



Robert Haken

software & cloud architect, HAVIT, s.r.o.

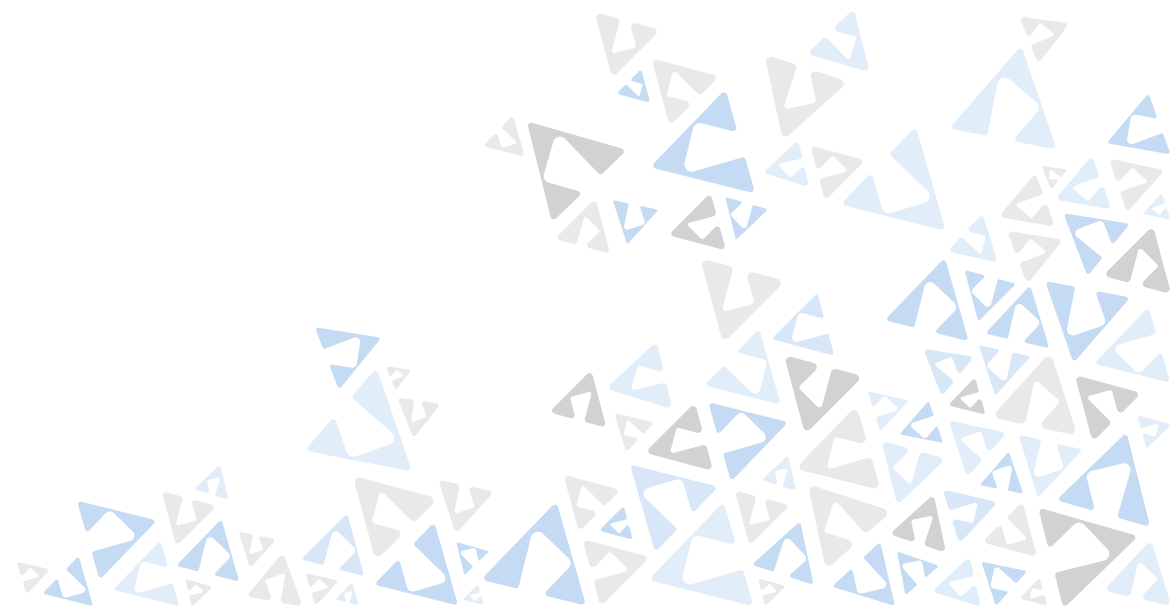
haken@havit.cz, @RobertHaken, <https://knowledge-base.havit.cz> + .eu

Microsoft MVP: Development, MCT, MCPD: Web, MC: Azure Architect

C# 8+

Visual Studio 2019+

C# 8.0



C# 8.0: Default interface methods

```
interface IEnumerable<T>
{
    int Count()
    {
        int count = 0;
        foreach (var x in this)
            count++; return count;
    }
}
```

```
interface IList<T> : IEnumerable<T>
{
    int Count { get; }
    override int IEnumerable<T>.Count() => this.Count;
}
```

<https://github.com/dotnet/csharplang/blob/master/proposals/default-interface-methods.md>

C# vNext



C# 9.0 Candidate: Primary Constructors, Records

```
class Person(string First, string Last);
```

```
class Person : IEquatable<Person>
```

```
{
```

```
    public string First { get; }
```

```
    public string Last { get; }
```

```
    public Person(string First, string Last) => (this.First, this.Last) = (First, Last);
```

```
    public void Deconstruct(out string First, out string Last)
```

```
        => (First, Last) = (this.First, this.Last);
```

```
    public bool Equals(Person other)
```

```
        => other != null && First == other.First && Last == other.Last;
```

```
    public override bool Equals(object obj) => obj is Person other ? Equals(other) : false;
```

```
    public override int GetHashCode() => GreatHashFunction(First, Last);
```

```
    ...
```

```
}
```

<https://github.com/dotnet/csharplang/blob/master/proposals/records.md>

C# 9.0 Candidate: pattern-based `with` expressions

```
class Point
{
    public readonly int X;
    public readonly int Y;
    public Point With(int x = this.X, int y = this.Y) => new Point(x, y);
    // etc
}
```

```
p2 = p with { X = 5 };
```

```
db.Persons.Find(id) with { Name = data.Name, ... };
```

