While Loop

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

SoftUni

sli.do

#prgm-for-qa

Table of Contents

1. **Review** of the Previous Lesson
2. **Introduction** to While Loops
3. The **While Loop**
4. **While** or **For**?
5. **Infinite** Loop "**`while (true)`**“
6. The "**`break`**" Operator
7. Practical Coding **Exercises**

Review

For Loops, Loops with a Step

Prefix and Postfix

- Prefix decrement

```
int a = 1;
Console.WriteLine(--a); // 0
Console.WriteLine(a);   // 0
```

Decrease the value, then print it

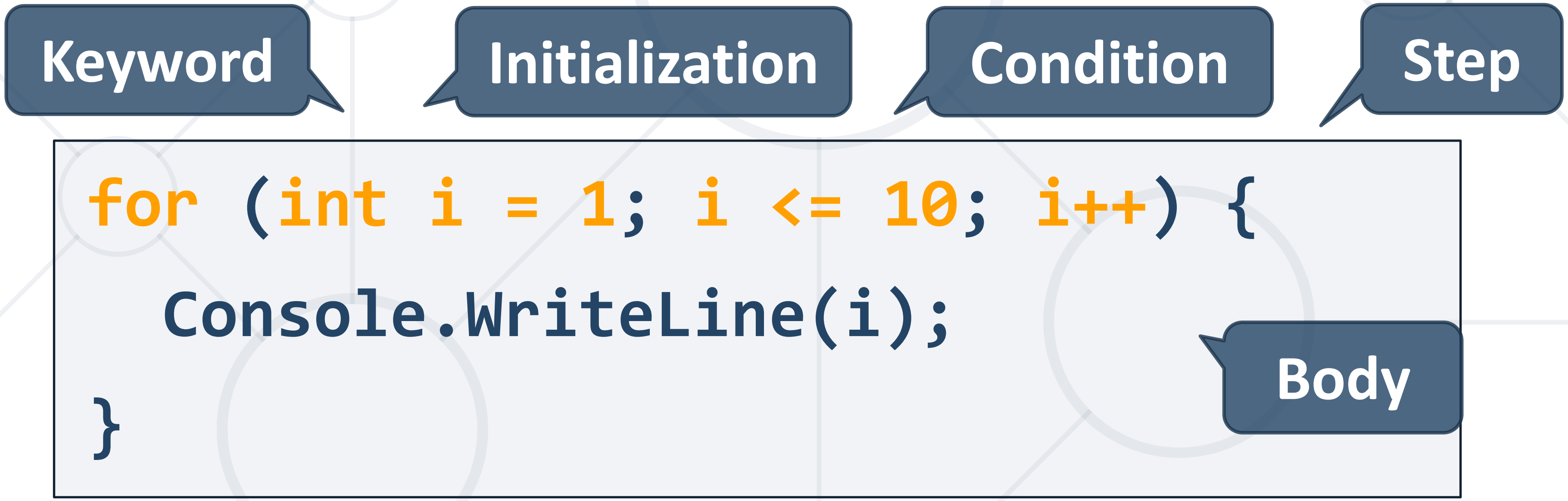
- Postfix decrement

```
int a = 1;
Console.WriteLine(a--); // 1
Console.WriteLine(a);   // 0
```

First print the value, then decrease it

Simple For-Loop

- **For loops** repeat a certain code block a known number of times
- Printing the numbers **from 1 to 10**



