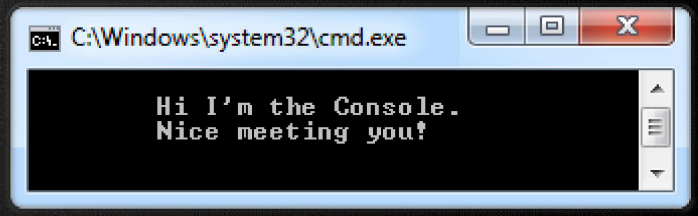
Telerik Academy

# Console Input / Output

## Reading and Writing to the Console




## C# Fundamentals

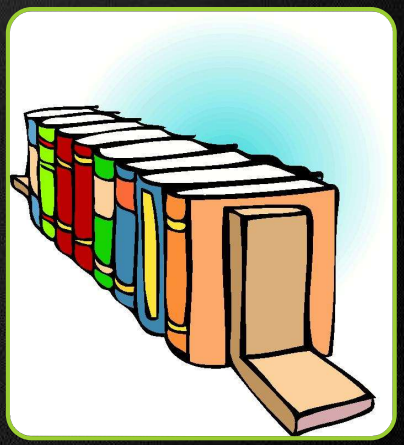
Telerik Software Academy

https://telerikacademy.com

Telerik Academy

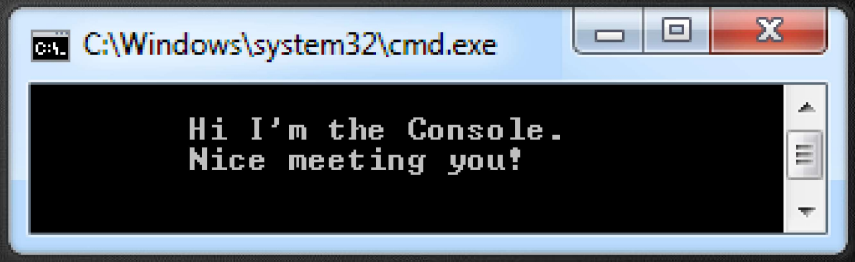
# Table of Contents





Telerik Academy

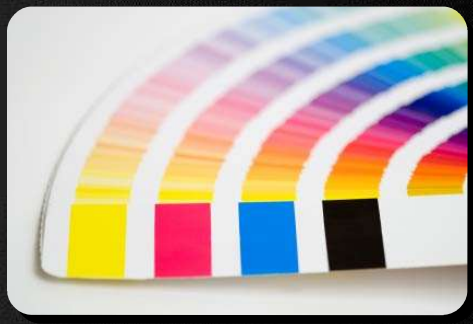
# Printing to the Console



Telerik Academy

# Printing to the Console

- The Console is used to display information in a text window
- Can display different values:
  - Strings
  - Numeric types
  - All primitive data types

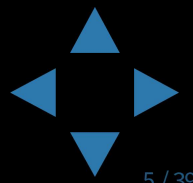


- To print to the console use the class Console (Found in namespace System)

Telerik Academy

# The Console Class

- Provides methods for console input and output
  - Input
    - `Read(…)` – reads a single character
    - `ReadKey(…)` – reads a combination of keys
    - `ReadLine(…)` – reads a single line of characters
  - Output
    - `Write(…)` – prints the specified argument on the console
    - `WriteLine(…)` – prints specified data to the console and moves to the next line



Telerik Academy

# Console.Write

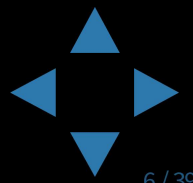
- Printing an integer variable

```
int a = 15;

Console.Write(a); // 15
```

- Printing more than one variable using a formatting string

```
double a = 15.5;
int b = 14;

Console.Write("{0} + {1} = {2}", a, b, a + b);
// 15.5 + 14 = 29.5
```

- Next print operation will start from the same line



Telerik Academy

# Console.WriteLine

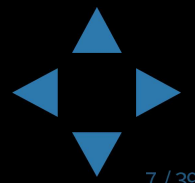
- Printing a string variable

```
string str = "Hello C#!";

Console.WriteLine(str);
```

- Printing more than one variable using a formatting string

```
string name = "Marry";
int year = 1987;

Console.WriteLine("{0} was born in {1}.", name, year);
// Marry was born in 1987.
```

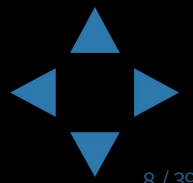
- Next printing will start from the new line



# Printing to the Console – *Example*

```
static void Main()
{
    string name = "Peter";
    int age = 18;
    string town = "Sofia";

    Console.Write("{0} is {1} years old from {2}.",
                  name, age, town);
    // Result: Peter is 18 years old from Sofia.

    Console.Write("This is on the same line!");

    Console.WriteLine("Next sentence will be" +
                      " on a new line.");

    Console.WriteLine("Bye, bye, {0} from {1}.",
                      name, town);
}
```

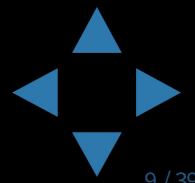




Telerik Academy

# Formatting Strings

- {index[,alignment][:formatString]}
- index
  - The zero-based index of the argument whose string representation is to be included at this position in the string
- alignment
  - A signed integer that indicates the total length of the field into which the argument is inserted
    - a positive integer – right-aligned
    - a negative integer – left-aligned



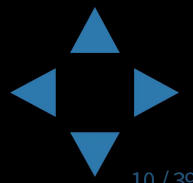
Telerik Academy

# Formatting Strings

- {index[,alignment][:formatString]}
- formatString
  - Specifies the format of the corresponding argument's result string, e.g. "X", "C", "0.00"
- *Example*:

```
static void Main()
{
    double pi = 1.234;
    Console.WriteLine("{0:0.000000}", pi);
    // 1.234000
}
```



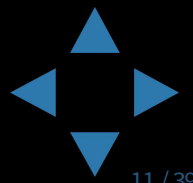


Telerik Academy

# Formatting Strings – *Example*

```
static void Main()
{
    int a=2;
    int b=3;
    Console.Write("{0} + {1} =", a, b);
    Console.WriteLine(" {0}", a + b);
    // 2 + 3 = 5

    Console.WriteLine("{0} * {1} = {2}", a, b, a * b);
    // 2 * 3 = 6

    float pi = 3.14159206;
    Console.WriteLine("{0:F2}", pi); // 3,14

    Console.WriteLine("Bye - Bye!");
}
```



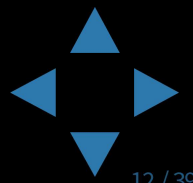


Telerik Academy

# Printing a Menu – *Example*

```
string cola = "Coca Cola";
double colaPrice = 1.20;

string fanta = "Fanta Dizzy";
double fantaPrice = 1.20;

string zagorka = "Zagorka";
double zagorkaPrice = 1.50;

Console.WriteLine("Menu:");
Console.WriteLine("1. {0} - {1}", cola, colaPrice);
Console.WriteLine("2. {0} - {1}", fanta, fantaPrice);
Console.WriteLine("3. {0} - {1}", zagorka, zagorkaPrice);
Console.WriteLine("Have a nice day!");
```





Telerik Academy

# Printing to the Console

Demo

Telerik Academy

# Reading from the Console

## Reading Strings and Numeral Types



Telerik Academy

# Reading from the Console

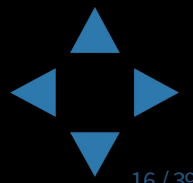
- We use the console to read information from the command line
- We can read:
  - `Characters`
  - `Strings`
  - `Numeric types` (after conversion)
- To read from the console we use the methods `Console.Read()` and `Console.ReadLine()`

Telerik Academy

# Console.Read()

- Gets a single character from the console (after `[Enter]` is pressed)
  - Returns a result of type `int`
  - Returns `-1` if there aren't more symbols
- To get the actually read character we need to cast it to `char`

```
int i = Console.Read();
char ch = (char) i;  // Cast the int to char

// Gets the code of the entered symbol
Console.WriteLine("The code of '{0}' is {1}.", ch, i);
```

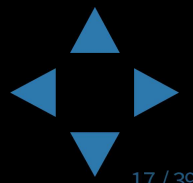


Telerik Academy

# Console.ReadKey

- Waits until a combination of keys is pressed
  - Reads a single character from console or a combination of keys
- Returns a result of type ConsoleKeyInfo
  - KeyChar – holds the entered character
  - Modifiers – holds the state of [Ctrl], [Alt], …

```
ConsoleKeyInfo key = Console.ReadKey();
Console.WriteLine();
Console.WriteLine("Character entered: " + key.KeyChar);
Console.WriteLine("Special keys: " +  key.Modifiers);
```



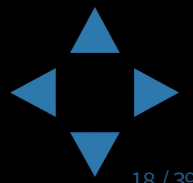




Telerik Academy

# Reading Keys from the Console

Demo



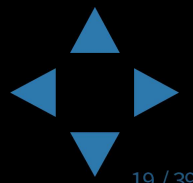
Telerik Academy

# Console.ReadLine

- Gets a line of characters
- Returns a `string` value
- Returns `null` if the end of the input is reached

```
Console.Write("Please enter your first name: ");
string firstName = Console.ReadLine();

Console.Write("Please enter your last name: ");
string lastName = Console.ReadLine();

Console.WriteLine("Hello, {0} {1}!", firstName, lastName);
```





Telerik Academy

# Reading Strings from the Console

Demo

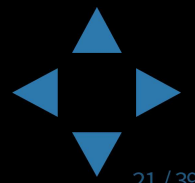


Telerik Academy

# Reading Numeral Types

- Numeric types can not be read directly from the console
- To read a numeral type do the following:
  1. Read a `string` value
  2. Convert (parse) it to the required numeric type
- `int.Parse(string)`
  - Parses (converts) a `string` to `int`

```
string input = Console.ReadLine()
int number = int.Parse(input);

Console.WriteLine("You entered: {0}", number);
```



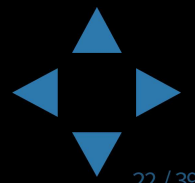




# Converting Strings to Numbers

- Numeric types have a method Parse(…) for extracting the numeral value from a string
  - int.Parse(string) – string → int
  - long.Parse(string) – string → long
  - float.Parse(string) – string → float
  - Causes FormatException in case of error

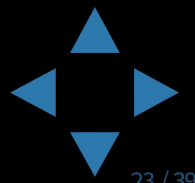
```
string s = "123";
int i = int.Parse(s); // i = 123
long l = long.Parse(s); // l = 123L

string invalid = "xxx1845";
int value = int.Parse(invalid); // FormatException is thrown
```



# Reading Numbers from the Console *Example*

```
static void Main()
{
    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());

    Console.WriteLine("{0} + {1} = {2}", a, b, a + b);
    Console.WriteLine("{0} * {1} = {2}", a, b, a * b);

    float f = float.Parse(Console.ReadLine());
    Console.WriteLine("{0} * {1} / {2} = {3}",
                      a, b, f, a * b / f);
}
```



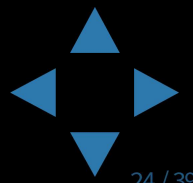


# Telerik Academy Converting Strings to Numbers

- Converting can also be done using the methods of the Convert class
  - `Convert.ToInt32(string)` - string → int
  - `Convert.ToSingle(string)` - string → float
  - `Convert.ToInt64(string)` - string → long
  - It uses the parse methods of the numeric types

```
string s = "123";
int i = Convert.ToInt32(s); // i = 123
long l = Convert.ToInt64(s); // l = 123L

string invalid = "xxx1845";
int value = Convert.ToInt32(invalid); // FormatException
```





Telerik Academy

# Reading Numbers from the Console

Demo



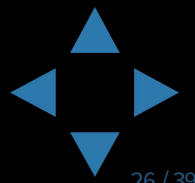
Telerik Academy

# Error Handling when Parsing

- Sometimes we want to handle the errors when parsing a number
  - Two options: use `try-catch` block or `TryParse` method
- Parsing with `TryParse`:

```
string str = Console.ReadLine();
int number;

if (int.TryParse(str, out number))
{
   Console.WriteLine("Valid number: {0}", number);
}
else
{
   Console.WriteLine("Invalid number: {0}", str);
}
```





Telerik Academy

# Parsing with TryParse

Demo



Telerik Academy

# Regional Settings

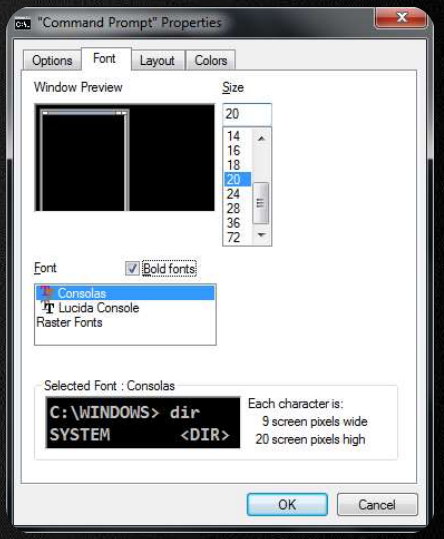
## Printing and Reading Special Characters

## Regional Settings and the Number Formatting



# How to Print Special Characters on the Console?

- Printing special characters on the console needs two steps:
  - Change the console properties to enable Unicode-friendly font
  - Enable Unicode for the Console by adjusting its output encoding
    - Prefer UTF8 (Unicode)




```
using System.Text;
…
Console.OutputEncoding = Encoding.UTF8;
Console.WriteLine("Това е кирилица: ☺");
```

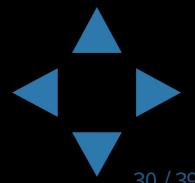
Telerik Academy

# Decimal Separator

- The currency and number formats are different in different countries
  - E.g. the decimal separator could be "." or ","
- To ensure the decimal separator is "." use the following code:

```
using System.Threading;
using System.Globalization;
…
Thread.CurrentThread.CurrentCulture =
  CultureInfo.InvariantCulture;

Console.WriteLine(3.54); // 3.54
decimal value = decimal.Parse("1.33");
```





Telerik Academy

# Regional Settings

Demo

Telerik Academy

# Reading and Printing to the Console

## Various examples

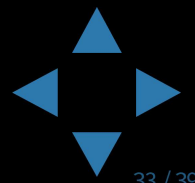


Telerik Academy

# Printing a Letter – *Example*

```
Console.Write("Enter person name: ");
string person = Console.ReadLine();

Console.Write("Enter company name: ");
string company = Console.ReadLine();

Console.WriteLine("  Dear {0},", person);
Console.WriteLine("We are pleased to tell you " +
                  "that {1} has chosen you to take part " +
                  "in the \"Introduction To Programming\" " +
                  "course. {1} wishes you good luck!",
                  person, company);

Console.WriteLine("  Yours,");
Console.WriteLine("  {0}", company);
```





Telerik Academy

# Printing a Letter

Demo

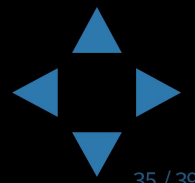


Telerik Academy

# Calculating Area – *Example*

```
Console.WriteLine("This program calculates the area " +
                  "of a rectangle or a triangle");

Console.Write("Enter a and b (for rectangle) " +
              "or a and h (for triangle): ");

int a = int.Parse(Console.ReadLine());
int b = int.Parse(Console.ReadLine());

Console.Write("Enter 1 for a rectangle or 2 " +
              "for a triangle: ");

int choice = int.Parse(Console.ReadLine());
double area = (double)(a * b) / choice;
Console.WriteLine("The area of your figure " +
                  "is {0}", area);
```





Telerik Academy

# Calculating Area

Demo



Telerik Academy

# Summary

- We have discussed the basic input and output methods of the class `Console`
  - `Write` and `WriteLine` methods
    - Used to write values to the console
  - `Read` and `ReadLine` methods
    - Used to read values from the console
- Parsing numbers to strings
  - `int.Parse`, `double.Parse`, and so on methods



Telerik Academy

# Console Input / Output


# Questions?


Telerik Academy

# Free Trainings @ Telerik Academy

- Fundamentals of C# Programming Track of Courses
  - csharpfundamentals.telerik.com
    - Telerik Software Academy
  - telerikacademy.com
    - Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy
    - Telerik Academy Learning System
  - telerikacademy.com