C# 9.0 Candidate: Top level statements and member declarations

```
public class Functions
{
    public int Fac(int x) => x < 0 ? x * Fac(x-1) : 1; // Top-level function
}
public class Program
{
    static void Main()
    {
        Console.WriteLine(Functions.Fac(5)); // Top-level statement
    }
}

// C# vNext (scripting dialect replacement)
public int Fac(int x) => x < 0 ? x * Fac(x-1) : 1; // Top-level function
Console.WriteLine(Fac(5)); // Top-level statement
```

C# 9.0 Candidate: Covariant return types

```
class Compilation ...  
{  
    virtual Compilation WithOptions(Options options)...  
}
```

```
class CSharpCompilation : Compilation  
{  
    override CSharpCompilation WithOptions(Options options)...  
}
```


C# 9.0 Candidate: Nullable-Enhanced Common Type

```
// C# now
```

```
int? i = condition ? 1 : (int?)null;
```

```
// C# vNext
```

```
int? i = condition ? 1 : null;
```

```
// C# vNext - Other implicit conversions?
```

```
int i = condition ? myInt : myDouble;
```

C# 9.0 Candidate: Target typed null coalescing (`??`) expression

```
M(List<int> list, uint? u)
{
    IEnumerable<int> x = list ?? Array.Empty<int>();

    long l = u ?? -1;
}
```

C# 9.0 Candidate: and, or, and not patterns

```
switch (o)
{
    case 1 or 2:
    case Point(0, 0) or null:
    case Point(var x, var y) and var p:
}
```

C# 9.0 Candidate: Switch expression as a statement expression

```
void M(bool c, ref int x, ref string s)
{
    c switch { true => x = 1, false => s = null };
}
```

C# 9.0 Candidate: allow comparison operators in switch cases

```
int iq = DoIqTest();

switch (iq)
{
    case <= 69:
        ProcedureExtremelyLow();
        break;
    case 70 to 129:
        Something();
        break;
    case >= 130:
        VerySuperiorProcedure();
        break;
}
```

C# 9.0 Candidate: Native-Sized Number Types

```
// now
IntPtr ptr;
UIntPtr uptr;
IntPtr.Size

// vNext
IntN i;
UIntN u;
FloatN f;
```

<https://github.com/dotnet/corefxlab/blob/master/docs/specs/nativesized.md>

C# 9.0 Candidates

Discriminated Unions

Static Delegates

Module Initializers

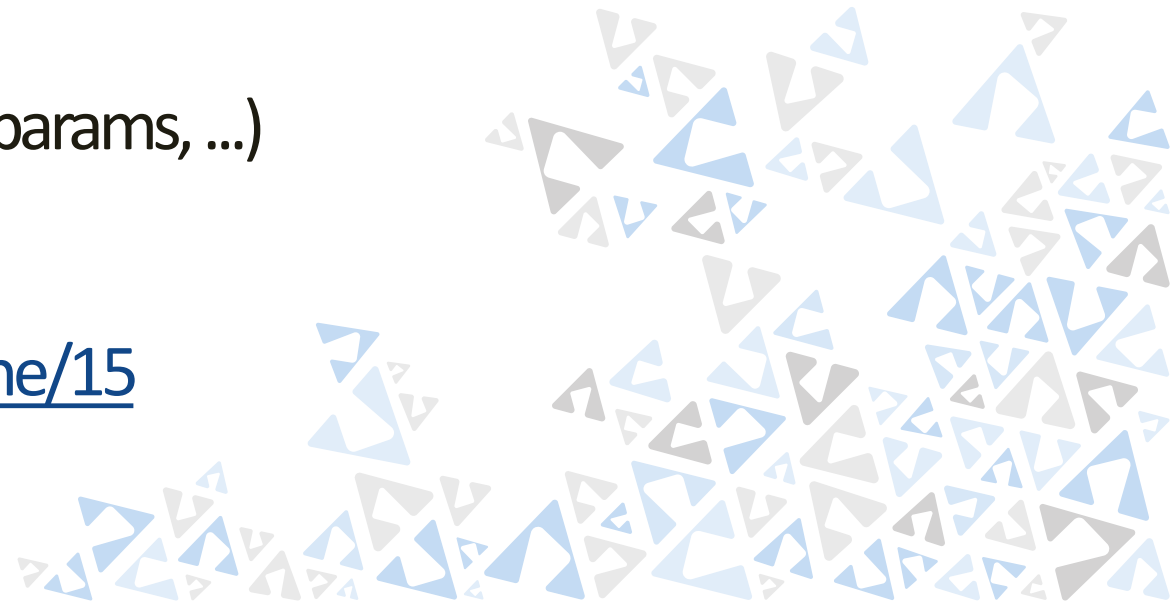
defer statement

Permit attributes on local functions

Efficient Params and String Formatting (Span as params, ...)