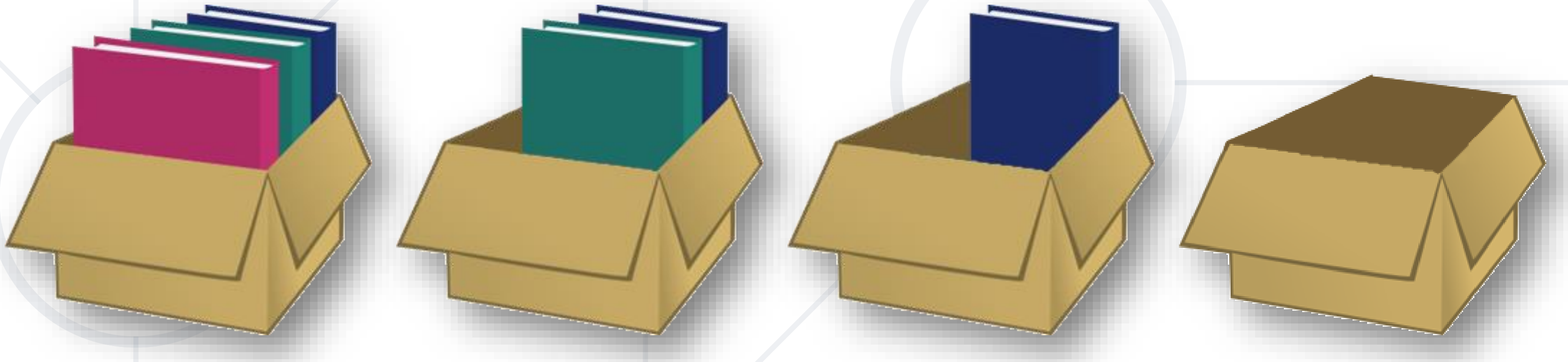
```
for (int i = 1; i <= 10; i++) {
  Console.WriteLine(i);
}
```

Introduction

The Need of While Loops

Real-Life Example: Box of Books

- Unpack a box of books
  - Remove the first book from the box
  - Keep removing books **until** the box is emptied

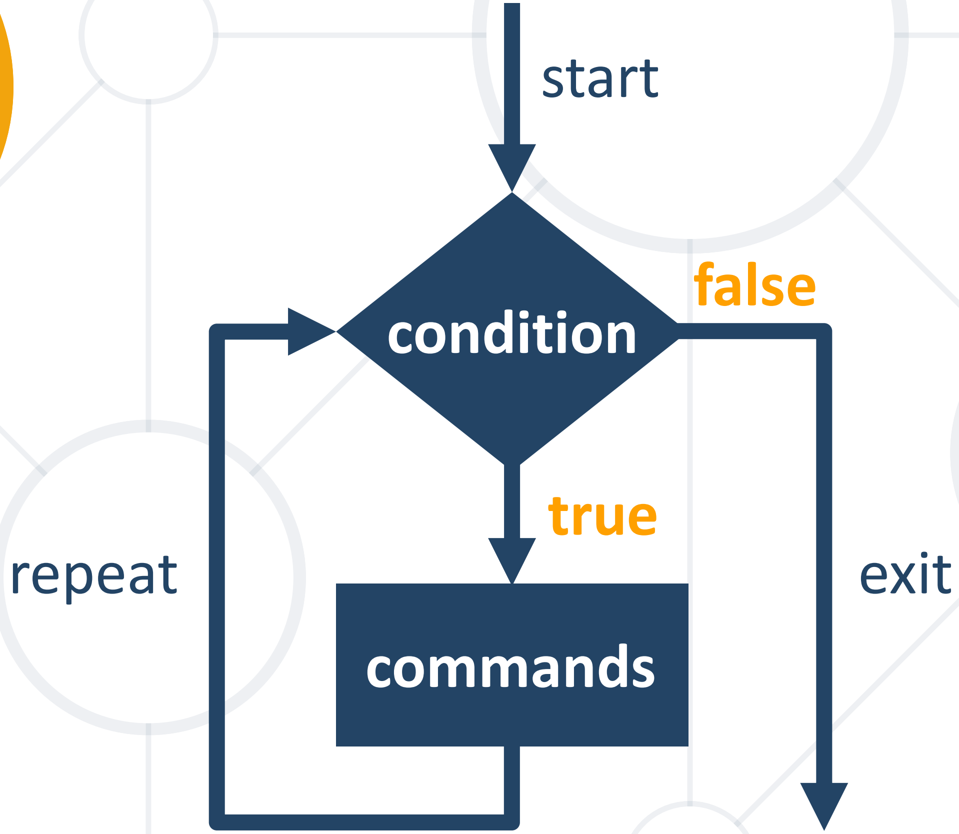


While Loop

Control Flow Statement

While Loop

- Used to **repeat a code** block until an **exit condition** is met



```
while (condition)
{
    // commands
}
```

