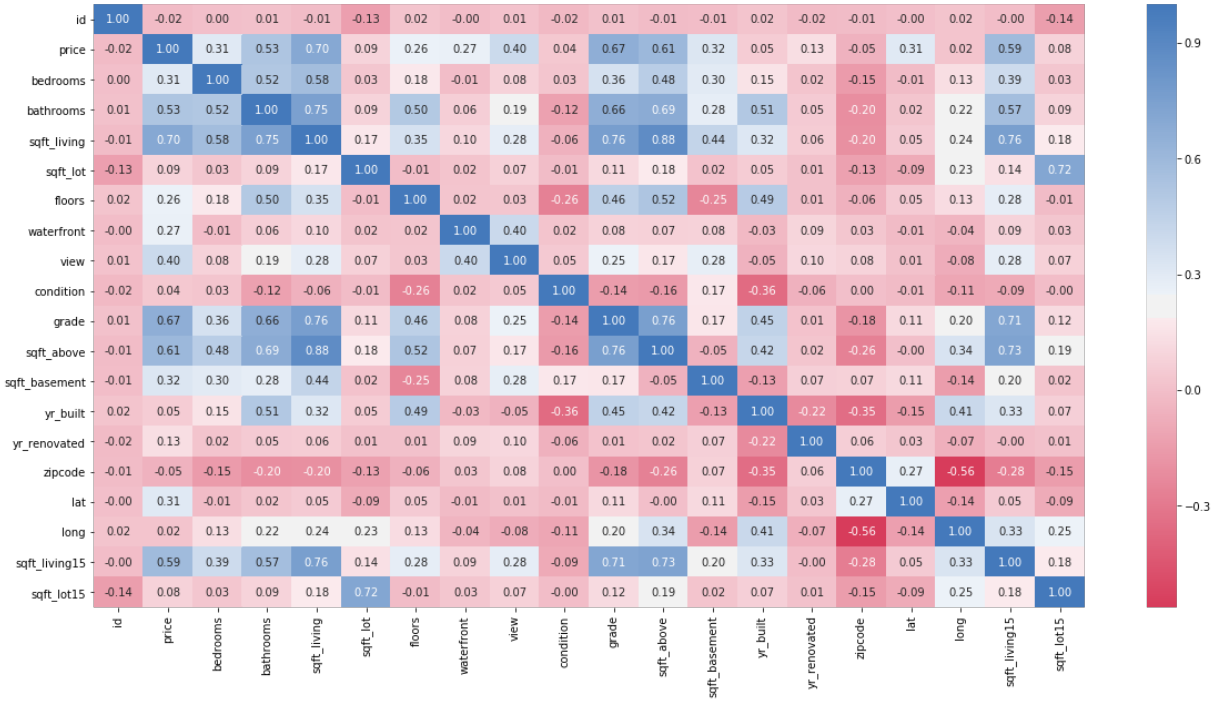
In [2]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.neighbors import KNeighborsRegressor
import matplotlib.pyplot as plt
from sklearn import metrics
import seaborn as sns

df = pd.read_csv("kc_house_data.csv")

