In [15]:

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

In [16]:

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

In [17]:

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

Out[17]:

	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 [18]:

```
data.columns = boston.feature_names
data.head()
```

Out[18]:

	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

In [19]:

```
data['MEDV'] = boston.target
data.head()
```

Out[19]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
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

In [20]:

```
data.shape
```

Out[20]:

```
(506, 14)
```

In [21]:

```
data.columns
```

Out[21]:

```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV'],
      dtype='object')
```

In [22]:

```
data[data.isnull().any(axis=1)]
```

Out[22]:

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV

In [23]:

```
X = data.drop(['MEDV'], axis = 1)
y = data['MEDV']
```

In [24]:

```
corr = data.corr()
corr.shape
```

Out[24]:

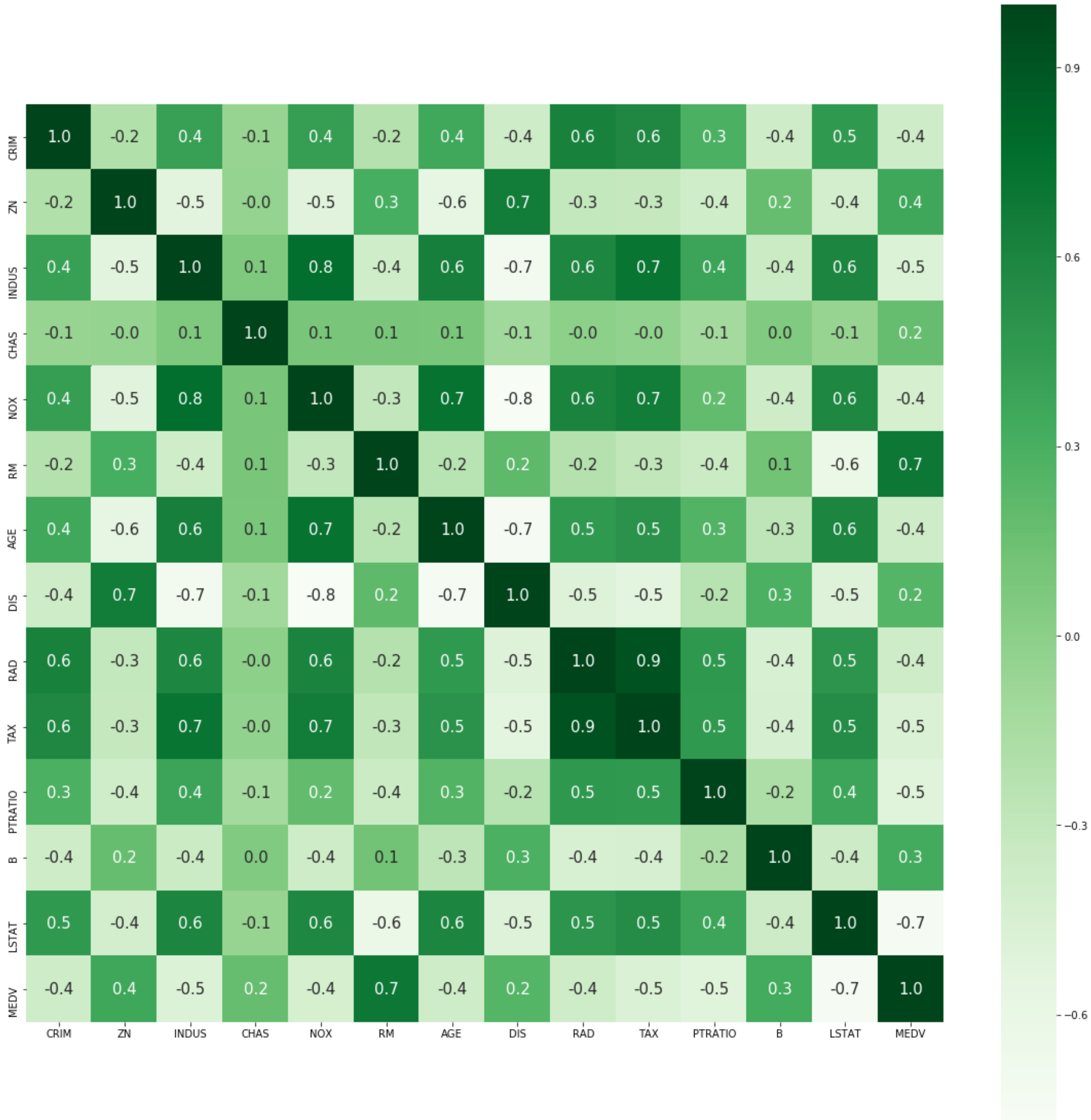
```
(14, 14)
```

In [25]:

```
plt.figure(figsize=(20,20))
sns.heatmap(corr, cbar=True, square= True, fmt='.1f', annot=True, annot_kws={'size':15}, cmap='Greens'
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x95a610>
```