...and more

<https://github.com/dotnet/csharpplang/milestone/15>



C# vNext (??): Extension everything

```
extension Enrollee extends Person
{
    // static field
    static Dictionary<Person, Professor> enrollees = new Dictionary<Person, Professor>();

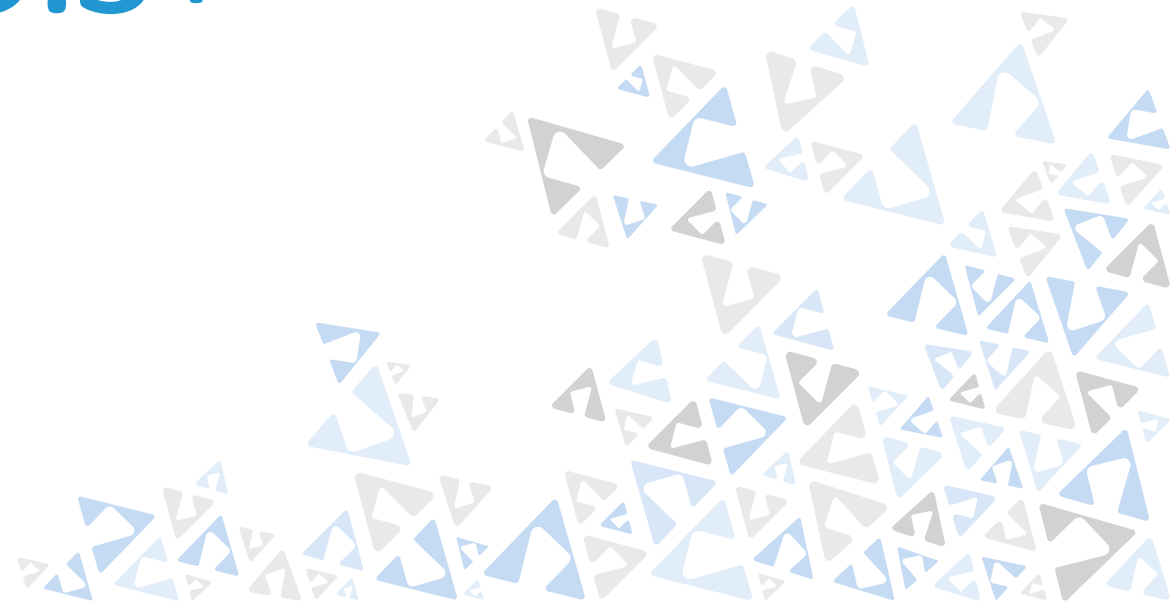
    // instance method
    public void Enroll(Professor supervisor) { enrollees[this] = supervisor; }