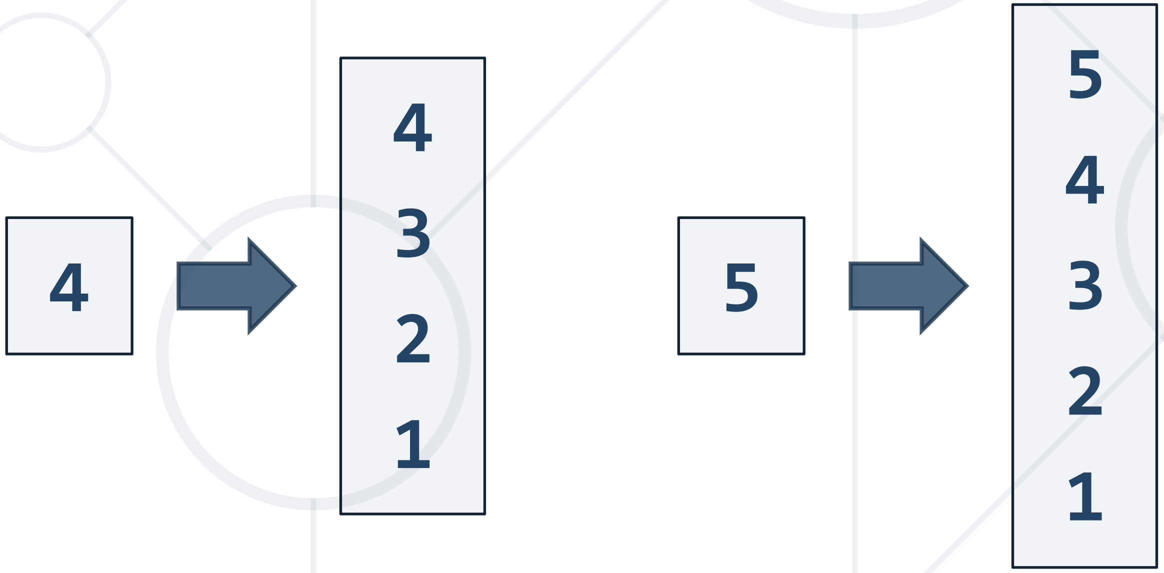
Example: While Loop

- Print the numbers from 1 to 5

```
int i = 1;
while (i <= 5)
{
    Console.WriteLine(i);
    i++;
}
```

Problem: Decreasing Numbers

- Print the **numbers from N down to 1**, using a **while** loop
  - Write a program which receives number: **n**
  - Print the numbers **n … 1**



4 → 4 3 2 1

5 → 5 4 3 2 1

Solution: Decreasing Numbers

```
int number =
int.Parse(Console.ReadLine());
while (number >= 1)
{
    Console.WriteLine(number);
    number--;
}
```

While or For Loop?