# Plotting correlation heatmap to get the factors highly correlated
correlation = df.corr()
fig, ax = plt.subplots(figsize=(20, 10))
cm = sns.diverging_palette(5, 250, as_cmap=True)
sns.heatmap(correlation, cmap=cm, annot=True, fmt=".2f")
plt.show()
plt.show()
```



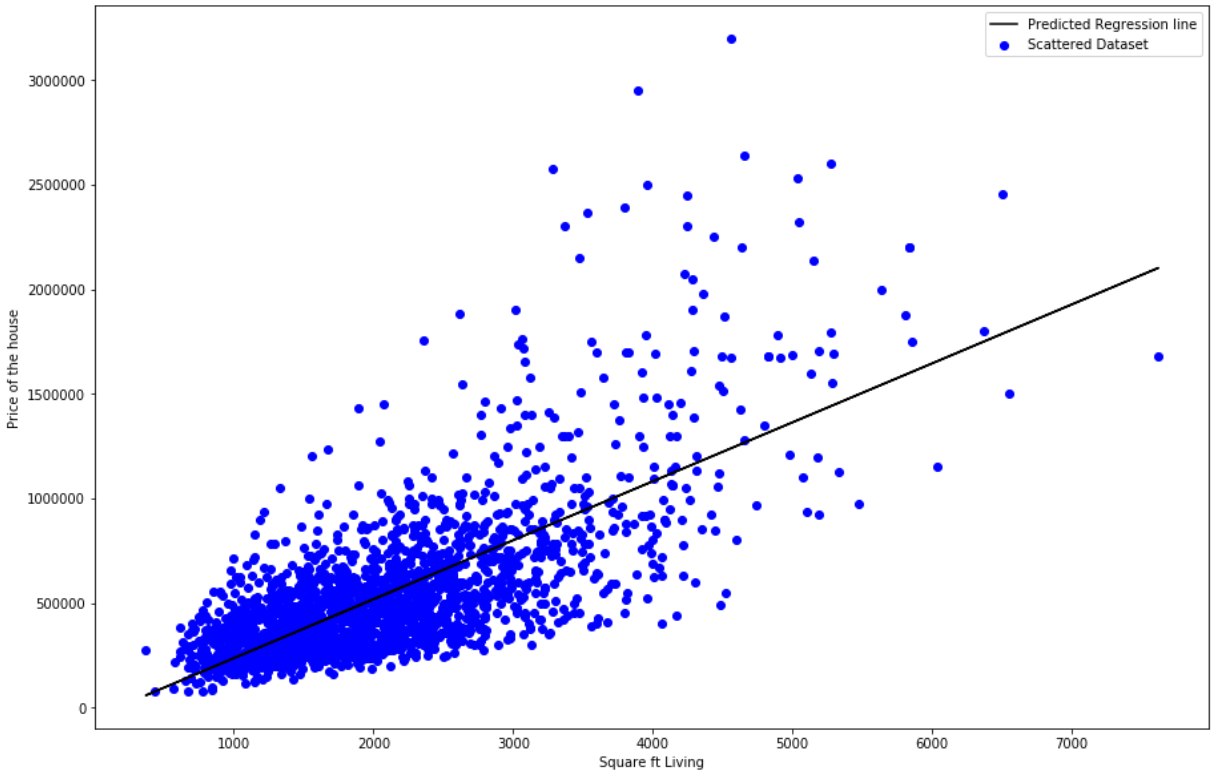
In [3]:
```
# Linear regression using sqft_living as X

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn import metrics

df = pd.read_csv("kc_house_data.csv")

rf = df.drop(['id','date','price'],1)
```

```
13
14 #  From the heat map we identified that sqft_living and grade have
15 # So we did two models
16 # -> Using all attributes against price
17 # -> Using sqft_living and grade attributes against price
18
19
20 column_selected = ['sqft_living']
21 predicted_array = []
22
23 train, test = train_test_split(df, train_size =0.90, random_state =
24
25 lm = linear_model.LinearRegression()
26
27 X_train = np.array(train[column_selected]).reshape(-1,1)
28 X_test = np.array(test[column_selected]).reshape(-1, 1)
29
30 Y_train = np.array(train['price']).reshape(-1, 1)
31 Y_test = np.array(test['price']).reshape(-1, 1)
32
33 lm.fit(train[column_selected], train['price'])
34
35 prediction = lm.predict(test[column_selected])
36
37 print("Model using {} as X" .format(column_selected))
38 mse = metrics.mean_squared_error(Y_test, prediction)
39 error = np.sqrt(mse)
40 intercept = lm.intercept_
41 accuracy = lm.score(test[column_selected], test['price'])
42
43 print("\nThe root mean squared error is ", np.round(error, 2))
44 print("\nThe coefficient array is ", np.round(lm.coef_,2))
45 print("\nThe intercept is ", np.round(intercept,2))
46 print("\nThe accuracy is given by is ", round(accuracy, 2))
47
48 fig, ax = plt.subplots(figsize= (15, 10))
49 plt.scatter(X_test, Y_test, color= 'blue', label = 'Scattered Datas
50 plt.plot(X_test, lm.predict(X_test), color='black', label= 'Predict
51 plt.xlabel('Square ft Living')
52 plt.ylabel('Price of the house')
53 plt.legend()
54
55
56
57
58
59
```

