In [5]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

Uploading the dataset

In [9]:
```
data = pd.read_csv("StudentsPerformance.csv")
```

In [10]:
```
data
```

Out[10]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

1. Understanding the data

In [11]:
```
data.head()
```

Out[11]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [13]:
```
data.tail()
```

Out[13]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

In [14]:
```
data.shape
```

Out[14]:
```
(1000, 8)
```

In [15]:
```
data.describe()
# only for integer values
```

Out[15]:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [16]:
```
data.columns
```

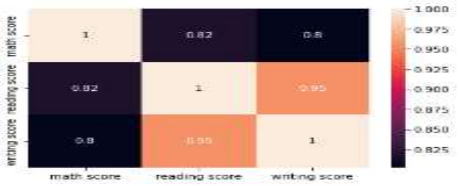
Out[16]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

3. Relationship analysis

In [26]:
```
# using correlation between variables
corelation = student.corr()
```

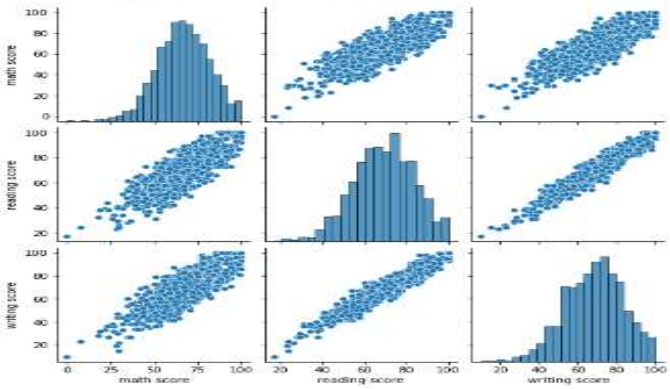
In [27]:
```
sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True)
```

Out[27]: <AxesSubplot:>



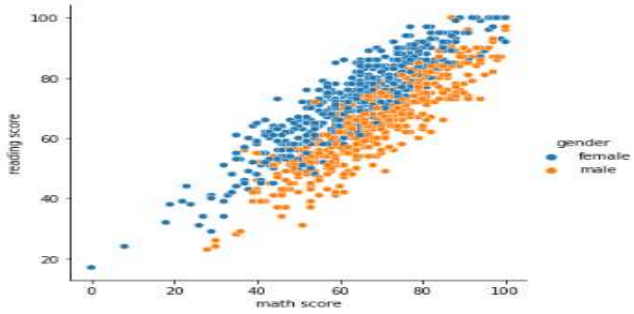
In [28]:
```
sns.pairplot(student)
# sued to view realtionship between any two variables : continous, categorical, boolean
```

Out[28]: <seaborn.axisgrid.PairGrid at 0x11f9c9053d0>



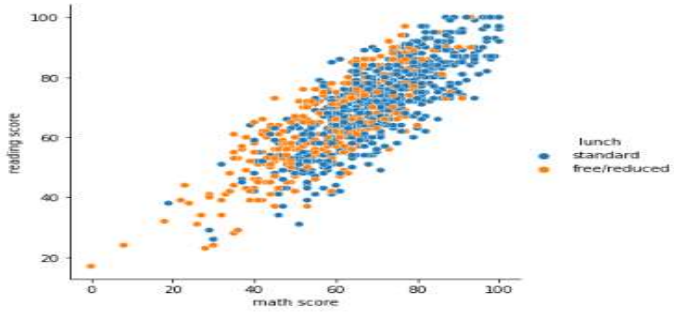
In [ ]:
```
# use scatter plot to see relationship between two numerical variables
# use relation plot
sns.relplot(x = 'math score', y = 'reading score', hue = 'gender', data = student)
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x11f9d90b4c0>



[ ]:
```
sns.relplot(x = 'math score', y = 'reading score', hue = 'lunch', data = student)
```

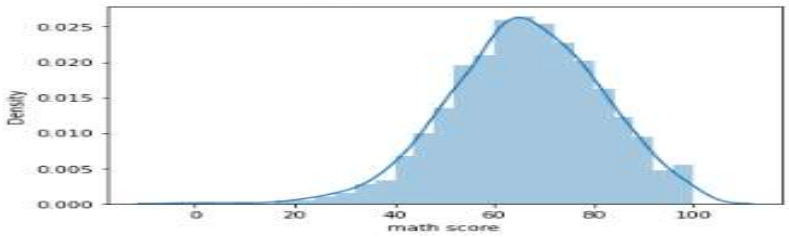
[ ]: <seaborn.axisgrid.FacetGrid at 0x11f9d919d60>



```
# using histograms
sns.distplot(student['math score'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions
d will be removed in a future version. Please adapt your code to
xibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

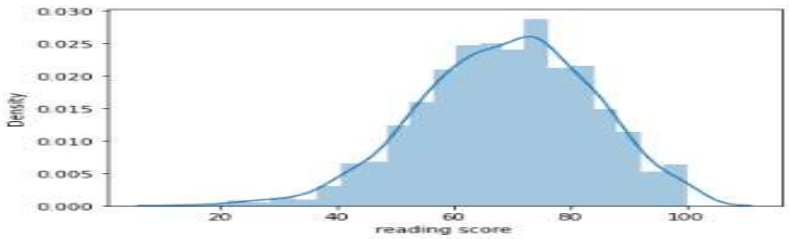
```
<AxesSubplot:xlabel='math score', ylabel='Density'>
```



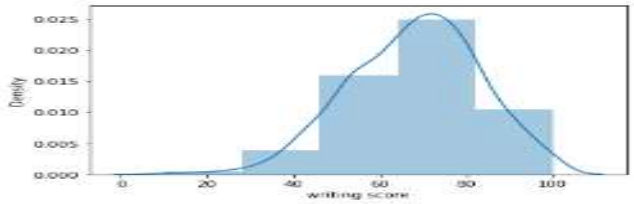

```
sns.distplot(student['reading score'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions
d will be removed in a future version. Please adapt your code to
xibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='reading score', ylabel='Density'>
```

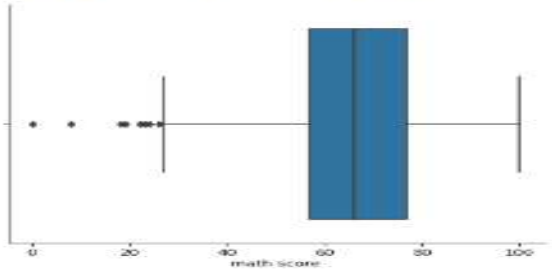


Out[33]:
```
<AxesSubplot:xlabel='writing score', ylabel='Density'>
```



In [34]:
```
# categorical plot
sns.catplot(x = 'math score', kind= 'box', data = student)
```

Out[34]:
```
<seaborn.axisgrid.FacetGrid at 0x11f9f03cd30>
```



In [45]:
```
sns.catplot(x = 'writing score', kind= 'box', data = student)
```

Out[45]:
```
<seaborn.axisgrid.FacetGrid at 0x7fd6816cabb0>
```