《VB程序设计》

教 师：田翔华（副教授）
Email：xj303@163.com

新疆医科大学

教学目标及重难点

- 教学目标

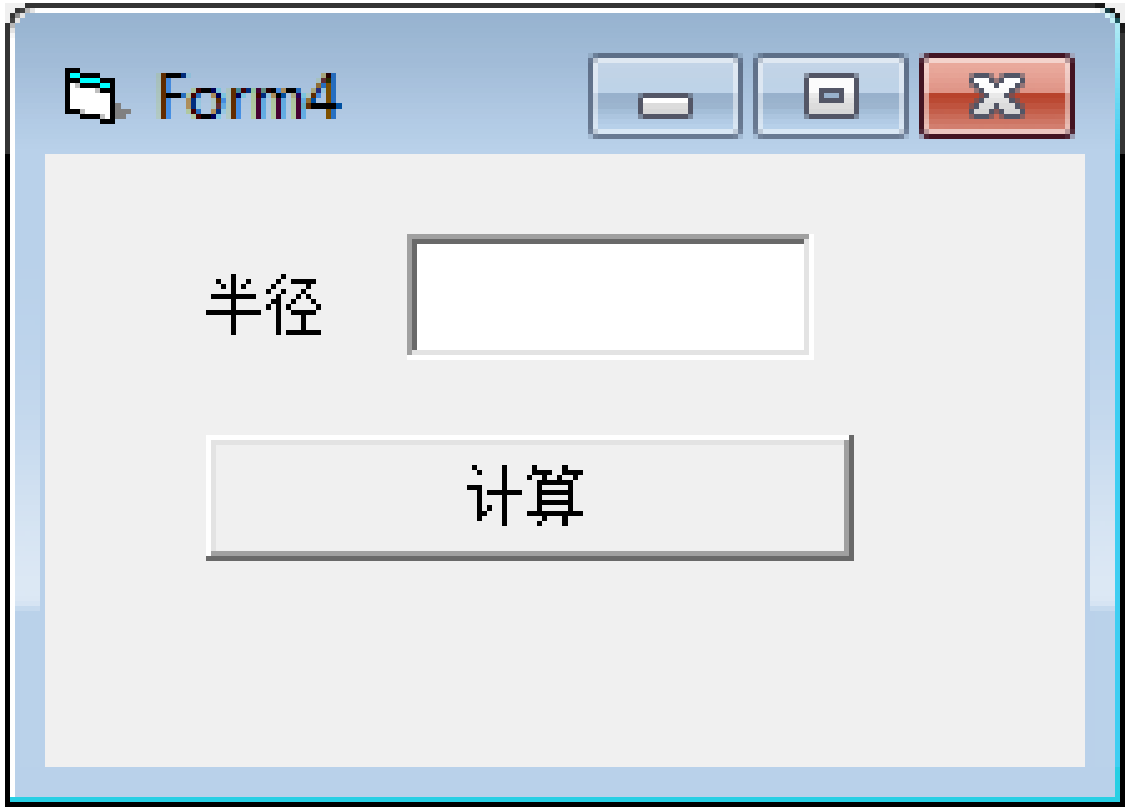
掌握VB数据类型、变量或常量的命名规则、变量和常量的声明使用；运算符和表达式、常用函数、功能及使用方法；编码规则。

- 重点和难点

VB数据类型；变量或常量的命名规则、变量和常量的声明使用；运算符和表达式、常用函数、功能及使用方法

教学案例

❖案例:给定圆的半径，计算圆的周长和面积。



Visual Basic 语言基础

2.1 数据类型

2.1.1 数值型数据

2.1.2 字符型数据

2.1.3 布尔型数据

2.1.4 日期型数据

2.1.5 变体型数据

2.1.6 对象型数据

2.1.7 自定义类型

Visual Basic 语言基础

2.2 变量

2.3 常量

2.4 数组（后面单独介绍）

2.5 运算符

Visual Basic 语言基础

2.1 数据类型

- 数据类型决定了数据的存储和运算方式。

数据类型	存储空间	数据范围
Byte（字节型）	1 个字节	0～255
Boolean（布尔型）	2 个字节	True 或 False
Integer（整型）	2 个字节	-32,768 ～ 32,767
Long（长整型）	4 个字节	-2,147,483,648 ～ 2,147,483,647
Single（单精度浮点型）	4 个字节	负数： -3.402823E38 ～ -1.401298E-45 正数： 1.401298E-45 ～ 3.402823E38
Double（双精度浮点型）	8 个字节	负数： -1.79769313486232E308 ～ -4.94065645841247E-324 正数： 4.94065645841247E-324 ～ 1.79769313486232E308
Currency（货币型）	8 个字节	-922337203685477.5808 ～ 922337203685477.5807
Date（日期型）	8 个字节	100 年 1 月 1 日 ～ 9999 年 12 月 31 日
Object（对象型）	4 个字节	任何 Object 引用
String（字符型）	变长：10 个字节加字符串长度	0 ～ 20 亿字节
	定长：字符串长度	1 ～ 65535 字节
Variant（变体型）	数字：16 个字节	任何数字，最大范围与 Double 类型相同
	字符：22 个字节加字符串长度	与变长字符串的范围相同
用户自定义（用 Type 定义）	所有元素占用的存储空间	每个元素的范围与它本身的数据类型的范围相同