```
Model using ['sqft_living'] as X

The root mean squared error is  249846.53

The coefficient array is  [282.03]
```

```
The intercept is  -46861.8

The accuracy is given by is  0.49

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_spli
t.py:2026: FutureWarning: From version 0.21, test_size will always c
omplement train_size unless both are specified.
  FutureWarning)
```

Out[3]: <matplotlib.legend.Legend at 0x1a2369f7f0>

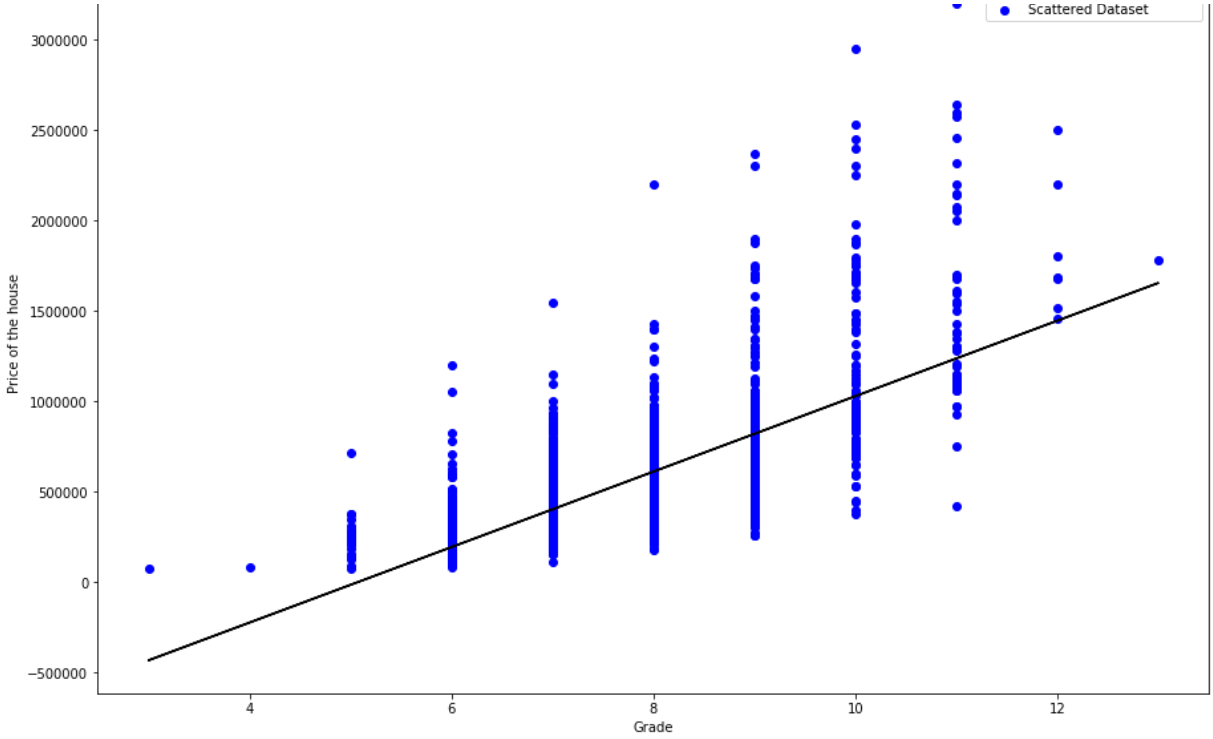


In [5]:

```
# Linear regression using sqft_living as X

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn import metrics
import seaborn as sns


df = pd.read_csv("kc_house_data.csv")

rf = df.drop(['id','date','price'],1)


column_selected = ['grade']
predicted_array = []

train, test = train_test_split(df, train_size =0.90, random_state =
```

```
lm = linear_model.LinearRegression()

