var (C# Reference)

Beginning in Visual C# 3.0, variables that are declared at method scope can have an implicit "type" var. An implicitly typed local variable is strongly typed just as if you had declared the type yourself, but the compiler determines the type. The following two declarations of i are functionally equivalent:1

Copy

var i = 10; // implicitly typed

int i = 10; //explicitly typed

For more information, see [Implicitly Typed Local Variables](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/implicitly-typed-local-variables) and [Type Relationships in LINQ Query Operations](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/type-relationships-in-linq-query-operations).

Example

The following example shows two query expressions. In the first expression, the use of var is permitted but is not required, because the type of the query result can be stated explicitly as an IEnumerable<string>. However, in the second expression, var must be used because the result is a collection of anonymous types, and the name of that type is not accessible except to the compiler itself. Note that in Example #2, the foreach iteration variable item must also be implicitly typed.

C#Copy

// Example #1: var is optional because

// the select clause specifies a string

string[] words = { "apple", "strawberry", "grape", "peach", "banana" };

var wordQuery = from word in words

where word[0] == 'g'

select word;

// Because each element in the sequence is a string,

// not an anonymous type, var is optional here also.

foreach (string s in wordQuery)

{

Console.WriteLine(s);

}

// Example #2: var is required because

// the select clause specifies an anonymous type

var custQuery = from cust in customers

where cust.City == "Phoenix"

select new { cust.Name, cust.Phone };

// var must be used because each item

// in the sequence is an anonymous type

foreach (var item in custQuery)

{

Console.WriteLine("Name={0}, Phone={1}", item.Name, item.Phone);

}