2.1.1 数值型数据

❖ **整型**(Integer)　整型数据用来表示不带小数点的数值。

如：*10、+256、0、-68*

整型数也可以用十六进制数表示，以&H开头，数据位数不超过4位，范围为 &H0 ～ &HFFFF，如：&H64表示十进制数100，&HA6表示十进制数166。

❖ **长整型**(Long)　与整型类似，但取值范围比整型数大。

整型或长整型数据的运算速度较快，而且比其它数据类型占据的内存要少。

❖ **字节型**(Byte)　可用来表示无符号的整数，不能表示负数。除一元减法外，所有可对整数进行操作的运算符均可操作 Byte 数据类型。

Byte 数据类型在存储二进制数据时很有用。

2.1.1 数值型数据

❖ **单精度浮点型**(Single)　浮点型数据用来表示一个实数，小数点可以放在数字的任何位置。

单精度数的最大有效位数为7，超过7位时，四舍五入。如：

-4.5、12.3、+12.34、0.0069、-31.24683

❖ **双精度浮点型**(Double)　双精度数比单精度数的取值范围大，最大有效位数为15 。如：

-12345678901234 5、0.98765432101234、1234567890.12345

❖ **货币型**(Currency)　货币型数据可表示15位整数和4位小数，是一个定点数据类型（小数点位置固定），适用于货币计算。其精度高于单精度和双精度浮点数。

2.1.2 字符型数据

❖ **字符型数据**(String) 是用双引号（”“）括起来的一串字符，包括汉字、字母、数字、符号等，如：

"Visual Basic"、"教学管理系统V1.0"、"123456789"

❖ **字符串的类型**

- **变长字符串** 是指字符串的长度可以改变。当给一个变长字符串变量赋予不同的值时，其长度可增可减。
- **定长字符串** 是指在程序执行过程中，长度始终保持不变的字符串。例如，定义一个长度为20个字符的字符串变量：

```
Dim strTemp As String *20
```

如果赋予字符串的字符少于20个，则不足部分用空格填充；如果超过20个，则多出的字符被截去。

2.1.2 字符型数据

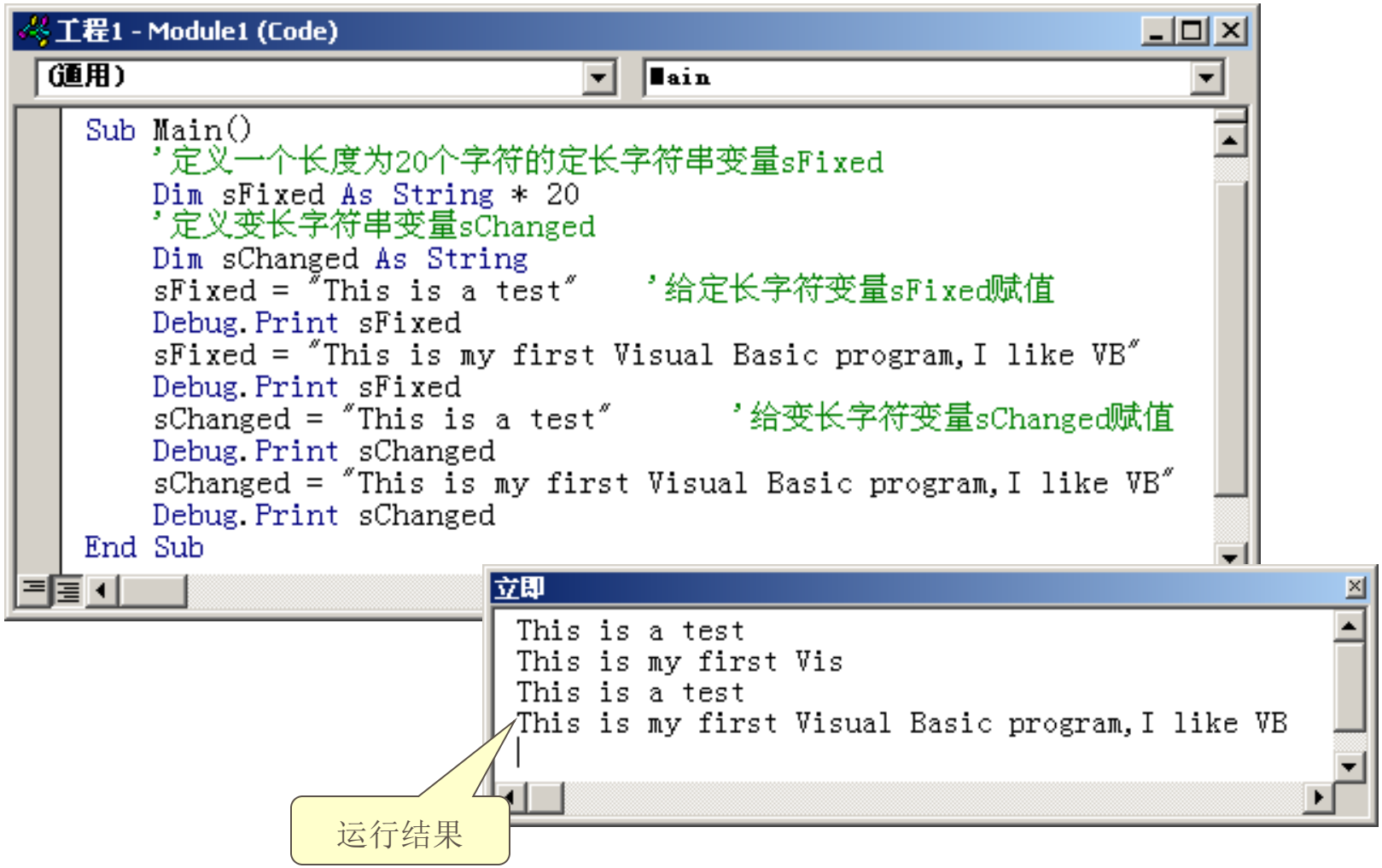
【例3.2】 字符型数据的使用。

① 选择“文件|新建工程”命令，新建一个“标准EXE”类型的工程文件。然后执行“工程|添加模块”命令，在当前工程中添加一个名为“Module1(Module1)”的模块。

② 在工程资源管理器中右击“工程1”项，从快捷菜单中选择“工程1属性”命令，打开“工程属性”对话框，选择“通用”选项卡，从“启动对象”下拉列表框中选择“Sub Main”选项。当编写无窗体的应用程序时，必须将启动对象设置为“Sub Main”过程，作为程序运行的起始点。

③ 在代码窗口中输入代码。结果在立即窗口中输出。

2.1.2 字符型数据


工程1 - Module1 (Code)
(通用)
ain
Sub Main()
'定义一个长度为20个字符的定长字符串变量sFixed
Dim sFixed As String * 20
'定义变长字符串变量sChanged
Dim sChanged As String
sFixed = "This is a test" '给定长字符变量sFixed赋值
Debug.Print sFixed
sFixed = "This is my first Visual Basic program,I like VB"
Debug.Print sFixed
sChanged = "This is a test" '给变长字符变量sChanged赋值
Debug.Print sChanged
sChanged = "This is my first Visual Basic program,I like VB"
Debug.Print sChanged
End Sub
立即
This is a test
This is my first Vis
This is a test
This is my first Visual Basic program,I like VB
运行结果

2.1.2 字符型数据

❖ 说明：

- VB是用双引号来标识字符串的，当遇到第一个“时，系统就认为是字符串的开始，当遇到下一个”时，就认为是字符串的结束。
- 当字符串本身要包含“时，可以输入两个连续的”“，这时VB就会判定它是一个单纯的双引号，而不会将它作为字符串的开始或结束标志。

例如，要显示字符串This is my first "VB" program，

可以在立即窗口中输入以下语句：

```
? "This is my first ""VB"" program"
```

2.1.3 布尔型数据

❖ **布尔型数据**（Boolean） 只有两个值：真(True)和假(False)，通常用来表示逻辑判断的结果，也称之为逻辑型数据。

任何只有两种状态的数据，如真/假、是/否、开/关等，都可以用Boolean型表示。

❖ **说明：**

- 当把数值型数据转换为Boolean型时，0会转换为False，其他非0值转换为True。
- 当把Boolean值转换为数值型时，False转换为0，True转换为-1。

2.1.4 日期型数据

- **日期型数据**（Date） 用来表示日期和时间。Date型数据要用两个“#”号作为定界符，例如：

  2003年1月12日表示为：

  #1/12/2003#

  2003年1月12日下午1时20分45秒表示为：

  #1/12/2003 1:20:45 pm#

- 如果输入的日期或时间是非法的，系统将显示出错信息。

2.1.5 变体型数据

❖ **变体型数据**（Variant） 能够表示任何数据类型。

把常量赋予Variant型数据时，VB按以下规则判定其类型：

- 用双引号括起来的数据是字符型
- 用#号括起来的数据是日期型
- 值为True或False的数据是布尔型
- 对于数值型数据，如果不含小数位，且大小不超出整型数范围就被看作是整型；如果数值大于整型数范围但不超过长整型数范围，就被看作是长整型；除此以外，其它的数值型数据都被看作双精度浮点型。

2.1.6 对象型数据

❖ **对象型数据**(Object)　用来存储应用程序中的某个对象，用4个字节来表示该对象在内存中的地址引用。

一个被声明为 Object 的变量可以引用应用程序所识别的任何对象。例如：

```
Dim objDb As Object
Set objDb = OpenDatabase ("c:\Vb6\Biblio.mdb")
```

上述两条语句首先声明一个名为objDb的对象型变量，然后引用了应用程序中的一个数据库对象。

2.1.7 自定义类型

❖ **自定义类型**　用Type语句创建，可以包含多个不同类型的数据元素、数组或一个已定义过的自定义类型。

例如，自定义一个名为Student的数据类型：

```
Type Student
    Dim Number   As Long      ' 声明Number为长整型变量
    Dim Name     As String    ' 声明Name为字符型变量
    Dim Birthday As Date      ' 声明Birthday为日期型变量
    Dim Sex As Boolean        ' 声明Sex为布尔变量
End Type
```

