**语句（C# 编程指南）**

程序所执行的操作以“语句”表达。常见操作包括声明变量、赋值、调用方法、循环访问集合，以及根据给定条件分支到一个或另一个代码块。语句在程序中的执行顺序称为“控制流”或“执行流”。根据程序对运行时所收到的输入的响应，在程序每次运行时控制流可能有所不同。

语句可以是以分号结尾的单行代码，或者是语句块中的一系列单行语句。语句块括在括号 {} 中，并且可以包含嵌套块。下面的代码演示两个单行语句示例和一个多行语句块：

static void Main()

{

// Declaration statement.

int counter;

// Assignment statement.

counter = 1;

// Error! This is an expression, not an expression statement.

// counter + 1;

// Declaration statements with initializers are functionally

// equivalent to declaration statement followed by assignment statement:

int[] radii = { 15, 32, 108, 74, 9 }; // Declare and initialize an array.

const double pi = 3.14159; // Declare and initialize constant.

// foreach statement block that contains multiple statements.

foreach (int radius in radii)

{

// Declaration statement with initializer.

double circumference = pi \* (2 \* radius);

// Expression statement (method invocation). A single-line

// statement can span multiple text lines because line breaks

// are treated as white space, which is ignored by the compiler.

System.Console.WriteLine("Radius of circle #{0} is {1}. Circumference = {2:N2}",

counter, radius, circumference);

// Expression statement (postfix increment).

counter++;

} // End of foreach statement block

} // End of Main method body.

} // End of SimpleStatements class.

/\*

Output:

Radius of circle #1 = 15\. Circumference = 94.25

Radius of circle #2 = 32\. Circumference = 201.06

Radius of circle #3 = 108\. Circumference = 678.58

Radius of circle #4 = 74\. Circumference = 464.96

Radius of circle #5 = 9\. Circumference = 56.55

\*/

**语句的类型**

下表列出 C# 中的各种语句类型及其关联的关键字，并提供指向包含更多信息的主题的链接：

| **类别** | **C# 关键字/说明** |
| --- | --- |
| 声明语句 | 声明语句引入新的变量或常量。变量声明可以选择为变量赋值。在常量声明中必须赋值。 |

// Variable declaration statements.

double area;

double radius = 2;

// Constant declaration statement.

const double pi = 3.14159;

|  |  |
| --- | --- |
| 表达式语句 | 用于计算值的表达式语句必须在变量中存储该值。 |

// Expression statement (assignment).

area = 3.14 \* (radius \* radius);

// Error. Not statement because no assignment:

//circ \* 2;

// Expression statement (method invocation).

System.Console.WriteLine();

// Expression statement (new object creation).

System.Collections.Generic.List&lt;string&gt; strings =

new System.Collections.Generic.List&lt;string&gt;();

