Code Clock

Day 2: Loops and Lists

Learn.to.code

Programming with C#

@ QUB

Loops

Loops in any programming language allow us to repeat a set of instructions again and again. C# supports several types of loops. Today, we will look at three types:

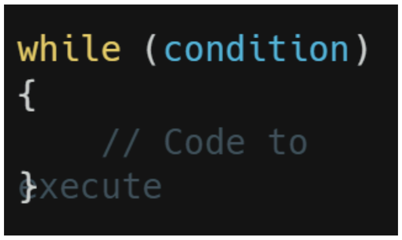
- ***While Loop***
- ***For Loop***
- ***Do While Loop***
- ***Nested Loop***

WHILE LOOP

A while loop is used to repeat a set of statements multiple times until a condition becomes false. It is generally used when we don't know how many times the loop should run.

If the condition stays true, the loop keeps running. If the condition is false, the loop stops.

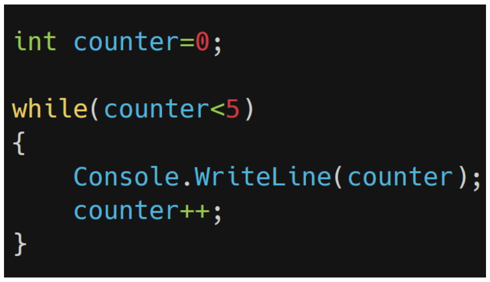
Syntax:



```
while (condition)
{
    // Code to
}execute
```

Example:

This loop will print numbers 1 to 5. The counter starts at 1 and is increased by 1 in each loop. When the counter is greater than 5, the loop stops.



```
int counter=0;

while(counter<5)
{
    Console.WriteLine(counter);
    counter++;
}
```

Challenge: Using a While Loop, print the following:

1. All numbers from 1 to 10
2. All numbers from 10 to 1
3. All even numbers from 1 to 10
4. All odd numbers from 1 to 10
5. All numbers exactly divisible by 4 between 1 and 40 inclusive
6. All even numbers between 6 and 30
7. Add a counter to a total until the total is greater than 100

DO WHILE LOOP

A do while loop is like a while loop, but it runs the code **at least once**, even if the condition is false the first time.

Syntax:

```
do
{

  // Code to execute

} while (condition);
```

Example:

This prints 1 to 5. Even if the condition is false at the start, the code inside runs once.

```
int count = 1;
do
{
 Console.WriteLine("Count is: " + count);
 count++;
}
while (count <= 5);
```

FOR LOOP

A for loop is used when we know how many times to repeat the code. It's commonly used to loop through lists, arrays, and strings.

Syntax:

```
for (start; condition; increment)
{
    // Code to execute
}
```

Example:

```
for (int x = 0; x < 6;
x++)
    Console.WriteLine(x);
}
```

This prints numbers from 0 to 5. The loop starts at 0 and increases x by 1 each time.

You can also change the start, end, and how much you increase by:

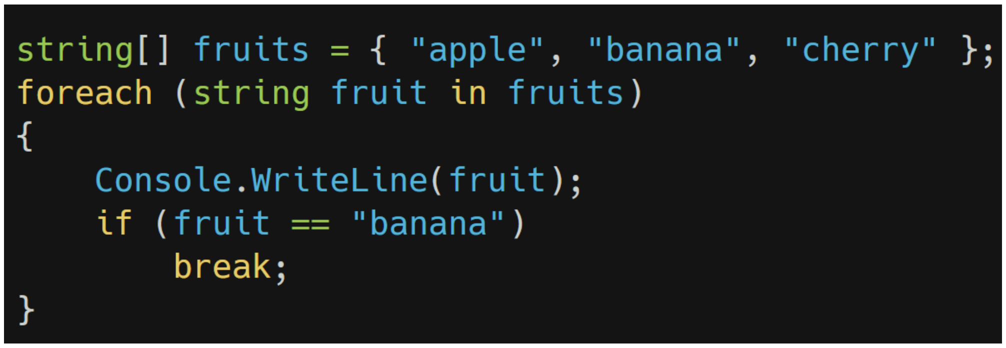
```
for (int x = 2; x < 6; x++) // prints 2 to 5
{
    Console.WriteLine(x);
}

for (int x = 2; x < 30; x += 3) // prints numbers from 2 to 29, jumping by
{
    Console.WriteLine(x);
}
```

BREAK STATEMENT

We use the break statement to stop a loop early.

Example:

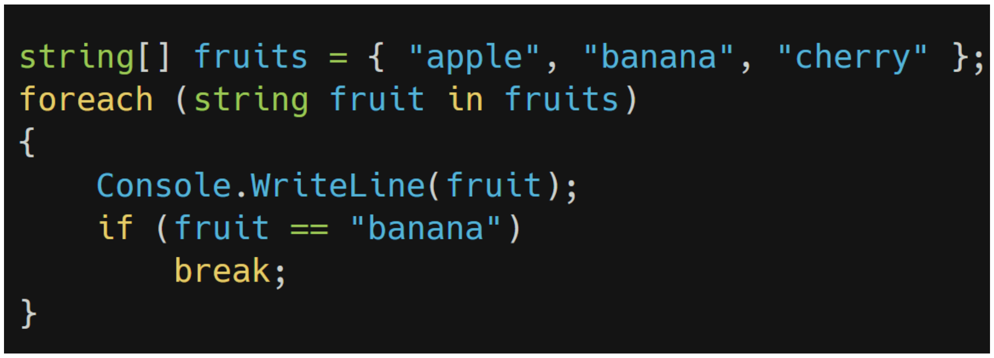


```
string[] fruits = { "apple", "banana", "cherry" };
foreach (string fruit in fruits)
{
    Console.WriteLine(fruit);
    if (fruit == "banana")
        break;
}
```

This stops the loop once it reaches "banana".