    // instance property
    public Professor Supervisor => enrollees.TryGetValue(this, out var supervisor) ? supervisor : null;

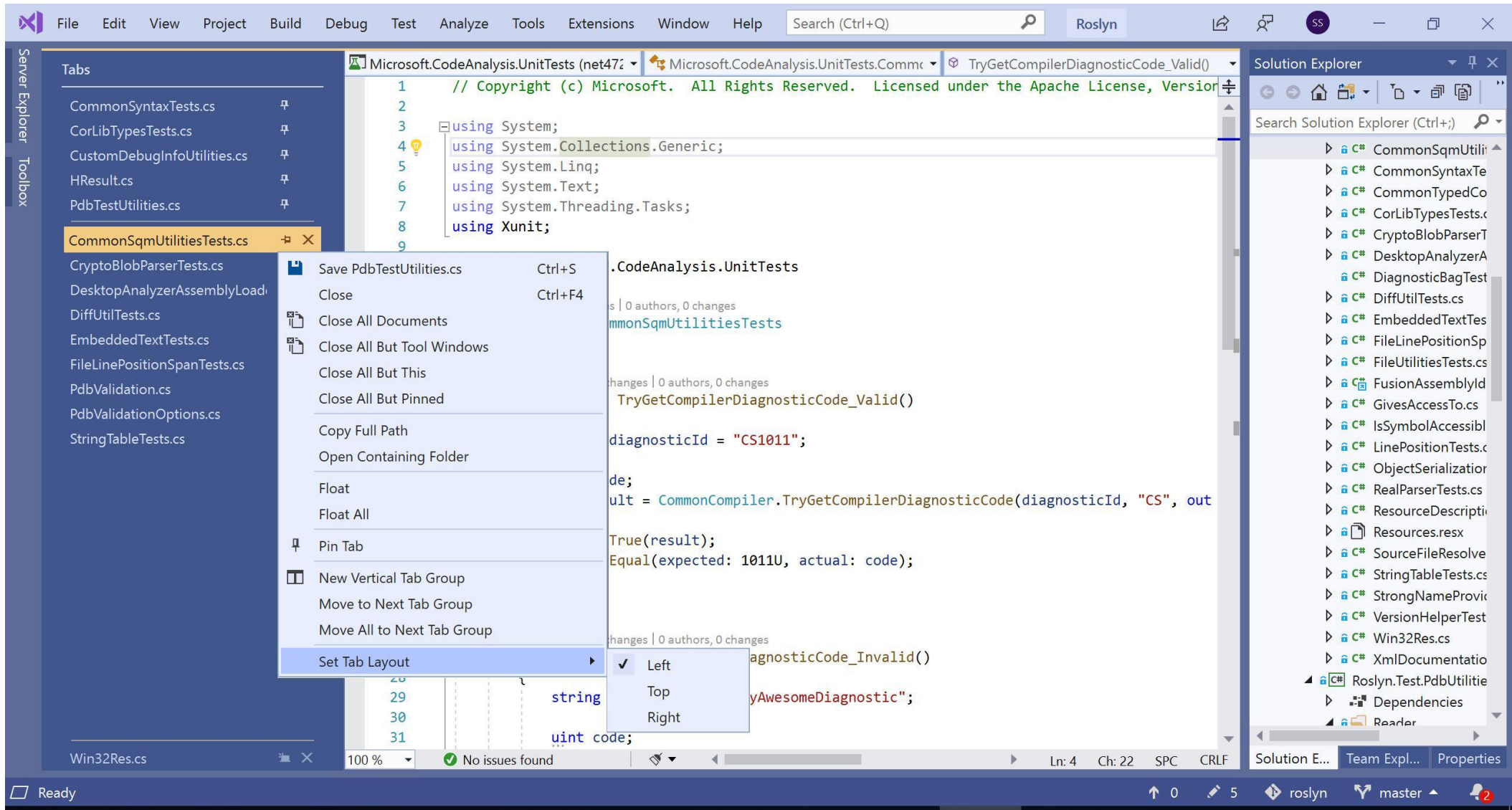
    // static property
    public static ICollection<Person> Students => enrollees.Keys;

    // instance constructor
    public Person(string name, Professor supervisor) : this(name) { this.Enroll(supervisor); }
}
```

