**C# 2.0**

* Generics[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Partial types[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* [Anonymous methods](https://en.wikipedia.org/wiki/Anonymous_method)[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Iterators[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Nullable value types[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Getter/setter separate accessibility[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Method group conversions (delegates)[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Co- and Contra-variance for delegates[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Static classes[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Delegate inference[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)
* Null coalescing operator[[43]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv2_msdn-46)

**C# 3.0**

* Implicitly typed local variables[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Object and collection initializers[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Auto-Implemented properties[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Anonymous types[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* [Extension methods](https://en.wikipedia.org/wiki/Extension_method)[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Query expressions[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Lambda expression[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Expression trees[[44]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_msdn-47)
* Partial methods[[45]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv3_partmeths-48)

**C# 4.0**

* Dynamic binding[[46]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv4_msdn-49)
* Named and optional arguments[[46]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv4_msdn-49)
* Generic co- and contravariance[[46]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv4_msdn-49)
* Embedded interop types ("NoPIA")[[46]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv4_msdn-49)

**C# 5.0**[[47]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-50)

* [Asynchronous methods](https://en.wikipedia.org/wiki/Async/await)[[48]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv5_msdn-51)
* Caller info attributes[[48]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-csv5_msdn-51)

**C# 6.0**

* Compiler-as-a-service ([Roslyn](https://en.wikipedia.org/wiki/Microsoft_Roslyn))
* Import of static type members into namespace[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Exception filters[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Await in catch/finally blocks[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Auto property initializers[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Default values for getter-only properties[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Expression-bodied members[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Null propagator ([null-conditional operator](https://en.wikipedia.org/wiki/Safe_navigation_operator), succinct null checking)[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* String interpolation[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* nameof operator[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)
* Dictionary initializer[[49]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-roslyn_6-52)

**C# 7.0**[[50]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-53)[[51]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-new_features_in_7-54)

* Inline out variable declaration
* [Pattern matching](https://en.wikipedia.org/wiki/Pattern_matching)
* Tuple types and tuple literals
* Deconstruction
* [Local functions](https://en.wikipedia.org/wiki/Local_function)
* Digit separators
* Binary literals
* Ref returns and locals
* Generalized async return types
* Expression bodied constructors and finalizers
* Expression bodied getters and setters
* Throw can also be used as expression

**C# 7.1**[[52]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-55)

* Async main
* Default literal expressions
* Inferred tuple element names

**C# 7.2**[[53]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-56)

* Reference semantics with value types
* Non-trailing named arguments
* Leading underscores in numeric literals
* private protected access modifier

**C# 7.3**[[54]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-57)

* Accessing fixed fields without pinning
* Reassigning ref local variables
* Using initializers on stackalloc arrays
* Using fixed statements with any type that supports a pattern
* Using additional generic constraints

**C# 8.0**[[55]](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)#cite_note-C#8-58)

* readonly struct members
* default interface members
* switch expressions
* Property, Tuple, and positional patterns
* using declarations
* static local functions
* Disposable ref struct
* Nullable reference types
* Indices and Ranges
* Null-coalescing assignment
* Async Streams