

## C# 基本语法

C# 是一种面向对象的编程语言。在面向对象的程序设计方法中，程序由各种相互交互的对象组成。相同种类的对象通常具有相同的类型，或者说，是在相同的 class 中。

例如，以 Rectangle（矩形）对象为例。它具有 length 和 width 属性。根据设计，它可能需要接受这些属性值、计算面积和显示细节。

让我们来看看一个 Rectangle（矩形）类的实现，并借此讨论 C# 的基本语法：

### 实例

```
using System;
namespace RectangleApplication
{
    class Rectangle
    {
        // 成员变量
        double length;
        double width;
        public void Acceptdetails()
        {
            length = 4.5;
            width = 3.5;
        }
        public double GetArea()
        {
            return length * width;
        }
        public void Display()
        {
            Console.WriteLine("Length: {0}", length);
            Console.WriteLine("Width: {0}", width);
            Console.WriteLine("Area: {0}", GetArea());
        }
    }

    class ExecuteRectangle
    {
        static void Main(string[] args)
        {
            Rectangle r = new Rectangle();
            r.Acceptdetails();
            r.Display();
            Console.ReadLine();
        }
    }
}
```

[尝试一下 »](#)

当上面的代码被编译和执行时，它会产生下列结果：

```
Length: 4.5
Width: 3.5
Area: 15.75
```

## using 关键字

在任何 C# 程序中的第一条语句都是：

```
using System;
```

**using** 关键字用于在程序中包含命名空间。一个程序可以包含多个 using 语句。

## class 关键字

**class** 关键字用于声明一个类。

## C# 中的注释

注释是用于解释代码。编译器会忽略注释的条目。在 C# 程序中，多行注释以 `/*` 开始，并以字符 `*/` 终止，如下所示：

```
/* This program demonstrates
The basic syntax of C# programming
Language */
```

单行注释是用 `//` 符号表示。例如：

```
//end class Rectangle
```

## 成员变量

变量是类的属性或数据成员，用于存储数据。在上面的程序中，*Rectangle* 类有两个成员变量，名为 *length* 和 *width*。

## 成员函数

函数是一系列执行指定任务的语句。类的成员函数是在类内声明的。我们举例的类 *Rectangle* 包含了三个成员函数：*AcceptDetails*、*GetArea* 和 *Display*。

## 实例化一个类

在上面的程序中，类 *ExecuteRectangle* 是一个包含 *Main()* 方法和实例化 *Rectangle* 类的类。

## 标识符

标识符是用来识别类、变量、函数或任何其它用户定义的项目。在 C# 中，类的命名必须遵循如下基本规则：

- 标识符必须以字母、下划线或 @ 开头，后面可以跟一系列的字母、数字（ 0 - 9 ）、下划线（ \_ ）、@。
- 标识符中的第一个字符不能是数字。
- 标识符必须不包含任何嵌入的空格或符号，比如 ? - + ! # % ^ & \* ( ) [ ] { } . ; : " ' / \。
- 标识符不能是 C# 关键字。除非它们有一个 @ 前缀。例如，@if 是有效的标识符，但 if 不是，因为 if 是关键字。
- 标识符必须区分大小写。大写字母和小写字母被认为是不同的字母。
- 不能与C#的类库名称相同。

## C# 关键字

关键字是 C# 编译器预定义的保留字。这些关键字不能用作标识符，但是，如果您想使用这些关键字作为标识符，可以在关键字前面加上 @ 字符作为前缀。

在 C# 中，有些关键字在代码的上下文中有特殊的意义，如 get 和 set，这些被称为上下文关键字（contextual keywords）。下表列出了 C# 中的保留关键字（Reserved Keywords）和上下文关键字（Contextual Keywords）：

保留关键字						
abstract	as	base	bool	break	byte	case
catch	char	checked	class	const	continue	decimal
default	delegate	do	double	else	enum	event
explicit	extern	false	finally	fixed	float	for
foreach	goto	if	implicit	in	in (generic modifier)	int
interface	internal	is	lock	long	namespace	new
null	object	operator	out	out (generic modifier)	override	params
private	protected	public	readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc	static	string	struct
switch	this	throw	true	try	typeof	uint
ulong	unchecked	unsafe	ushort	using	virtual	void
volatile	while					

上下文关键字

add	alias	ascending	descending	dynamic	from	get
global	group	into	join	let	orderby	partial (type)
partial (method)	remove	select	set			

← C# 程序结构

C# 数据类型 →



8 篇笔记

 写笔记