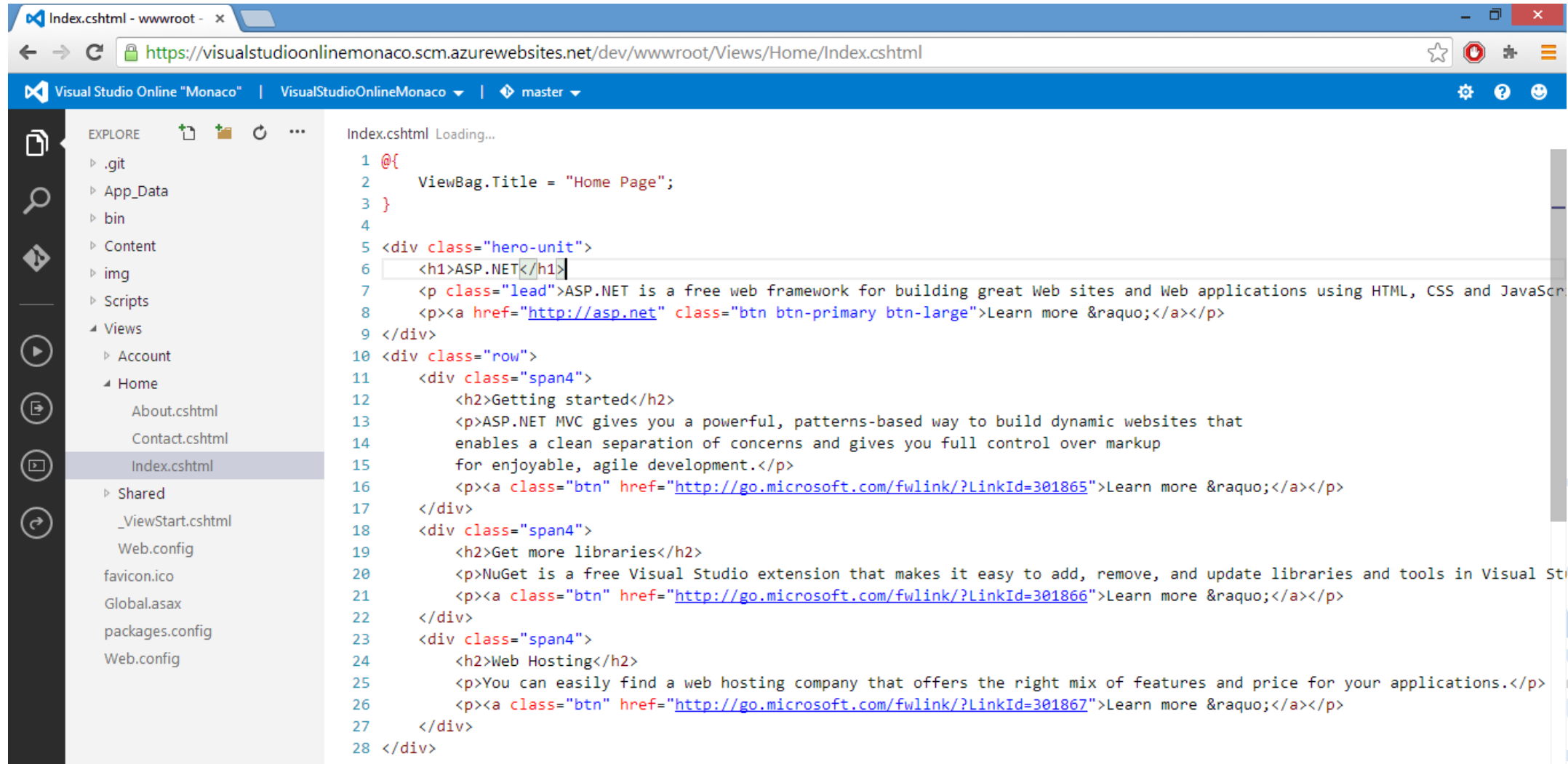

Visual Studio 2019.3+



Vertical Document Tabs



Visual Studio Online (“Monaco”)



