1. 创建内插字符串

var test = "Bill";

Console.WriteLine( $"{test}" );

每个内插字符串必须具备的两个元素：

* 字符串文本以 $ 字符开头，后接左双引号字符。 $ 符号和引号字符之间不能有空格。 （如果希望看到包含空格会发生什么情况，请在交互式窗口中的 $ 字符后面插入一个空格，然后运行更新的代码。 C# 编译器报错“意外的字符 '$'”。）
* 一个或多个内插表达式。 左大括号和右大括号（{ 和 }）指示内插表达式。 可将任何返回值的 C# 表达式置于大括号内（包括 null）。

1. 内插字符串表达式的结果可以是任何数据类型。

如果内插表达式的计算结果为 null，则会使用一个空字符串（"" 或 [String.Empty](https://docs.microsoft.com/zh-cn/dotnet/api/system.string.empty)）。

如果内插表达式的计算结果不是 null，通常会调用结果类型的 ToString 方法。 可以通过更新 Vegetable.ToString 方法的实现来进行测试。 你甚至不用实现 ToString 方法，因为每个类型都有一些此方法的实现。 可通过注释掉示例中 Vegetable.ToString 方法的定义（在它前面添加注释符号 // 即可）来进行测试。 在输出中，字符串“eggplant”被替换为完全限定的类型名称（由 C# REPL 定义的名称空间以及类型名称），这是 [Object.ToString()](https://docs.microsoft.com/zh-cn/dotnet/api/system.object.tostring#System_Object_ToString) 方法的默认行为。 对于枚举值的 ToString 方法，其默认行为是返回该值的字符串表示形式。

1. 控制内插表达式的格式

示例代码：

var date = DateTime.Now;

var price = 100;

var unit = "kilo";

var item = "西红柿";

Console.WriteLine($"On {date:d}, the price of {item} was {price:C2} per {unit}.");

可通过在内插表达式后接冒号（“:”）和格式字符串来指定格式字符串。 “d”是[标准日期和时间格式字符串](https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/standard-date-and-time-format-strings#the-short-date-d-format-specifier)，表示短日期格式。“C2”是[标准数值格式字符串](https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/standard-numeric-format-strings#the-currency-c-format-specifier)，用数字表示货币值（精确到小数点后两位）。

.NET 库中的许多类型支持一组预定义的格式字符串。 这些格式字符串包括所有数值类型以及日期和时间类型。 有关支持格式字符串的完整类型列表，请参阅 [.NET 中的格式化类型](https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/formatting-types)文章中的[格式字符串和. NET 类库类型](https://docs.microsoft.com/zh-cn/dotnet/standard/base-types/formatting-types#stringRef)。

尝试修改示例中的格式字符串，查看更改如何影响日期、时间和数值的格式。 将 {date:d} 中的“d”更改为“t”（显示短时间格式）、“y”（显示年份和月份）和“yyyy”（显示四位数年份）。 将 {price:C2} 中的“C2”更改为“e”（用于指数计数法）和“F3”（使数值在小数点后保持三位数字）。

除了控制格式之外，还可以控制结果字符串中包含

1. 控制内插表达式的字段宽度和对齐方式:

通过在内插表达式后面添加一个逗号（“,”）并指定“最小”字段宽度来指定对齐方式。 如果指定的值是正数，则该字段为右对齐。 如果它为负数，则该字段为左对齐。