语句,表达式,运算符

# 语句

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”。

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。

## 语句的类型

<https://msdn.microsoft.com/zh-cn/library/ms173143.aspx>

# 表达式

语句的一种

表达式”是由一个或多个操作数以及零个或零个以上的运算符所组成的序列，可以通过计算得到一个值、对象、方法或命名空间等结果。

表达式可以包含文本值、方法调用、运算符及其操作数，或简单名称。

简单名称可以是变量、类型成员、方法参数、命名空间或类型的名称。

## 表达式值

## 溢出

计算结果 的值溢出操作

## 运算符的优先级和结合

计算表达式的方式由结合性和运算符优先级控制。有关更多信息，请参见 [运算符（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/ms173145.aspx)。性

## 文本和简单名称

最简单的两种表达式类型是文本和简单名称。文本是没有名称的常数值。

// Expression statements.

int i = 5;

string s = "Hello World";

等式左边为简单名称,右边为文本;

## 调用表达式

DoWork();

## 查询表达式

## lambda表达式

## 表达式树

# 运算符

运算符是应用于表达式或语句中的一个或多个操作数的程序元素。

接受一个操作数的运算符称为一元运算符，例如递增运算符 (**++**) 或**new**。

接受两个操作数的运算符称为二元运算符，例如算术运算符（**+**、**-**、**\***、**/**）。

条件运算符 **?:** 接受三个操作数，是 C# 中唯一的三元运算符。

## 运算符,计算和运算符优先级

每个运算符都具有已定义的优先级。 在包含具有不同优先级级别的多个运算符的表达式中，运算符的优先级确定运算符的计算顺序。

## 运算符类型/优先级

<https://msdn.microsoft.com/zh-cn/library/ms173145.aspx>

## 结合性

无论表达式中的运算符是左结合运算符还是右结合运算符，都将先从左至右评估各表达式的操作数。

当表达式中出现两个或两个以上具有相同优先级的运算符时，将根据结合性计算它们。 左结合运算符按从左到右的顺序计算。 例如，x \* y / z 将计算为 (x \* y) / z。 右结合运算符按从右到左的顺序计算。 例如，赋值运算符是右关联的。   
添加括号

优先级顺序

括号 <-- 优先级<--结合性

## 运算符重载

# 匿名函数

匿名函数是一个“内联”语句或表达式，可在需要委托类型的任何地方使用。

 可以使用匿名函数来初始化命名委托，或传递命名委托（而不是命名委托类型）作为方法参数。

## 两种匿名函数

### ?Lambda 表达式

Lambda 表达式可以绑定到表达式树，也可以绑定到委托。

### ?匿名方法