In [26]:

```python
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 = 4)
print("X_train: ",X_train)
print("Y_train: ",y_train)
```

```
500    16.8
313    21.6
503    23.9
311    22.1
94     20.6
109    19.4
359    22.6
252    29.6
58     23.3
393    13.8
306    33.4
456    12.7
87     22.2
360    25.0
385     7.2
197    30.3
439    12.8
174    22.6
122    20.5
```

In [27]:

```python
#Import library for Linear Regression
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

Out[27]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [28]:

```
predictions = lm.predict(X_test)
print(predictions)
```

```
[11.07380893 26.47910329 17.34489869 19.1948608  36.36170735 24.77095832
 31.00051311 19.94226881 19.22375105 24.42998435 28.31512637 28.40796034
 19.27427968 33.82295207 21.28596487 15.11171444 20.97688767 11.28556596
 11.8611348  13.88444387  5.37422679 17.55278177 20.58171204 22.59849951
 16.07544265 20.45924503 19.1068775  14.37832191 21.23235601 17.52186564
 14.40725559 23.68483414 33.7410661  22.02733357 17.62139147 19.97241153
 30.24069397 34.69718954 23.85821534 24.30715093 36.13378112 31.97532293
 19.626175   31.61097971 34.58127809 25.62718797 39.95041812 17.60880538
 19.90319708 23.40417501 33.70182396 25.62491083 18.25559302 27.27317174
 13.46377871 23.43470656 24.43721849 33.52056736 16.99896935 37.94464404
 15.94567818 19.32528916 31.84088262 15.25081303 38.40344789 27.45372884
 34.36154312  9.37353936 19.42580066 21.99218459 22.79983394 22.50810313
 22.30918714 27.84395887 16.40818345 22.55507669 16.5147332  25.11106836
 13.76991927 19.78656399 22.10247463 20.26663237 28.15165586 19.52050766
 30.33254364 22.79109999 29.2663436  19.43113706 24.7968264  37.46275648
 31.05503576 41.3372879  18.46365381 36.67964528 19.40842405 23.61810063
 27.93475362 24.41825213  9.4599059  20.68088677  8.99426788 28.4492398
 31.88237066 14.04302958 24.8347909  19.94124425 36.90271393 31.06556982
 33.91883403 28.64591536 31.1007263  22.82363163 11.58125942 29.46902405
 37.06066106 23.01945872 41.79865192 18.44334162  3.433324   18.57485663
 22.21257489 16.71192648 28.00473344 28.42374739 19.6417452  18.76090758
 35.37631447 13.12349548 14.73923539 18.16202333 38.26604753 15.97821613
 41.91544265 30.44631625 28.65848089 24.19590457 12.06559683 26.01408744
 23.25012698 18.92506857 17.05016777 17.50245392 20.89247338 24.62630514
  1.82167558 23.03969555 19.35693345 17.89193065 38.43943954 19.7075262
 31.67181183 19.0130913 ]
```

In [29]:

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

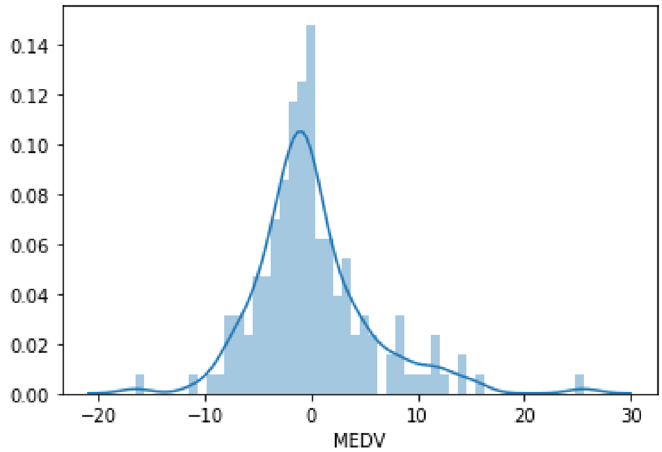
In [30]:

```
from sklearn import metrics
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
RMSE: 5.482152251362979
```

In [31]:

```
sns.distplot((y_test-predictions),bins=50);
```



In [32]:

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