LIST FUNCTIONS


```
transactions = [10.44, 20.56, 200.14, 1242.66, 2.07, 8.01]
```

len(list_name) *Returns the number of elements in a list*

```
len(transactions)
```

```
6
```

sum(list_name) *Returns the sum of the elements in the list*

```
sum(transactions)
```

```
1483.88
```

min(list_name) *Returns the smallest element in the list*

```
min(transactions)
```

```
2.07
```

max(list_name) *Returns the largest element in the list*

```
max(transactions)
```

```
1242.66
```

SORTING LISTS

There are two ways to **sort** lists elements:

1. The **.sort()** method sorts the list permanently *(in place)*

```
transactions = [10.44, 20.56, 200.14, 1242.66, 2.07, 8.01]

transactions.sort()
transactions
```

```
[2.07, 8.01, 10.44, 20.56, 200.14, 1242.66]
```

PRO TIP: Don't sort in place until you're positive the code works as expected and you no longer need to preserve the original list

2. The **sorted** function returns a sorted list, but does not change the original *(not in place)*

```
transactions = [10.44, 20.56, 200.14, 1242.66, 2.07, 8.01]
sorted(transactions)
```

```
[2.07, 8.01, 10.44, 20.56, 200.14, 1242.66]
```

```
transactions
```

```
[10.44, 20.56, 200.14, 1242.66, 2.07, 8.01]
```

LIST METHODS


```
transactions = [10.44, 20.56, 200.14, 1242.66, 2.07, 8.01]
```

.index(*value*)

Returns the index of a specified value within a list (returns -1 if not found)

```
transactions.index(200.14)
```

```
2
```

.count()

Counts the number of times a given value occurs in a list

```
transactions.count(200.14)
```

```
1
```

.reverse()

Reverses the order of the list elements in place

```
transactions.reverse()
transactions
```

```
[8.01, 2.07, 1242.66, 200.14, 20.56, 10.44]
```

PRO TIP: Use a negative slice to reverse the order "not in place"