In [4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

dataset = pd.read_excel("HousePricePrediction.xlsx")
```

In [5]:

```
dataset
```

Out[5]:

	Id	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	YearBuilt	YearR
0	0	60	RL	8450	Inside	1Fam	5	2003	
1	1	20	RL	9600	FR2	1Fam	8	1976	
2	2	60	RL	11250	Inside	1Fam	5	2001	
3	3	70	RL	9550	Corner	1Fam	5	1915	
4	4	60	RL	14260	FR2	1Fam	5	2000	
...	...	...	...	...	...	...	...	...	
2914	2914	160	RM	1936	Inside	Twnhs	7	1970	
2915	2915	160	RM	1894	Inside	TwnhsE	5	1970	
2916	2916	20	RL	20000	Inside	1Fam	7	1960	
2917	2917	85	RL	10441	Inside	1Fam	5	1992	
2918	2918	60	RL	9627	Inside	1Fam	5	1993	

2919 rows × 13 columns

In [6]:

```
dataset.shape
```

Out[6]:

```
(2919, 13)
```

In [7]:

```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:",len(object_cols))

int_ = (dataset.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:",len(num_cols))

fl = (dataset.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:",len(fl_cols))
```

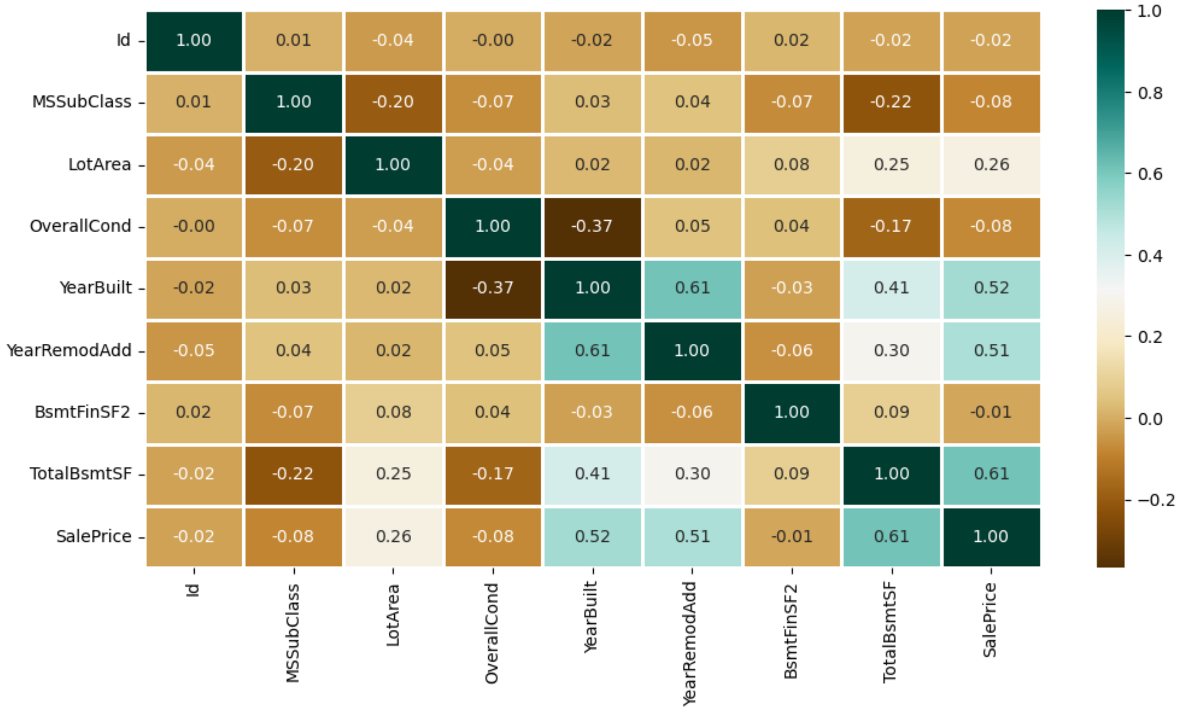
```
Categorical variables: 4
Integer variables: 0
Float variables: 3
```

In [8]:

```
plt.figure(figsize=(12, 6))
sns.heatmap(dataset.corr(),
            cmap = 'BrBG',
            fmt = '.2f',
            linewidths = 2,
            annot = True)
```

Out[8]:

```
<AxesSubplot:>
```



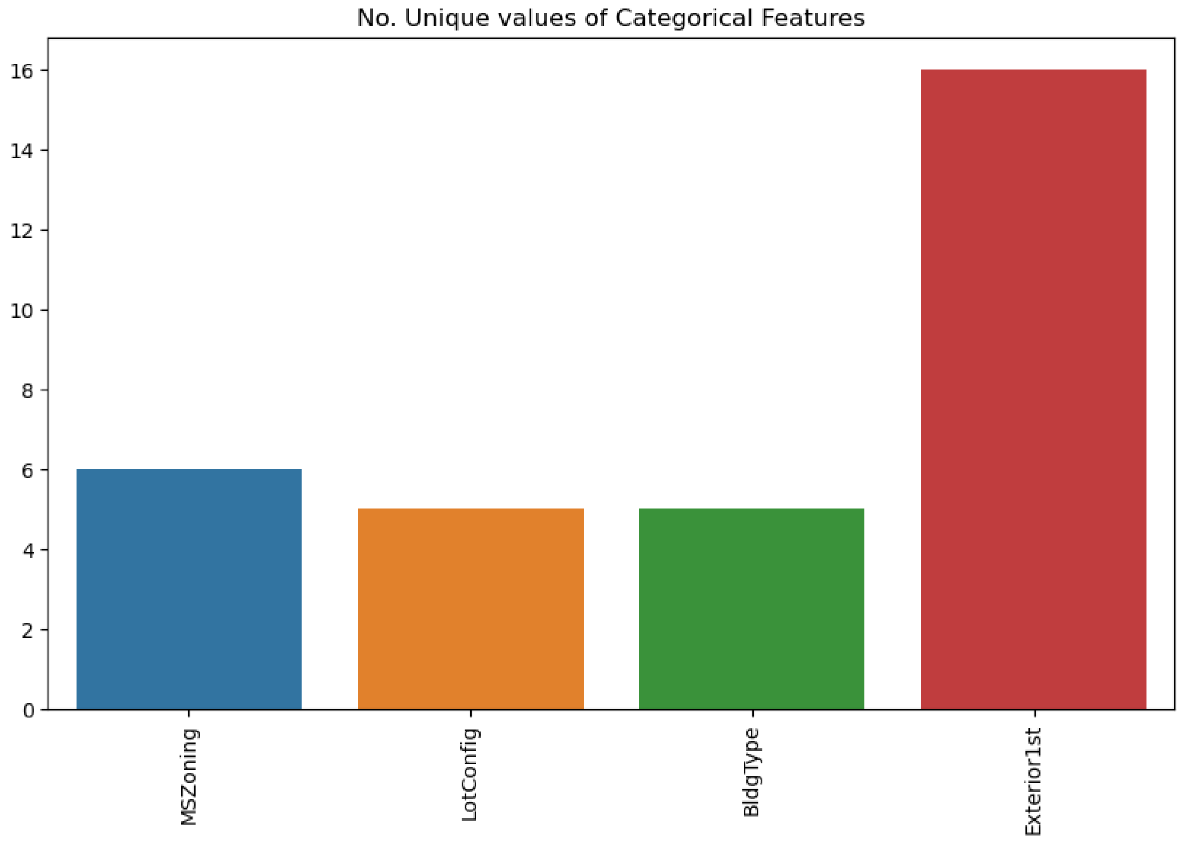
	Id	MSSubClass	LotArea	OverallCond	YearBuilt	YearRemodAdd	BsmtFinSF2	TotalBsmtSF	SalePrice
Id	1.00	0.01	-0.04	-0.00	-0.02	-0.05	0.02	-0.02	-0.02
MSSubClass	0.01	1.00	-0.20	-0.07	0.03	0.04	-0.07	-0.22	-0.08
LotArea	-0.04	-0.20	1.00	-0.04	0.02	0.02	0.08	0.25	0.26
OverallCond	-0.00	-0.07	-0.04	1.00	-0.37	0.05	0.04	-0.17	-0.08
YearBuilt	-0.02	0.03	0.02	-0.37	1.00	0.61	-0.03	0.41	0.52
YearRemodAdd	-0.05	0.04	0.02	0.05	0.61	1.00	-0.06	0.30	0.51
BsmtFinSF2	0.02	-0.07	0.08	0.04	-0.03	-0.06	1.00	0.09	-0.01
TotalBsmtSF	-0.02	-0.22	0.25	-0.17	0.41	0.30	0.09	1.00	0.61
SalePrice	-0.02	-0.08	0.26	-0.08	0.52	0.51	-0.01	0.61	1.00

In [9]:

```
unique_values = []
for col in object_cols:
  unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
plt.title('No. Unique values of Categorical Features')
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=unique_values)
```

Out[9]:

```
<AxesSubplot:title={'center':'No. Unique values of Categorical Features'}>
```

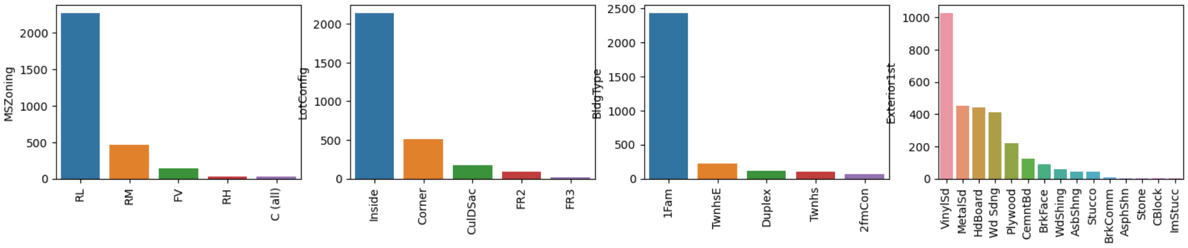


In [10]:

```
plt.figure(figsize=(18, 36))
plt.title('Categorical Features: Distribution')
plt.xticks(rotation=90)
```

```
index = 1

for col in object_cols:
    y = dataset[col].value_counts()
    plt.subplot(11, 4, index)
    plt.xticks(rotation=90)
    sns.barplot(x=list(y.index), y=y)
    index += 1
```



In [11]:
```
dataset.drop(['Id'],
             axis=1,
             inplace=True)
```

In [12]:
```
dataset['SalePrice'] = dataset['SalePrice'].fillna(
  dataset['SalePrice'].mean())
```

In [13]:
```
new_dataset = dataset.dropna()
new_dataset
```

Out[13]:

	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	YearBuilt	YearRemodA
0	60	RL	8450	Inside	1Fam	5	2003	20
1	20	RL	9600	FR2	1Fam	8	1976	19
2	60	RL	11250	Inside	1Fam	5	2001	20
3	70	RL	9550	Corner	1Fam	5	1915	19
4	60	RL	14260	FR2	1Fam	5	2000	20
...	...	...	...	...	...	...	...	
2914	160	RM	1936	Inside	Twnhs	7	1970	19
2915	160	RM	1894	Inside	TwnhsE	5	1970	19
2916	20	RL	20000	Inside	1Fam	7	1960	19
2917	85	RL	10441	Inside	1Fam	5	1992	19
2918	60	RL	9627	Inside	1Fam	5	1993	19

2913 rows × 12 columns

In [19]:
```
new_dataset.isnull().sum()
```

Out[19]:

```
MSSubClass      0
MSZoning        0
LotArea         0
LotConfig       0
BldgType        0
OverallCond     0
YearBuilt       0
YearRemodAdd    0
Exterior1st     0
BsmtFinSF2      0
TotalBsmtSF     0
SalePrice       0
dtype: int64
```

In [22]:

```
from sklearn.preprocessing import OneHotEncoder

s = (new_dataset.dtypes == 'object')
object_cols = list(s[s].index)
print("Categorical variables:")
print(object_cols)
print('No. of. categorical features: ',
      len(object_cols))
```

```
Categorical variables:
['MSZoning', 'LotConfig', 'BldgType', 'Exterior1st']
No. of. categorical features:  4
```

In [ ]:

```
OH_encoder = OneHotEncoder(sparse=False)
OH_cols = pd.DataFrame(OH_encoder.fit_transform(new_dataset[object_cols]))
OH_cols.index = new_dataset.index
OH_cols.columns = OH_encoder.get_feature_names()
df_final = new_dataset.drop(object_cols, axis=1)
df_final = pd.concat([df_final, OH_cols], axis=1)
```

In [24]:

```
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split

X = df_final.drop(['SalePrice'], axis=1)
Y = df_final['SalePrice']

# Split the training set into
# training and validation set
X_train, X_valid, Y_train, Y_valid = train_test_split(
    X, Y, train_size=0.8, test_size=0.2, random_state=0)
```

In [25]:

```
from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import mean_absolute_percentage_error

model_SVR = svm.SVR()
model_SVR.fit(X_train,Y_train)
Y_pred = model_SVR.predict(X_valid)

print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.1870512931870423
```

In [26]:

```
from sklearn.ensemble import RandomForestRegressor

model_RFR = RandomForestRegressor(n_estimators=10)
model_RFR.fit(X_train, Y_train)
Y_pred = model_RFR.predict(X_valid)
```

```
print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.19824961872591532
```

In [27]:

```
from sklearn.linear_model import LinearRegression

model_LR = LinearRegression()
model_LR.fit(X_train, Y_train)
Y_pred = model_LR.predict(X_valid)

print(mean_absolute_percentage_error(Y_valid, Y_pred))
```

```
0.1874168384159999
```