|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Example | [C#](https://roslyn.codeplex.com/wikipage?title=CSharp%20Language%20Design%20Notes&referringTitle=Language%20feature%20status) | [VB](https://roslyn.codeplex.com/wikipage?title=Visual%20Basic%20Language%20Design%20Notes&referringTitle=Language%20feature%20status) |
| Primary constructors | class Point**(int x, int y)** { … } | No | No |
| Auto-property initializers | public int X { get; set; } **= x;** | Done | Exists |
| Getter-only auto-properties | public int Y**{ get; }**= y; | Done | Done |
| [Ctor assignment to getter-only autoprops](https://roslyn.codeplex.com/discussions/568824) | Y = 15 | Done | Done |
| Parameterless struct ctors | Structure S : Sub New() : End Sub : End Structure | Done | Done |
| Using static members | using System.**Console**; … **Write**(4); | Done | Exists |
| Dictionary initializer | new JObject { **["x"]** = 3, **["y"]** = 7 } | Done | No |
| Indexed member initializer | new JObject { **$x** = 3, **$y** = 7 } | No | No |
| Indexed member access | c.**$name** = c.**$first** + " " + c.**$last**; | No | Exists |
| Declaration expressions | int.TryParse(s, out **var x**); | No | No |
| Await in catch/finally | try … catch { **await** … } finally { **await** … } | Done | Maybe |
| Exception filters | catch(E e) **if (e.Count > 5)** { … } | Done | Exists |
| Typecase | Select Case o : Case **s As String** : … | No | Maybe |
| Guarded cases | Select Case i : Case Is > 0 **When i Mod 2 = 0** | No | No |
| Partial modules | **Partial Module** M1 | N/A | Done |
| Partial interfaces | **Partial Interface** I1 | Exists | Done |
| Multiline string literals | "Hello***<newline>***World" | Exists | Done |
| Year-first date literals | Dim d = #**2014-04-03**# | N/A | Done |
| Binary literals | **0b00000100** | No | No |
| Digit separators | 0xEF**\_**FF**\_**00**\_**A0 | No | No |
| Line continuation comments | Dim addrs = From c in Customers **' comment** | N/A | Done |
| TypeOf IsNot | If **TypeOf** x **IsNot** Customer Then … | N/A | Done |
| Expression-bodied members | public double Dist **=> Sqrt(X \* X + Y \* Y);** | Done | No |
| Event initializers | new Customer { **Notify +=**MyHandler }; | No | No |
| [Null propagation](https://roslyn.codeplex.com/discussions/540883) | customer**?**.Orders**?**[5]**?**.$price | Done | Planned |
| Semicolon operator | (var x = Foo()**;** Write(x)**;** x \* x) | No | No |
| Private protected | **private protected** string GetId() { … } | No | No |
| Params IEnumerable | int Avg(**params IEnumerable<int>** numbers) { … } | No | No |
| Constructor Inference | new **Tuple**(3, "three", true); | No | No |
| [String interpolation](https://roslyn.codeplex.com/discussions/540869) | "**\{**p.First**} \{**p.Last**}**is **\{**p.Age**}**years old." | Planned | Planned |
| TryCast for nullable | Dim x = **TryCast**(u, **Integer?**) | Exists | No |
| Delegate combination with + | d1 **+=** d2 | Exists | No |
| Implicit implementation | Class C : **Implicitly** Implements I | Exists | No |
| [nameof operator](https://roslyn.codeplex.com/discussions/552377) | string s = **nameof(**Console.Write**)**; | Done | Planned |
| [Strict modules](https://roslyn.codeplex.com/discussions/540507) | **Strict** Module M | Exists | No |
| [Faster CInt](https://roslyn.codeplex.com/discussions/543475) | Dim x = CInt(Math.Truncate(d)) | | Exists | No |
| [#pragma](https://roslyn.codeplex.com/discussions/543476) | #Disable Warning BC40008 | Done | Done |
| Checked and Unchecked blocks | Checked : x += 1 : End Checked | Exists | No |
| Field targets on autoprops | **<Field:** Serializable> Property p As Integer | Maybe | [Maybe](https://roslyn.codeplex.com/discussions/545260) |
| If(b,e1,e2) uses type context | Dim x As Integer? = If(b,1,**Nothing**) | N/A | No |