X_train = np.array(train[column_selected]).reshape(-1,1)
X_test = np.array(test[column_selected]).reshape(-1, 1)

Y_train = np.array(train['price']).reshape(-1, 1)
Y_test = np.array(test['price']).reshape(-1, 1)

lm.fit(train[column_selected], train['price'])

prediction = lm.predict(test[column_selected])

print("Model using {} as X" .format(column_selected))
mse = metrics.mean_squared_error(Y_test, prediction)
error = np.sqrt(mse)
intercept = lm.intercept_
accuracy = lm.score(test[column_selected], test['price'])

print("\nThe root mean squared error is ", np.round(error, 2))
print("\nThe coefficient array is ", np.round(lm.coef_,2))
print("\nThe intercept is ", np.round(intercept,2))
print("\nThe accuracy is given by is ", round(accuracy, 2))


fig, ax = plt.subplots(figsize= (15, 10))
plt.scatter(X_test, Y_test, color= 'blue', label = 'Scattered Datas
plt.plot(X_test, lm.predict(X_test), color='black', label= 'Predict
plt.xlabel('Grade')
plt.ylabel('Price of the house')
plt.legend()
plt.show()
```

```
/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_spli
t.py:2026: FutureWarning: From version 0.21, test_size will always c
omplement train_size unless both are specified.
  FutureWarning)

Model using ['grade'] as X

The root mean squared error is  252130.8

The coefficient array is  [208591.32]

The intercept is  -1057294.06

The accuracy is given by is  0.48
```



In [ ]:
```
1
```

In [6]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.neighbors import KNeighborsRegressor
import matplotlib.pyplot as plt

df = pd.read_csv("kc_house_data.csv")

nf = df
date_string = nf['date'].tolist()
date_int = [int(str[:4]) for str in date_string]
df['year'] = date_int

rf = df.drop(['id','date','price', 'zipcode'],1)

#  From the heat map we identified that sqft_living and grade have
# So we did two models
# -> Using all attributes against price
# -> Using sqft_living and grade attributes against price

column_selected = list(rf.columns.values)

predicted_array = []

train, test = train_test_split(df, train_size =0.90, random_state =

lm = linear_model.LinearRegression()

X_train = np.array(train[column_selected]).reshape(-1,1)
```

```
X_test = np.array(test[column_selected]).reshape(-1, 1)

Y_train = np.array(train['price']).reshape(-1, 1)
Y_test = np.array(test['price']).reshape(-1, 1)

lm.fit(train[column_selected], train['price'])

prediction = lm.predict(test[column_selected])

print("Model using {} as X" .format(column_selected))
mse = metrics.mean_squared_error(Y_test, prediction)
error = np.sqrt(mse)
intercept = lm.intercept_
accuracy = lm.score(test[column_selected], test['price'])

print("\nThe root mean squared error is ", np.round(error, 2))
print("\nThe coefficient array is ", np.round(lm.coef_,2))
print("\nThe intercept is ", np.round(intercept,2))
print("\nThe accuracy is given by is ", round(accuracy, 2))

```

```
Model using ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'fl
oors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sq
ft_basement', 'yr_built', 'yr_renovated', 'lat', 'long', 'sqft_livin
g15', 'sqft_lot15', 'year'] as X

The root mean squared error is  193425.01

The coefficient array is  [-3.3978050e+04  4.4056360e+04  1.0924000e
+02  1.3000000e-01
  9.5241000e+02  6.0859354e+05  4.9747160e+04  3.2698890e+04
  9.6312430e+04  7.1300000e+01  3.7940000e+01 -2.4791000e+03
  2.3030000e+01  5.6349587e+05 -1.1762568e+05  2.7110000e+01
 -3.9000000e-01  2.9429070e+04]

The intercept is  -96292145.69

The accuracy is given by is  0.7

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_spli
t.py:2026: FutureWarning: From version 0.21, test_size will always c
omplement train_size unless both are specified.
  FutureWarning)
```

In [ ]:

```
1
```