

C# 可空类型 (Nullable)

C# 单问号 ? 与 双问号 ??

? : 单问号用于对 int,double,bool 等无法直接赋值为 null 的数据类型进行 null 的赋值，意思是这个数据类型是 Nullable 类型的。

```
int? i = 3
等同于
Nullable<int> i = new Nullable<int>(3);

int i; //默认值0
int? ii; //默认值null
```

?? : 双问号 可用于判断一个变量在为 null 时返回一个指定的值。

接下来我们详细说明。

C# 可空类型 (Nullable)

C# 提供了一个特殊的数据类型，**nullable** 类型（可空类型），可空类型可以表示其基础值类型正常范围内的值，再加上一个 null 值。

例如，Nullable< Int32 >，读作“可空的 Int32”，可以被赋值为 -2,147,483,648 到 2,147,483,647 之间的任意值，也可以被赋值为 null 值。类似的，Nullable< bool > 变量可以被赋值为 true 或 false 或 null。

在处理数据库和其他包含可能未赋值的元素的数据类型时，将 null 赋值给数值类型或布尔型的功能特别有用。例如，数据库中的布尔型字段可以存储值 true 或 false，或者，该字段也可以未定义。

声明一个 **nullable** 类型（可空类型）的语法如下：

```
< data_type> ? <variable_name> = null;
```

下面的实例演示了可空数据类型的用法：

实例

```
using System;
namespace CalculatorApplication
{
    class NullablesAtShow
    {
        static void Main(string[] args)
        {
            int? num1 = null;
            int? num2 = 45;
            double? num3 = new double?();
            double? num4 = 3.14157;
```

```
bool? boolval = new bool?();

// 显示值

Console.WriteLine("显示可空类型的值: {0}, {1}, {2}, {3}",
                  num1, num2, num3, num4);
Console.WriteLine("一个可空的布尔值: {0}", boolval);
Console.ReadLine();

    }
}
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
显示可空类型的值:  , 45,  , 3.14157
一个可空的布尔值:
```

Null 合并运算符 (??)

Null 合并运算符用于定义可空类型和引用类型的默认值。Null 合并运算符为类型转换定义了一个预设值，以防可空类型的值为 Null。Null 合并运算符把操作数类型隐式转换为另一个可空（或不可空）的值类型的操作数的类型。

如果第一个操作数的值为 null，则运算符返回第二个操作数的值，否则返回第一个操作数的值。下面的实例演示了这点：

实例

```
using System;
namespace CalculatorApplication
{
    class NullablesAtShow
    {
        static void Main(string[] args)
        {
            double? num1 = null;
            double? num2 = 3.14157;
            double num3;
            num3 = num1 ?? 5.34;      // num1 如果为空值则返回 5.34
            Console.WriteLine("num3 的值: {0}", num3);
            num3 = num2 ?? 5.34;
            Console.WriteLine("num3 的值: {0}", num3);
            Console.ReadLine();
        }
    }
}
```

当上面的代码被编译和执行时，它会产生下列结果：

num3 的值: 5.34
num3 的值: 3.14157

← C# 方法

C# 数组 (Array) →

 点我分享笔记