

What's New in C# 8.0 and .NET Core 3.0

INTRODUCING C# 8 AND .NET CORE 3.0



Matt Honeycutt

HEROIC CONSULTING

@matthoneycutt www.trycatchfail.com



What to Expect



Slides and demos



You need Visual
Studio 2019 and .NET
Core 3.0



You need to know the
basics

.NET Core 3.0:
<http://bit.ly/2kNorFn>

VS 2019:
<http://bit.ly/2kQuSaL>



Need to Brush Up?



C# Fundamentals

<https://app.pluralsight.com/library/courses/csharp-fundamentals-dev/table-of-contents>

Visual Studio 2019 Getting Started

<https://app.pluralsight.com/library/courses/visual-studio-2019-getting-started/table-of-contents>

C# skill path

<https://app.pluralsight.com/paths/skill/csharp>



A History of C#



1.0: The (Absolute) Basics

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



2.0: Generics Arrive

```
static void Main(string[] args)
{
    List<string> names = new List<string>();
    names.Add("John");
    names.Add("Jane");

    int? age = null;

    foreach (string name in names)
    {
        Console.WriteLine("Name: {0}", name);
    }
}
```



3.0: LINQ!

```
static void Main(string[] args)
{
    var widgets = from w in Widgets
                  where w.IsActive
                  select new
                  {
                      w.Name
                  };

    foreach (var widget in widgets)
    {
        Console.WriteLine("Name: {0}", widget.Name);
    }
}
```



4.0: Dynamic

```
static void Main(string[] args)
{
    dynamic animal = new Dog();

    animal.ThisCompilesButIsNotGoingToWork();
}
```



5.0: Async and Await

```
public async Task MarkWidgetAsInactive(int id)
{
    var target = await GetWidget(id);

    target.IsActive = false;

    await SaveChanges();
}
```



6.0: Many Small Improvements

```
public async Task<bool> IsWidgetActive(int id)
{
    var target = await GetWidget(id);

    // My favorite thing EVER
    return (target?.IsActive) ?? false;
}
```



7.0: Expressions and Tuples

```
static void Main(string[] args)
{
    var pairs = new[]
    {
        (Left: Mark, Right: Mary),
        (Left: Jane, Right: John)
    };

    foreach (var pair in pairs)
    {
        Console.WriteLine($"Pair: {pair.Left} - {pair.Right}");
    }
}
```



C# 8.0

The Future



Nullable Reference Types

```
BlogPost post = null;  
PrintTitle(post);
```



[🔍] (local variable) `BlogPost post`

Possible null reference argument for parameter 'post' in 'void Program.PrintTitle(BlogPost post)'.

[Show potential fixes](#) (Alt+Enter or Ctrl+.)



Pattern Matching Improvements

```
public static string DisplayShapeInfo(object shape) =>
    shape switch
    {
        Rectangle r => r switch
        {
            _ when r.Length == r.Width => "Square!",
            _ => "",
        },
        Circle { Radius: 1 } c => $"Small Circle!",
        Circle c => $"Circle (r={c.Radius})",
        Triangle t => $"Triangle ({t.Side1}, {t.Side2}, {t.Side3})",
        _ => "Unknown Shape"
    };
```



Indices and Ranges

```
var numbers = new[] { 1, 2, 3, 4, 5 };
```

```
var lastTwo = numbers[^2..];
```

```
var position = 2;
```

```
Index startIndex = ^position;
```

```
Range range = startIndex..;
```

```
var lastTwo = numbers[startIndex];
```



Default Interface Members

```
public interface IAnimalWidget
{
    private static int AmountToFeed = 10;

    static void SetAmountToFeed(int amount)
    {
        AmountToFeed = amount;
    }

    string Name { get; }

    int Happiness { get; set; }

    void Feed()
    {
        Happiness += AmountToFeed;
    }
}
```



Async Streams

```
await foreach (var order in orderFactory.MakeOrdersAsync(5))  
{  
    Console.WriteLine($"[{ThreadId}]Received order {order.Id}.");  
}
```



There are even *more*
improvements in C# 8!

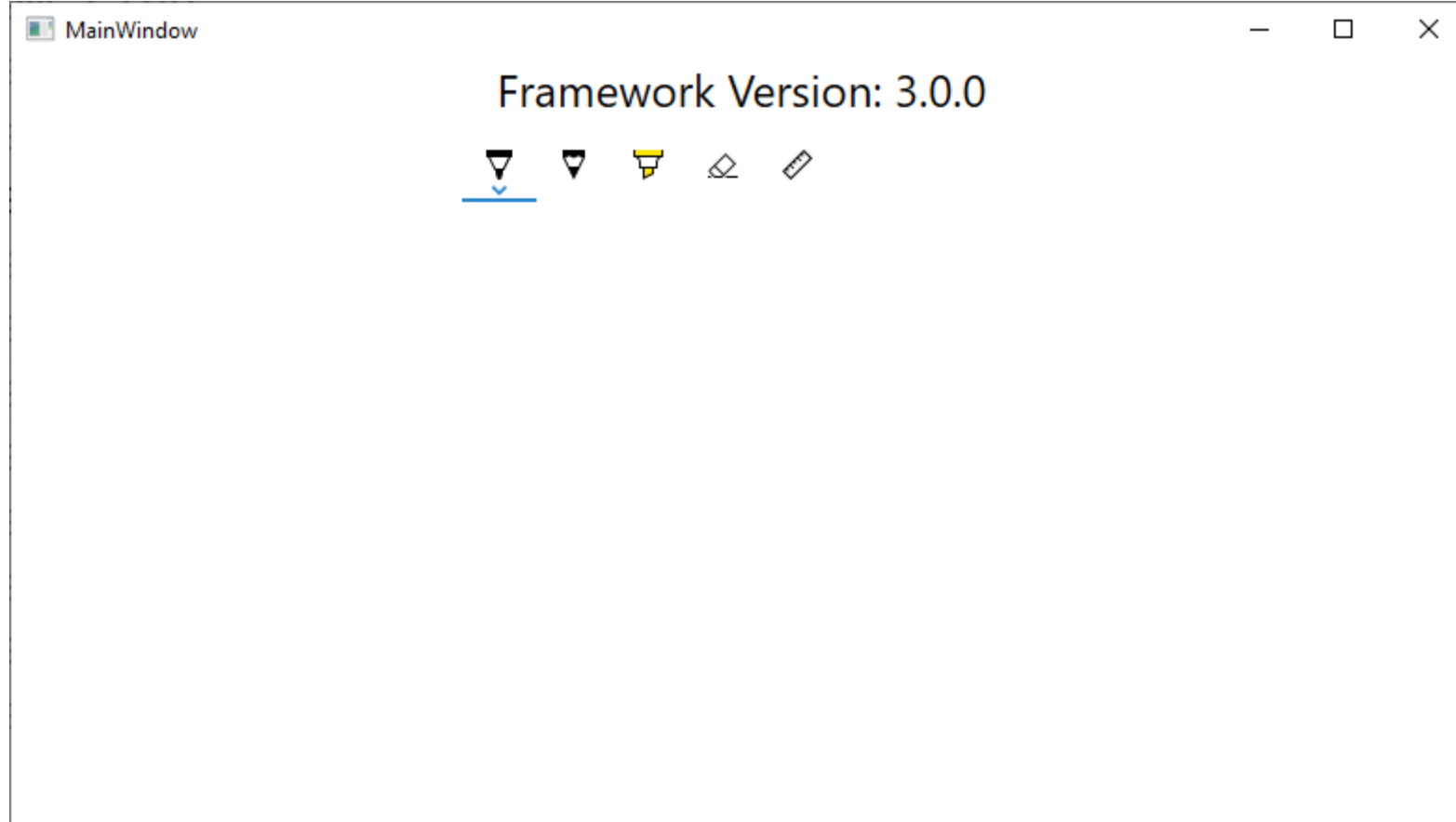


Now Included: Fast JSON Processing

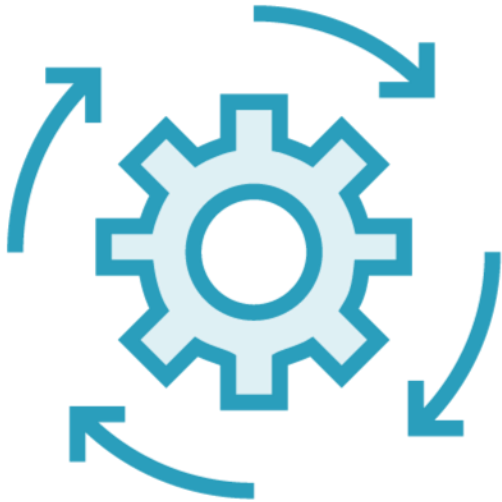
```
private static string GetTokenDesc(Utf8JsonReader json) =>
    json.TokenType switch
    {
        JsonTokenType.StartObject => "START OBJECT",
        JsonTokenType.EndObject => "END OBJECT",
        JsonTokenType.StartArray => "START ARRAY",
        JsonTokenType.EndArray => "END ARRAY",
        JsonTokenType.PropertyName => $"PROPERTY: {json.GetString()}",
        JsonTokenType.Comment => $"COMMENT: {json.GetString()}",
        JsonTokenType.String => $"STRING: {json.GetString()}",
        JsonTokenType.Number => $"NUMBER: {json.GetInt32()}",
        JsonTokenType.True => $"BOOL: {json.GetBoolean()}",
        JsonTokenType.False => $"BOOL: {json.GetBoolean()}",
        JsonTokenType.Null => $"NULL",
        _ => $"**UNHANDLED TOKEN: {json.TokenType}"
    };
```



Windows Desktop Support



Build, Pack, and Deploy Improvements



Course Roadmap

Nullable
reference
types

Pattern
matching

Indices and
ranges

Built-in
JSON
support

Windows
desktop
support

Build and
deployment
changes

More language
improvements

Other .NET
Core platform
improvements



Summary

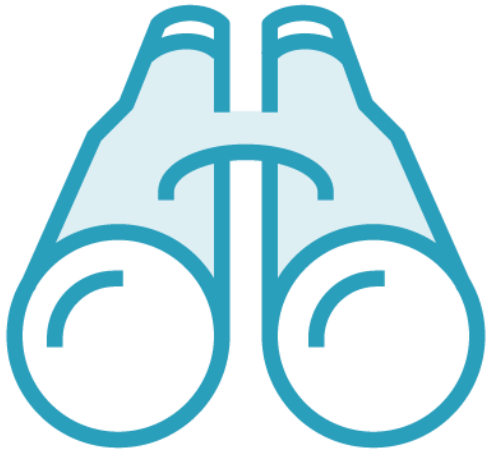


Course overview and roadmap

Are you excited?



Up Next



Nullable reference types

