Not Your Mother or Father's C#

Brendan Enrick
🐘: @Brendoneus@our.devchatter.com
🐦: @Brendoneus
📓 : Brendoneus.com
▶: YouTube and Twitch: @DevChatter

Disclaimers

- My father never used C# and was mostly using PHP before retiring
- My mother is not a programmer, so also did not use C#
- IS EVERYTHING A LIE?!?!
    - Probably
- IS YOUR FIRST NAME EVEN BRENDAN?!
    - No

Your Father or Mother’s C#

Outdated C# information.

What Some People Think is True About C#

What Some People Think is True About C#

C# will require Visual Studio and Windows to develop.

What Some People Think is True About C#

C# will only run on Windows.

What Some People Think is True About C#

My C# web app will need IIS to host it.

What Some People Think is True About C#

C# is just {Curly Braces} and indentation halfway across the screen.

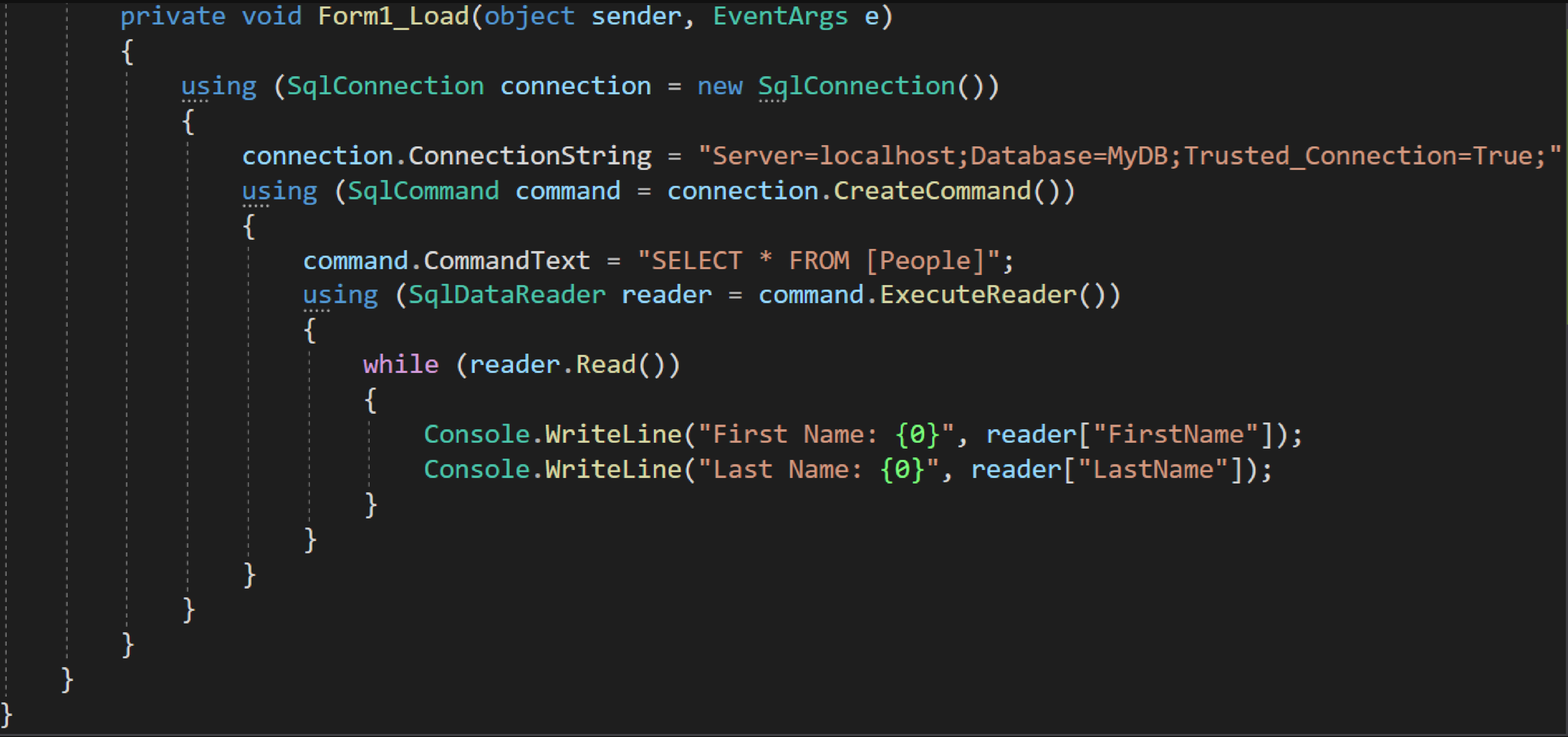
What Some People Think is True About C#

C# is just like Java

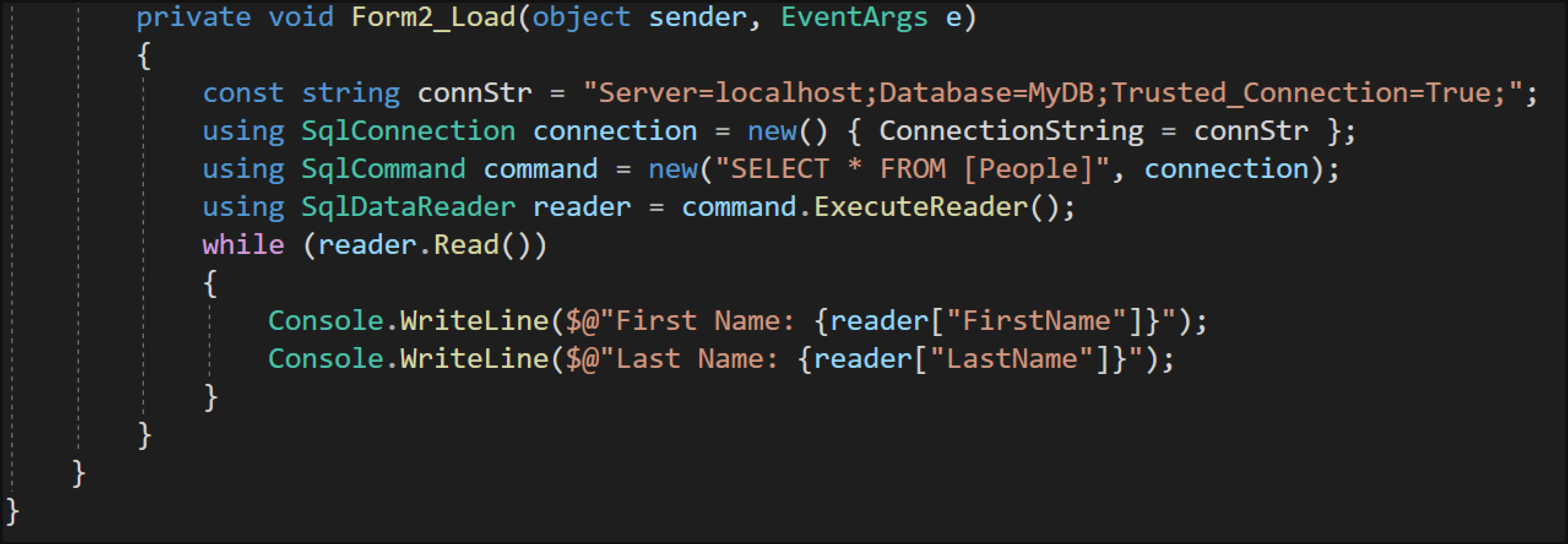
What Some People Think is True About C#

C# is just for monolithic business applications.