|  |  |
| --- | --- |
| [选择语句](https://msdn.microsoft.com/zh-CN/library/676s4xab.aspx) | 选择语句用于根据一个或多个指定条件分支到不同的代码段。有关更多信息，请参见下列主题：[if](https://msdn.microsoft.com/zh-CN/library/5011f09h.aspx), [else](https://msdn.microsoft.com/zh-CN/library/5011f09h.aspx), [switch](https://msdn.microsoft.com/zh-CN/library/06tc147t.aspx), [case](https://msdn.microsoft.com/zh-CN/library/06tc147t.aspx) |
| [迭代语句](https://msdn.microsoft.com/zh-CN/library/32dbftby.aspx) | 迭代语句用于遍历集合（如数组），或重复执行同一组语句直到满足指定的条件。有关更多信息，请参见下列主题：[do](https://msdn.microsoft.com/zh-CN/library/370s1zax.aspx), [for](https://msdn.microsoft.com/zh-CN/library/ch45axte.aspx), [foreach](https://msdn.microsoft.com/zh-CN/library/ttw7t8t6.aspx), [in](https://msdn.microsoft.com/zh-CN/library/ttw7t8t6.aspx), [while](https://msdn.microsoft.com/zh-CN/library/2aeyhxcd.aspx) |
| [跳转语句](https://msdn.microsoft.com/zh-CN/library/d96yfwee.aspx) | 跳转语句将控制转移给另一代码段。有关更多信息，请参见下列主题：[break](https://msdn.microsoft.com/zh-CN/library/adbctzc4.aspx), [continue](https://msdn.microsoft.com/zh-CN/library/923ahwt1.aspx), [default](https://msdn.microsoft.com/zh-CN/library/06tc147t.aspx), [goto](https://msdn.microsoft.com/zh-CN/library/13940fs2.aspx), [return](https://msdn.microsoft.com/zh-CN/library/1h3swy84.aspx)，[yield](https://msdn.microsoft.com/zh-CN/library/9k7k7cf0.aspx) |
| [异常处理语句](https://msdn.microsoft.com/zh-CN/library/s7fekhdy.aspx) | 异常处理语句用于从运行时发生的异常情况正常恢复。有关更多信息，请参见下列主题：[throw](https://msdn.microsoft.com/zh-CN/library/1ah5wsex.aspx), [try-catch](https://msdn.microsoft.com/zh-CN/library/0yd65esw.aspx), [try-finally](https://msdn.microsoft.com/zh-CN/library/zwc8s4fz.aspx), [try-catch-finally](https://msdn.microsoft.com/zh-CN/library/dszsf989.aspx) |
| [检查和未检查](https://msdn.microsoft.com/zh-CN/library/khy08726.aspx) | 检查和未检查语句用于指定当将结果存储在变量中、但该变量过小而不能容纳结果值时，是否允许数值运算导致溢出。有关更多信息，请参见[检查](https://msdn.microsoft.com/zh-CN/library/74b4xzyw.aspx)和[未检查](https://msdn.microsoft.com/zh-CN/library/a569z7k8.aspx)。 |
| **await** 语句 | 如果标记与 [异步](https://msdn.microsoft.com/zh-CN/library/hh156513.aspx) 修饰符的方法，在方法可以使用 [等待](https://msdn.microsoft.com/zh-CN/library/hh156528.aspx) 运算符。当控件移到在异步方法中的一个 **await** 表达式，控件回调用方，因此，在方法的进度挂起，直到等待任务完成。当任务完成后，执行在方法可以恢复。有关简单示例，请参见 [方法（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/ms173114.aspx)“Async "方法”一节。有关更多信息，请参见[使用 Async 和 Await 的异步编程（C# 和 Visual Basic）](https://msdn.microsoft.com/zh-CN/library/hh191443.aspx)。 |
| **yield return** 语句 | 迭代器对集合的自定义迭代，如列表或数组。迭代器使用 [将返回](https://msdn.microsoft.com/zh-CN/library/9k7k7cf0.aspx) 语句返回每个元素一个节点。当 **yield return** 语句时，代码的当前位置确保。迭代器，当下次时，调用执行从该位置进行重新启动。有关更多信息，请参见[迭代器（C# 和 Visual Basic）](https://msdn.microsoft.com/zh-CN/library/dscyy5s0.aspx)。 |
| **fixed** 语句 | Fixed 语句禁止垃圾回收器重定位可移动的变量。有关更多信息，请参见 [fixed](https://msdn.microsoft.com/zh-CN/library/f58wzh21.aspx)。 |
| **lock** 语句 | lock 语句用于限制一次仅允许一个线程访问代码块。有关更多信息，请参见 [lock](https://msdn.microsoft.com/zh-CN/library/c5kehkcz.aspx)。 |
| 标记语句 | 可以为语句指定一个标记，然后使用 [goto](https://msdn.microsoft.com/zh-CN/library/13940fs2.aspx) 关键字跳转到该标记语句。（参见下一行中的示例。） |
| 空语句 | 空语句只含一个分号。空语句不执行任何操作，可以在需要语句但不需要执行任何操作的地方使用。下面的示例演示空语句的两种用法： |

void ProcessMessages()

{

while (ProcessMessage())

; // Statement needed here.

}

void F()

{

//...

if (done) goto exit;

//...

exit:

; // Statement needed here.

}

**嵌入语句**

一些语句（例如 [do](https://msdn.microsoft.com/zh-CN/library/370s1zax.aspx)、[while](https://msdn.microsoft.com/zh-CN/library/2aeyhxcd.aspx)、[for](https://msdn.microsoft.com/zh-CN/library/ch45axte.aspx) 和 [foreach](https://msdn.microsoft.com/zh-CN/library/ttw7t8t6.aspx)）后面始终跟有一条嵌入语句。此嵌入语句可以是单个语句，也可以是语句块中括在括号 {} 内的多个语句。甚至可以在括号 {} 内包含单行嵌入语句，如下面的示例所示：

// Recommended style. Embedded statement in block.

foreach (string s in System.IO.Directory.GetDirectories(

System.Environment.CurrentDirectory))

{

System.Console.WriteLine(s);

}

// Not recommended.

foreach (string s in System.IO.Directory.GetDirectories(

System.Environment.CurrentDirectory))

System.Console.WriteLine(s);

未括在括号 {} 内的嵌入语句不能作为声明语句或标记语句。下面的示例演示了这种情况：

if(pointB == true)

//Error CS1023:

int radius = 5;

将该嵌入语句放在语句块中以修复错误：

if (b == true)

{

// OK:

System.DateTime d = System.DateTime.Now;

System.Console.WriteLine(d.ToLongDateString());

}

**嵌套语句块**

语句块可以嵌套，如以下代码所示：

foreach (string s in System.IO.Directory.GetDirectories(

System.Environment.CurrentDirectory))

{

if (s.StartsWith("CSharp"))

{

if (s.EndsWith("TempFolder"))

{

return s;

}

}

}

return "Not found.";

**无法访问的语句**

如果编译器认为在任何情况下控制流都无法到达特定语句，将生成警告 CS0162，如下面的示例所示：

// An over-simplified example of unreachable code.

const int val = 5;

if (val < 4)

{

System.Console.WriteLine("I'll never write anything."); //CS0162

}