List

Permutations Returns a list of all possible combinations for a list of items.

```
List<int> list = new List<int> { 1, 2, 3 };
List<List<int>> listOfCombinations = list.Permutations();
```

Index	Value
0	1
1	2
2	1, 2
3	3
4	1, 3
5	2, 3
6	1, 2, 3

Pop Removes the last item from a list and returns that item.

```
List<float> list = new List<float>() { 1, 2, 3, 4, 5 };
float lastItemInList = list.Pop();
```

Random Returns a random item from a List.

```
List<float> list = new List<float>() { 1, 2, 3, 4, 5 };
float randomItemFromList = list.Random();
```

Shift Removes the first item from a list and returns that item.

```
List<float> list = new List<float>() { 1, 2, 3, 4, 5 };
float firstItemInList = list.Shift();
```

Shuffle Creates a new copy of a list and shuffles the values.

```
List<float> list = new List<float>();
List<float> shuffledList = list.Shuffle();
```

Shuffle with a specific seed.

```
List<float> list = new List<float>();
List<float> shuffledList = list.Shuffle(10);
```

Slice Returns a shallow copy of a portion of a list.

```
List<float> list = new List<float>();
List<float> specificItems = list.Slice(0, 1);
```

Splice Removes and returns a shallow copy of a portion of a list.

```
List<float> list = new List<float>();
List<float> removedItems = list.Splice(0, 1);
```

Unshift Adds a range of items to the beginning of a List.

```
List<int> list = new List<int> { 1, 2, 3, 4, 5 };

list.Unshift(new List<int> { -1, 0 });
```

```
List<int> list = new List<int> { 1, 2, 3, 4, 5 };

list.Unshift(new ObservableList<int> { -1, 0 });
```

Adds an item to the beginning of a List.

```
List<int> list = new List<int> { 1, 2, 3, 4, 5 };

list.Unshift(0);
```