Curly Braces and Indentation

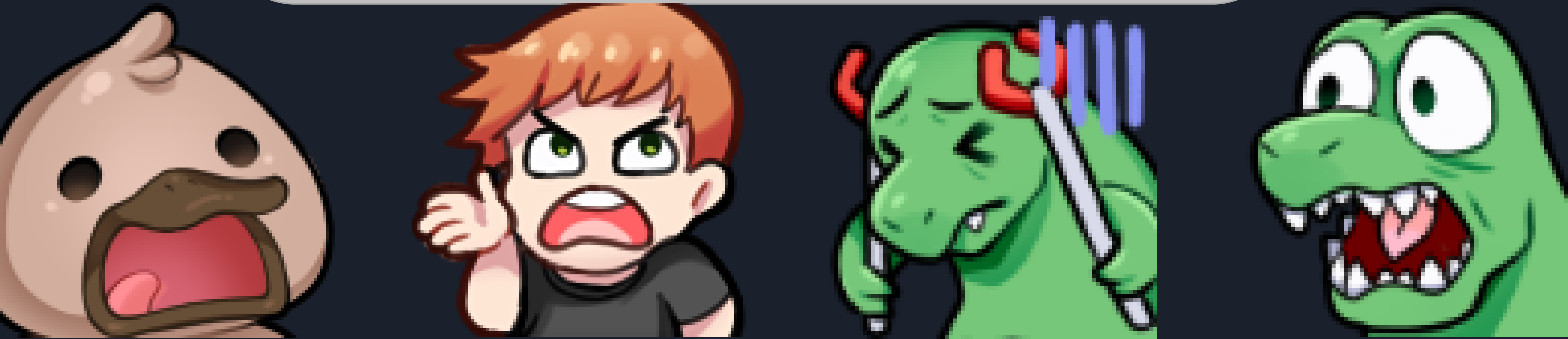


```
        private void Form1_Load(object sender, EventArgs e)
        {
            using (SqlConnection connection = new SqlConnection())
            {
                connection.ConnectionString = "Server=localhost;Database=MyDB;Trusted_Connection=True;"
                using (SqlCommand command = connection.CreateCommand())
                {
                    command.CommandText = "SELECT * FROM [People]";
                    using (SqlDataReader reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            Console.WriteLine("First Name: {0}", reader["FirstName"]);
                            Console.WriteLine("Last Name: {0}", reader["LastName"]);
                        }
                    }
                }
            }
        }
    }
}
```



```
        private void Form2_Load(object sender, EventArgs e)
        {
            const string connStr = "Server=localhost;Database=MyDB;Trusted_Connection=True;";
            using SqlConnection connection = new() { ConnectionString = connStr };
            using SqlCommand command = new("SELECT * FROM [People]", connection);
            using SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                Console.WriteLine($@"First Name: {reader["FirstName"]}");
                Console.WriteLine($@"Last Name: {reader["LastName"]}");
            }
        }
    }
}
```

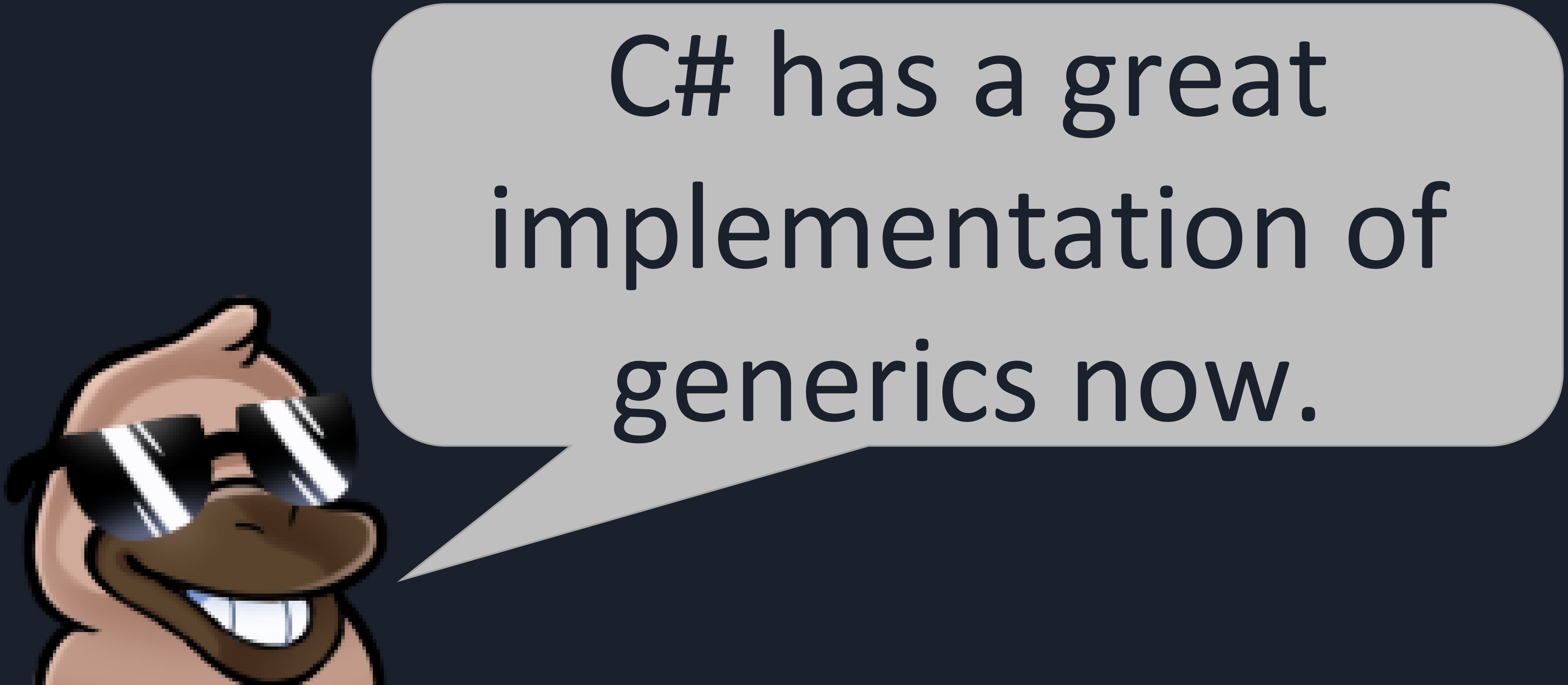
C# really was missing some basics.



Parent's C#

No generic types.

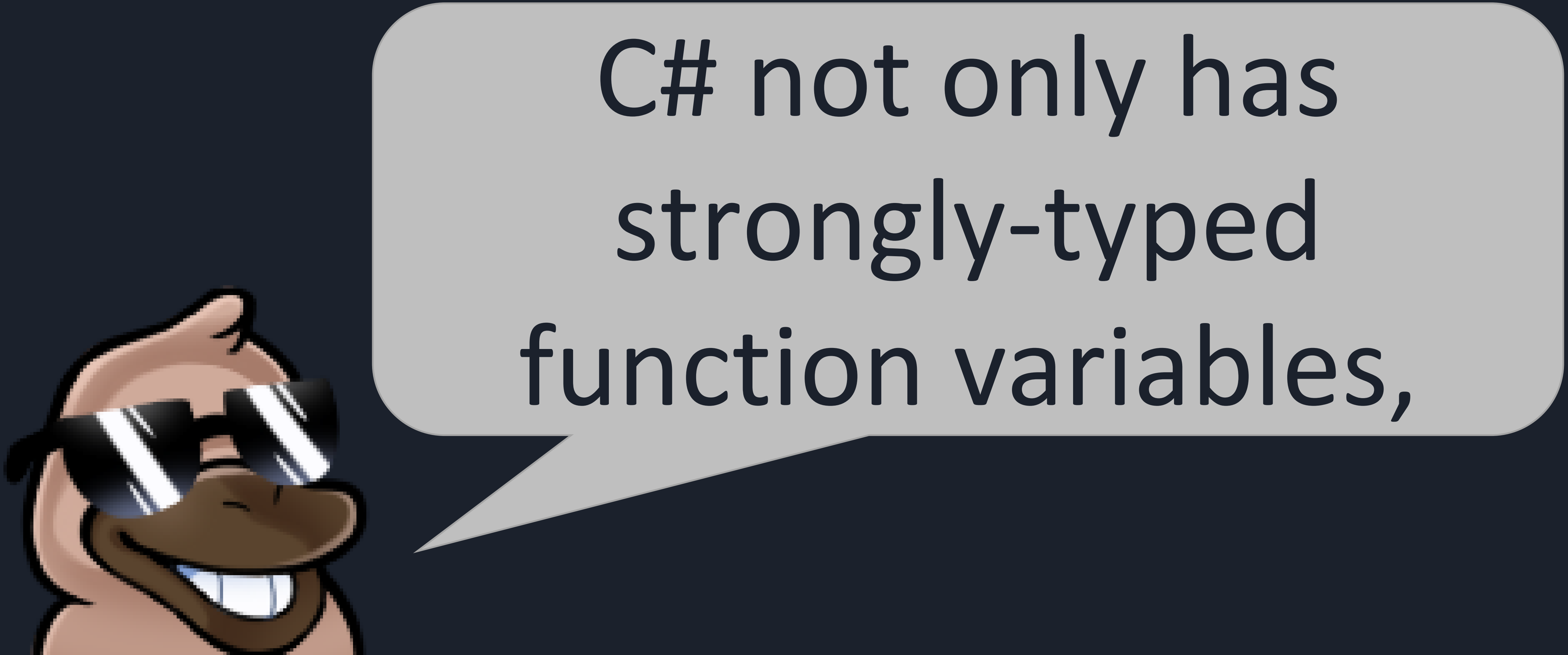
Our C#



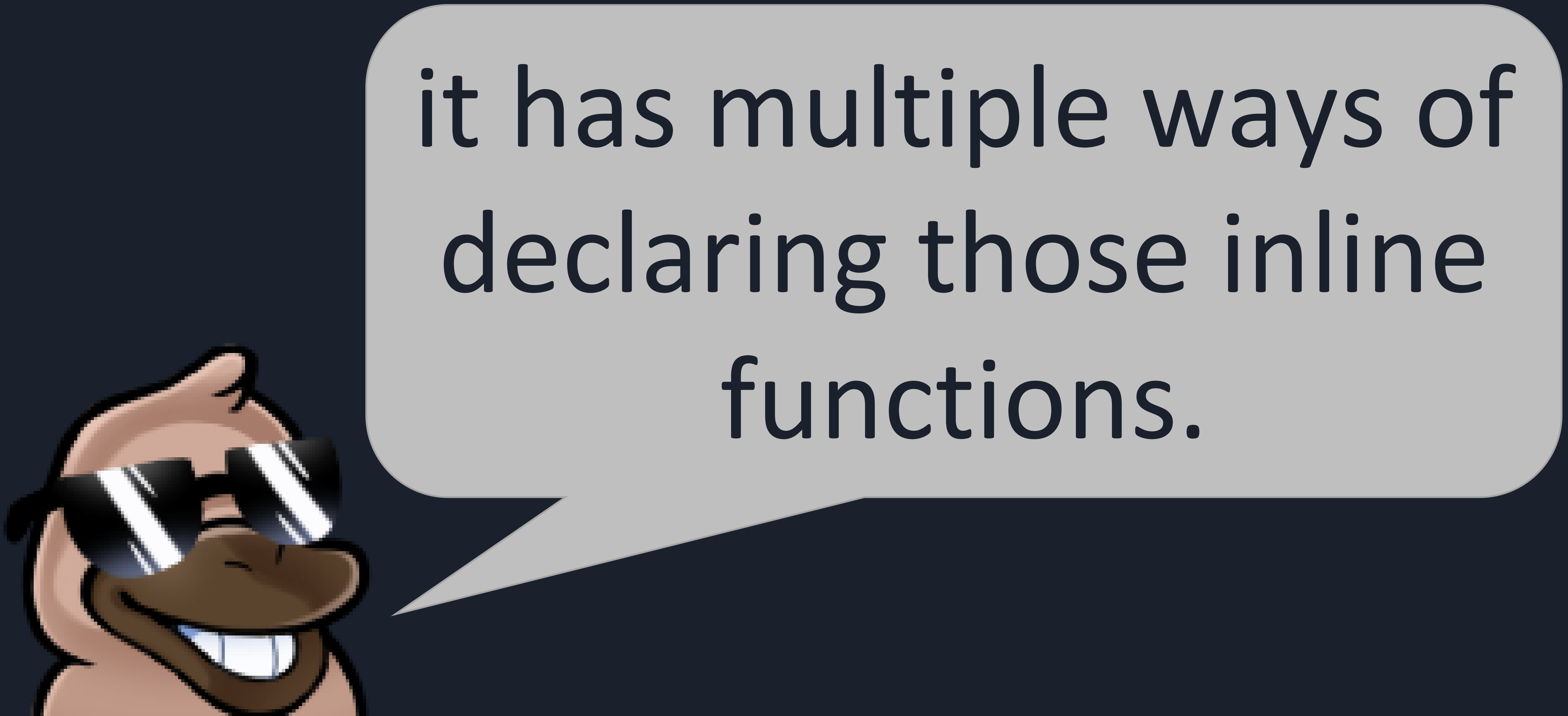
Parent’s C#

No lambda functions!

Our C#



Our C#



The “Tuple” type is a joke.

We got our first “real” tuples 4 years ago.

And each new version improves something.

What about async and await?

That’s nearing a decade, and other languages

adopted the syntax
used by C#
(sometimes indirectly)

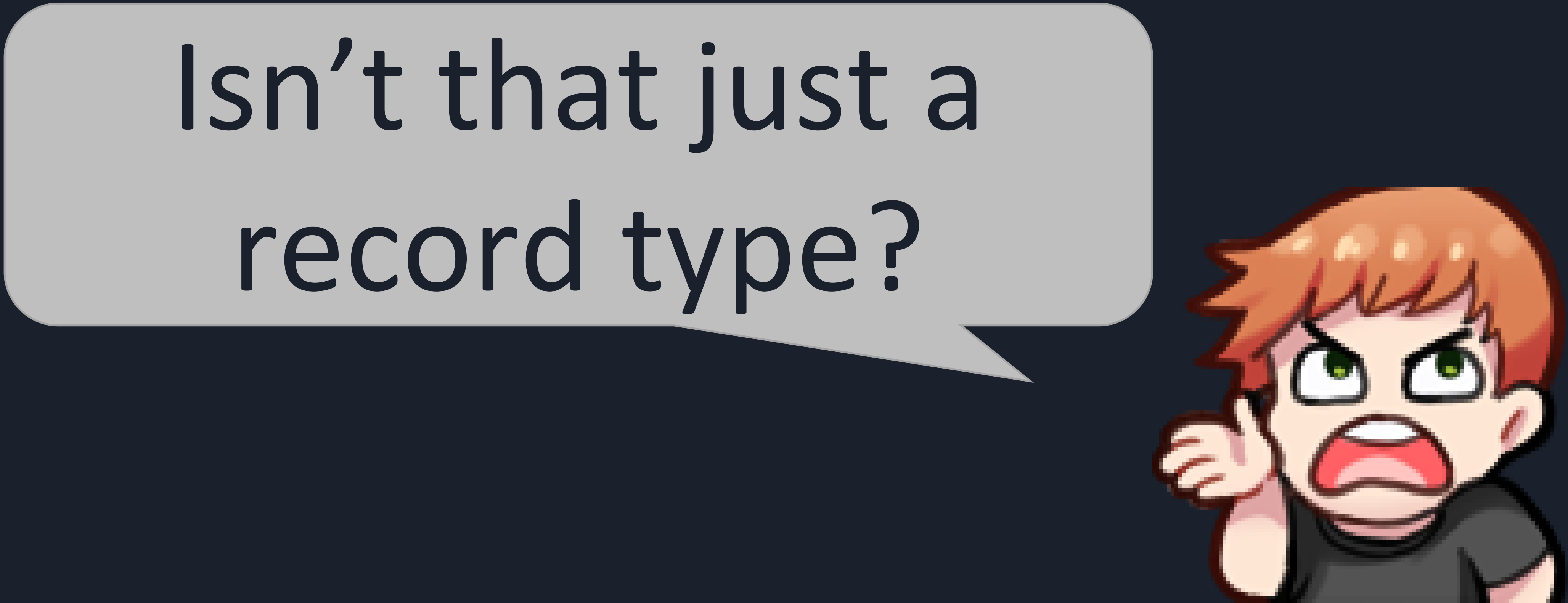
C#

Recent New Features

Primary Constructors C# 12

Primary Constructor

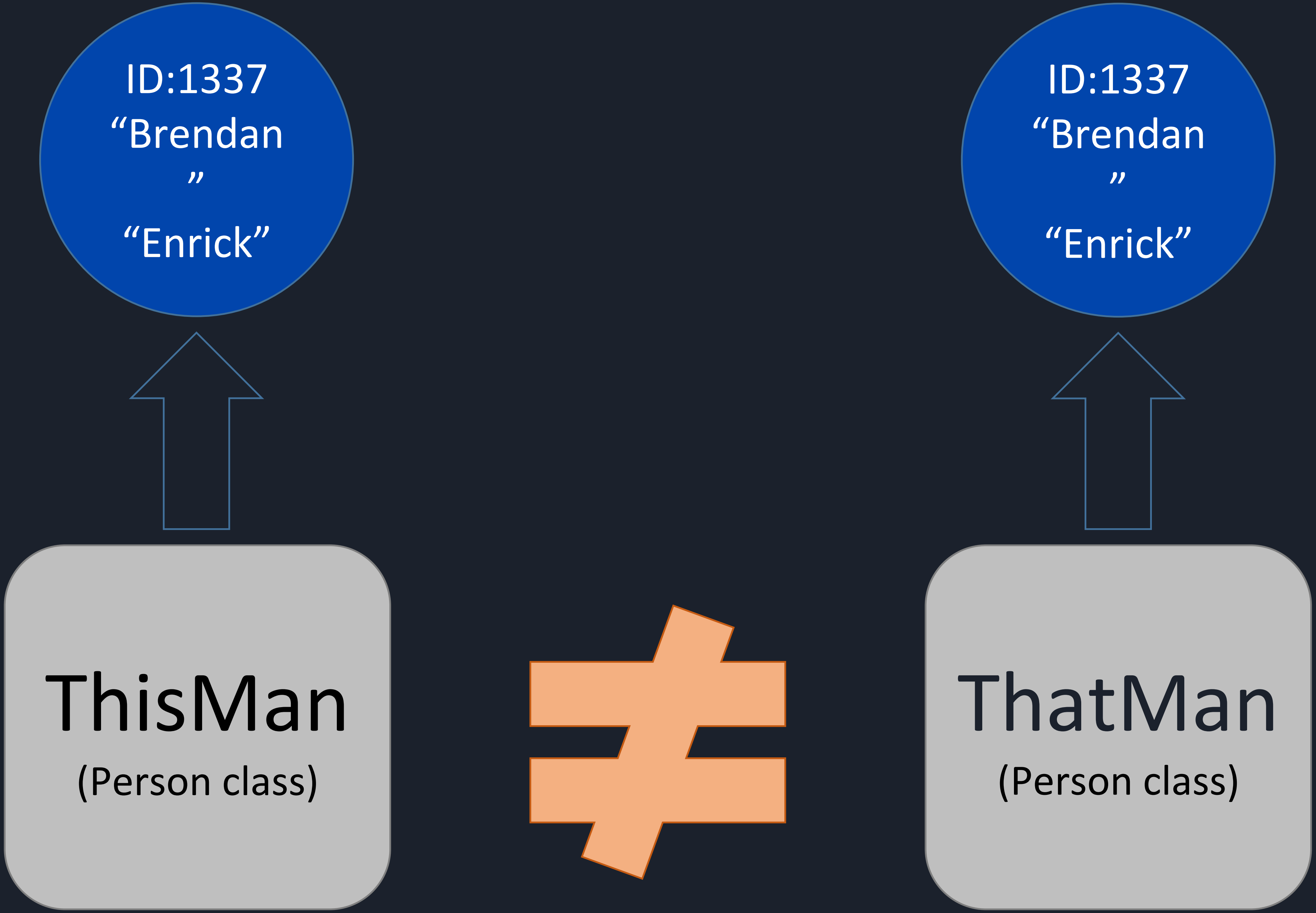
```
public readonly class Rectangle(int length, int width)
{


}
```

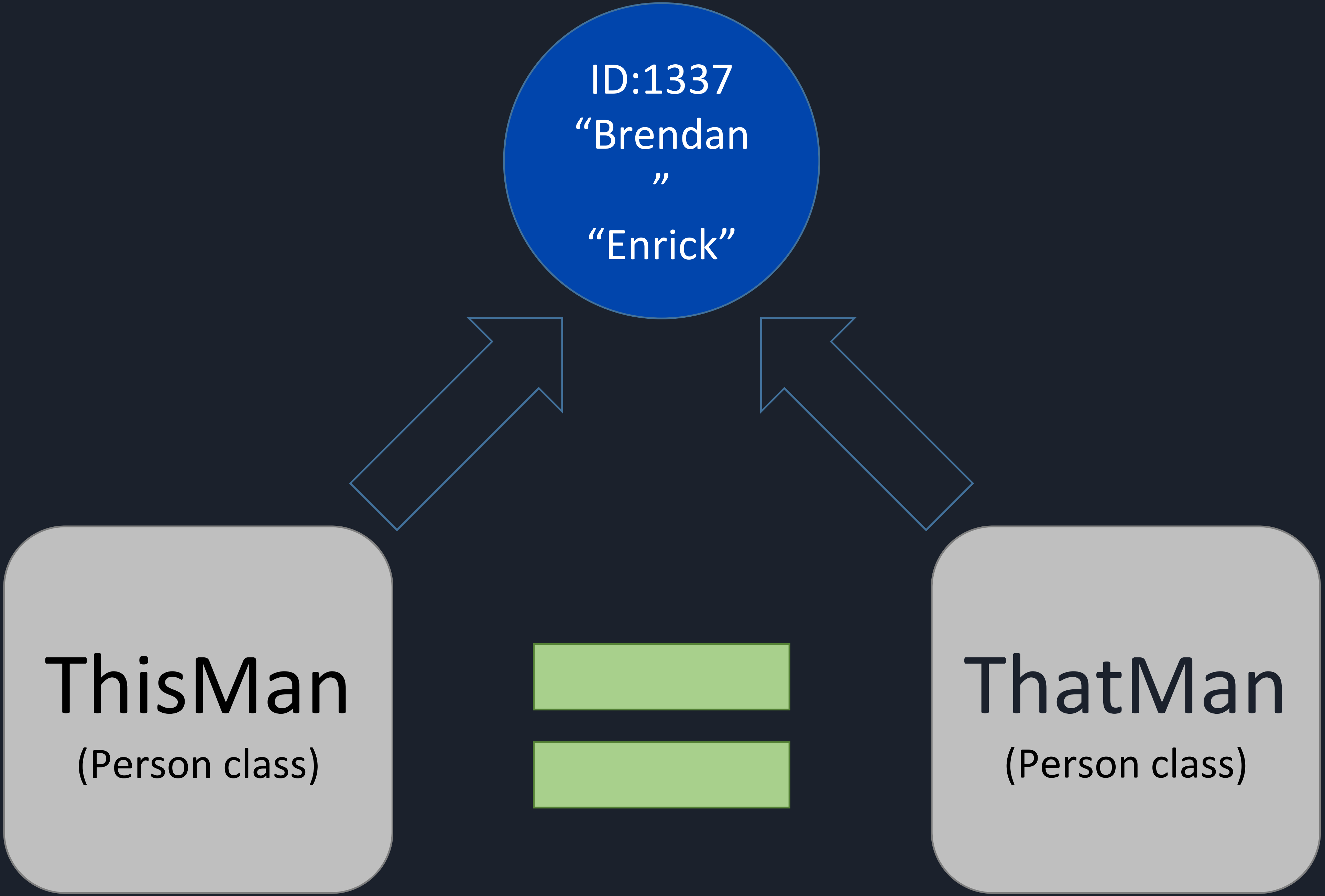


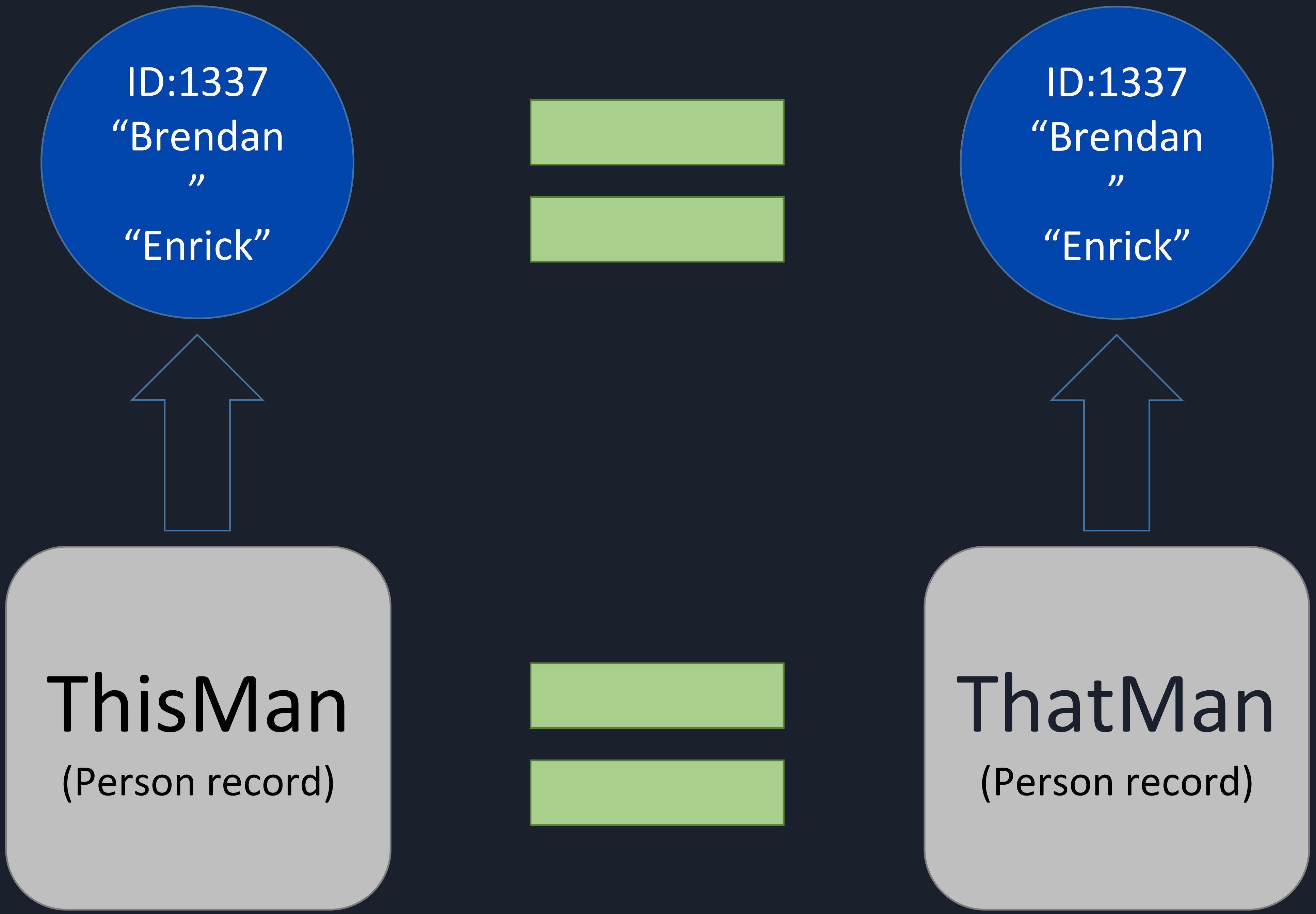
Record Types in C# 9

Record Type Declaration

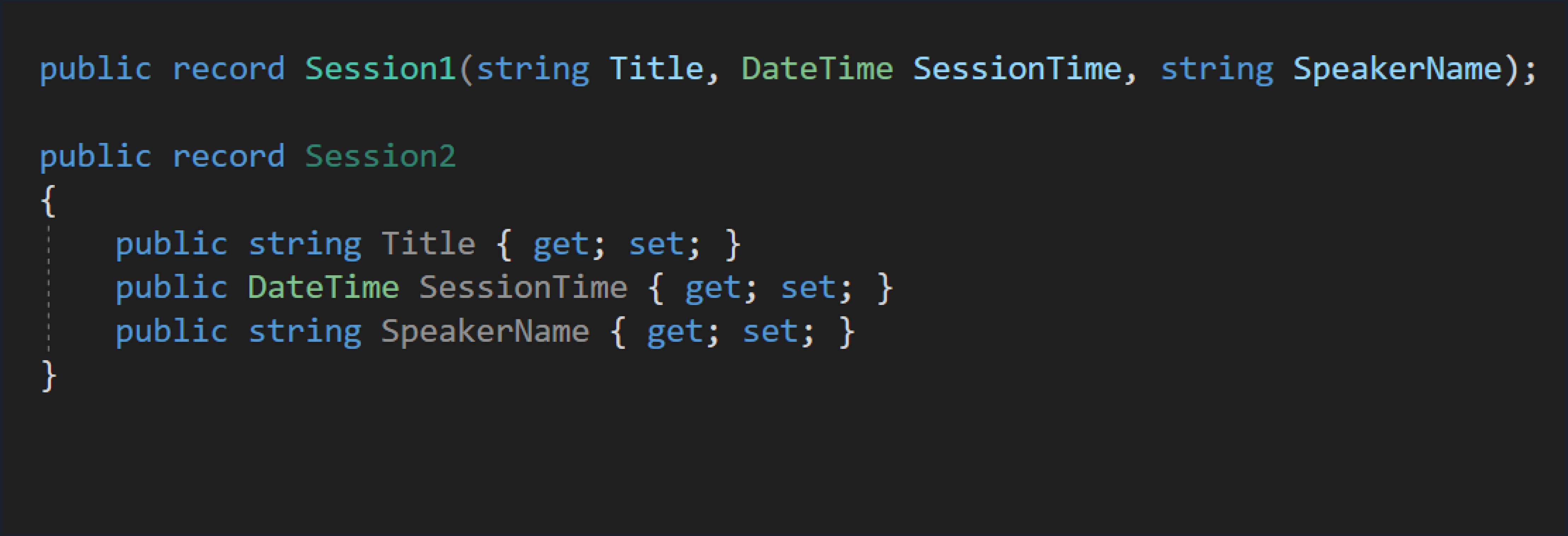
```
public record Session1(string Title, DateTime SessionTime, string SpeakerName);
```


ID:1337
“Brendan”
“Enrick”
ID:1337
“Brendan”
“Enrick”
ThisMan
(Person class)
≠
ThatMan
(Person class)


ID:1337
“Brendan”
“Enrick”
ThisMan
(Person class)
ThatMan
(Person class)

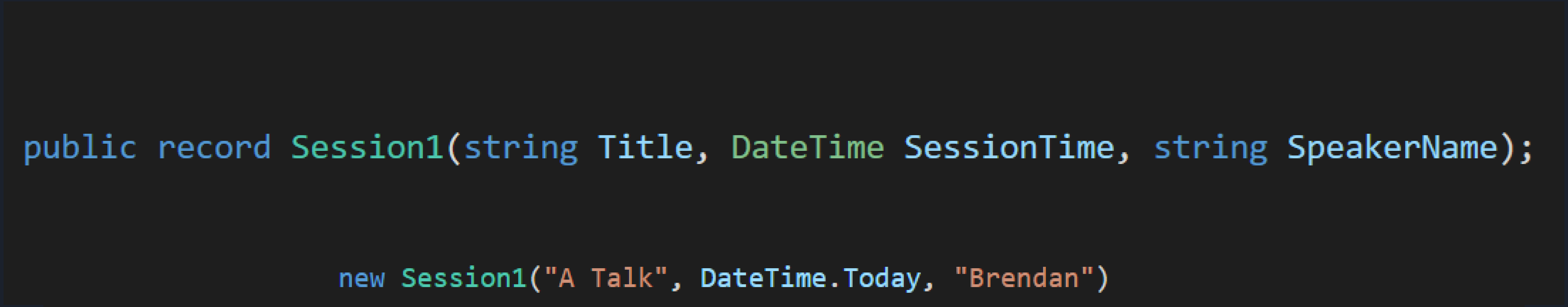

ID:1337
“Brendan”
“Enrick”
ID:1337
“Brendan”
“Enrick”
=
ThisMan
(Person record)
=
ThatMan
(Person record)

Record Without Constructor

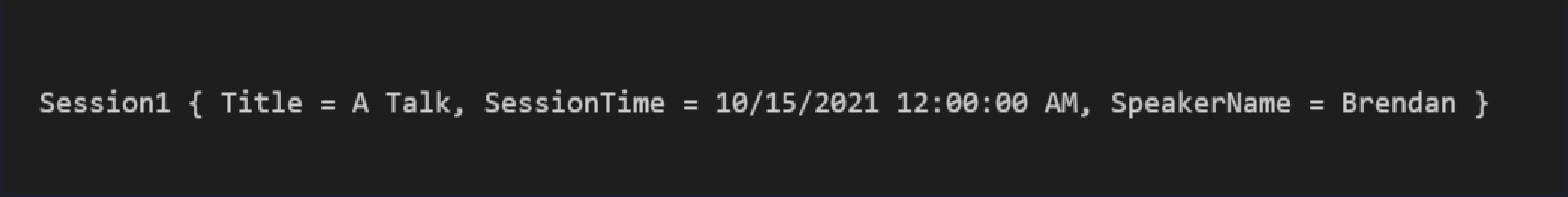


```
public record Session1(string Title, DateTime SessionTime, string SpeakerName);

public record Session2
{
    public string Title { get; set; }
    public DateTime SessionTime { get; set; }
    public string SpeakerName { get; set; }
}
```

Default ToString() Example



```
public record Session1(string Title, DateTime SessionTime, string SpeakerName);

            new Session1("A Talk", DateTime.Today, "Brendan")
```



```
Session1 { Title = A Talk, SessionTime = 10/15/2021 12:00:00 AM, SpeakerName = Brendan }
```

You can create new instances from existing records.

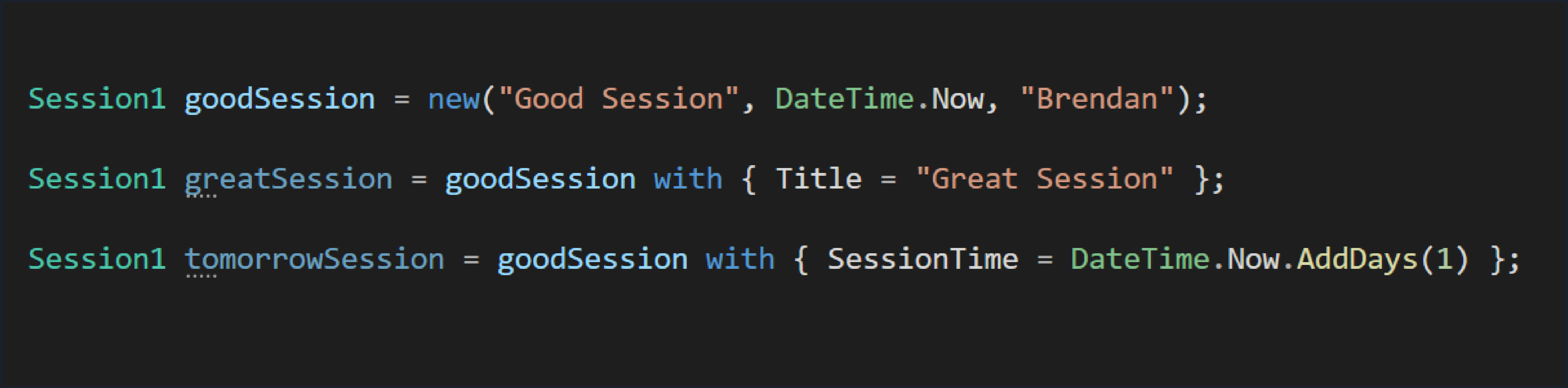
What do you do with it?

Yes, exactly.

What?

No, not “what”, “with”.

Example of using “with”



```
Session1 goodSession = new("Good Session", DateTime.Now, "Brendan");

Session1 greatSession = goodSession with { Title = "Great Session" };

Session1 tomorrowSession = goodSession with { SessionTime = DateTime.Now.AddDays(1) };
```

Constructor Inferences

Implicit Constructor Examples

```
Session mySession = new("This", DateTime.Today, "Brendan");

Session more = new("This", new(2021, 10, 15), "Brendan");
```

Init-Only Setters in C# 9

Init-Only Setters (Person Example)

```
// Set Only In Constructor or Inline
public DateTime BirthDate { get; }

// Set In Construction or Initialization
public string BirthName { get; init; }

// Set At Any Time
public string Name { get; set; }
```

Init-Only Setters Example

```
Person result = new(DateTime.Now)
{
    BirthName = "Brendan"
};
result.Name = "Brendoneus";
return result;
```

String Interpolation

String Interpolation in C# 6

```
string session = $"{title} - presented by {speaker}"
```

String Interpolation in C# 10

```
const string session = $"{title} - presented by {speaker}"
```

Verbatim Strings (before C# 11)

```
string review = @"
    My favorite sessions at CodeMash 2024:
     - ""Mastering TDD in Legacy Code""
     - ""Not Your Mother's or Father's C#""
     - ""Balloon Animals: Blowing things Up at Codemash""
    ";
```

Raw String Literals in C# 11

```
string review = """
    My favorite sessions at CodeMash 2024:
     - "Mastering TDD in Legacy Code"
     - "Not Your Mother's or Father's C#"
     - "Balloon Animals: Blowing things Up at Codemash"
    """;
```

Interpolating Raw String Literals in C# 11

```
string review = $$"""
    My favorite sessions at CodeMash 2024:
     - "Mastering TDD in Legacy Code"
     - "Not Your Mother's or Father's C#"
     - "Balloon Animals: Blowing things Up at Codemash"
     - "{favoriteSession}"
    """;
```

Pattern Matching

Pattern Matching History

C# 7 - Added

C# 7.1 - Generics

C# 8 - Switches, Properties, Tuples, Positional

C# 9 - Types, Boolean on multiple patterns, Relational (< > >= <=)

C# 10 - Nested Property Patterns

C# 11 - Span<char>

Pattern Matching

Pattern Matching – Is Expressions

```
Rectangle rec = shape as Rectangle;
if (rec != null)
    return rec.Length * rec.Width;
```

Pattern Matching – Is Expressions

```
if (shape is Rectangle rec)
    return rec.Length * rec.Width;
```

Pattern Matching – Is Expressions

```
public static int CalculateArea(Shape shape)
{
    if (shape is null) return 0;
    if (shape is Rectangle rec)
        return rec.Length * rec.Width;
    if (shape is Triangle tri)
        return tri.Base * tri.Height / 2;
    return 0;
}
```

Pattern Matching – Property Patterns (C# 8)

```csharp
if (session is { State: "OH", Name: "Momentum" })
{
    return "You are here.";
}
```

Nested Property Patterns (C# 10)

C# 8

```
is { Addr: { State: "OH"} }
```

C# 10

```
is { Addr.State: "OH" }
```

List Patterns (C# 11)

```
string[] grades = { "A", "A", "B", "A", "B" };

// True
numbers is ["A", "A", "B", "A", "B"] // Full Match
numbers is ["A", .., "B"] // Check start and end
numbers is [.., "B", "A", "B"] // Check end
numbers is ["A", "A", ..] // Check start
numbers is ["A" or "B", ..] // Boolean Patterns (C# 9)
```

List Patterns (C# 11)

```
string[] grades = { "A", "A", "B", "A", "B" };

// False
numbers is ["A", "A", "B", "A", "A"];  // Wrong Value
numbers is ["A", .., "A"];  // Wrong Value
numbers is ["A", "A", "B", "A", "B", "A"];  // Too Many Items
```

Span<char> Patterns on string (C# 11)

```
public bool IsBrendan(Span<char> name)
{
    return name is "Brendan";
}
public bool StartsWithB(Span<char> name)
{
    return name is ['B', ..]; // with List Pattern
}
```

required modifier

required modifier on properties - C# 11

```
public User() { }

[SetsRequiredMembers]
public User(int id, string name) => (Id, Name) = (id, name);

public required int Id { get; init; }
public required string Name { get; init; }
public string? Email { get; set; }
```

file Access Modifier

file Access Modifier - C# 11

```
file enum RelatedEnum // Used Internally Only
{
    Enabled,
    Disabled,
    Unknown
}
```

Collection Expressions

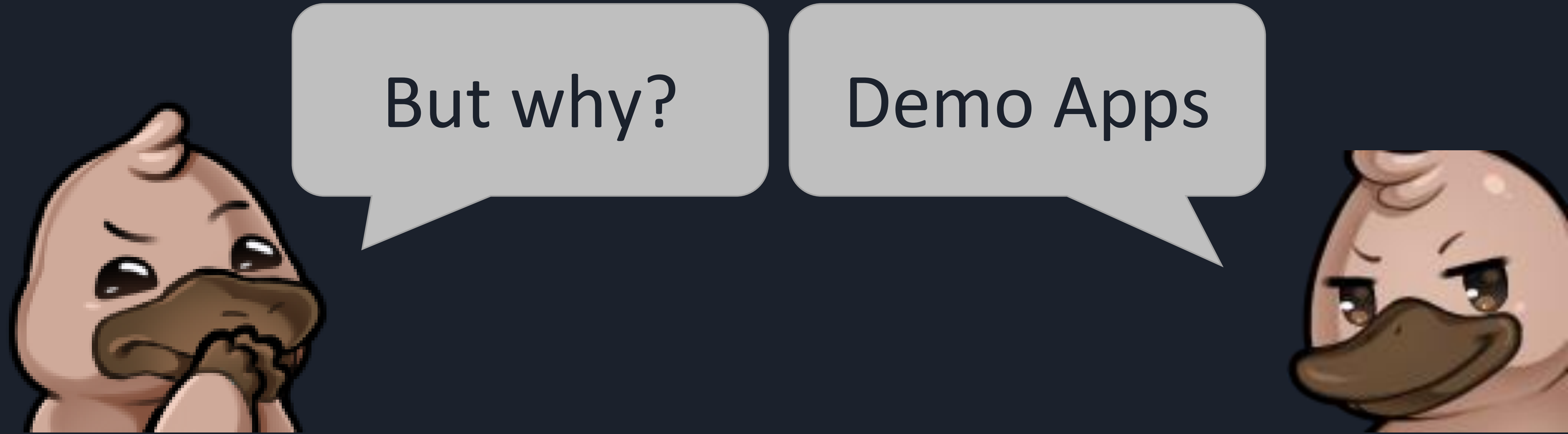
Collection Expressions

```
int[] first = [1, 2, 3];
int[] second = [4, 5, 6];
int[] all = [.. first, .. second, 7, 8, 9];

// [1, 2, 3, 4, 5, 6, 7, 8, ,9];
```

Top-Level Statements

Top-Level Statements Example

```
using System;

Console.WriteLine("Hello World");
```



Static Using Example

```
using System;
using static System.Console;

WriteLine("Hello World");
```

Tuples

Tuple Construction Example

```
(bool success, int number) SafeParse(string s)
    => (int.TryParse(s, out int n), n);

var r = SafeParse("123");

Console.WriteLine($"{r.number} {r.success}");
```

Tuple Deconstruction Example

```
(bool success, int number) SafeParse(string s)
    => (int.TryParse(s, out int n), n);

var (number, success) = SafeParse("123");

Console.WriteLine($"{number} {success}");
```

Tuple Combined Example

```
private int _x;
private int _y;
private int _z;

public Coord3D(int x, int y, int z)
{
    (_x, _y, _z) = (x, y, z);
}
```

Declare and Assign in Deconstruction (C# 10)

C# 7, 8, and 9

```
// Declaration:
(int x, int y) = point;

// Assignment:
int x = 0;
int y = 0;
(x, y) = point;
```

C# 10

```
// Both:
int x = 0;
(x, int y) = point;
```

Alias Any Type - C# 12

Alias Any Type - C# 12

```
using File = MyCode.MyFile; // Before C# 12
using MyUser = (int id, string name); // C# 12
```

Alias Any Type - C# 12

```
using MyUser = (int id, string name);

(int id, string name) myUser = (1, "Brendan");
string message = "Hello and welcome!";
SendMessage(message, myUser);
```

Alias Any Type - C# 12

```
using MyUser = (int id, string name);

MyUser myUser = (1, "Brendan");
string message = "Hello and welcome!";
SendMessage(message, myUser);
```

Global Usings

Global Usings (C# 10)

In `GlobalUsings.cs`

```
global using CodeMash.Samples;
global using static System.Console;
```

Global Usings (C# 10)

```csharp
WriteLine("Hello CodeMash!");
Session session = SessionCollection.Get(1);
WriteLine($"Welcome to {session.Name}");
```

Using Declarations

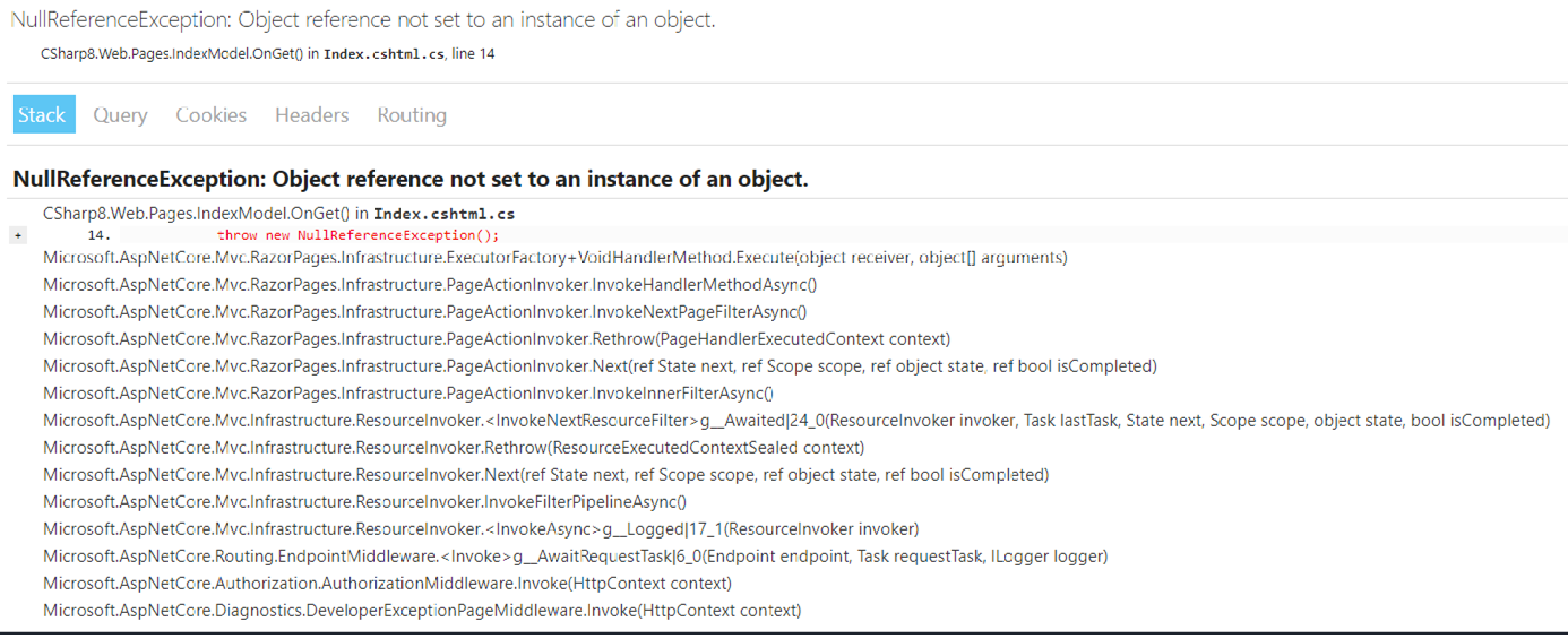
Using Statement Example (Old Way)

```
public void Example()
{
    using (FileStream fs = File.Create("File.txt"))
    {
        fs.WriteByte(42);
    }
}
```

Using Declaration Example (New Way)

```csharp
public void Example()
{
    using (FileStream fs = File.Create("File.txt"));

    fs.WriteByte(42);
}
```

Null References



```
NullReferenceException: Object reference not set to an instance of an object.
    CSharp8.Web.Pages.IndexModel.OnGet() in Index.cshtml.cs, line 14

Stack   Query   Cookies   Headers   Routing

NullReferenceException: Object reference not set to an instance of an object.
    CSharp8.Web.Pages.IndexModel.OnGet() in Index.cshtml.cs
+       14.             throw new NullReferenceException();
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.ExecutorFactory+VoidHandlerMethod.Execute(object receiver, object[] arguments)
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeHandlerMethodAsync()
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeNextPageFilterAsync()
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.Rethrow(PageHandlerExecutedContext context)
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.Next(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
    Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeInnerFilterAsync()
    Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeNextResourceFilter>g__Awaited|24_0(ResourceInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
    Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Rethrow(ResourceExecutedContextSealed context)
    Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Next(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
    Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.InvokeFilterPipelineAsync()
    Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeAsync>g__Logged|17_1(ResourceInvoker invoker)
    Microsoft.AspNetCore.Routing.EndpointMiddleware.<Invoke>g__AwaitRequestTask|6_0(Endpoint endpoint, Task requestTask, ILogger logger)
    Microsoft.AspNetCore.Authorization.AuthorizationMiddleware.Invoke(HttpContext context)
    Microsoft.AspNetCore.Diagnostics.DeveloperExceptionPageMiddleware.Invoke(HttpContext context)
```

Avoiding Null References

Null-Coalescing

A ?? B

If not null, use A.

Else, use B.

Null Coalescing

```
return Nickname ?? "";
```

Null Conditionals

(Null Propogation)

A?.B

If A is null, propagate that null instead of accessing B.

Null Conditionals (Null Propogation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    if (user == null)
    {
        return string.Empty;
    }
    return user.Nickname;
}
```

Null Conditionals (Null Propogation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    return user?.Nickname;
}
```

Null Conditionals (Null Propogation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    return user?.Nickname ?? "";
}
```

Null Conditionals (Null Propogation)

```
public string GetFirstChildName()
{
    return this.Children?[0]?.Name;
}
```

Null-Coalescing Assignment

??=

If not null, use value.
Else, assign
then use value.

Lazy Initialization Example

```
private Thing _thing;
public Thing Thing => _thing ??= new Thing();
```

Default Value Example

```
int? num = null;
num ??= 42;
Console.WriteLine(num ??= 1337);
```

Nullable Annotation Contexts
(“Nullable Reference Types”)

Nullable Context – Project Level

```
<Nullable>enable</Nullable>
```

Nullable Context – Processor Directive

```
#nullable enable

public class Person
{
}

#nullable disable
```

Reverse Indexing

Reverse Indexing (Hat Operator)

A	B	C	D	E	F	G	H	I
0	1	2	3	4	5	6	7	8
^9	^8	^7	^6	^5	^4	^3	^2	^1

Range Indexing

Range Indexing Example

```
var numbers = new[] {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

var middle = numbers[3..7];
var firstFew = numbers[..3];
var lastBunch = numbers[4..];

var lastFew = numbers[^3..];
var lastElement = numbers[^1];
```

File-Scoped Namespaces

Namespace Example

```
namespace ExampleNamespace
{
    public class Something
    {
        public int MyProperty { get; set; }
    }
}
```

File-Scoped Namespace Example

```
namespace ExampleNamespace;

public class Something
{
    public int MyProperty { get; set; }
}
```

Future Cool Stuff

Possibly Coming Soon or Not. Maybe. Who knows?

Params Collection - C# Future?

```
public bool Something(params int[] numbers) // Current
{
    // Do Something
}

Something(1, 2, 3);
```

Params Collection - C# Future?

```
public bool Something(params string letters) //Future?
{
    // Do Something
}

Something('a', 'b', 'c');
Something("abc");
```

Thanks for being here!

Questions?