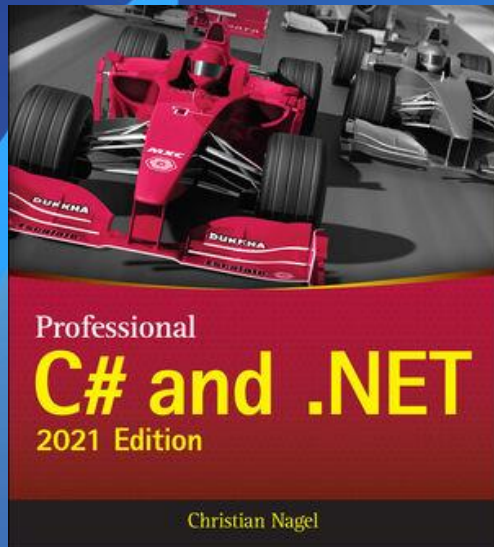




Don't miss a Beat with C# 11 and C# 12

Christian Nagel

<https://www.cinnovation.com>





Christian Nagel

- Training
 - Coaching
 - Consulting
 - Development
-
- Microsoft MVP
 - www.cninnovation.com
 - csharp.christiannagel.com
 - @christiannagel

Professional

C# and .NET

2021 5th



What's new with...

- Strings
- Types
- Interfaces
- Arrays and Collections
- Something special...





New String Features





UTF-8 String Literals

- Windows, JavaScript, Java, .NET use UTF-16
- Web uses UTF-8 (Unicode Text Format 8)
- **U8** suffix
- `ReadOnlySpan<byte>`
- .NET 6 added **`ISpanFormattable`**
`bool TryFormat(Span<char> destination, out int charsWritten, ReadOnlySpan<char> format, IFormatProvider? provider)`
- .NET 8 adds **`IUtf8SpanFormattable`**
`bool TryFormat(Span<byte> destination, out int bytesWritten, ReadOnlySpan<char> format, IFormatProvider? provider)`

Raw String Literals



- Three or more quotes """"
- Spans multiple lines
- Indentation based on end quotes
- Combine with interpolated strings \$\$



Type Enhancements

Alias any type

- **using** alias relaxed with C# 12
- alias tuples, pointers, arrays, generic types...



Required Modifier

- Nullable reference types solves problems
- Constructors adds complexity
- **required**: Initialize with *constructor* or *object initializer*
- Modifier for properties and fields
- ***SetsRequiredMembers*** Attribute on constructor

Primary Constructors

- Class records
 - get & init accessors
- Struct records
 - get & set accessors
- Readonly struct records
 - get & init accessors
- Classes and structs
 - Parameters

A wooden letter 'N' is positioned on the left side of the frame, resting on a surface with a complex geometric pattern of intersecting lines. The letter is light-colored with a visible wood grain. The background pattern consists of thin, dark lines forming a series of triangles and polygons, creating a sense of depth and perspective. The lighting is soft, casting a gentle shadow from the letter onto the patterned surface.

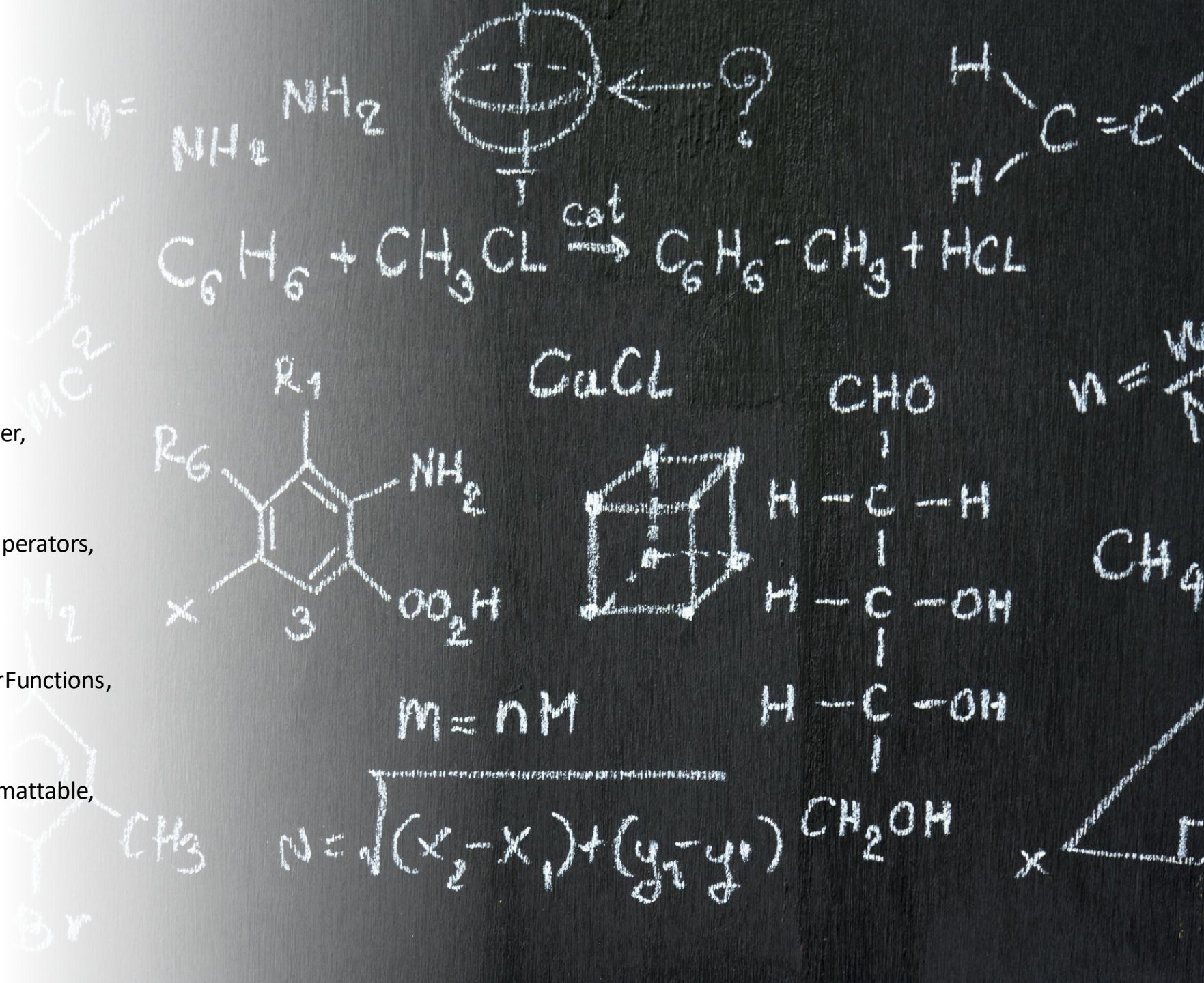
Updates with Interfaces

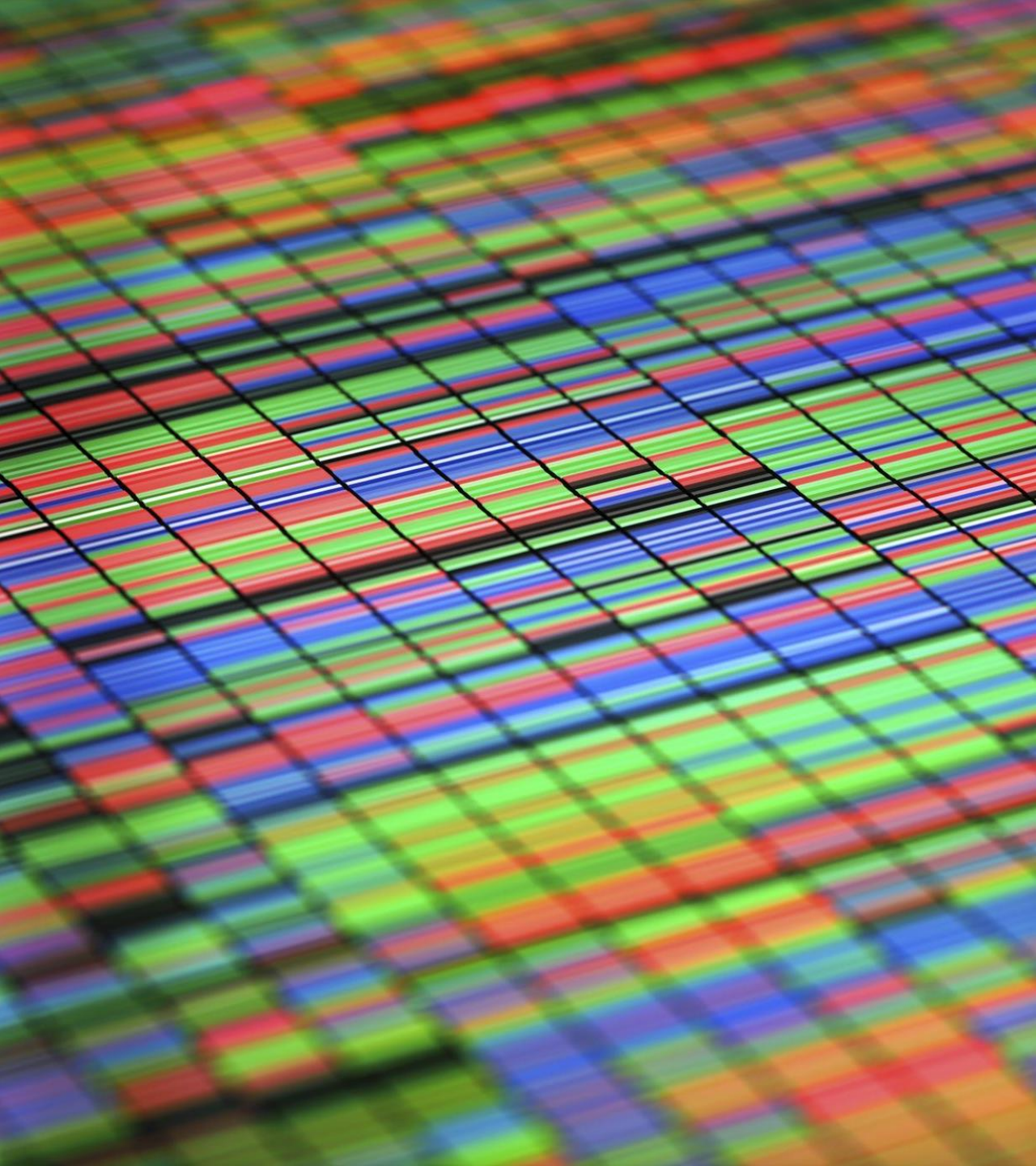
Interface Enhancements

- C# 8 added implementation in interfaces
 - Versioning
 - Mixin functionality
- C# 11 adds static members
 - Constraints with generic types

Generic Math

- Numeric interfaces
 - INumber, IMinMaxNumber, IAdditiveIdentity, IBinaryInteger, IFloatingPoint...
- Operator interfaces
 - IAdditionOperators, IBitwiseOperators, IIncrementOperators...
- Function interfaces
 - IExponentialFuncations, ILogarithmicFunctions, IPowerFunctions, ITrigonometricFunctions...
- Parsing and formatting interfaces
 - IParsable, ISpanParsable, IFormattable, ISpanFormattable





Arrays and Collections

Collection Expressions (Collection Literals)

- Conversion to many different collection types using square brackets `[]`
- Spread Element `..`
- *CollectionBuilder* Attribute – use collection expressions with your type

Pattern Matching Enhancements

- List Patterns
 - Apply a pattern on any element in the sequence
 - Discard `_` pattern to match any element
 - Slice `..` pattern to match zero or more elements
- Matching on `Span<T>`

Inline Arrays

- Optimized creation for fixed sizes
- Directly assign *Span<T>*
- *InlineArray* attribute

```
[InlineArray(10)]  
public struct Buffer  
{  
    private int _x;  
}
```




Something
special...

Unsafe Accessor

- With reflection it was possible to access private members of a type
- Serialization, EF Core...
- *UnsafeAccessor*
- Compiler-Feature
- Access private members

```
internal class Changelit
{
    [UnsafeAccessor(UnsafeAccessorKind.Field,
        Name = "_title")]
    public extern static ref string GetTitle(Book @this);
}
```


Interceptors

- Replace implementation
- Usually used by source generators
- Pre-release with .NET 8, but used from Native AOT source generators
- *InterceptsLocation* Attribute

Native AOT

- Compile .NET to native code
 - Self-contained
 - Quick startup, less memory usage
 - Can run where JIT is not allowed
 - Compilation to a single file
- No dynamic loading
 - No reflection emit
 - No C++/CLI
 - Requires trimming
 - Compilation into a single file
 - Create minimal APIs
 - Many libraries don't support native AOT (yet)

Source Generators

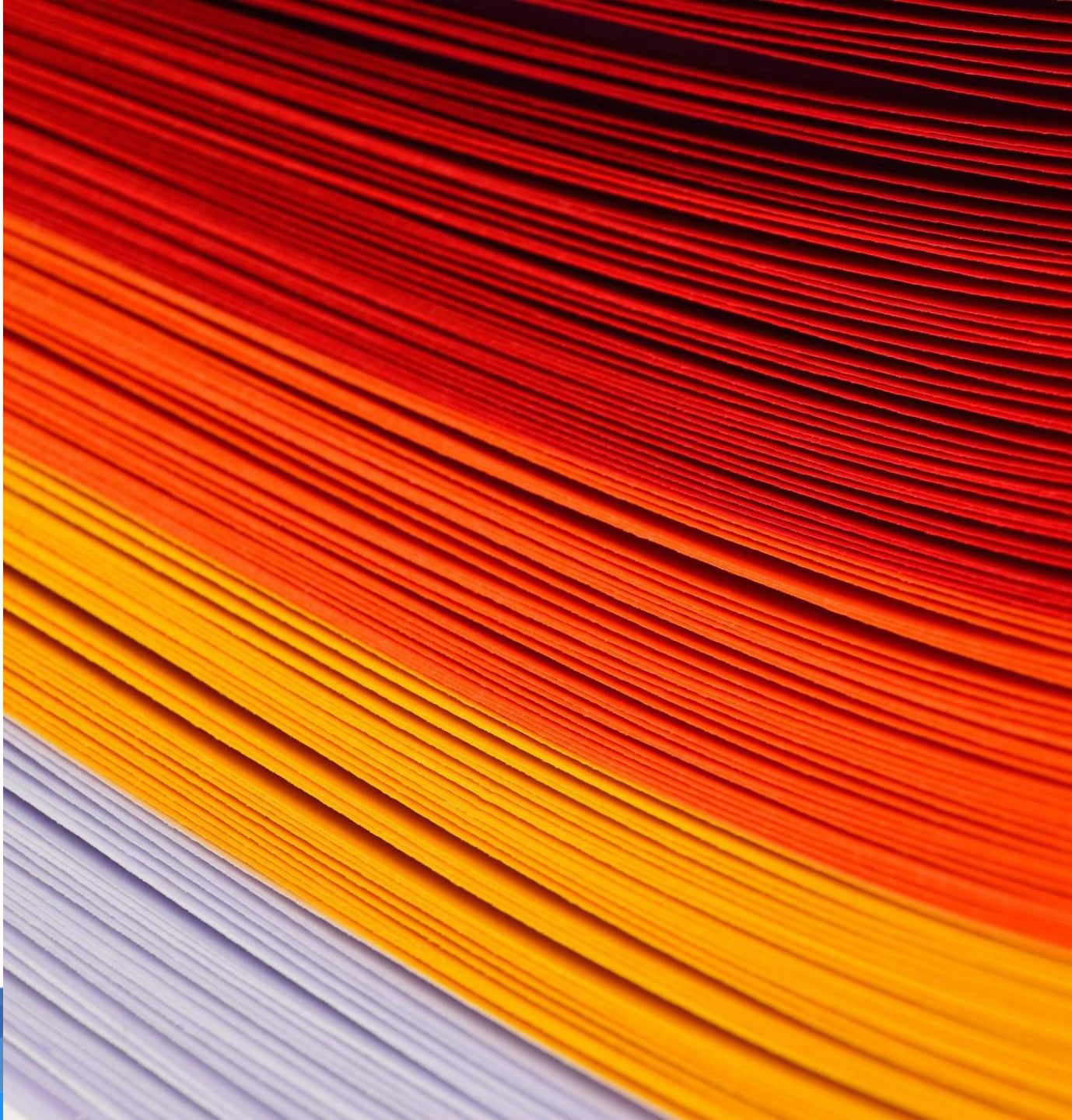
- Regular expressions
- JSON Serialization
- JavaScript
- Minimal API
- Platform invoke
- COM Interop
- Configuration binding





Summary

- Productivity
 - Raw string literals
 - Primary constructors
 - Collection expressions
- Performance
 - Span enhancements
 - Inline Array
 - Native AOT
 - Source generators



Questions?

- <https://github.com/cnilearn/bastamainz2023>
- <https://csharp.christiannagel.com>
- <https://www.cninnovation.com>