2.2 变量

❖ **变量** 是指在数据处理过程中其值可以改变的量。

每个变量都有一个名字（变量名）和一种数据类型，（确定该变量能够存储哪种数据）。

❖ **变量的命名**

- 变量必须以字母开头，由汉字、字母、数字或下划线组成，不能包含空格、!、#、@、$、%、&、.等字符，且长度不超过256个字符。
- 不能使用VB中的关键字、对象名称或属性名称。

2.2 变量

❖ 变量的声明

- 格式：

  Dim〈变量1〉[As 数据类型][,〈变量 2〉[As 数据类型] ……]

- 功能：声明变量并分配存储空间。

使用AS类型子句可以指定变量的数据类型，如果没有指定数据类型，默认为Variant型。

声明一个变量后，该变量的初始值默认为：数值型数据为0，字符型数据为空字符串（""），日期时间型数据为0:00:00，逻辑型数据为False。

2.2 变量

例如，下列语句分别声明了String 、Date和 Single类型的变量：

```
Dim  strName  As String
Dim  datBirthday  As  Date
Dim  fltScore  As Single
```

在 Dim 语句中声明多个变量时，如果没有用As子句指定类型，就被认为是Variant类型。例如：

```
Dim strName As String, datBirthday As Date
```

该语句分别声明了一个字符型变量和一个日期型变量。

```
Dim strName, strAddress As String
```

该语句将strAddress声明为字符型变量，strName默认为Variant型变量。

2.2 变量

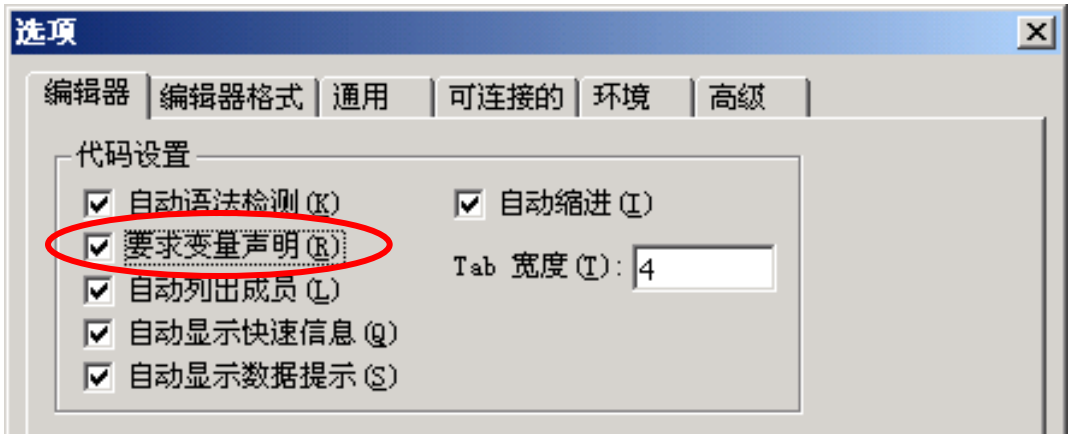
❖ 变量的隐式声明

- 先用Dim语句声明一个变量，然后再使用该变量的方式称作显式声明。
- 不事先声明某个变量而直接在程序中使用的方式称作隐式声明。在执行程序时，VB会根据该变量名自动创建一个Variant型变量。

2.2 变量

❖ **强制声明变量** 方法是：

- 在程序开头加上一条语句：*Option Explicit*。
- 选择“工具|选项”命令，选中“要求变量声明”选项。VB将会在任何一个新建立的模块中自动插入Option Explicit语句。



2.2 变量

❖ **类型声明字符** 在变量名的后面加一个类型声明字符，也可以指定变量的数据类型。

常用类型声明字符

数据类型	声明字符
String	$
Integer	%
Long	&
Single	!
Double	#

2.2 变量

- 在Dim语句中可以使用类型声明符号来代替“As”子句。

  例如：*Dim iTemp%*　　等同于

  Dim iTemp As Integer

- 当一个变量未经声明第一次使用时，若后面加上了类型声明字符，就会被自动声明为该类型声明字符所代表的数据类型。

  例如：*a% = 55*　　' 隐式声明变量a并指定其数据类型为整型

- 给Variant型变量赋值时，可以使用类型声明字符指定类型。

  例如：*Dim varTemp As Variant*

  varTemp = 10&　　'指定varTemp中存储的数据为长整型

2.3 常量

- **常量**　指在程序运行过程中其值始终保持不变的量。
- **类型**
  - **系统常量**　是系统内部预定义的常量，以小写的vb字母开头，如：vbRed（代表红色数值）。在程序中可以直接使用系统常量。
  - **用户定义的常量**　使用Const语句自定义的常量。
    - 格式：*Const〈常量名〉[As 数据类型]＝〈表达式〉*
    - 功能：声明用于代替文字量的常数。

例如，自定义一个双精度浮点型常量Pi：

```
Const Pi = 3.1415926
```

用Const语句定义常量后，就可以在程序的其他位置使用这个常量，但不能再用任何代码去改变该常量的值。

2.5 运算符

- 运算符是对相同类型的数据进行运算操作的符号。
- 用运算符将常量、变量和函数等数据连接起来的式子称为表达式。表达式的类型通常由运算符的类型决定，每个表达式按照规定的运算规则产生一个惟一的值。

