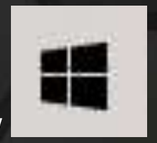
Microsoft
.NET

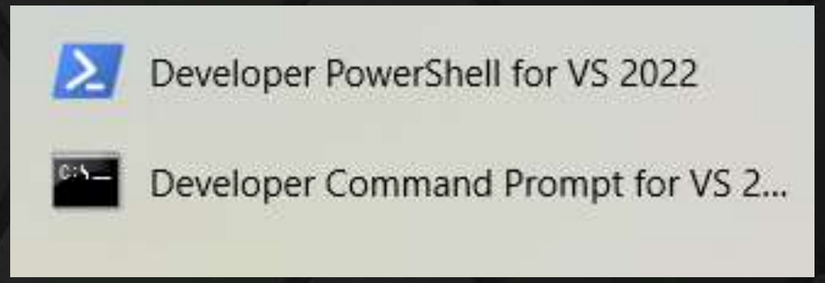
C# Basics

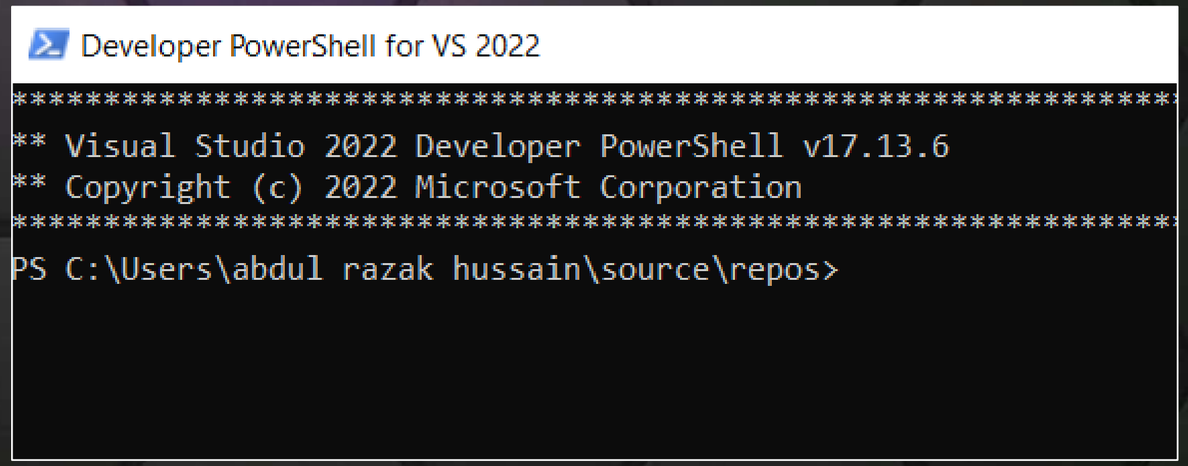
24 April 2025

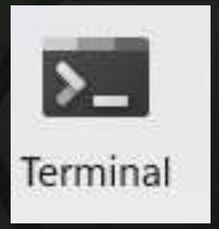
TABLE OF CONTENTS

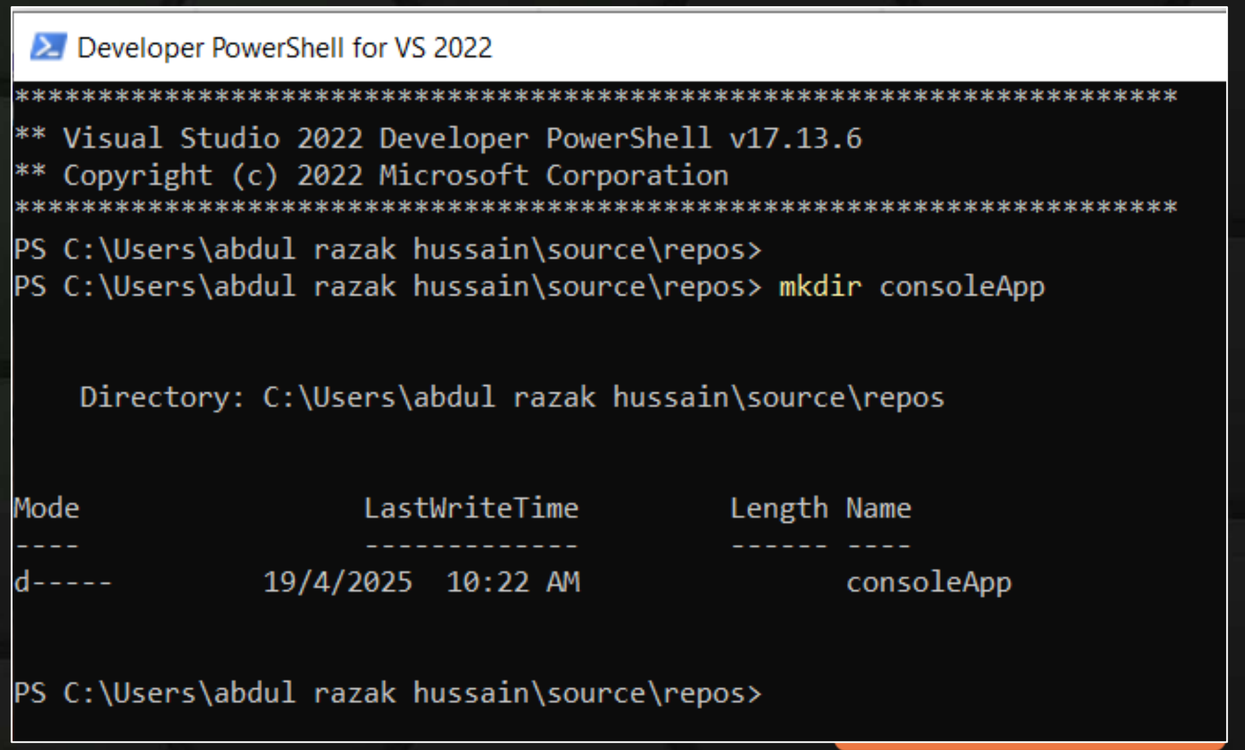
01 C# Console Application: Command prompt

- Click the Windows icon,  select Developer PowerShell



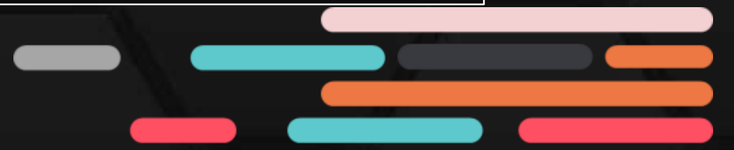


In Windows 11, use Terminal 



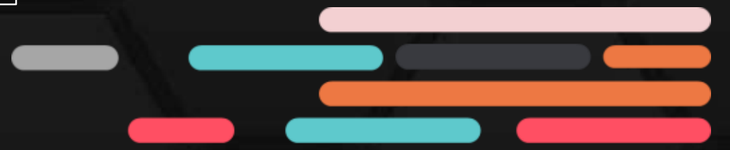
C:> cd consoleApp

```
PS C:\Users\abdul razak hussain\source\repos\consoleApp> dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help         Display help.
  --info            Display .NET information.
  --list-sdks       Display the installed SDKs.
  --list-runtimes   Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.
```

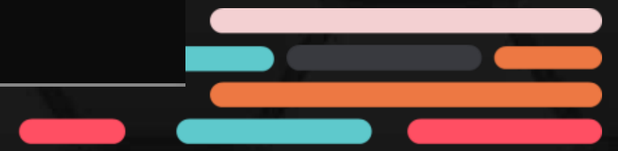


```
PS C:\Users\abdul razak hussain\source\repos> dotnet new
The 'dotnet new' command creates a .NET project based on a template.

Common templates are:
Template Name      Short Name  Language    Tags
---------------    ----------  ----------  --------------------
Blazor Web App     blazor      [C#]        Web/Blazor/WebAssembly
Class Library      classlib    [C#],F#,VB  Common/Library
Console App        console     [C#],F#,VB  Common/Console
Windows Forms App  winforms    [C#],VB     Common/WinForms
WPF Application    wpf         [C#],VB     Common/WPF

An example would be:
   dotnet new console

Display template options with:
   dotnet new console -h
Display all installed templates with:
   dotnet new list
Display templates available on NuGet.org with:
   dotnet new search web
```

C:> consoleApp> dotnet new console



```
PS C:\Users\abdul razak hussain\source\repos\consoleApp> dotnet new console
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring C:\Users\abdul razak hussain\source\repos\consoleApp\consoleApp.csproj:
Restore succeeded.

PS C:\Users\abdul razak hussain\source\repos\consoleApp> dir


    Directory: C:\Users\abdul razak hussain\source\repos\consoleApp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         19/4/2025  10:47 AM                obj
-a----         19/4/2025  10:47 AM            252 consoleApp.csproj
-a----         19/4/2025  10:47 AM            105 Program.cs

PS C:\Users\abdul razak hussain\source\repos\consoleApp> dotnet run
Hello, World!
PS C:\Users\abdul razak hussain\source\repos\consoleApp>
```



```
PS C:\Users\abdul razak hussain\source\repos\consoleApp> notepad .\Program.cs
```

Program.cs - Notepad

File Edit Format View Help

```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

```
PS C:\Users\abdul razak hussain\source\repos\consoleApp>
```

File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search Solution1

Attach...

Program.cs

Miscellaneous Files

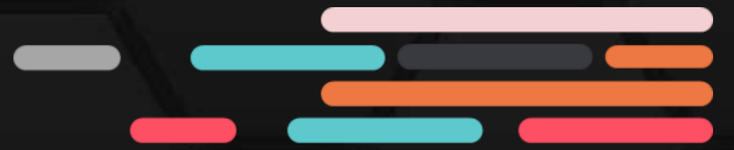
```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

C# Interactive (.NET Core)

```
Microsoft (R) Visual C# Interactive Compiler version 4.13.0-3.25167.3 ()
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> Console.WriteLine("Hello, World!");
Hello, World!
>
```

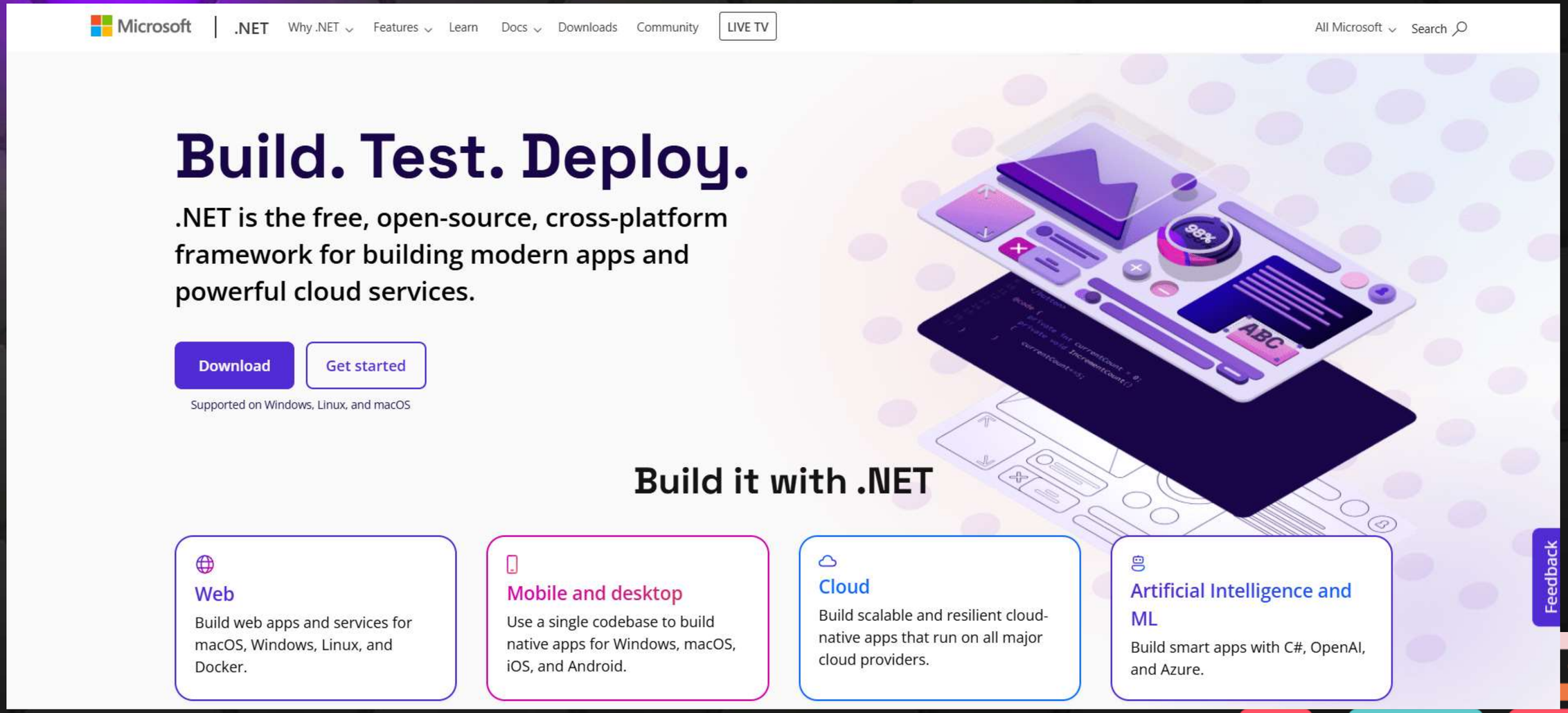
Quick Actions and Refactorings...	Ctrl+.
Rename...	Ctrl+R, Ctrl+R
Peek Definition	Alt+F12
Go To Definition	F12
Go To Base	Alt+Home
Go To Implementation	Ctrl+F12
Find All References	Shift+F12
View Call Hierarchy	Ctrl+K, Ctrl+T
Track Value Source	
Run Tests	Ctrl+R, T
Debug Tests	Ctrl+R, Ctrl+T
Show in Test Explorer	Ctrl+E, S
Breakpoint	▸
Execute in Interactive	Ctrl+E, Ctrl+E
Snippet	▸
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Annotation	▸
Outlining	▸

```
PS C:\Users\abdul razak hussain\source\repos\consoleApp> dir


    Directory: C:\Users\abdul razak hussain\source\repos\consoleApp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         19/4/2025  10:53 AM                bin
d-----         19/4/2025  10:53 AM                obj
-a----         19/4/2025  10:47 AM            252 consoleApp.csproj
-a----         19/4/2025  11:28 AM            105 Program.cs
```



02 C# Console Application: Visual Studio

https://dotnet.microsoft.com/en-us/



Everything you need to start your .NET learning journey

.NET offers a library of learning resources. Access videos, tutorials, code samples, and content from Microsoft Learn to help you build better.

Explore learning portal

.NET for Beginners videos

Getting started with .NET development? We have you covered with our .NET for Beginners videos. Explore videos on web, mobile, desktop, C#, machine learning, containers/Docker, data access, and more.

Browse beginner videos →

Microsoft Learn

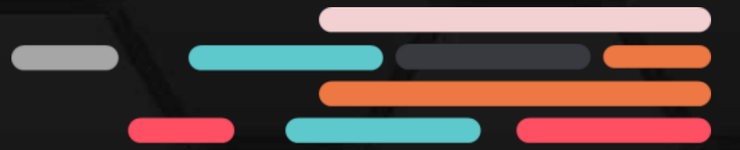
Discover your path to build apps with .NET on Microsoft Learn. Whether you're just starting or an experienced professional, Microsoft Learn's hands-on approach helps you arrive at your goals faster, with more confidence and at your own pace for free.

Get started →

LinkedIn Learning

Get an introduction to the programming skills needed for a career as a .NET software developer. Experience .NET learning courses that provide a broad perspective on core technologies leveraging .NET.

Explore courses →



What do you want to learn?

C#

Programming with C#

Learn the building blocks of programming using C#, .NET's open-source, cross-platform, object-oriented programming language.

</>

Front-end web

Use Blazor to build interactive web pages with HTML, CSS, and C#.

Back-end web

Start building powerful, versatile APIs with ASP.NET Core.

Cloud-ready apps

Build modern, distributed, and powerful apps for the cloud with .NET Aspire.

Mobile and desktop

Use a single codebase to build native mobile apps for iOS, Android, and more.

F#

Programming with F#

Write succinct, robust, and performant code.

Artificial Intelligence and ML

Build smart apps with C#, OpenAI, and Azure.

Game development

Develop 2D and 3D games for the most popular desktops, phones, and consoles.

Microsoft | .NET Why .NET Features Learn Docs Downloads Community LIVE TV All Microsoft Search

Home / Learn to code

Web	Native Desktop & Mobile	Languages	More
ASP.NET Core	.NET Multi-platform App UI	C#	.NET
Web APIs	Windows Forms	F#	Cloud
Blazor	Windows Presentation Foundation	Visual Basic	Data
Razor Pages			Internet of Things
MVC			Machine learning
SignalR			

👋 Hi friend

Get certified

Showcase your C# knowledge with the new Foundational C# Certification, made in partnership with freeCodeCamp. The certification is comprehensive, globally accessible, and, most importantly, free, ensuring that learners everywhere can benefit from a robust C# training on Microsoft Learn.

The certification includes a full C# training course.

Learn more on freeCodeCamp →

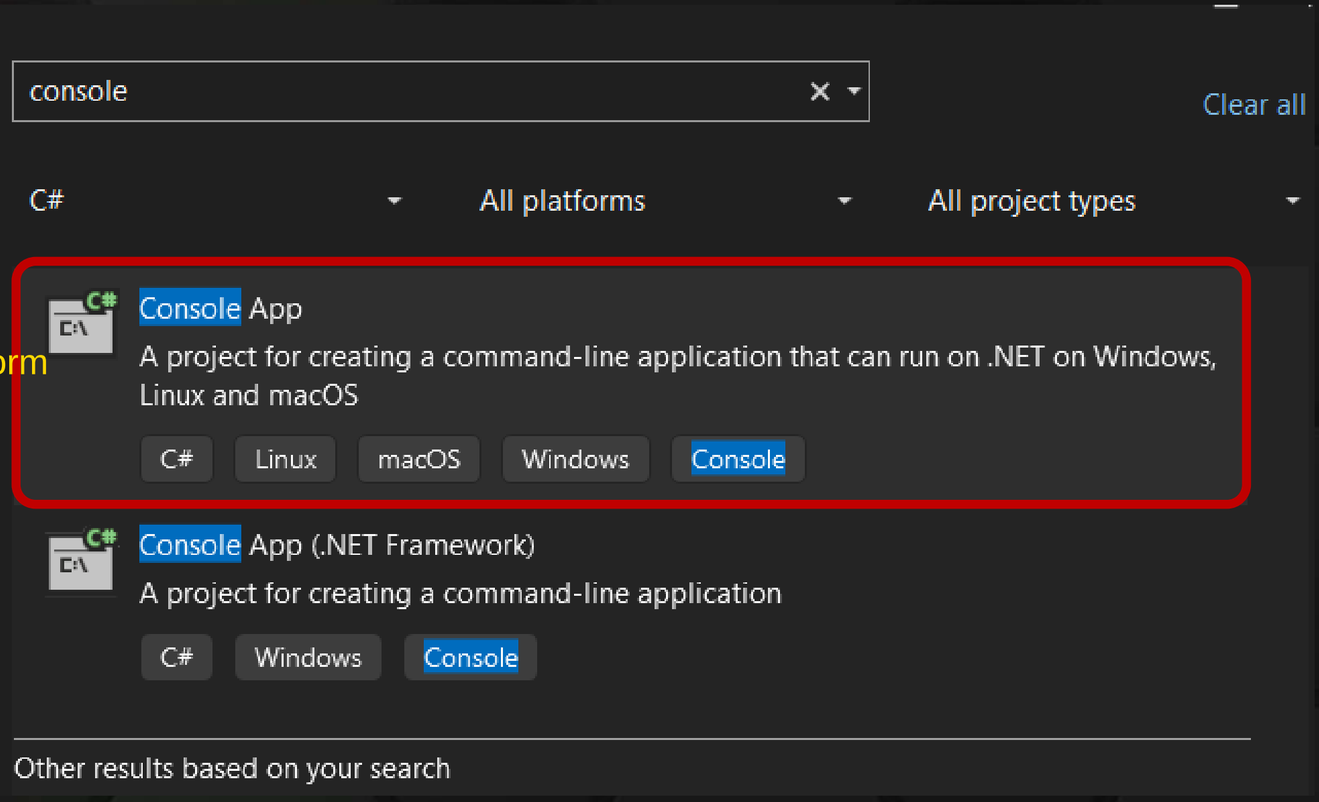
Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

refers to a .NET application that is cross-platform and can run on Windows, macOS, and Linux.

refers to a .NET application that can only runs on Windows.



Additional information

Console App

C# Linux macOS Windows Console

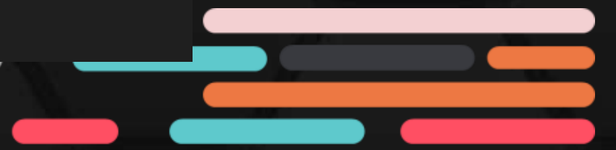
Framework

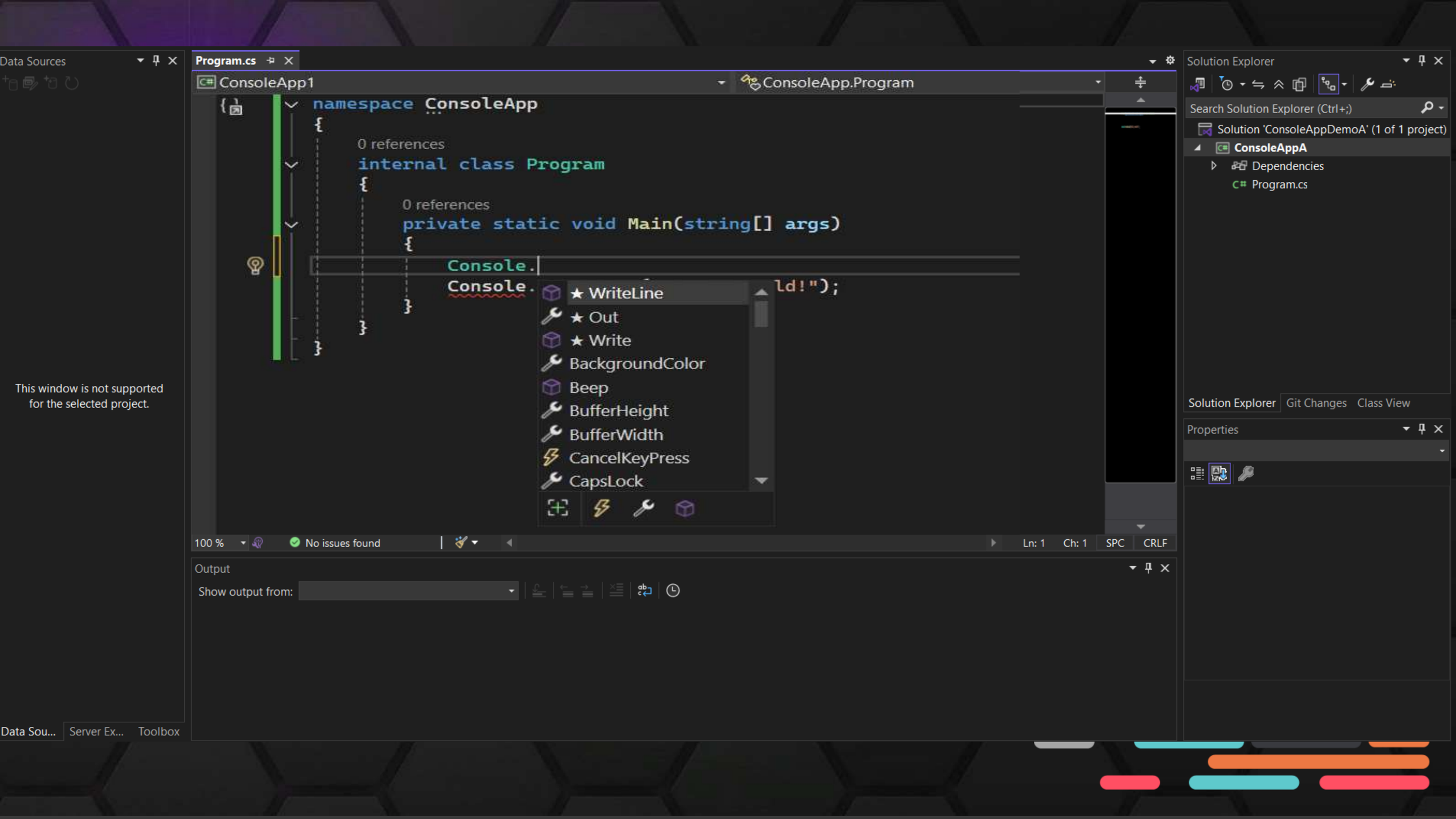
.NET 8.0 (Long Term Support)

- [ ] Do not use top-level statements
- [ ] Enable native AOT publish

- [ ] Place solution and project in the same directory

Project will be created in "C:\Users\abdul razak hussain\source\repos\ConsoleAppDemoA\ConsoleAppA
\"




Data Sources
This window is not supported for the selected project.
Data Sou...
Server Ex...
Toolbox
Program.cs
ConsoleApp1
ConsoleApp.Program
namespace ConsoleApp
{
0 references
internal class Program
{
0 references
private static void Main(string[] args)
{
Console.
Console.
ld!");
}
}
}
★ WriteLine
★ Out
★ Write
BackgroundColor
Beep
BufferHeight
BufferWidth
CancelKeyPress
CapsLock
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Output
Show output from:
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ConsoleAppDemoA' (1 of 1 project)
ConsoleAppA
Dependencies
Program.cs
Solution Explorer
Git Changes
Class View
Properties

C# Online Compiler

‹ Previous Next ›

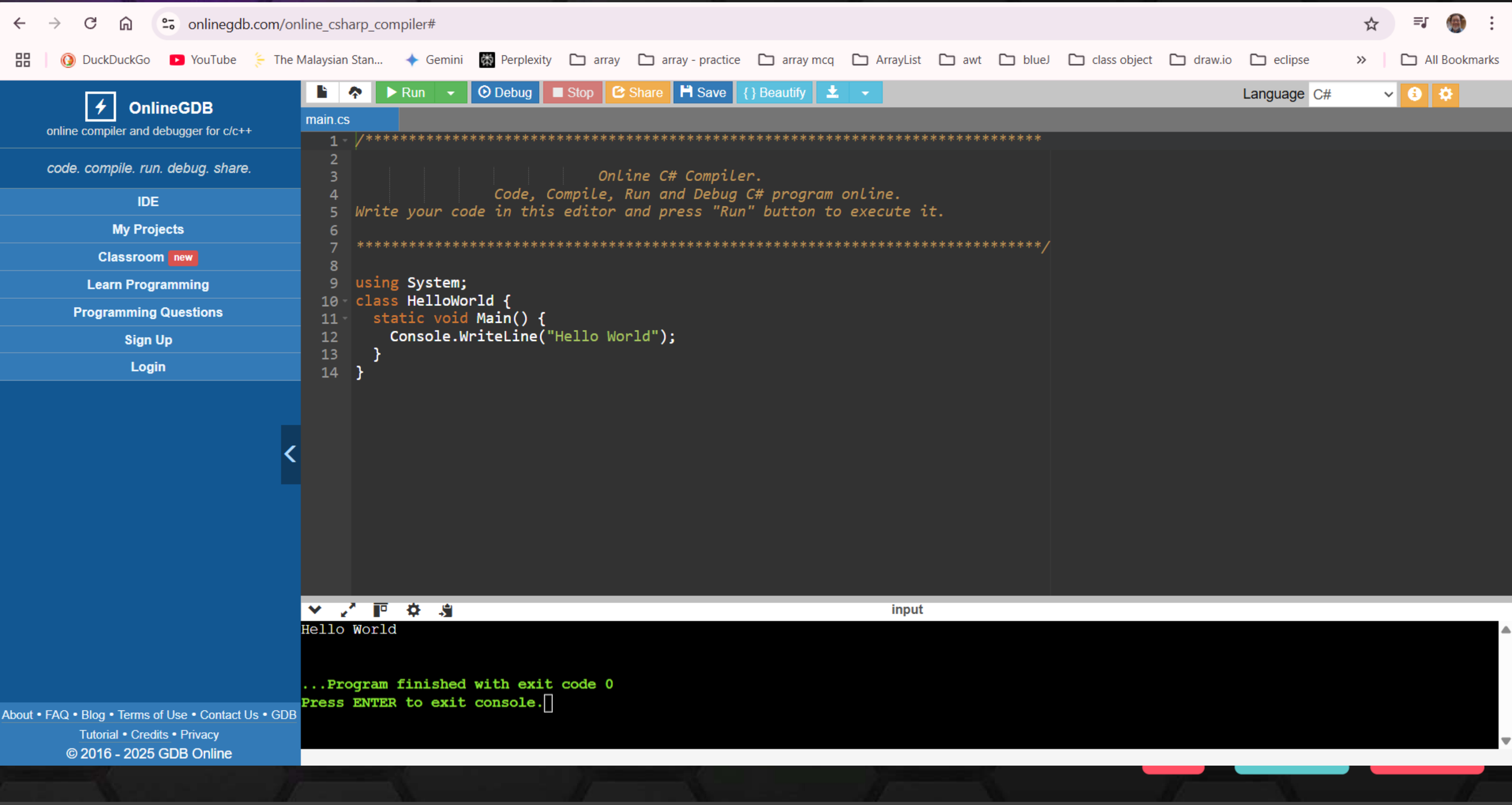
C# Compiler (Editor)

With our online C# compiler, you can edit C# code, and view the result in your browser.

Run »

```
using System;

namespace HelloWorld
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

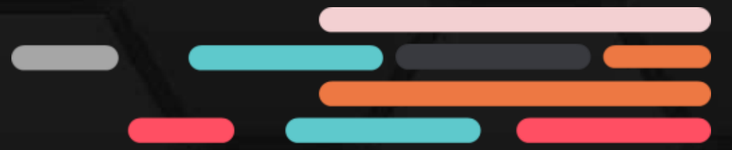
```
Hello World!
```


onlinegdb.com/online_csharp_compiler#
DuckDuckGo
YouTube
The Malaysian Stan...
Gemini
Perplexity
array
array - practice
array mcq
ArrayList
awt
blueJ
class object
draw.io
eclipse
All Bookmarks
OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C#
main.cs
/**
Online C# Compiler.
Code, Compile, Run and Debug C# program online.
Write your code in this editor and press "Run" button to execute it.
**/
using System;
class HelloWorld {
static void Main() {
Console.WriteLine("Hello World");
}
}
input
Hello World
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

03 On string ...

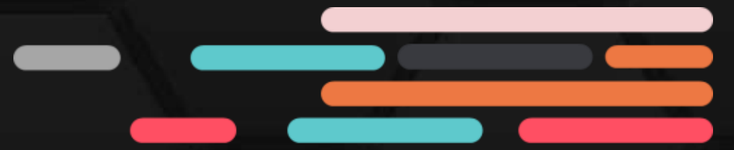
Methods	Description
`Format()`	returns a formatted string
`Split()`	splits the string into substring
`Substring()`	returns substring of a string
`Compare()`	compares string objects
`Replace()`	replaces the specified old character with the specified new character
`Contains()`	checks whether the string contains a substring
`Join()`	joins the given strings using the specified separator
`Trim()`	removes any leading and trailing whitespaces
`EndsWith()`	checks if the string ends with the given string
`IndexOf()`	returns the position of the specified character in the string
`Remove()`	returns characters from a string

`ToUpper()`	converts the string to uppercase
`ToLower()`	converts the string to lowercase
`PadLeft()`	returns string padded with spaces or with a specified Unicode character on the left
`PadRight()`	returns string padded with spaces or with a specified Unicode character on the right
`StartsWith()`	checks if the string begins with the given string
`ToCharArray()`	converts the string to a `char` array
`LastIndexOf()`	returns index of the last occurrence of a specified string



```
// Write(), WriteLine()
Console.Write("Visual Studio welcomes you, ");
Console.WriteLine("Razak");

Console.Write("Congratulations!");
Console.Write(" ");
Console.Write("You wrote your first lines of code.\n");
```

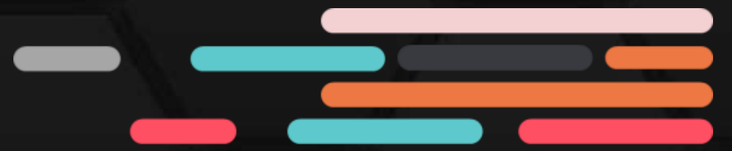
```
Visual Studio welcomes you, Razak
Congratulations! You wrote your first lines of code.
```



```
// GETING INPUT FROM USER: ReadLine()

// Type your username and press enter
Console.Write("\nEnter username:");

// Create a string variable and get user input from the keyboard and store it in the variable
string userName = Console.ReadLine();

// Print the value of the variable (userName), which will display the input value
Console.WriteLine("Username is: " + userName);

//===

Console.Write("\nEnter your age: ");
int age = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Your age is: " + age);
```

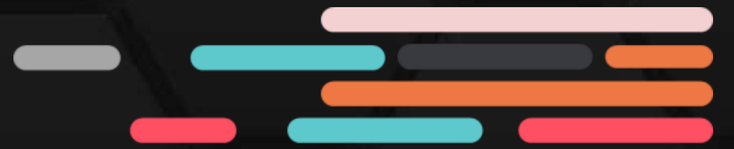
In C#, the string keyword is an alias for String; therefore, String and string are equivalent

```
Enter username:kiki
Username is: kiki

Enter your age: 26
Your age is: 26
```



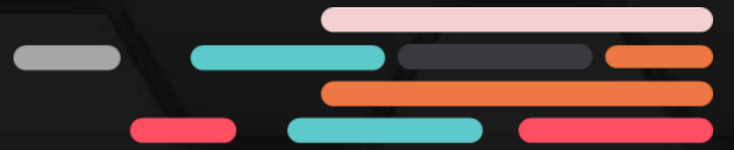
```
// CHECKING IF A STRING REPRESENTS NUMERIC VALUE

string numString = "1287543"; //"1287543.0" will return false for a long
long number1 = 0;
bool canConvert = long.TryParse(numString, out number1);
if (canConvert == true)
    Console.WriteLine($"number1 now = {number1}");
else
    Console.WriteLine("numString is not a valid long");
```

```
number1 now = 1287543
```



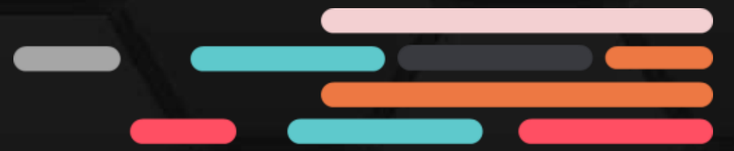
```
byte number2 = 0;
numString = "256"; // A value of 256 will return false
canConvert = byte.TryParse(numString, out number2);
if (canConvert == true)
    Console.WriteLine($"number2 now = {number2}");
else
    Console.WriteLine("numString is not a valid byte");
```

```
numString is not a valid byte
```



```
decimal number3 = 0;
numString = "27.3"; //"27" is also a valid decimal
canConvert = decimal.TryParse(numString, out number3);
if (canConvert == true)
    Console.WriteLine($"number3 now = {number3}");
else
    Console.WriteLine("number3 is not a valid decimal");
```

```
number3 now = 27.3
```



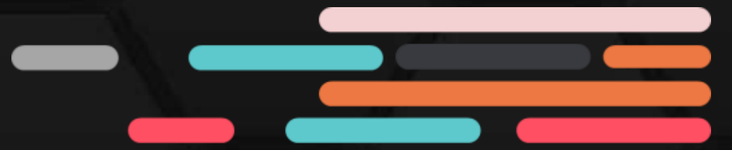
```
// GET THE LENGTH OF STRING
// create string
string str = "Melaka pernah jatuh ke tangan Portugis, Belanda, Inggeris, Jepun ...";
Console.WriteLine("\nstring: \n" + str);

// get length of str
int length = str.Length;
Console.WriteLine("Length: " + length + "\n");

string str2 = "Di mana bumi kupijak\r\nDi situ langit kujunjung\r\nAlang-alang menyeluk pekasam\r
length = str2.Length;
Console.WriteLine("string: \n" + str2);
Console.WriteLine("Length: " + length);
```

```
string:
Melaka pernah jatuh ke tangan Portugis, Belanda, Inggeris, Jepun ...
Length: 68

string:
Di mana bumi kupijak
Di situ langit kujunjung
Alang-alang menyeluk pekasam
Biar sampai ke pangkal lengan
Length: 107
```

```
// STRING JOINING (CONCATENTATION)

string str1 = "Nasi Lemak ";
Console.WriteLine("string str1: " + str1);

string str2 = "Istimewa Kg Morten";
Console.WriteLine("string str2: " + str2);

// join two strings
string joinedString = string.Concat(str1, str2);
Console.WriteLine("Joined string: " + joinedString);
```

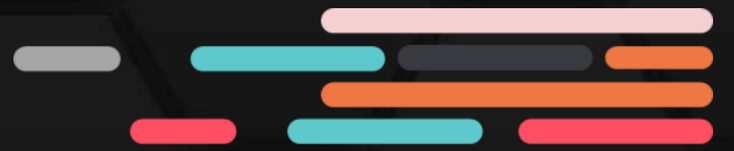
```
string str1: Nasi Lemak
string str2: Istimewa Kg Morten
Joined string: Nasi Lemak Istimewa Kg Morten
```

```
Console.WriteLine($"{str1.Trim()} {str2.Trim()}");
```



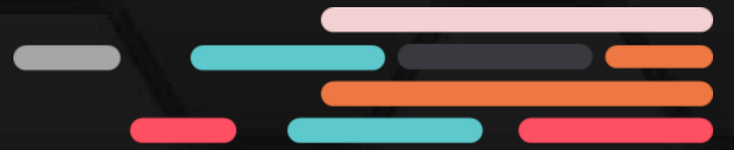
```
// STRING METHOD: TRIM()
string str1 =


    "         Nasi Lemak      ";
Console.WriteLine("string str1: " + str1);

string str2 = "Istimewa Kg Morten



    ;
Console.WriteLine("string str2: " + str2);
```

```
string str1:          Nasi Lemak
string str2: Istimewa Kg Morten                    .
Nasi Lemak Istimewa Kg Morten                       .
```



```
// STRING INTERPOLATION
// to insert values into a single string
// to control text formatting
string bookTitle1 = "Thinking, Fast and Slow";
string bookTitle2 = "Freakonomics";

Console.WriteLine("I\'m reading " + bookTitle1 + " and " + bookTitle2);
```

```
I'm reading Thinking, Fast and Slow and Freakonomics
I'm reading Thinking, Fast and Slow and Freakonomics
```



```
// STRING INTERPOLATION
// to insert values into a single string
// to control text formatting
var name = "Aisyah";
Console.WriteLine($"Hello, {name}. It's a pleasure to meet you!");

var item = (Name: "Nasi Lemak Cun", Price: 1.99m, perPackage: 3); // m = money!
var date = DateTime.Now;
Console.WriteLine($"On {date}, the price of {item.Name} was RM {item.Price} per {item.perPackage} items.");
```

```
Hello, Aisyah. It's a pleasure to meet you!
On 20/4/2025 12:17:48 PM, the price of Nasi Lemak Cun was RM 1.99 per 3 items.
```

```
Console.WriteLine($"On {date:d}, the price of {item.Name} was {item.Price:C2} per {item.perPackage} items");
```

```
Hello, Aisy
On 20/4/202
```

```
rd date and
ay the short
ay the year
splay the ye
umeric value
umeric value
```

```
figuration: Debug Any
an execute 'Build' or
```

d	short date
d	day of the month (1-2 digits)
D	long date
dd	day of the month (2 digits)
ddd	day of the week (abbreviated)
dddd	day of the week (full)
f	full short date/time
F	full long date/time
g	general short date/time

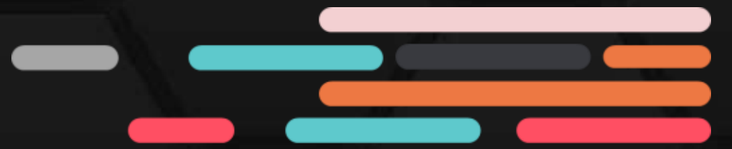
Examples:
d (en-MY) → 15/6/2009
d (en-US) → 6/15/2009

The "d" standard format specifier represents a custom date and time format string
defined by a specific culture's DateTimeFormatInfo.ShortDatePattern property. For
example, the custom format string that is returned by the ShortDatePattern proper
the invariant culture is "MM/dd/yyyy".

string methods

Methods	Description
`Format()`	returns a formatted string
`Split()`	splits the string into substring
`Substring()`	returns substring of a string
`Compare()`	compares string objects
`Replace()`	replaces the specified old character with the specified new character
`Contains()`	checks whether the string contains a substring
`Join()`	joins the given strings using the specified separator
`Trim()`	removes any leading and trailing whitespaces
`EndsWith()`	checks if the string ends with the given string
`IndexOf()`	returns the position of the specified character in the string
`Remove()`	returns characters from a string

`ToUpper()`	converts the string to uppercase
`ToLower()`	converts the string to lowercase
`PadLeft()`	returns string padded with spaces or with a specified Unicode character on the left
`PadRight()`	returns string padded with spaces or with a specified Unicode character on the right
`StartsWith()`	checks if the string begins with the given string
`ToCharArray()`	converts the string to a `char` array
`LastIndexOf()`	returns index of the last occurrence of a specified string

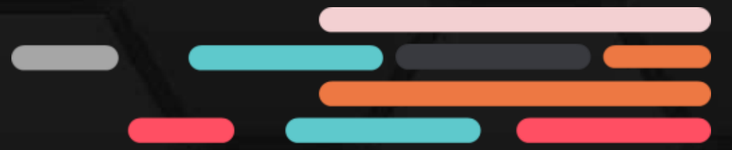


```
//STRING INTERPOLATION - FIELD WIDTH & ALIGNMENT
var inventory = new Dictionary<string, int>()
{
    ["hammer, ball pein"] = 18,
    ["hammer, cross pein"] = 5,
    ["screwdriver, Phillips #2"] = 14
};

Console.WriteLine($"Inventory on {DateTime.Now:d}");
Console.WriteLine(" ");
Console.WriteLine("+-----------------------------------+");
Console.WriteLine($"|{"Item",-25}|{"Quantity",10}|");
Console.WriteLine("+-----------------------------------+");
foreach (var item in inventory)
    Console.WriteLine($"|{item.Key,-25}|{item.Value,10}|");
Console.WriteLine("+-----------------------------------+");
```

```
Inventory on 20/4/2025

+-----------------------------------+
|Item                       |  Quantity|
+-----------------------------------+
|hammer, ball pein          |        18|
|hammer, cross pein         |         5|
|screwdriver, Phillips #2   |        14|
+-----------------------------------+
```

```
byte number = 2;
int count = 10;
float totalPrice = 145_890.00f;
char gender = 'M';
string address = "Olak Lempit, Selangor";
bool isDelivered = false;

Console.WriteLine("number is " + number);
Console.WriteLine("count is " + count);
Console.WriteLine("totalPrice is " + totalPrice);
Console.WriteLine("gender is " + gender);
Console.WriteLine("address is " + address);
Console.WriteLine("isDelivered is " + isDelivered);
```

```
number is 2
count is 10
totalPrice is 145890
gender is M
address is Olak Lempit, Selangor
isDelivered is False
```



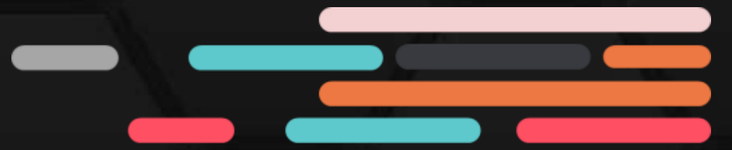
```
var number = 2;
var count = 10;
var totalPrice = 145_890.00f;
var gender = 'M';
va                                    ;
va
```

readonly struct System.Char

Represents a character as a UTF-16 code unit.

```
Console.WriteLine("number is " + number);
Console.WriteLine("count is " + count);
Console.WriteLine("totalPrice is " + totalPrice);
Console.WriteLine("gender is " + gender);
Console.WriteLine("address is " + address);
Console.WriteLine("isDelivered is " + isDelivered);
```

```
number is 2
count is 10
totalPrice is 145890
gender is M
address is Olak Lempit, Selangor
isDelivered is False
```



04 On numbers...

- Implicit type conversion

```
byte b = 1;

int i = b;
```

```
int i = 1;

float f = i;
```

- Explicit type conversion (casting)

```
int i = 1;

byte b = (byte)i;
```

```
string s = "1";

int i = (int)s;
```

```
float f = 1.0f;

int i = (int)f;
```

- Conversion between non-compatible types

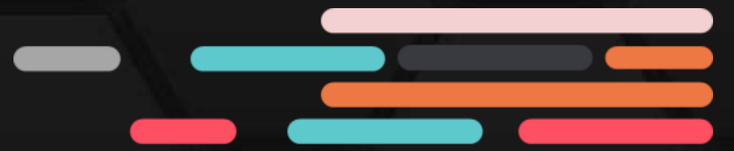
```
string s = "1";

int i = Convert.ToInt32(s);

int j = int.Parse(s);
```

```
int anInteger = 300;
int anInteger2 = 255;

byte aByte = (byte)anInteger;
Console.WriteLine($"integer: {anInteger}, byte : {aByte}");
aByte = (byte)anInteger2;
Console.WriteLine($"integer: {anInteger2}, byte : {aByte}");
```

```
integer: 300, byte : 44
integer: 255, byte : 255
```

```
float aFloat = 34.897f;
int anInteger3 = (int)aFloat;
Console.WriteLine($"float: {aFloat}, integer: {anInteger3}");
```

```
float: 34.897, integer: 34
```

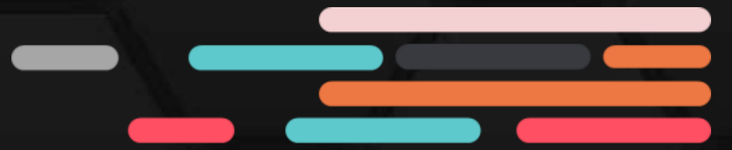


```
string aString = "23";
int anInteger4 = Convert.ToInt32(aString);
int anInteger5 = int.Parse(aString);
Console.WriteLine($"string: {aString}, integer: {anInteger4}");
Console.WriteLine($"string: {aString}, integer: {anInteger5}");
```

```
string: 23, integer: 23
string: 23, integer: 23
```

```
int number5;
numString = "5627.1";
decimal aDecimal = Convert.ToDecimal(numString);
number5 = (int)aDecimal;
Console.WriteLine($"number5 is now {number5}");
```

```
number5 is now 5627
```



```
int a = 2100000000;
int b = 2100000000;
int c = a + b;
Console.WriteLine(c);
```

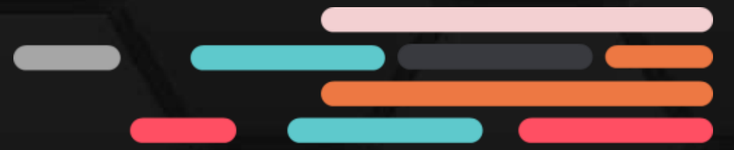
```
-94967296
```

```
int a = 2100000000;
int b = 2100000000;
long c = a + b;
Console.WriteLine(c);
```

```
-94967296
```

```
int a = 2100000000;
int b = 2100000000;
long c = (long)a + (long)b;
Console.WriteLine(c);
```

```
4200000000
```

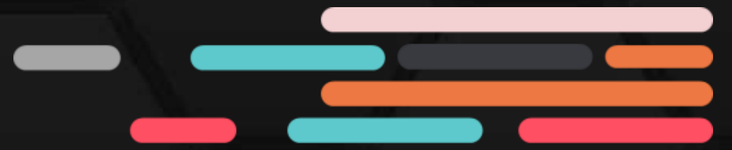


```
int a = 2100000000;
int b = 2100000000;
long c = checked(a + b);
Console.WriteLine(c);
```

```
Unhandled exception. System.OverflowException: Arithmetic operation resulted in an overflow.
   at ConsoleApp.Program.Main(String[] args) in C:\Users\hp\source\repos\ConsoleApp\ConsoleApp
```

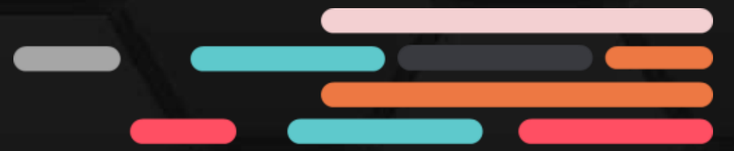
```
int a = 78.3;
int b = 21.7;
long c = checked(a + b);
Console.WriteLine(c);
```

```
double a = 78.3;
float b = 21.7;
long c = checked(a + b);
Console.WriteLine(c);
```



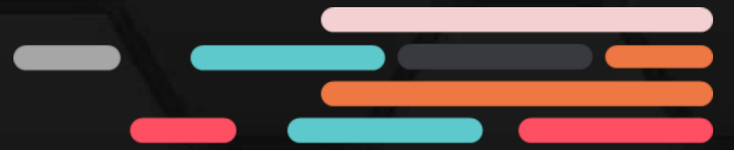
```
decimal a = 78.3m;
decimal b = 21.7m;
decimal c = a + b;
Console.WriteLine($"{a} + {b} equals to {c}");
```

```
78.3 + 21.7 equals to 100.0
```



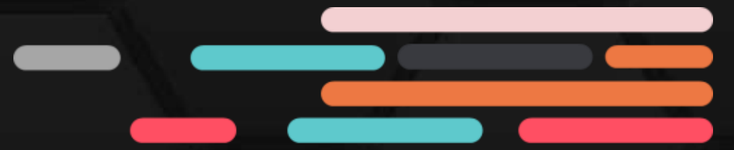
05 Data Collection with List<T>

```
var names = new List<string>() {"Tuah", "Jebat", "Lekir","Le Que","Kasturi" };
names.Add("Kiki");
names.Add("Kaka");      //.Remove
names.Add("Kuku");

foreach (var name in names)
{
    Console.WriteLine($"Hello {name.ToUpper()}");
}

Console.WriteLine(names[0]);
```

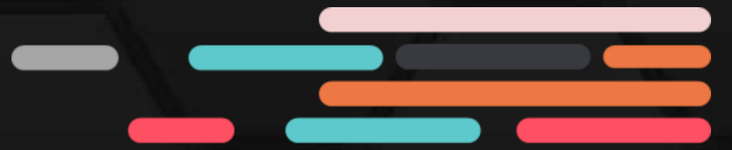


```
var salesMonthly = new List<decimal>() { 230_900.906m, 300_215.459m  };


foreach (var sales in salesMonthly)
{
    Console.WriteLine($" {sales.ToString("N2")}");
}
```

```
230,900.91
300,215.46
```



```
public class Customer
{
    2 references
    public string name { get; set; }
    2 references
    public int custId { get; set; }
}

var customer = new List<Customer>();

customer.Add(new Customer { name = "Alfred, Hitchcock", custId = 207 } );

foreach (var cust in customer)
{
    //Console.WriteLine($"{customer}");
    Console.WriteLine($"Name: {cust.name}, Customer ID: {cust.custId}");
}
```



06