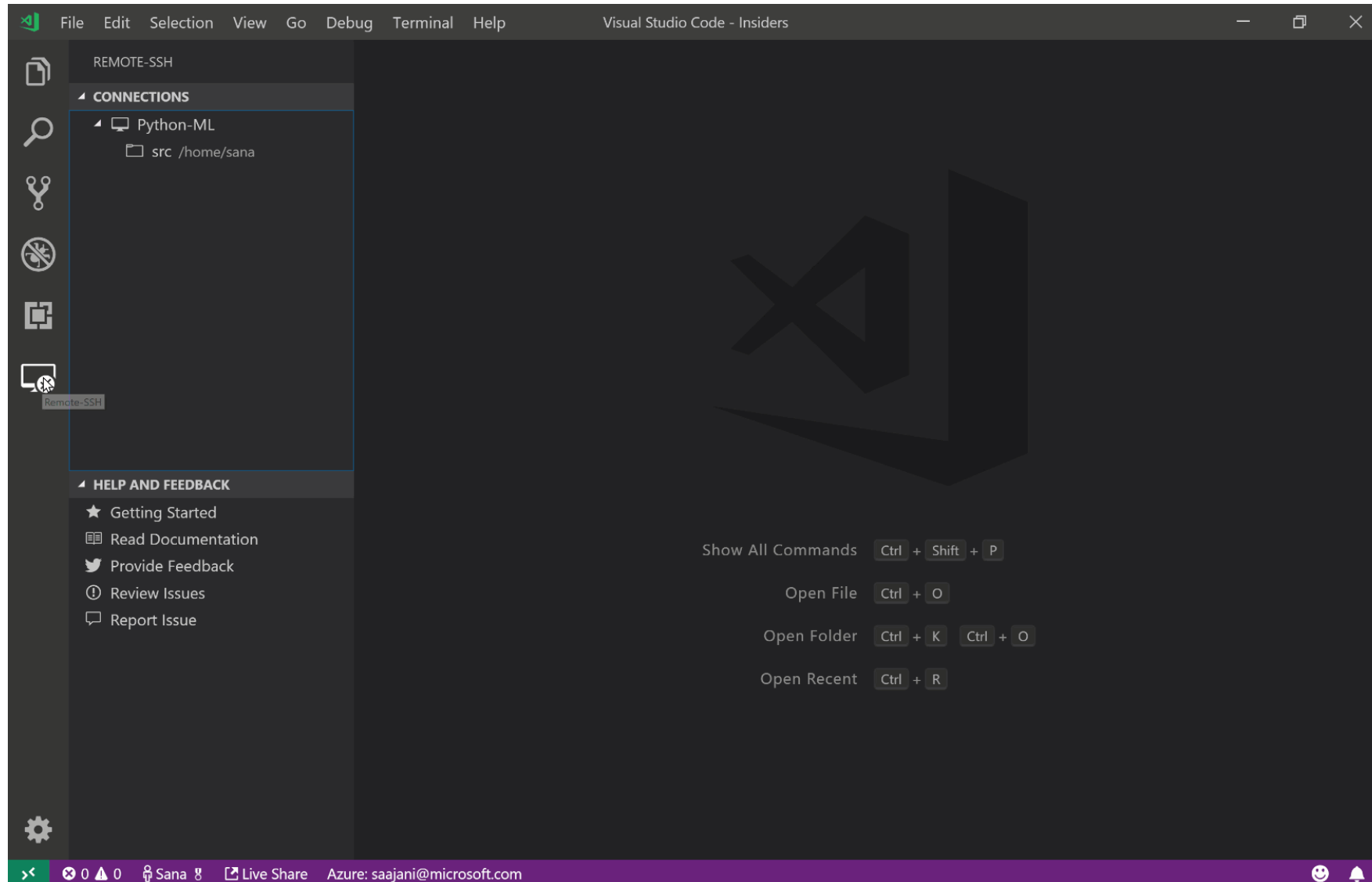
The screenshot displays the Visual Studio Online Monaco editor interface. The browser address bar shows the URL: `https://visualstudioonlinemonaco.scm.azurewebsites.net/dev/wwwroot/Views/Home/Index.cshtml`. The editor title bar indicates the file is `Index.cshtml` and is in the `master` branch. The left sidebar shows the file explorer with the following structure:

- EXPLORE
 - .git
 - App_Data
 - bin
 - Content
 - img
 - Scripts
 - Views
 - Account
 - Home
 - About.cshtml
 - Contact.cshtml
 - Index.cshtml (selected)
 - Shared
 - _ViewStart.cshtml
 - Web.config
 - favicon.ico
 - Global.asax
 - packages.config
 - Web.config

The main editor area shows the source code for `Index.cshtml`, which is a Razor view. The code is as follows:

```
1 @{
2     ViewBag.Title = "Home Page";
3 }
4
5 <div class="hero-unit">
6     <h1>ASP.NET</h1>
7     <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript</p>
8     <p><a href="http://asp.net" class="btn btn-primary btn-large">Learn more &raquo;</a></p>
9 </div>
10 <div class="row">
11     <div class="span4">
12         <h2>Getting started</h2>
13         <p>ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.</p>
14         <p><a class="btn" href="http://go.microsoft.com/fwlink/?LinkId=301865">Learn more &raquo;</a></p>
15     </div>
16     <div class="span4">
17         <h2>Get more libraries</h2>
18         <p>NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio.</p>
19         <p><a class="btn" href="http://go.microsoft.com/fwlink/?LinkId=301866">Learn more &raquo;</a></p>
20     </div>
21     <div class="span4">
22         <h2>Web Hosting</h2>
23         <p>You can easily find a web hosting company that offers the right mix of features and price for your applications.</p>
24         <p><a class="btn" href="http://go.microsoft.com/fwlink/?LinkId=301867">Learn more &raquo;</a></p>
25     </div>
26 </div>
```

Visual Studio Remote Development



REFERENCES

Demos

<https://github.com/hakenr/CSharp8Demo>

Blog – HAVIT Knowledge Base- <http://knowledge-base.havit.cz/> + .eu

Twitter - [@RobertHaken](https://twitter.com/RobertHaken)

YouTube - <https://www.youtube.com/user/HAVITcz>