2.5.1 赋值运算符

- **赋值运算符** 就是“=”号。
  - 格式：*〈变量〉=〈表达式〉*
  - 说明：“=”右边可以是一个常量或已赋过值的变量，也可以是一个表达式。如果是表达式，VB将先计算表达式的值，再将此值赋给“=”左边的变量。例如：

```
a = 3        ' 将数值常量3赋予变量a
b = a        ' 将变量a的值赋予变量b
' 将表达式2 * a + b的运算结果赋予变量c
c = 2 * a + b
```

变量还可以用它当前的值参加运算，再将运算结果赋给变量自身。例如：X=X+1 表示先计算表达式 X+1，再将运算结果存回变量 X中，替换原来的值。

2.5.2 算术运算符

❖ **算术运算符** 用来进行数学计算的运算符。运算优先级依次为：括号→指数运算→负数→乘、除、整型除法、求模→加、减，其中乘、除、整型除法和求模同级，加和减同级，分别从左到右进行计算。

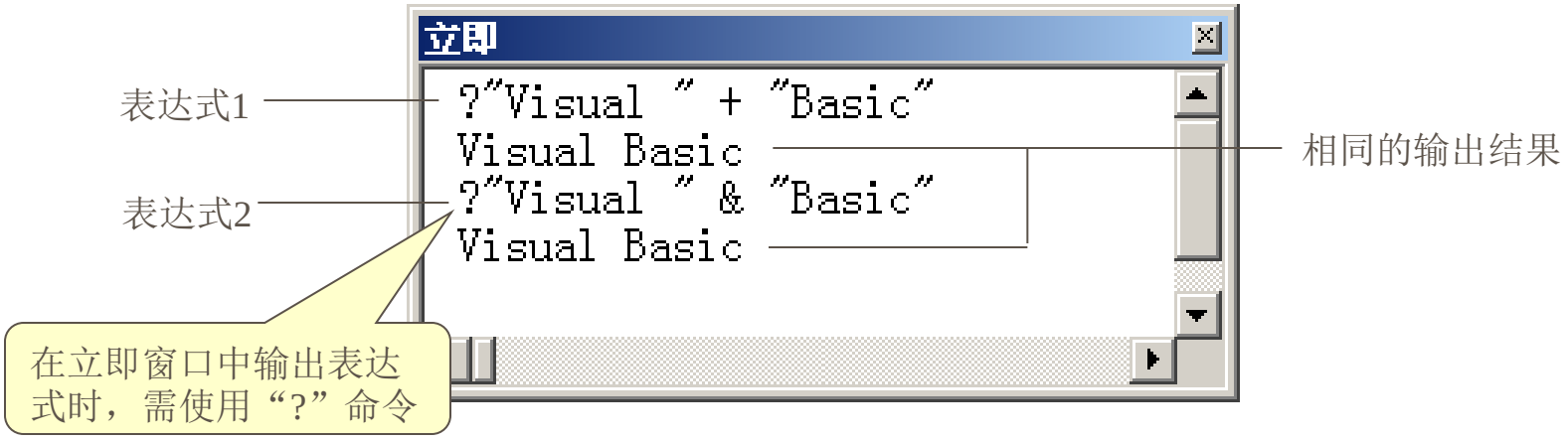
算术运算符

名称	运算符	名称	运算符
括号	()	整型除法	\
指数运算	^	求模运算	Mod
负号	-	加法	+
乘法	*	减法	-
除法	/		

2.5.3 连接运算符

- **连接运算符** 将两个字符串连接起来合并为一个字符串。
- **连接运算符的类型** 有 “+”和“&” 两种
  - 当运算符前后的数据都是字符串时，两者完全相同。

例如， 在立即窗口中输入两个表达式，结果是一样的。



在立即窗口中输出表达式

2.5.3 连接运算符

- 运算符前后的数据不都是字符串时，两者的处理方式不同。
  - ✓ “+” 有处理数值运算的作用。当一个字符串与一个数值数据相加时，如果该字符串是一个数值格式的字符串，则VB先将字符串转换成数值，然后再相加，此时进行的是数值相加操作。如果字符串不能转换成数值的格式，就会发生错误。例如，

```
? “12” + 45        ' 结果为 57（12+45）
? "abc"+ 45        ' 错误
```

  - ✓ “&” 是专门用来进行字符串连接的运算符。使用“&”时，不管是何种类型的数据，都会先转换成字符串再进行连接。例如，

```
? 12.34 & 43       ' 结果为1 2.3443
? “a12bc” & 56     ' 结果为 a12bc56
```

2.5.4 日期运算符

- 日期型运算符有“+”和“-”两种。
  - 〈日期时间〉±〈数值〉，结果为一个新的日期时间。〈数值〉的单位是“天”。
  - 〈日期时间〉-〈日期时间〉，结果为两个日期相差的天数。

例如，

```
? #2003-4-28# + 10
```

结果是：原来的日期加上10天，为 2003-5-8

```
? #3-24# - #1-21#
```

结果是：两个日期相减，相差62天

2.5.5 比较运算符

❖ **比较运算符**　用来对两个表达式进行比较运算。运算结果为Boolean值（True或False）。比较运算符有 6 种，优先级相同。

比较运算符

等于	大于	小于	大于等于	小于等于	不等于
=	>	<	>=、=>	<=、=<	<>

❖ **说明**

- 参加比较的数据的类型必须一致。
- 数值型数据按数值的大小比较，西文字符按ASCII码的值比较，汉字按国标码的值比较，日期时间型数据分别按年月日和时分秒的值比较。

2.5.5 比较运算符

例如，

① 数值的比较

? 1=2, 1<=2, 1<>1.1

结果分别是 False，True，True

② 字符串的比较

? “张” > “帐”

结果是 True

③ 日期时间的比较

? #2003-2-2 12:12:12# < #2003-3-1#

结果是 True

? #2003-3-1 12:12:12# < #2003-3-1#

结果是 False

2.5.6 逻辑运算符

- **逻辑运算符** 用来对表达式执行逻辑运算。运算结果为Boolean型（True或False）。
- **类型**
  - AND（与）：参与运算的两个表达式的值均为真，结果才为真，否则结果为假。
  - OR（或）：参与运算的两个表达式只要有一个值为真，结果就为真，否则结果为假。
  - NOT（取反）：表达式为真时，结果为假；表达式为假时，结果为真。
  - XOR（异或）：两个表达式的值都为真或假时，结果为假，否则，结果为真。
- 逻辑运算符的优先级：NOT→AND→OR→XOR

2.5.6 逻辑运算符

- 同一表达式中各类运算符的优先顺序由高到低依次为：

括号→算术运算符→字符串运算符→关系运算符→逻辑运算符

例如，

```
? 20>5 AND 8<4
```

结果为 False

```
? 20>5 XOR 8<4
```

结果为 True

```
? True XOR (2=3) OR (3=3) AND NOT(5=5)
```

结果为 True

2.2.7 位运算符

- **位运算** 先将整型数据转换为二进制数，然后将其中的每一位分别进行运算处理，其中0相当于False，1相当于True。
- 逻辑运算符可以作为位运算符使用。

例如，

```
Dim a, b As Byte
a = 5
b = 9
? a And b        ' 结果为1
? a Or b         ' 结果为13
? Not b          ' 结果为246
? a XOR b        ' 结果为12
```

2.2.7 位运算符

❖ **说明：** 进行位运算时，VB先将十进制数5和9转换成二进制，$(5)_{10}=(00000101)_2$，$(9)_{10}=(00001001)_2$，运算过程下表所示。

位运算示例

A=5	0	0	0	0	0	1	0	1	结果
B=9	0	0	0	0	1	0	0	1	
AND	0	0	0	0	0	0	0	1	1
OR	0	0	0	0	1	1	0	1	13
NOT	1	1	1	1	0	1	1	0	246
XOR	0	0	0	0	1	1	0	0	12

2.6 函数

- **函数**（Function） 是一种能够完成某种特定操作或功能的程序段。函数的运算结果称为函数值。
- **函数调用的格式**
  - *Call <函数名>([参数1][,参数2][,……])*
  - *<函数名> [参数1][,参数2][,……]*
  - *变量 = <函数名>([参数1][,参数2][,……])*

  说明：任何可以使用表达式的地方都可以使用函数，表达式将函数的返回值作为运算对象。
- **函数分类**
  - 系统函数：VB提供的内部函数。
  - 用户自定义函数：用户根据程序设计的需要自行编写的函数。

2.6.1 输入输出函数

使用输入输出函数可以实现一些简单的人机对话。

❖ 输入函数

- **格式：** *InputBox(〈提示信息〉[,〈标题〉][,〈默认值〉]*
- **功能：** 显示一个用户自定义的对话框，等待用户输入文本或按下按钮，并返回用户在文本框中输入的字符串。

例如，在立即窗口中输入以下语句：

？InputBox(“请输入学生性别”，“学生情况”，“男”)

结果如图所示。

2.6.1 输入输出函数

❖ 输出函数

- **格式：** *MsgBox(<提示信息>[, Buttons] [, <标题>])*
- **功能：** 显示一个消息对话框，并等待用户单击按钮，作为继续执行程序的依据。
- **说明：**
  - ✓ <Buttons> 参数是一个数值表达式，指定消息框中的按钮类型、图标样式和默认按钮等，默认值为0。
  - ✓ 用户在消息框中单击某一按钮后，该函数返回一个整数值，表示某个按钮被选中。

2.6.1 输入输出函数

按钮类型

常量名	值	显示按钮
VbOKOnly	0	确定
VbOKCancel	1	确定、取消
VbAbortRetryIgnore	2	终止、重试、忽略
VbYesNoCancel	3	是、否、取消
VbYesNo	4	是、否
VbRetryCancel	5	重试、取消

2.6.1 输入输出函数

图标样式和默认按钮

常量名	值	图标样式	常量名	值	默认按钮
VbCritical	16		VbDefaultButton1	0	第一个按钮
VbQuestion	32		VbDefaultButton2	256	第二个按钮
VbExclamation	48		VbDefaultButton3	512	第三个按钮
VbInformation	64		VbDefaultButton4	768	第四个按钮

2.6.1 输入输出函数

MsgBox 函数的返回值

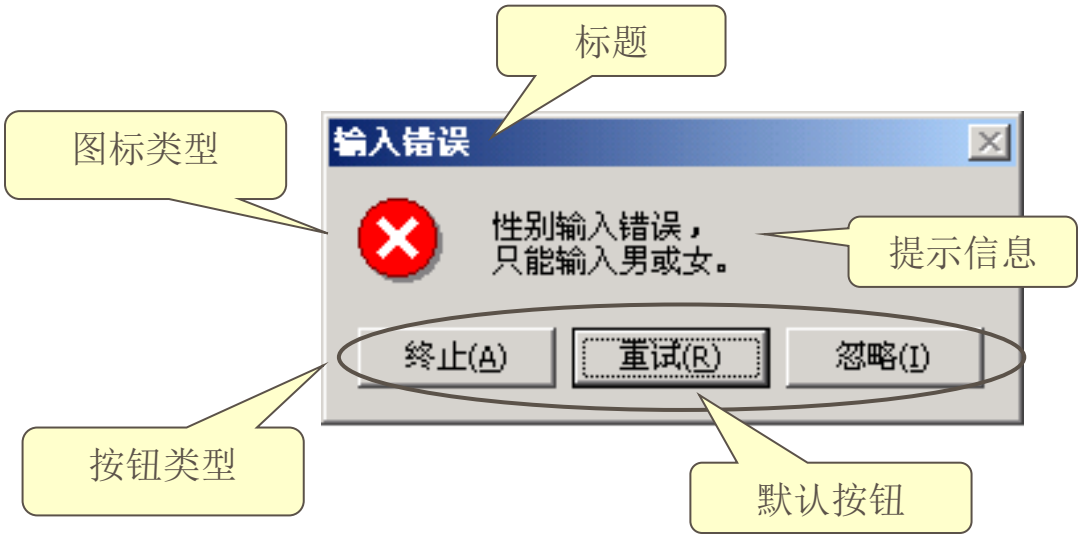
常量名	值	选择按钮
vbOK	1	确定
vbCancel	2	取消
vbAbort	3	终止
vbRetry	4	重试
vbIgnore	5	忽略
vbYes	6	是
vbNo	7	否

2.6.1 输入输出函数

例如，在立即窗口中输入以下语句：

```
?MsgBox("性别输入错误," & Chr(10)& "只能输入男或女。", _
        2 + 256 + 16, _
        "输入错误")
```

结果如图所示。



2.6.2 数学函数

常用数学函数

函 数	功 能	示 例
Abs(<数值>)	求指定数值的绝对值	abs(-1) 结果为 1
Round(<数值>[,小数位数])	四舍五入运算	Round(1.26,1) 结果为 1.3
Int(<数值>)	取整运算	Int(-1.8) 结果为 -2
Fix(<数值>)	求指定数值的整数部分	Fix(-1.8) 结果为 -1
Sin(<数值>)	求指定数值的正弦值	
Cos(<数值>)	求指定数值的余弦值	
Tan(<数值>)	求指定数值的正切值	
Atn(<数值>)	求指定数值反正切值	
Sqr(<数值>)	求正数的算术平方根	Sqr(25) 结果为 5
Log(<数值>)	求正数的自然对数	
Exp(<数值>)	求 e 的幂次方	
Rnd(<数值>)	求一个大于 0 且小于 1 的 Single 类型的随机数	Rnd(1) 结果为 0.7055475

2.6.3 字符处理函数

函 数	功 能	示 例
Len(<字符串>)	求字符串中的字符个数	len(" 北京大学 123") 结果为 8
Left(<字符串>, <字符数>)	从字符串左边选取子字符串	left("abcdefgh",5) 结果为 abcde
Right(<字符串>, <字符数>)	从字符串右边选取子字符串	right("abcdefgh",5) 结果为 defgh
Mid(<字符串>, <位置>, <字符数>])	从字符串的指定位置处选取子字符串	mid("abcdefgh",3,5) 结果为 cdefg
InStr(<字符串 1>, <字符串 2>)	查找字符串 2 在字符串 1 中首次出现的位置	InStr("abcdefghde","de") 结果为 4
InstrRev(<字符串 1>, <字符串 2>)	从后往前查找字符串 2 在字符串 1 中首次出现的位置	InStrRev("abcdefghde","de") 结果为 9
Replace(<字符串 1>, <字符串 2>, <字符串 3>)	在字符串 1 中查找所有的字符串 2，并用字符串 3 替换，然后返回替换后的字符串	Replace("ab12ab","12","XY") 结果为 abXYab
LTrim(<字符串>)	删除字符串左边的空格	x = " abc def " LTrim(x) 结果为"abc def " RTrim(x) 结果为" abc def" Trim(x) 结果为"abc def"
RTrim(<字符串>)	删除字符串右边的空格	
Trim(<字符串>)	删除字符串两边的空格	
UCase(<字符串>)	将小写字母转换成大写字母	UCase("a1BC") 结果为 A1BC
LCase(<字符串>)	将大写字母转换成小写字母	LCase("a1BC") 结果为 a1bc
Space(<数值>)	生成若干空格组成的字符串	Space(3) 结果为" "
String(<n>,<字符>)	生成由 n 个字符组成的字符串	String(3,"x") 结果为"xxx"