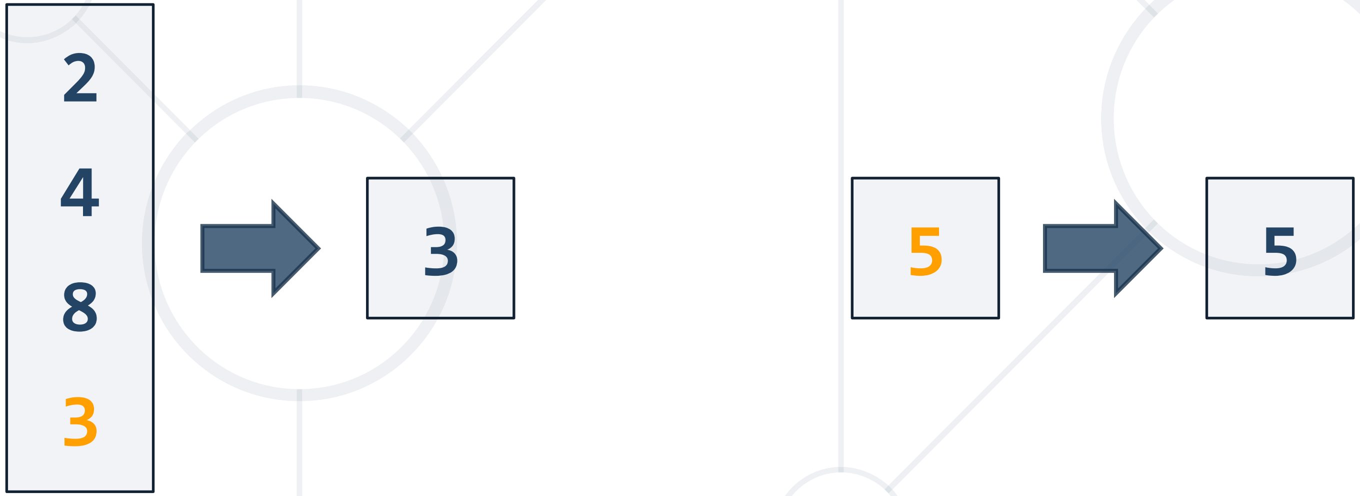
Choosing the Right Loop Type

While or For Loop?

- The `while` and `for` loops both **repeat a block of code**
- Use `for`-loop when you preliminary know the **number of iterations**
  - E.g. repeat exactly **n** times
- Use `while` if you don't know when the exit condition will be met
  - E.g. repeat until stopped

Problem: Odd Number

- Write a program to **enter an odd number**
  - Read numbers from the console until an **odd number** is entered
  - Print the **odd** number as output



```
2
4
8
3
```

→

```
3
```

```
5
```

→

```
5
```

Solution: Odd Number

```
int num = int.Parse(Console.ReadLine());
while (num % 2 == 0)
{
  // The number is not odd → read a new one
  num = int.Parse(Console.ReadLine());
}
Console.WriteLine(num);
```

Infinite While Loop

Using `while (true) { … }`

Infinite While Loop

Software University

- **Infinite loop** = repeating a block of code **an infinite number** of times
- Infinite **while** loops: use **true** as loop condition

```
while (true)
{
   // Commands
}
```

Example: Infinite While Loop (Bug)

```
string command = Console.ReadLine();
while (command != "End")
{
  Console.WriteLine("Executing: " + command);
}
```

always **true**

Example: Finite While Loop (Bug Fixed)

```
string command = Console.ReadLine();
while (command != "End")
{
    Console.WriteLine("Executing: " + command);
    command = Console.ReadLine();
}
```

The "break" Operator

Exiting from a Loop

The "break" Operator

- Used for prematurely **exiting** the loop
- Can only be executed from the loop's **body**
- When break is executed, the code inside the loop's body after it **is skipped** (does not execute)

```
while (true)
{
  // Some code …
  if (…) break;
  // More code …
}
```

Example: Break Operator

```
int sum = 0;
while (true)
{
  int nextNum = int.Parse(Console.ReadLine());
  if (nextNum == 0)
  {
    // The last number was reached
    break;
  }
  sum += nextNum;
  Console.WriteLine("Sum: " + sum);
}
```

