [31]:

```
y
```

Out[31]:

```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: Price, Length: 506, dtype: float64
```

In [32]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=100)
```

In [33]:

```
from sklearn.linear_model import LinearRegression
Lm = LinearRegression()
```

In [34]:

```
Lm.fit(X_train,y_train)
```

Out[34]:

```
LinearRegression()
```

In [35]:

```
Lm.intercept_
```

Out[35]:

```
33.11584094298617
```

In [36]:

```
y_pred = Lm.predict(X_train)
```

In [37]:

```
y_pred
```

```
Out[37]: array([21.83658529, 23.49584711, 18.78472049, 29.44524553, 24.95757314,
                17.36071897, 20.61155508, 15.54124686, 19.07132899, 31.69059701,
                25.69522401, 13.65894201, 23.59091867, 25.18794008, 14.95408521,
                27.39240249, 16.61614088, 15.16322007, 27.98061764, 27.1112327 ,
                21.79599763, 33.68949582, 28.10912125, 24.32568146, 26.30449731,
                19.79043408, 31.29437017, 22.44797258, 30.29007294, 17.57026365,
                17.01183476, 28.54325066, 21.67624279, 15.1692288 , 20.84718467,
                27.81287241, 38.92371294, 22.48502759, 28.74634666, 17.78106461,
                 2.40187439, 18.48670856, 32.58187824, 13.13674448, 10.53072706,
                10.25303966, 32.72374458, 18.7658457 , 17.46652738, 11.21655555,
                22.59325322, 29.76647072, 23.16268617, 33.02605784, 19.50552824,
                17.54187526, 18.03746555, 13.42998591, 26.69421014, 34.4820332 ,
                19.49849579, 25.69564503, 12.76526966, 13.79755932, 28.62269021,
                18.5409617 , 19.794523  , 23.35714737, 22.16946357, 21.12912701,
                18.85855973, 30.06029071, 19.10203637, 24.85947948, 35.36656836,
                16.88301506, 19.77278234, 33.0791375 , 22.47918777, 12.04342801,
                24.33881115,  9.68255027, 23.54429911, 19.57435193, 10.59905211,
                15.78635035, 31.82868184, 23.69545656,  8.41811354, 30.88462655,
                42.91125504, 25.14257859, 24.98714041, 13.75495648, 36.46691223,
                19.5802772 , 25.85484879, 23.57531892, 34.54732106,  7.27290968,
                15.63953301, 29.65287007, 18.59888365, 21.5759628 , 25.14422204,
                42.1953393 , 32.58798343, 18.76704525, 16.54206146, 18.29257845,
                25.00401083, 30.50872864, 25.80602856, 29.94546009, 10.13448779,
                31.2022734 , 17.1458857 , 22.45583784, 21.90760212, 11.70882825,
                19.52455392, 20.21138462, 19.38004062, 27.56070101, 28.5876323 ,
                24.26709182, 27.70720743, 16.53804721, 16.33115465, 39.06369595,
                28.64808482, 21.72720782,  7.70689204, 14.94791322, 36.11288852,
                20.60170861, 23.77671754, 20.58395664, 20.1503321 , 21.2357496 ,
                25.58041974, 24.75909535,  7.20050143, 22.71051025,  8.34821794,
                 6.20034916, 19.87965392, 17.7328021 , 13.77154908, 24.02561592,
                32.69591234, 25.73788323, 30.05731407, 19.25992522, 25.14420308,
                21.6674345 , 18.14956443, 20.67630025, 31.09453437, 34.01680265,
                26.38987725, 21.72738177, 25.86509172, 14.22255826, 22.38485446,
                12.25570734, 25.33278764, 37.50915865, 15.85629076, 25.01764004,
                20.07787065, 13.6752788 ,  2.303429  , 19.64960412, 16.00615184,
                14.96593864, 20.09714444, 22.45735958, 31.29910935, 16.08187748,
                25.64107174, 11.69231836, 10.51325604, 13.05799691, 15.75630305,
                22.94969366, 22.04794215, 14.20948316, 10.77182695, 22.89389227,
                33.20203139, 37.22428666, 24.16009841, 14.28772338, 32.29220641,
                21.72726681, 23.22027063, 28.32835329, 22.2298012 , 16.47332068,
                29.99314606, 35.59876327, 14.12306879, 35.0276192 , 29.17153732,
                32.59457001, 21.15531601, 19.70156123, 22.17140396, 21.66887855,
                20.92405528, 33.63767972, 30.53062808, 28.61548906, 16.54854815,
                27.36391814, 20.8717546 , 20.39574862, 32.16604684, 31.75261746,
                20.57573971, 28.19798172, 35.72908118, 23.13081585, 20.76552116,
                21.96450185, 24.50280751, 40.6395194 , 34.3619416 , 19.31991453,
                30.0975294 , 34.91895215, 35.27910072, 23.42564435, 27.09816727,
                17.08541985, 24.2857531 , 36.53529683, 18.98252436, 25.19206453,
                26.82643274, 22.30421547, 20.76051001, 32.11434595, 24.58862356,
                28.22734896, 21.76920708, 22.07871059, 15.04834357, 26.32627635,
                13.25587055, 23.36813159, 19.95315846, 36.91796217, 22.43477952,
                31.42776554, 34.41589197, 25.80405626, 18.75030638, 13.54650107,
                24.60422425, 21.49351704, 18.33066272, 23.15551291, 13.70325311,
                25.25323387, 19.04974801, 17.61828927, 25.11541745, 19.60683801,
                29.68751886, 22.09450072, 17.18536326, 24.90376238, 17.69371019,
                21.15099166, 17.31126609, 12.35026389, 17.33418433, 19.70150605,
                24.48685234, 35.19339247, 36.33781918,  7.67470716, 32.21727259,
                35.43002957, 22.9309059 , 24.3159047 , 13.15436691, 22.54528424,
                32.36711495, 18.71539219, -2.1063874 , 30.79244452, 28.27768086,
                23.06691832, 43.89332253, 27.82213684, 22.80291431, 13.48422685,
                18.42218006, 20.27722872, 30.15895837, 20.73801452, 24.94964805,
                15.44845981, 30.33584504, 22.69988587, 18.8771101 , 31.84644066,
                17.94207875, 17.81438937, 17.2657805 , 27.1275138 , 18.24243795,
                27.13268404, 27.97233567, 24.16289494, 17.72097068, 13.85663836,
                 5.40782404,  9.25041549, 10.15148285,  7.49662992, 36.38969817,
                20.76102018,  7.29317077, 22.61083346, 28.607299  , 19.12742023,
                12.96093586, 21.93317192, 18.17344765, 20.02018248, 35.37530462,
                13.98215078, 25.73553681, 27.16812569, 32.63609052, 19.27567858,
                36.56111691, 25.1187651 , 24.35224039, 14.52798761, 14.40529562,
                20.41695883, 17.83684721, 20.26335084, 22.07988126, 27.18026529,
                19.09734323, 19.49422878, 38.03124219, 12.79480036])
```

In [38]:

```
print('R^2:',metrics.r2_score(y_train, y_pred))
```

```
R^2: 0.752890983596846
```

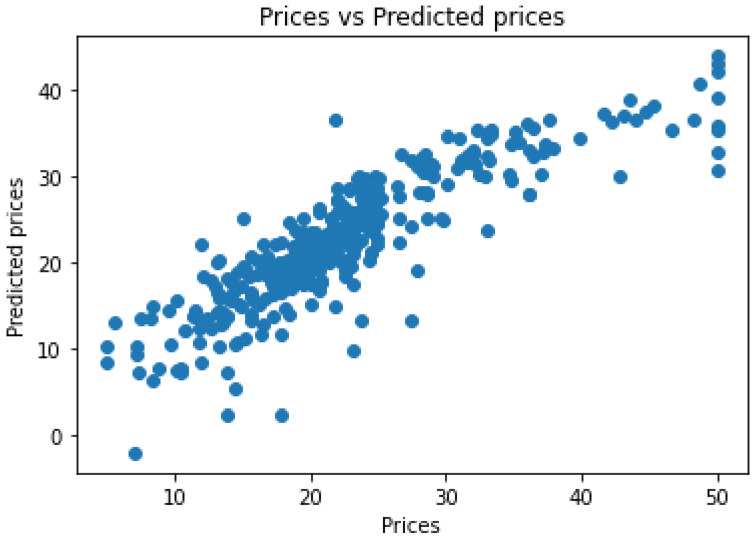
In [39]:

```
print('MAE:',metrics.mean_absolute_error(y_train, y_pred))
print('MSE:',metrics.mean_squared_error(y_train, y_pred))
```

```
MAE: 3.127349805330665
MSE: 19.067391155385046
```

In [40]:

```
plt.scatter(y_train, y_pred)
plt.xlabel("Prices")
plt.ylabel("Predicted prices")
plt.title("Prices vs Predicted prices")
plt.show()
```



In [ ]: