



Adrian D. Finlay

@thewipprogrammer. Writer @hackernoon. Code, LOTS of it. Mangos, LOVE THEM! Barbering. Health. Travel. Business. & more! Network w/ me @ [adriandavid.me/network](https://adriandavid.me/network)

Nov 20, 2017 · 7 min read

## New Language Features in C# 7, 7.1



SD Times C#7

In this article, we will discuss the new features of one of my favourite programming languages, Microsoft's C#! C# is an object oriented, multi-paradigm programming language developed by Microsoft by a team lead by Anders Herjlsberg in 2000. It is a **magnificent language**. It is very similar to the Java programming language.

**An obligatory Hello World for good measure:**

```
using System;

namespace HW //optional
{
    class HelloWorld
    {
        static void Main(string[] args) // args[] also
optional
        {
            Console.WriteLine("Hello, world! C#
Style!");
        }
    }
}
```

---

## New Language Features

### C#7

1. Binary Literals & Digit Separators
2. Tuples & Deconstruction
3. Local Functions
4. Pattern Matching
5. Out variables
6. Expression bodied getters and setters
7. Expression bodied constructors and finalizers
8. Throw Expressions
9. Discards
10. Generalized async return types
11. Ref returns and locals

### C#7.1

1. Async main
2. Default literal expressions
3. Inferred tuple element names

## The Compilers I'll be using

I will be using **.NET Core version 2.0.2** and **Mono JIT compiler version 5.4.0.20**. While .NET Core covers up to C#7.1, mono only partially covers C#7. I will be compiling and running the code on bash on my SUSE Linux box. **I recommend using .NET Core**, as it is the fully featured, official release. I am using Mono here for curiosity's sake and also because I am a fan of the project and I like how quickly the code compiles.

```
adrian@localhost:~$ dotnet --info
.NET Command Line Tools (2.0.2)

Product Information:
  Version: 2.0.2
  Commit SHA-1 hash: a04b4bf512

Runtime Environment:
  OS Name: opensuse
  OS Version: 20171108
  OS Platform: Linux
  RID: linux-x64
  Base Path: /usr/share/dotnet/sdk/2.0.2/

Microsoft .NET Core Shared Framework Host

  Version : 2.0.0
  Build : e0b8861ac7faf042c87a5c2f9f2d84c90b69f20d

adrian@localhost:~$ mono --version
Mono JIT compiler version 5.4.0.201 (tarball Sat Oct 28 13:29:06 UTC 2017)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-project.com
  TLS: __thread
  SIGSEGV: normal
  Notifications: epoll
  Architecture: amd64
  Disabling: none
  Misc: softdebug
  LLVM: supported, not enabled.
  GC: sgen (concurrent by default)

adrian@localhost:~$
```

## Basic use of .NET Core

```
C#7: bash -- Konsole

adrian@localhost:~/Desktop/CODE/C#7$ dotnet new console
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on /home/adrian/Desktop/CODE/C#7/C#7.csproj...
  Restoring packages for /home/adrian/Desktop/CODE/C#7/C#7.csproj...
  Generating MSBuild file /home/adrian/Desktop/CODE/C#7/obj/C#7.csproj.nuget.g.props.
  Generating MSBuild file /home/adrian/Desktop/CODE/C#7/obj/C#7.csproj.nuget.g.targets.
  Restore completed in 379.97 ms for /home/adrian/Desktop/CODE/C#7/C#7.csproj.

Restore succeeded.

adrian@localhost:~/Desktop/CODE/C#7$ dotnet build
Microsoft (R) Build Engine version 15.4.0.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/CODE/C#7/bin/Debug/netcoreapp2.0/C#7.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:03.55
adrian@localhost:~/Desktop/CODE/C#7$ dotnet run
Hello World!
adrian@localhost:~/Desktop/CODE/C#7$
```

## New Features in C#7

### 1) Binary Literals & Digit Separators

```
static void Main(string[] args)
{
    int b = 0b10101000; //This is a binary literal
    int z = 0b10_1010_0110; //These are digit separators
}
```

```

    Console.WriteLine("\nThis is a binary Literal!\t" +
b);
    Console.WriteLine("Binary Literal with digit
separators!:" + z);
}

```

Note that mono does not support the use of this feature while .NET Core does.

The screenshot shows a code editor with a C# file named `BinaryLit.cs`. The code defines a class `BinaryLiterals` with a `Main` method that prints two binary literals. Below the code editor is a terminal window titled `C#7: bash — Konsole`. The terminal shows the command `dotnet build` being executed, which succeeds. Then, the command `dotnet run` is executed, resulting in the output: `This is a binary Literal!: 168` and `Binary Literal with digit separators!: 678`. Finally, the command `mcs BinaryLit.cs -out:BinaryLit -langversion:7` is executed, which fails with two errors: `error CS1525: Unexpected symbol 'b10101000'` and `error CS1525: Unexpected symbol 'b10_1010_0110'`.

```

1  using System;
2
3  namespace C_Sharp_Blog
4  {
5      class BinaryLiterals
6      {
7          static void Main(string[] args)
8          {
9              int b = 0b10101000;
10             int z = 0b10_1010_0110; //These are digit separators
11             Console.WriteLine("\nThis is a binary Literal!\t" + b);
12             Console.WriteLine("Binary Literal with digit separators!:" + z);
13         }
14     }
15 }

```

```

C#7: bash — Konsole
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet build
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/CODE/C#7/C#7/bin/Debug/netcoreapp2.0/C#7.dll

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:02.78
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
This is a binary Literal!: 168
Binary Literal with digit separators!: 678
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd..
adrian@localhost:~/Desktop/CODE/C#7> mcs BinaryLit.cs -out:BinaryLit -langversion:7
BinaryLit.cs(9,12): error CS1525: Unexpected symbol 'b10101000'
BinaryLit.cs(10,12): error CS1525: Unexpected symbol 'b10_1010_0110'
Compilation failed: 2 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>

```

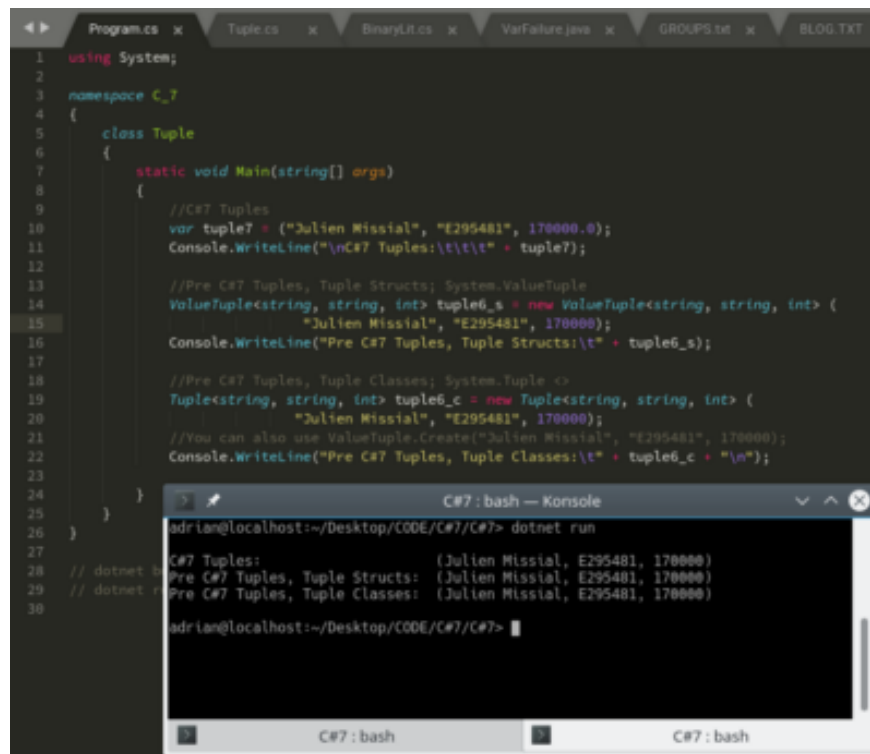
## 2)Tuples, Deconstruction

```

static void Main(string[] args)
{
    //C#7 Tuples
    var tuple7 = ("Julien Missial", "E295481", 170000.0);
    Console.WriteLine("\nC#7 Tuples:\t\t\t" + tuple7);
}

```

Note that mono does not support the use of this feature while .NET Core does.



```
1 using System;
2
3 namespace C_7
4 {
5     class Tuple
6     {
7         static void Main(string[] args)
8         {
9             //C#7 Tuples
10            var tuple7 = ("Julien Missial", "E295481", 170000.0);
11            Console.WriteLine("\nC#7 Tuples:\t\t\t" + tuple7);
12
13            //Pre C#7 Tuples, Tuple Structs; System.ValueTuple
14            ValueTuple<string, string, int> tuple6_s = new ValueTuple<string, string, int> (
15                "Julien Missial", "E295481", 170000);
16            Console.WriteLine("Pre C#7 Tuples, Tuple Structs:\t" + tuple6_s);
17
18            //Pre C#7 Tuples, Tuple Classes; System.Tuple <>
19            Tuple<string, string, int> tuple6_c = new Tuple<string, string, int> (
20                "Julien Missial", "E295481", 170000);
21            //You can also use ValueTuple.Create("Julien Missial", "E295481", 170000);
22            Console.WriteLine("Pre C#7 Tuples, Tuple Classes:\t" + tuple6_c + "\n");
23
24        }
25    }
26 }
27
28 // dotnet run
29 // dotnet run
```

C#7: bash — Konsole

```
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
C#7 Tuples: (Julien Missial, E295481, 170000)
Pre C#7 Tuples, Tuple Structs: (Julien Missial, E295481, 170000)
Pre C#7 Tuples, Tuple Classes: (Julien Missial, E295481, 170000)

adrian@localhost:~/Desktop/CODE/C#7/C#7>
```

```
static void Main(string[] args)
{
    //A Class implementing Deconstruct(), & a Tuple
    Sample sample = new Sample (9, 72.3, "Yellow");
    var sample2 = (21, 103.5, "Turquoise");

    //Deconstruction
    (int x, double y, string z) = sample;
    Console.WriteLine("\nDeconstruction:\t\t" + x +
        ",\t" + y + ",\t" + z + '.');

    //Deconstruction
    (int a, double b, string c) = sample2;
    Console.WriteLine("Deconstruction:\t\t" + a + ",\t" +
        b + ",\t" + c + ".\n");
}
```

```
Program.cs x Deconstruct.cs x LocalFunc.cs x Tuple.cs x BinaryLit.cs x VarFa
1 using System;
2 using System.Text;
3
4 namespace C_7
5 {
6     class Sample {
7         //Properties
8         public int x {get;}
9         public double y {get;}
10        public string z {get;}
11
12        //Constructor
13        public Sample (int x, double y, string z) {
14            this.x = x;
15            this.y = y;
16            this.z = z;
17        }
18
19        //Deconstruct Method
20        public void Deconstruct (out int x, out double y, out string z) {
21            x=this.x;
22            y=this.y;
23            z=this.z;
24        }
25    }
26
```

```
Program.cs x Deconstruct.cs x LocalFunc.cs x Tuple.cs x BinaryLit.cs x VarFa
27 class Tuple
28 {
29     static void Main(string[] args)
30     {
31         //A Class implementing Deconstruct(), & a Tuple
32         Sample sample = new Sample (9, 72.3, "Yellow");
33         var sample2 = (21, 103.5, "Turquoise");
34
35         //Deconstruction
36         (int x, double y, string z) = sample;
37         Console.WriteLine("Deconstruction:\t\t" + x + ",\t" + y + ",\t" + z + ".");
38
39         //Deconstruction
40         (int a, double b, string c) = sample2;
41         Console.WriteLine("Deconstruction:\t\t" + a + ",\t" + b + ",\t" + c + ".\n");
42     }
43 }
44
45
46 // dotnet build
47 // dotnet run
48 // dotnet build --framework
```

```
C#7 : bash — Konsole
adrian@localhost:~/Desktop/C00E/C#7/C#7> dotnet build
Microsoft (R) Build Engine version 15.4.8.50801 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/C00E/C#7/C#7/bin/Debug/netcoreapp2.0/C#7.dll
Build succeeded.
    0 Warning(s)
    0 Error(s)

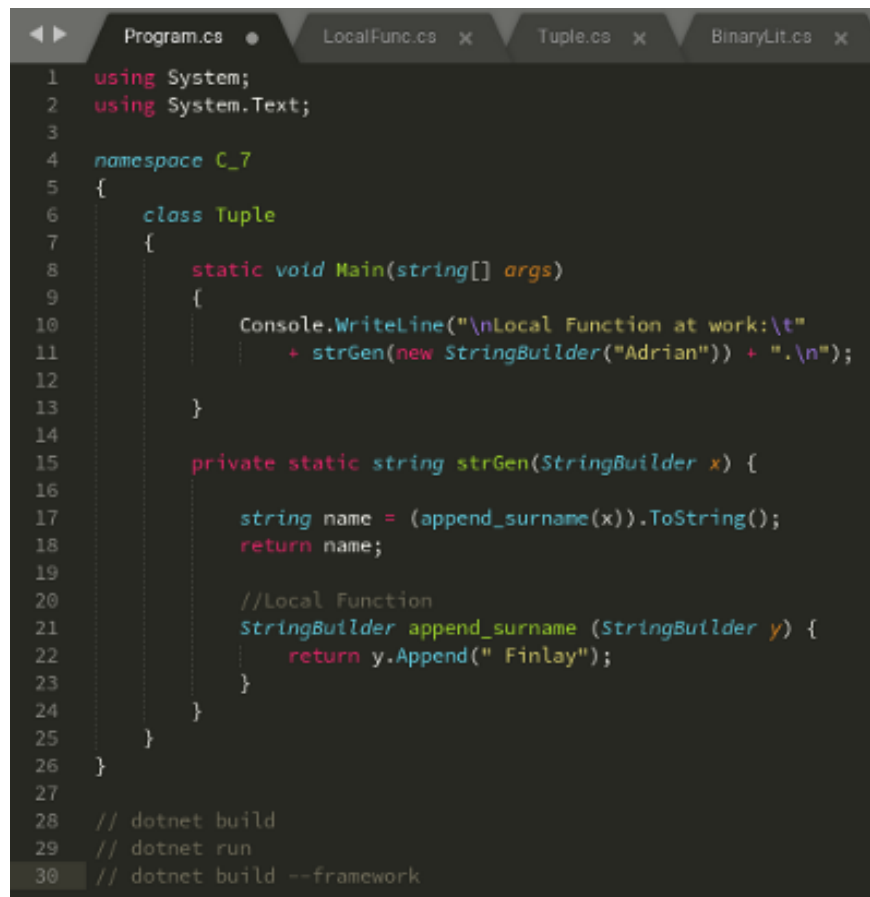
Time Elapsed 00:00:01.36
adrian@localhost:~/Desktop/C00E/C#7/C#7> dotnet run
Deconstruction:      9,      72.3,  Yellow.
Deconstruction:     21,     103.5,  Turquoise.

adrian@localhost:~/Desktop/C00E/C#7/C#7> █

C#7 : bash C#7 : bash
```

### 3) Local Functions

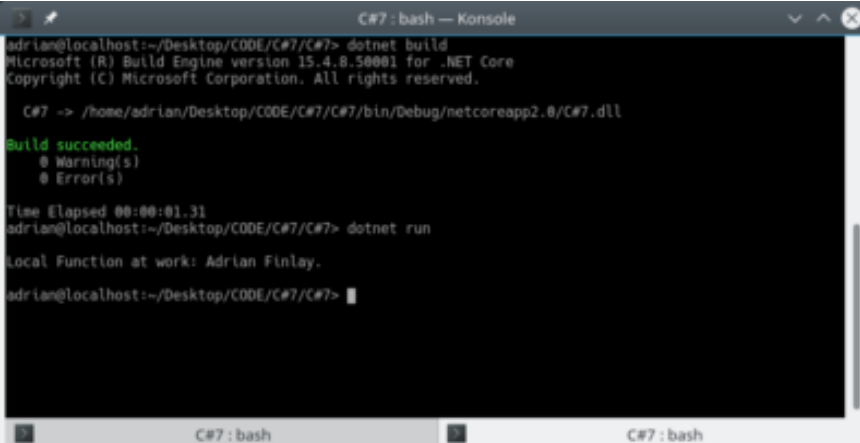
```
private static string strGen(StringBuilder x) {  
  
    string name = (append_surname(x)).ToString();  
    return name;  
  
    //Local Function  
    StringBuilder append_surname (StringBuilder y) {  
        return y.Append(" Finlay");  
    }  
}
```



The screenshot shows a code editor with four tabs: Program.cs, LocalFunc.cs, Tuple.cs, and BinaryLit.cs. The active tab is Program.cs, which contains the following C# code:

```
1  using System;  
2  using System.Text;  
3  
4  namespace C_7  
5  {  
6      class Tuple  
7      {  
8          static void Main(string[] args)  
9          {  
10             Console.WriteLine("\nLocal Function at work:\t"  
11                 + strGen(new StringBuilder("Adrian")) + ".\n");  
12  
13             }  
14  
15             private static string strGen(StringBuilder x) {  
16  
17                 string name = (append_surname(x)).ToString();  
18                 return name;  
19  
20                 //Local Function  
21                 StringBuilder append_surname (StringBuilder y) {  
22                     return y.Append(" Finlay");  
23                 }  
24             }  
25         }  
26     }  
27  
28     // dotnet build  
29     // dotnet run  
30     // dotnet build --framework
```

Note that mono does not support the use of this feature while .NET Core does.



```
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet build
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/CODE/C#7/C#7/bin/Debug/netcoreapp2.0/C#7.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:01.31
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run

Local Function at work: Adrian Finlay.

adrian@localhost:~/Desktop/CODE/C#7/C#7>
```

#### 4)Pattern Matching

```
foreach (var i in lc) {
    //Pattern Matching
    if (i is string ch) {
        Console.Write(ch);
    }
}
```



```
Program.cs x PatternMatch.cs x Deconstruct.cs x
1 using System;
2 using System.Collections.Generic;
3
4 namespace C_Sharp_Blog
5 {
6     class PatternMatch
7     {
8         static void Main(string[] args)
9         {
10             //A Collection
11             List<object> lc = new List<object> (13);
12             lc.AddRange(new List<object> {
13                 "\n", "A", 97, "D", 9,
14                 7, "R", 1, 2, "I", 4,
15                 "A", "N", 11, ".", "\n", "\n"
16             });
17
18             foreach (var i in lc) {
19                 //Pattern Matching
20                 if (i is string ch) {
21                     Console.Write(ch);
22                 }
23             }
24         }
25     }
26 }
27
```

Both mono and .NET Core support the use of this feature with if statements.

```
C#7 : bash — Konsole
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet build
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/CODE/C#7/C#7/bin/Debug/netcoreapp2.0/C#7.dll
Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:02.70
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
ADRIAN.

adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs PatternMatch.cs -out:PatternMatch -langversion:7
adrian@localhost:~/Desktop/CODE/C#7> mono ./PatternMatch
ADRIAN.

adrian@localhost:~/Desktop/CODE/C#7> █

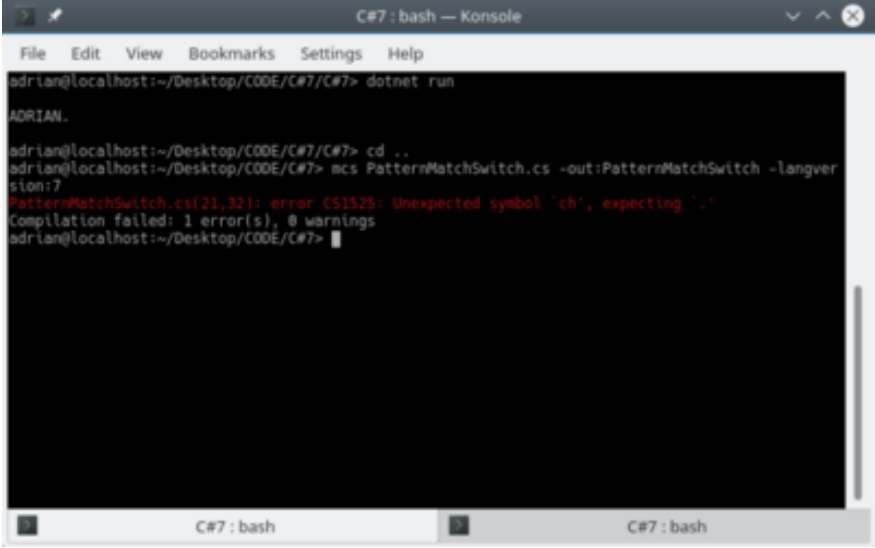
C#7 : bash C#7 : bash
```

```

foreach (var i in lc) {
    //Pattern Matching
    switch (i) {
        case string ch:
            Console.Write(ch);
            break;
    }
}

```

Note that mono does not support the use of this feature with switch statements while .NET Core does.



The screenshot shows a Windows console window titled "C#7: bash — Konsole". The window contains the following text:

```

File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
ADRIAN.
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs PatternMatchSwitch.cs -out:PatternMatchSwitch -langversion:7
PatternMatchSwitch.cs(21,32): error CS1525: Unexpected symbol 'ch', expecting '.'
Compilation failed: 1 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>

```

The error message indicates that the C# compiler (mcs) encountered an unexpected symbol 'ch' at line 21, column 32 of the file PatternMatchSwitch.cs, where it was expecting a period. This is likely due to the use of the 'ch' variable in a switch statement, which is not supported by the mono compiler used in this environment.

## 5)out variable declaration

```

static void Add(double x, double y, out double z)
{
    z = x + y;
}

static void main (string[] args) {
    //C#7
    Add(92.5, 7.49, out var v7); //we can say out double
v7 too
    Console.WriteLine("We did some addition:\t" + v7
+'\\n');
}

```

```
Program.cs x Out.cs x PatternMatchSwitch.cs x PatternMatch.cs x
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     class Out
6     {
7         static void Add(double x, double y, out double z)
8         {
9             z = x + y;
10        }
11
12        static void Main(string[] args)
13        {
14            //Pre-C#7
15            double v6;
16            Add(99, 1, out v6);
17            Console.WriteLine("\nWe did some addition:\t" + v6);
18
19            //C#7
20            Add(92.5, 7.49, out var v7);
21            Console.WriteLine("We did some addition:\t" + v7 + "\n");
22        }
23    }
24 }
25
```

Both compilers are fine with this feature.

```
C#7 : bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet build
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

C#7 -> /home/adrian/Desktop/CODE/C#7/C#7/bin/Debug/netcoreapp2.0/C#7.dll
Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:02.63
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run

We did some addition: 100
We did some addition: 99.99

adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs Out.cs -out:Out -langversion:7
adrian@localhost:~/Desktop/CODE/C#7> mono ./Out

We did some addition: 100
We did some addition: 99.99

adrian@localhost:~/Desktop/CODE/C#7>
```

## 6)Expression-Bodied getters & setters

```
private double VN = 7.1;

public double version_number {
    get => VN;
    set => this.VN = 7;
}
```

```
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     class EB
6     {
7
8         private double VN = 7.1;
9
10        public double version_number {
11            get => VN;
12            set => this.VN = 7;
13        }
14
15        static void Main(string[] args)
16        {
17            /* Expression-Bodied Getters/Setters */
18
19            //get
20            Console.WriteLine("\nThe Current Version Number is:\t" +
21                new EB().version_number);
22
23            //set
24            EB eb = new EB();
25            eb.version_number = 7;
26            Console.WriteLine("The Current Version Number is:\t" +
27                eb.version_number + '\n');
28        }
29    }
30 }
31
32 }
```

Both compilers are fine with this feature.

```
C#7 : bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
The Current Version Number is: 7.1
The Current Version Number is: 7

adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs EB.cs -out:EB -langversion:7
adrian@localhost:~/Desktop/CODE/C#7> mono ./EB
The Current Version Number is: 7.1
The Current Version Number is: 7

adrian@localhost:~/Desktop/CODE/C#7> █
```

## 7) Expression bodied constructors and finalizers

```
class Inner {
    /* Expression-Bodied Constructors/Finalizers */
}
```

```

    Inner () => Console.WriteLine("\nWe've created an
object.");
    ~Inner () => Console.WriteLine("We've finalized an
object.\n");
}

```

```

1  using System;
2
3  namespace C_Sharp_Blog
4  {
5      public class EB_CF
6      {
7          //Nested Class
8          class Inner {
9              //Allocate a lot of memory so hopefully we'll trigger the GC
10             private byte[] Ton = new byte[Int32.MaxValue/4];
11
12             /* Expression-Bodied Constructors/Finalizers */
13             public Inner () => Console.WriteLine("\nWe've created an object.");
14             ~Inner () => Console.WriteLine("We've finalized an object.\n");
15
16             //Will the GC clean up house? Who knows.
17             public void trigger () {
18                 GC.WaitForPendingFinalizers();
19                 GC.Collect();
20             }
21         }
22
23         //main()
24         static void Main(string[] args)
25         {
26             EB_CF.Inner eg = new EB_CF.Inner();
27             eg.trigger();
28         }
29     }
30 }

```

Note that mono does not support the use of this feature while .NET Core does.

```

C#7 : bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
We've created an object.
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs EB_CF.cs -out:EB_CF -langversion:7
EB_CF.cs(13,28): error CS1525: Unexpected symbol '=>', expecting '!', ';', or '{'
EB_CF.cs(14,24): error CS1519: Unexpected symbol '=>' in class, struct, or interface member declaration
EB_CF.cs(14,43): error CS1519: Unexpected symbol '(' in class, struct, or interface member declaration
Compilation failed: 3 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>

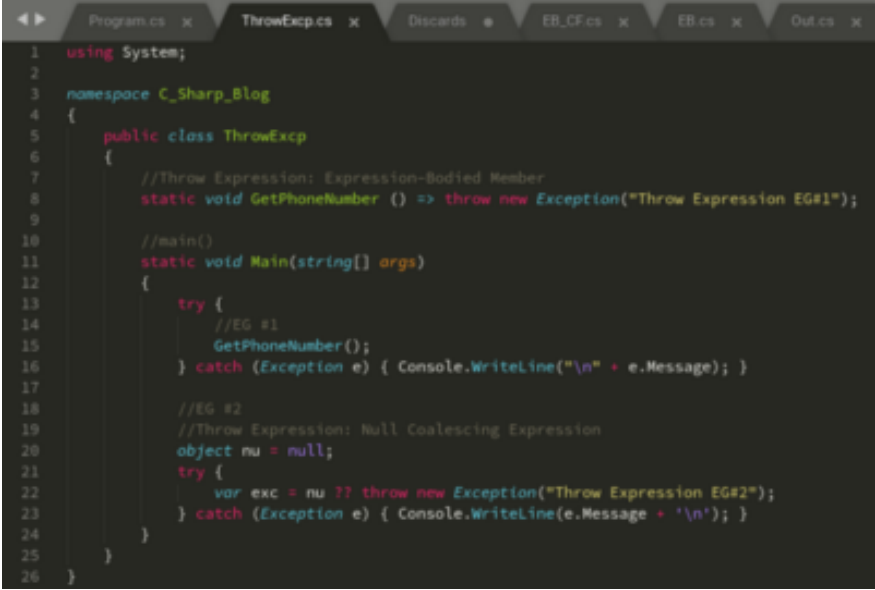
```

## 8)Throw Expressions

```
//Throw Expression: Expression-Bodied Member
static void GetPhoneNumber () => throw new
Exception("Throw Expression EG#1");

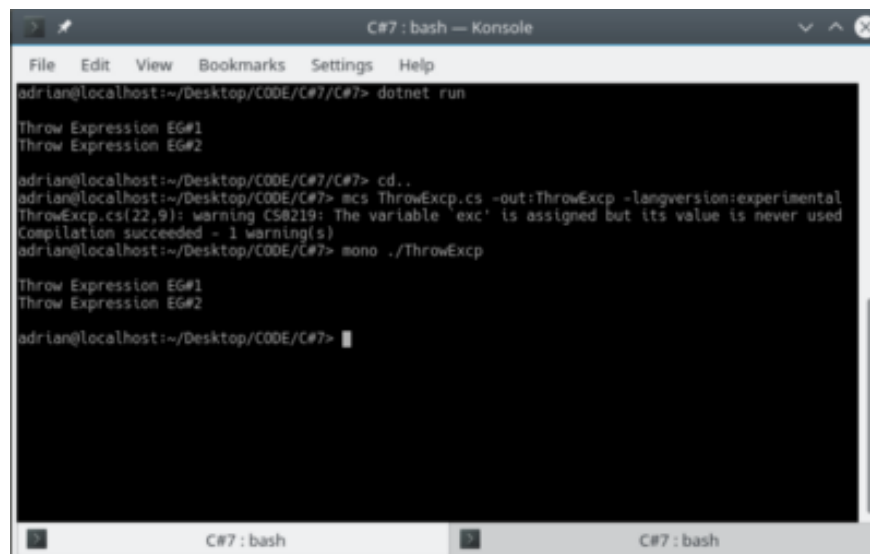
GetPhoneNumber();

//EG #2
//Throw Expression: Null Coalescing Expression
object nu = null;
var exc = nu ?? throw new Exception("Throw Expression
EG#2");
```



```
1  using System;
2
3  namespace C_Sharp_Blog
4  {
5      public class ThrowExcp
6      {
7          //Throw Expression: Expression-Bodied Member
8          static void GetPhoneNumber () => throw new Exception("Throw Expression EG#1");
9
10         //main()
11         static void Main(string[] args)
12         {
13             try {
14                 //EG #1
15                 GetPhoneNumber();
16             } catch (Exception e) { Console.WriteLine("\n" + e.Message); }
17
18             //EG #2
19             //Throw Expression: Null Coalescing Expression
20             object nu = null;
21             try {
22                 var exc = nu ?? throw new Exception("Throw Expression EG#2");
23             } catch (Exception e) { Console.WriteLine(e.Message + '\n'); }
24         }
25     }
26 }
```

Both compilers are fine with this feature. Note that we flipped the—  
langversion:experimental flag for the mono compiler.

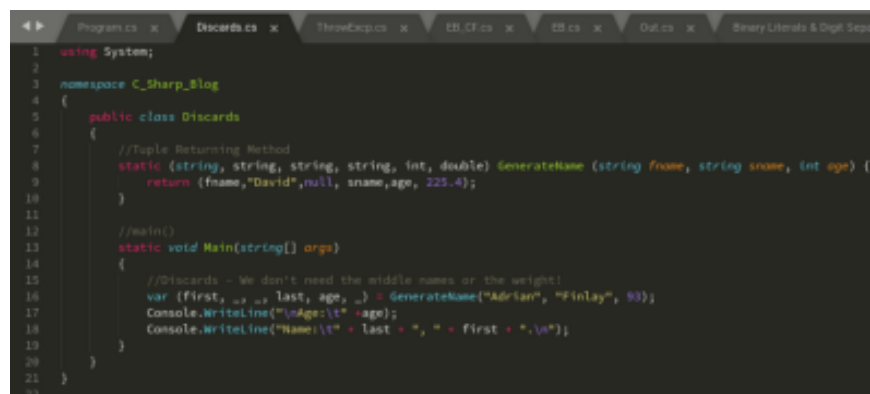


```
C#7: bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
Throw Expression EG#1
Throw Expression EG#2
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd..
adrian@localhost:~/Desktop/CODE/C#7> mcs ThrowExcp.cs -out:ThrowExcp -langversion:experimental
ThrowExcp.cs(22,9): warning CS0219: The variable 'exc' is assigned but its value is never used
Compilation succeeded - 1 warning(s)
adrian@localhost:~/Desktop/CODE/C#7> mono ./ThrowExcp
Throw Expression EG#1
Throw Expression EG#2
adrian@localhost:~/Desktop/CODE/C#7> 
```

## 9) Discards

```
//Tuple Returning Method
static (string, string, string, string, int, double)
GenerateName (string fname, string sname, int age) {
    return (fname,"David",null, sname,age, 225.4);
}

//Discards - We don't need the middle names or the
weight!
var (first, _, _, last, age, _) = GenerateName("Adrian",
"Finlay", 93);
```



```
Program.cs x Discards.cs x ThrowExcp.cs x TB.CF.cs x TB.cs x Out.cs x Binary Literals & Digit Separators
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     public class Discards
6     {
7         //Tuple Returning Method
8         static (string, string, string, string, int, double) GenerateName (string fname, string sname, int age) {
9             return (fname,"David",null, sname,age, 225.4);
10        }
11
12        //Main()
13        static void Main(string[] args)
14        {
15            //Discards - We don't need the middle names or the weight!
16            var (first, _, _, last, age, _) = GenerateName("Adrian", "Finlay", 93);
17            Console.WriteLine($"{age}\t" + age);
18            Console.WriteLine($"Name:\t" + last + ", " + first + ".\n");
19        }
20    }
21 }
22
```

Note that mono does not support the use of this feature while .NET Core does.

```
C#7: bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
Age: 93
Name: Finlay, Adrian.
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs Discards.cs -out:Discards -langversion:experimental
Discards.cs(8,13): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,20): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,28): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,36): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,44): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,49): error CS1519: Unexpected symbol '\',' in class, struct, or interface member de
claration
Discards.cs(8,57): error CS1519: Unexpected symbol '\'),' in class, struct, or interface member de
claration
Discards.cs(8,59): error CS1520: Class, struct, or interface method must have a return type
Discards.cs(9,19): error CS1026: Unexpected symbol '\',' expecting '\'),'
Discards.cs(9,27): error CS1026: Unexpected symbol '\',' expecting '\'),'
C#7: bash C#7: bash
```

## 10) Generalized async return types

```
public async ValueTask<int> Func() {
    await Task.Delay(100);
    return 5;
}
```

This example was **copied directly taken from this source**. If you are aware of classes implementing **GetAwaiter**, I would love to hear from you in the comments below :)

## 11) Ref returns and locals

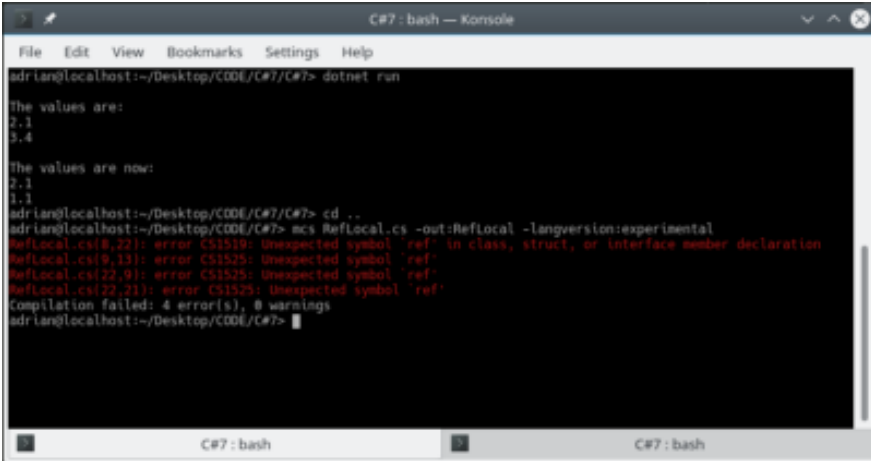
```
//Ref Method
public static ref double ref_local (double [] m) {
    return ref m[1];
}

//Ref Local - By reference, to the second index.
ref var i = ref ref_local (x);
```



```
Program.cs x RefLocal.cs x Discards.cs x ThrowExp.cs x EB_CF.cs x
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     public class RefLocal
6     {
7         //Ref Method
8         public static ref double ref_local (double [] m) {
9             return ref m[1];
10        }
11
12        //main()
13        static void Main(string[] args)
14        {
15            //Array[2] of double
16            double[] x = {2.1, 3.4};
17            //Let's print the contents - 2.1, 3.4
18            Console.WriteLine("\nThe values are:");
19            foreach (double a in x) { Console.WriteLine(a); }
20
21            //Ref Local - By reference, to the second index.
22            ref var i = ref ref_local (x);
23            //Let's change the value at this reference
24            //This will change the value at x[1]
25            i = 1.1;
26
27            //It is now 2.1, 1.1
28            Console.WriteLine("\nThe values are now:");
29            foreach (double a in x) { Console.WriteLine(a); }
30        }
31    }
32 }
```

Note that mono doesn't seem to support the use of this feature (although the website claims that it does) while .NET Core does.



```
C#7: bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
The values are:
2.1
3.4
The values are now:
2.1
1.1
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs RefLocal.cs -out:RefLocal -langversion:experimental
RefLocal.cs(8,22): error CS1519: Unexpected symbol 'ref' in class, struct, or interface member declaration
RefLocal.cs(9,13): error CS1525: Unexpected symbol 'ref'
RefLocal.cs(22,9): error CS1525: Unexpected symbol 'ref'
RefLocal.cs(22,23): error CS1525: Unexpected symbol 'ref'
Compilation failed: 4 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>
```

## New Features in C#7.1

### A Note on Compiler Support for C#7.1

Note that **mono** does not support C# 7.1 at all so far. You may try compiling with the `--langversion:experimental` flag but do not expect it to work.

Note that in order to get **.NET Core** to use C#7.1 you must edit your <project>.csproj file to include the following

```
<PropertyGroup>
  <LangVersion>7.1</LangVersion>
</PropertyGroup>
```

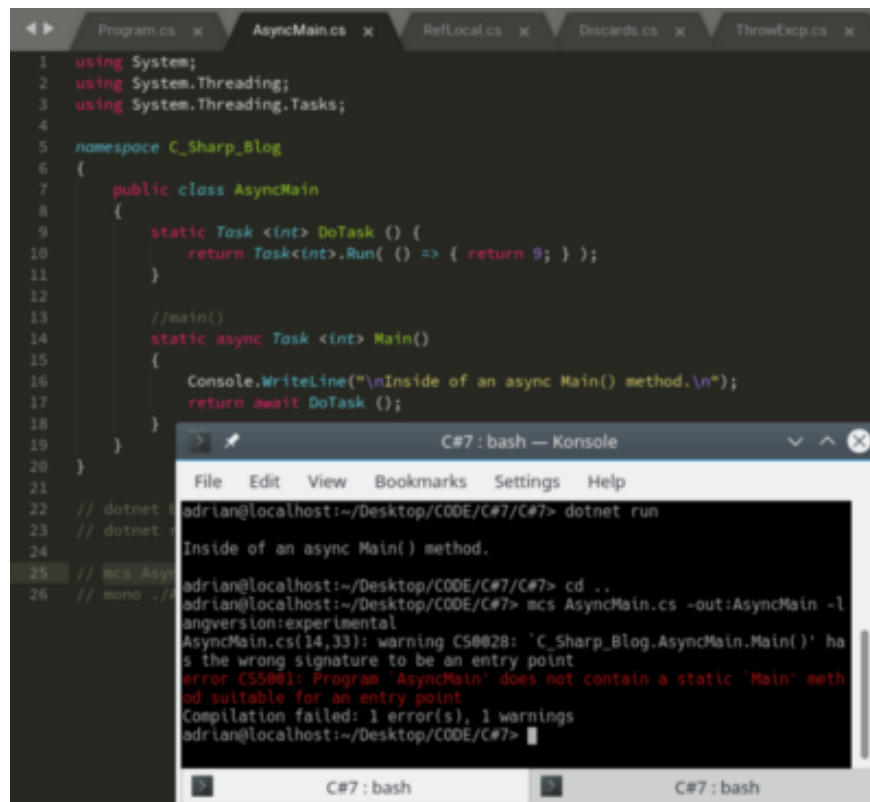
You can also substitute 7.1 for **Latest**, if you desire; This will only work because the latest version is **currently C#7.1**. The more explicit solution is to specifically mark 7.1, but of course, later versions will support everything included in 7.1, so it is of little practical effect in most cases.

After this, the code will compile against C# 7.1.

### 1)Async main

```
static Task <int> DoTask () {
    return Task<int>.Run( () => { return 9; } );
}

//main()
static async Task <int> Main()
{ return await DoTask (); }
```



The screenshot shows a C# IDE with several files open: Program.cs, AsyncMain.cs, RefLocal.cs, Discards.cs, and ThrowExcp.cs. The AsyncMain.cs file contains the following code:

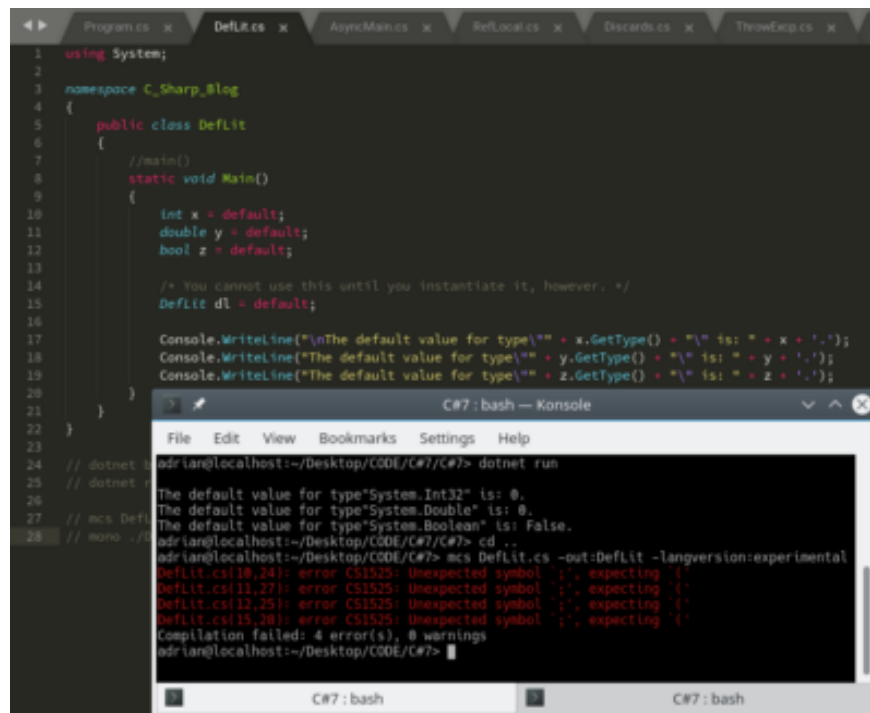
```
1 using System;
2 using System.Threading;
3 using System.Threading.Tasks;
4
5 namespace C_Sharp_Blog
6 {
7     public class AsyncMain
8     {
9         static Task<int> DoTask () {
10             return Task<int>.Run( () => { return 9; } );
11         }
12
13         //main()
14         static async Task<int> Main()
15         {
16             Console.WriteLine("\nInside of an async Main() method.\n");
17             return await DoTask ();
18         }
19     }
20 }
21
22 // dotnet run
23 // dotnet run
24
25 // mcs AsyncMain.cs
26 // mono AsyncMain.exe
```

The terminal window shows the output of the compilation process:

```
C#7: bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
Inside of an async Main() method.
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs AsyncMain.cs -out:AsyncMain -langversion:experimental
AsyncMain.cs(14,33): warning CS0028: 'C_Sharp_Blog.AsyncMain.Main()' has the wrong signature to be an entry point
error CS5001: Program 'AsyncMain' does not contain a static 'Main' method suitable for an entry point
Compilation failed: 1 error(s), 1 warnings
adrian@localhost:~/Desktop/CODE/C#7>
```

## 2)Default literal expressions

```
int x = default;
double y = default;
string z = default;
```



```
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     public class DefLit
6     {
7         //main()
8         static void Main()
9         {
10             int x = default;
11             double y = default;
12             bool z = default;
13
14             /* You cannot use this until you instantiate it, however. */
15             DefLit dl = default;
16
17             Console.WriteLine($"The default value for type{"" + x.GetType() + ""} is: " + x + ".");
18             Console.WriteLine($"The default value for type{"" + y.GetType() + ""} is: " + y + ".");
19             Console.WriteLine($"The default value for type{"" + z.GetType() + ""} is: " + z + ".");
20         }
21     }
22 }
23
24 // dotnet run
25 // dotnet run
26 // mcs DefLit.cs
27 // mono ./DefLit.exe
28 // mcs DefLit.cs -out:DefLit -langversion:experimental
```

```
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
The default value for type"System.Int32" is: 0.
The default value for type"System.Double" is: 0.
The default value for type"System.Boolean" is: False.
adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs DefLit.cs -out:DefLit -langversion:experimental
DefLit.cs(10,24): error CS1525: Unexpected symbol 'default', expecting '{'
DefLit.cs(11,27): error CS1525: Unexpected symbol 'default', expecting '{'
DefLit.cs(12,25): error CS1525: Unexpected symbol 'default', expecting '{'
DefLit.cs(15,20): error CS1525: Unexpected symbol 'default', expecting '{'
Compilation failed: 4 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>
```

### 3) Inferred tuple element names

```
//C#7
var (Istanbul, Dubai, Paris, Bangkok) =
("Istanbul","Dubai","Paris","Bangkok");

var cities = (Istanbul,Dubai,Paris,Bangkok);
//The names of Tuple Elements are inferred
```

```
1 using System;
2
3 namespace C_Sharp_Blog
4 {
5     public class InferTupleName
6     {
7         //main()
8         static void Main()
9         {
10             //This is how named tuples work
11             Console.WriteLine("\nThis is how named tuples work.");
12             var name = (first: "Adrian", sur:"Finlay");
13             Console.WriteLine("My name is: " + name.sur + ", " + name.first + ".");
14
15             //Pre C#7
16             var cities6 = (Istanbul: "Istanbul", Dubai: "Dubai", Paris: "Paris", Bangkok: "Bangkok");
17             //C#7
18             var (Istanbul, Dubai, Paris, Bangkok) = ("Istanbul","Dubai","Paris","Bangkok");
19             var cities = (Istanbul,Dubai,Paris,Bangkok); //The names of Tuple Elements are inferred
20
21             Console.WriteLine("\nFamous Cities around the world:");
22
23             //Observe the auto inferred Tuple Names: The name is inferred from the variable name
24             Console.WriteLine(cities.Istanbul);
25             Console.WriteLine(cities.Dubai);
26             Console.WriteLine(cities.Paris);
27             Console.WriteLine(cities.Bangkok);
28
29             Console.WriteLine();
30         }
31     }
32 }
```

```
C#7: bash — Konsole
File Edit View Bookmarks Settings Help
adrian@localhost:~/Desktop/CODE/C#7/C#7> dotnet run
This is how named tuples work.
My name is: Finlay,Adrian.

Famous Cities around the world:
Istanbul
Dubai
Paris
Bangkok

adrian@localhost:~/Desktop/CODE/C#7/C#7> cd ..
adrian@localhost:~/Desktop/CODE/C#7> mcs InferTupleName.cs -out:InferTupleName -langversion:experimental
InferTupleName.cs(12,26): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(12,41): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(16,32): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(16,51): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(16,67): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(16,85): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(18,60): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(18,68): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(18,76): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(18,86): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(19,31): error CS1026: Unexpected symbol ':', expecting ')'
InferTupleName.cs(19,51): error CS1525: Unexpected symbol ':', expecting ';' or ';'
Compilation failed: 12 error(s), 0 warnings
adrian@localhost:~/Desktop/CODE/C#7>
```

Want the source? Grab it here.

afinlay5/CSharp7

NET Core source code repository for C#7, 7.1 source code examples posted on personal blog...

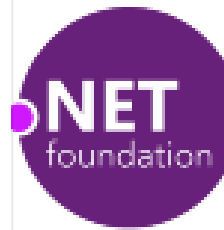
github.com



**Curious about upcoming releases of C#?  
See what's coming in C#7.2 and 8.0, here.**

dotnet/roslyn

roslyn - The .NET Compiler Platform ("Roslyn")  
provides open-source C# and Visual Basic...  
github.com



**Like the new features in C# 7, 7.1? Let me  
know in the comments below :)**



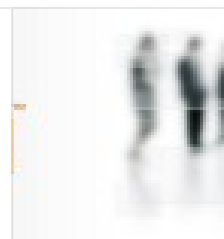
Looney Tunes Ending [4]

**Interested in Java? Join my Java group on  
Facebook:**

Join My Java Facebook Group

Interested in Java? Check out my Facebook Group:  
Java Software Development Group!

medium.com



## Like my Content? Subscribe to my mailing list:

This embedded content is from a site that does not comply with the Do Not Track (DNT) setting now enabled on your browser.

Please note, if you click through and view it anyway, you may be tracked by the website hosting the embed.

[Learn More about Medium's DNT policy](#)

**Don't forget to give it a.... ;)**



!Emoji.com