- Sum numbers until **0 is entered**

Example: Infinite Loop with Break

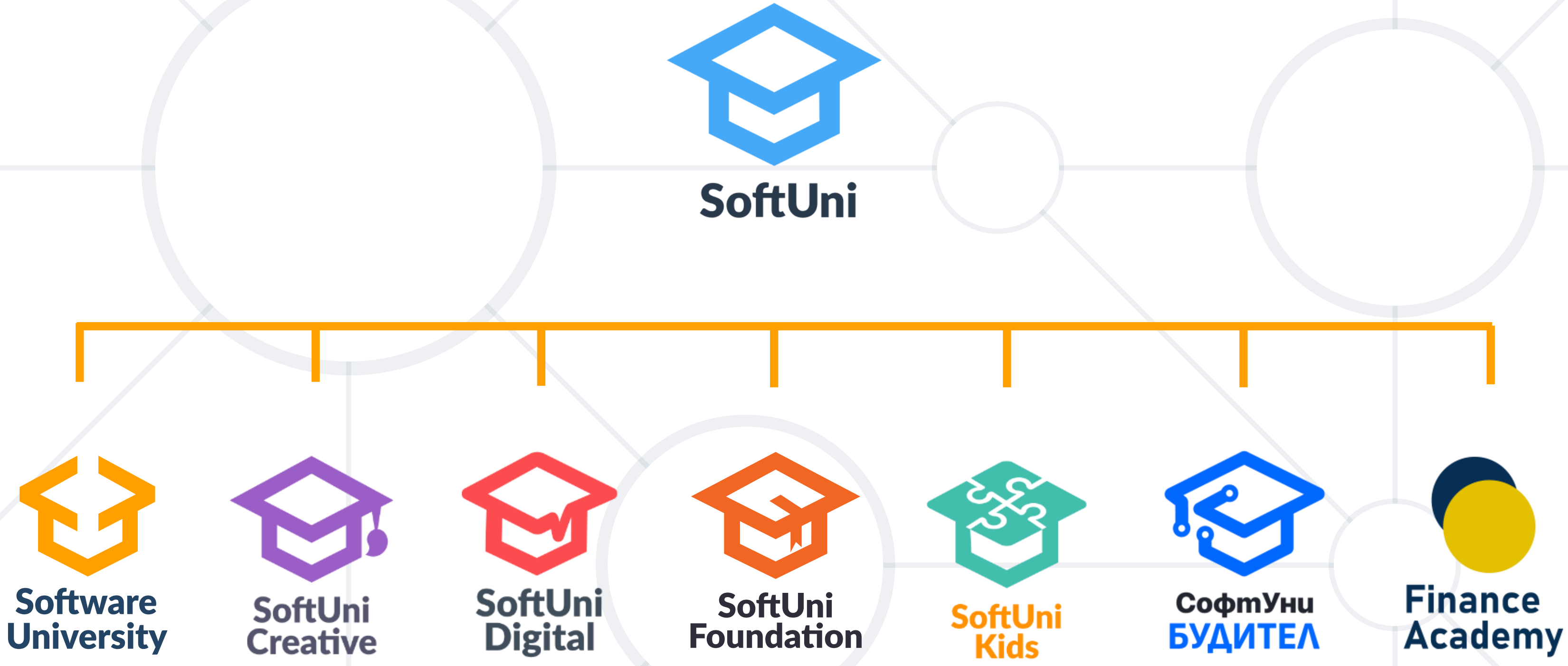
```
while (true)
{
  string command = Console.ReadLine();
  if (command == "End")
  {
     break;
  }
  Console.WriteLine("Executing: " + command);
}
```

Summary

Software University

- The `while` loop executes a block of code multiple times
  - While the loop condition is true
- Use `for` when you initially know the number of repetitions, `while` otherwise
- `while` loops can be infinite
  - Use the `break` operator to exit from the loop on certain condition

Questions?



SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни БУДИТЕЛ

Finance Academy

SoftUni Diamond Partners

SmartIT

createX

VIVACOM

INDEAVR
Serving the high achievers

Coca-Cola HBC
Bulgaria

Postbank
Решения за твоето утре

AMBITIONED

SOFTWARE
GROUP

SUPER
HOSTING
.BG

DRAFT
KINGS

PHAR
VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License