

0 references

```
public class TestScript : MonoBehaviour
```

```
{
```

1 reference

```
public string[] arrayItems;
```

Array  
Declaration

0 references

```
void Start()
```

```
{
```

```
    for(int i=0; i <= 4; i++)
```

```
    {
```

```
        print("Array Item: " + arrayItems[i]);
```

```
    }
```

```
}
```

Printing  
Array  
Items

```
// Update is called once per frame
```

0 references

```
void Update()
```

```
{
```

```
}
```

```
}
```

List

```

public List<string> vegetableNames;

0 references
void Start()
{
    vegetableNames.Add("Tomato");
    vegetableNames.Add("Potato");
    vegetableNames.Add("Onion");
}

0 references
void Update()
{
    //Press 'P' to call mom to change the list item
    //remove Onion and add Lady Finger
    if(Input.GetKeyDown(KeyCode.P))
    {
        print("Calling mom");
        vegetableNames.Remove("Onion");
        vegetableNames.Add("Lady Finger");
    }
}

```

**List Declaration**

**Adding Items to List**

**Removing List Item**

The basic difference between array and list is, in list we can remove and add another item in a list dynamically based upon the desired condition.

## What is an Array?

- A grouping of multiple variables of the same type
- Each item stored in an array is called an 'element'
- Each element can be accessed by its index number
- Counting starts at zero!

```

int[] oddNumbers = {1, 3, 5, 7, 9}

```

**Index numbers**

↑  
[ ] = array

↑ ↑ ↑ ↑ ↑  
0 1 2 3 4

```
int[] oddNumbers = new int[5]
oddNumbers[2] = 5;
```

```
public class QuestionSO : ScriptableObject
{
    1 reference
    [TextArea(2,6)] [SerializeField] string question = "Enter question text here";
    1 reference
    [SerializeField] string[] answers = new string[4];    // Array declaration
    1 reference
    [SerializeField] int correctanswerIndex;

    0 references
    public string GetQuestion()
    {
        return question;
    }

    0 references
    public int GetCorrectAnswerIndex()
    {
        return correctanswerIndex;
    }

    0 references
    public string GetAnswerIndex(int index)
    {
        return answers[index];
    }
}
```

will take question text

will provide correct answer index as entered by the designer in inspector panel

Will get the answer index and it's index number on clicking onto the answers while playing

## What is a List?

- They're kind of like arrays!
- A grouping of multiple variables of the same type
- Each item stored in an List is called an 'element'
- Each element can be accessed by its index number
- Counting starts at zero!
- They're mutable - meaning we can change their size!

# Syntax

## Array

```
Int[] oddNumbers = new int[5]
```

## List

```
List<int> oddNumbers = new List<int>()
```

# Useful Methods & Properties

Check item count:	<code>List.Count</code>
Check if item exists:	<code>List.Contains(3)</code>
Add an item:	<code>List.Add(3)</code>
Remove an item:	<code>List.Remove(3)</code>
Remove item at index:	<code>List.RemoveAt(0)</code>
Clear the list:	<code>List.Clear()</code>