LOOPING THROUGH INDEX NUMBERS

We can also loop through the index of a list or array:



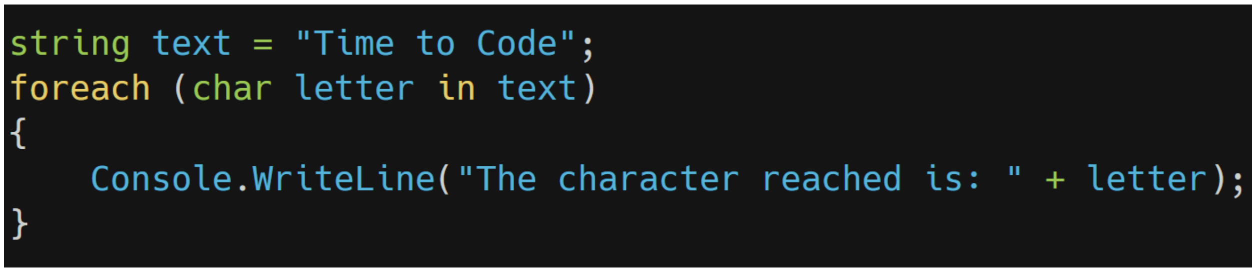
```
string[] fruits = { "apple", "banana", "cherry" };
foreach (string fruit in fruits)
{
    Console.WriteLine(fruit);
    if (fruit == "banana")
        break;
}
```

Challenge: Using a For Loop, print the following:

1. All numbers from 1 to 10
2. All numbers from 10 to 1
3. All even numbers from 1 to 10
4. All odd numbers from 1 to 10
5. All numbers exactly divisible by 4 between 1 and 40 inclusive
6. All even numbers between 6 and 30
7. Add a counter to a total until it exceeds 100
8. Run a maximum of ten times, each time asking the user to enter a number. Add this number to a total. When the total goes over 20, print “Combined total exceeds 20” and stop.

LOOPING OVER STRINGS

Strings are sequences of characters, so we can loop through each letter using a foreach loop:



```
string text = "Time to Code";
foreach (char letter in text)
{
    Console.WriteLine("The character reached is: " + letter);
}
```

Challenge: Using a loop, write code to:

1. Print each letter of a word input by the user
2. Count the number of letters in a word
3. Count the number of vowels in a word
4. Count the number of non-vowels in a word

LISTS

Lists are used to store many items in one variable.

In C#, we use the `List` class:

```
List<string> thisList = new List<string>() { "apple", "banana", "cherry" };
Console.WriteLine(thisList[0]); // prints apple
```

Lists are ordered, allow duplicates, and you can add or remove items easily.

Finding the length of a List:

```
Console.WriteLine(thisList.Count);
```

You can mix data types if needed:

```
List<object> mixedList = new List<object>() { "abc", 34, true
};
```

Negative Indexing: Not supported directly, but you can use:
Console.WriteLine(thisList[thisList.Count - 1]); // last item

Select a range of items in a List:

Use `GetRange(start, count)`:

```
List<string> fruits = new List<string>() { "apple", "banana", "cherry", "orange", "kiwi" };
var someFruits = fruits.GetRange(2, 3); // cherry, orange, kiwi
```

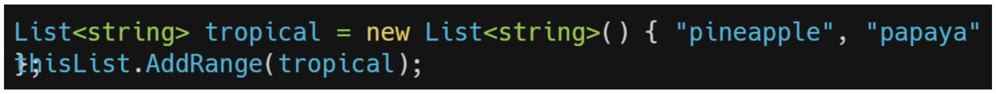
Check if Item exists in a List:

```
if (thisList.Contains("apple"))
{
    Console.WriteLine("Yes, apple is in the list");
}
```

Add Items to a List:

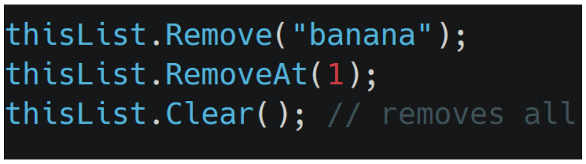
```
thisList.Add("orange");
thisList.Insert(1, "mango");
```

Extending a List:



```
List<string> tropical = new List<string>() { "pineapple", "papaya"
};
thisList.AddRange(tropical);
```

Remove Items from a List:



```
thisList.Remove("banana");
thisList.RemoveAt(1);
thisList.Clear(); // removes all
```

Challenge: Using Lists, do the following:

1. Create a list called `weekDays` with the first 3 working days
2. Add the remaining 2 days
3. Create another list called `weekend` and add Saturday and Sunday
4. Extend `weekDays` with the `weekend` list
5. Print how many days are in `weekDays`
6. Remove the first item and print the new total
7. Check if Monday is still in the list and print a message
8. Clear the list and confirm the length is now 0

EXTENSION: LOTTERY

1. Generate 6 random numbers between 1 and 20
2. Ask the user for 6 numbers between 1 and 20
3. Print the user’s numbers in order
4. - Compare with the random numbers
5. - Count how many match
6. - If 3+ match:
   - 3 matches: "You’ve won £10"
   - 4 matches: "You’ve won £100"
   - 5 matches: "You’ve won £1000"
   - 6 matches: "You’ve won £1,000,000"

- Ask the user if they want to play again or return to the main menu