CREATE LISTS

var names = new List<string> { "<name>", "Ana", "Felipe" };

foreach (var name in names)

{

Console.WriteLine($"Hello {name.ToUpper()}!");

}

Output:

Hello <NAME>!

Hello ANA!

Hello FELIPE!

MODIFY LIST CONTENTS:

var names = new List<string> { "<name>", "Ana", "Felipe" };

foreach (var name in names)

{

Console.WriteLine($"Hello {name.ToUpper()}!");

}

Console.WriteLine();

names.Add("Maria");

names.Add("Bill");

names.Remove("Ana");

foreach (var name in names)

{

Console.WriteLine($"Hello {name.ToUpper()}!");

}

OUTPUT:

Hello <NAME>!

Hello FELIPE!

Hello MARIA!

Hello BILL!

Console.WriteLine($"My name is {names[0]}.");

Console.WriteLine($"I've added {names[2]} and {names[3]} to the list.");

OUTPUT:

My name is <name>.

I've added Maria and Bill to the list.

Console.WriteLine($"The list has {names.Count} people in it");

OUTPUT:

The list has 4 people in it

SEARCH AND SORT LISTS

var index = names.IndexOf("Felipe");

if (index != -1)

Console.WriteLine($"The name {names[index]} is at index {index}");

var notFound = names.IndexOf("Not Found");

Console.WriteLine($"When an item is not found, IndexOf returns {notFound}");

OUTPUT:

The name Felipe is at index 1

When an item is not found, IndexOf returns -1

names.Sort();

foreach (var name in names)

{

Console.WriteLine($"Hello {name.ToUpper()}!");

}

OUTPUT:

Hello <NAME>!

Hello BILL!

Hello FELIPE!

Hello MARIA!

LIST OF OTHER TYPES:

var fibonacciNumbers = new List<int> {1, 1};

var previous = fibonacciNumbers[fibonacciNumbers.Count - 1];

var previous2 = fibonacciNumbers[fibonacciNumbers.Count - 2];

fibonacciNumbers.Add(previous + previous2);

foreach(var item in fibonacciNumbers)

Console.WriteLine(item);

OUTPUT:

1

1

2

var fibonacciNumbers = new List<int> {1, 1};

while (fibonacciNumbers.Count < 20)

{

var previous = fibonacciNumbers[fibonacciNumbers.Count - 1];

var previous2 = fibonacciNumbers[fibonacciNumbers.Count - 2];

fibonacciNumbers.Add(previous + previous2);

}

foreach(var item in fibonacciNumbers)

Console.WriteLine(item);

OUTPUT:

1

1

2

3

5

8

13

21

34

55

89

144

233

377

610

987

1597

2584

4181

6765