2.6.4 日期函数

常用的日期函数

函　数	功　能	示　例
Now()	取得系统当前的日期和时间	dt1=#2003-5-10 14:15:23# Year(dt1) 结果为 2003 Hour(dt1) 结果为 14 DatePart("h",dt1) 结果为 14 假设系统当前的日期时间为 2003-05-11 06:48:56 dt2=Now() DateValue(dt2) 结果为 2003-05-11 TimeValue(dt2) 结果为 06:48:56
Date()	取得系统当前的日期	
Time()	取得系统当前的时间	
Year(<Date>)	取得 Date 的年份	
Month(<Date>)	取得 Date 的月份	
Day(<Date>)	取得 Date 的日期	
WeekDay(<Date>)	取得 Date 的星期值	
Hour(<Date>)	取得 Date 的小时	
Minute(<Date>)	取得 Date 的分钟	
Second(<Date>)	取得 Date 的秒数	
DateValue(<Date>)	取得 Date 的日期	
TimeValue(<Date>)	取得 Date 的时间	
DatePart(时间单位,<Date>)	取得 Date 的各部分时间	

2.6.5 格式化函数

❖ 数值格式化函数

- 格式：*FormatNumber(数值[,小数位数[,前导0字符[,负数格式[,数字分组]]]])*
- 功能：按指定的数据格式对数值数据进行格式化。

例如，`? FormatNumber(-123456.789,2,,,-1)`

结果为 -123,456.79

❖ 日期时间格式化函数

- 格式：*FormatDateTime(日期[,格式])*
- 功能：按指定的日期时间格式对日期时间数据格式化。

2.6.5 格式化函数

❖ 通用格式化函数

- *格式：Format(表达式[, 格式])*
- 功能：按指定的格式对表达式进行格式化。

例如，

```
Dat = #5/10/2003 20:12#
? Format(Dat, "hh:mm am/pm")
```

结果为 08:12 pm

```
? Format(Dat,“现在是yyyy年m月”)
```

结果为：现在是2003年5月

```
? Format(514, “总分是#######.0”)
```

结果为：总分是514.0

2.6.6 数据类型转换函数

❖ 常用转换函数

- 强制将一个表达式转换成某种特定类型的数据。

函　数	功　能	示　例
Cbyte(<表达式>)	将表达式的值转换为 Byte 型数据	CLng(432.15) 结果为 432 CBool(20)　结果为 True CBool(0)　结果为 False CStr(3245.23) 结果为字符串 3245.23 CDate(37000.768)　结果为 2001-04-19 18:25:55
Cint(<表达式>)	将表达式的值转换为 Integer 型数据	
CLng(<表达式>)	将表达式的值转换为 Long 型数据	
CSng(<表达式>)	将表达式的值转换为 Single 型数据	
CDbl(<表达式>)	将表达式的值转换为 Double 型数据	
Ccur(<表达式>)	将表达式的值转换为 Currency 型数据	
Cbool(<表达式>)	当表达式的值为 0 时，结果为 False；否则都为 True	
Cdate(<表达式>)	将表达式的值转换为 Date 型数据	
CStr(<表达式>)	将表达式的值转换为字符串。	

2.6.6 数据类型转换函数

❖ 数值转换为字符串函数

- 格式：*Str(<数值>)*
- 功能：将数值型数据转换成字符串。

例如，`? str(123)` ' 结果为字符串 123

❖ 字符串转换为数值函数

- 格式：*Val(<字符串>)*
- 功能：将字符串转换为Double型的数值。

例如，`? Val("122.45")` ' 结果为数值 122.45

2.6.7 测试函数

❖ 数字测试函数

- 格式：*IsNumeric(<表达式>)*
- 功能：测试表达式的值是否为数值型数据或符合数值格式的字符串，若是，返回值为True；否则，返回值为False。

例如，
```
? isNumeric(12.25)      ' 结果为 True
? isNumeric("ab")       ' 结果为 False
```

❖ 日期测试函数

- 格式：*IsDate(<表达式>)*
- 功能：测试表达式的值是否为日期时间型数据或符合日期时间格式的字符串，若是，返回值为True；否则，返回值为False。

例如，
```
? isDate(#2003-5-20 8:30#)    ' 结果为 True
? isDate("#2003-5-20#")       ' 结果为 False
```

2.6.7 测试函数

❖ 数组测试函数

- 格式：*IsArray(<表达式>)*
- 功能：测试表达式是否为数组，若是，返回值为True；否则，返回值为False。

例如，
```
Dim XX(5) As Single
Dim YY As Single
? IsArray(XX)      ' 结果为True
? IsArray(YY)      ' 结果为False
```

❖ 数据类型测试函数

- 格式：*TypeName(<表达式>)*
- 功能：测试表达式的数据类型。

例如，
```
Dim X1 As Single
? TypeName(X1)     ' 结果为Single
```

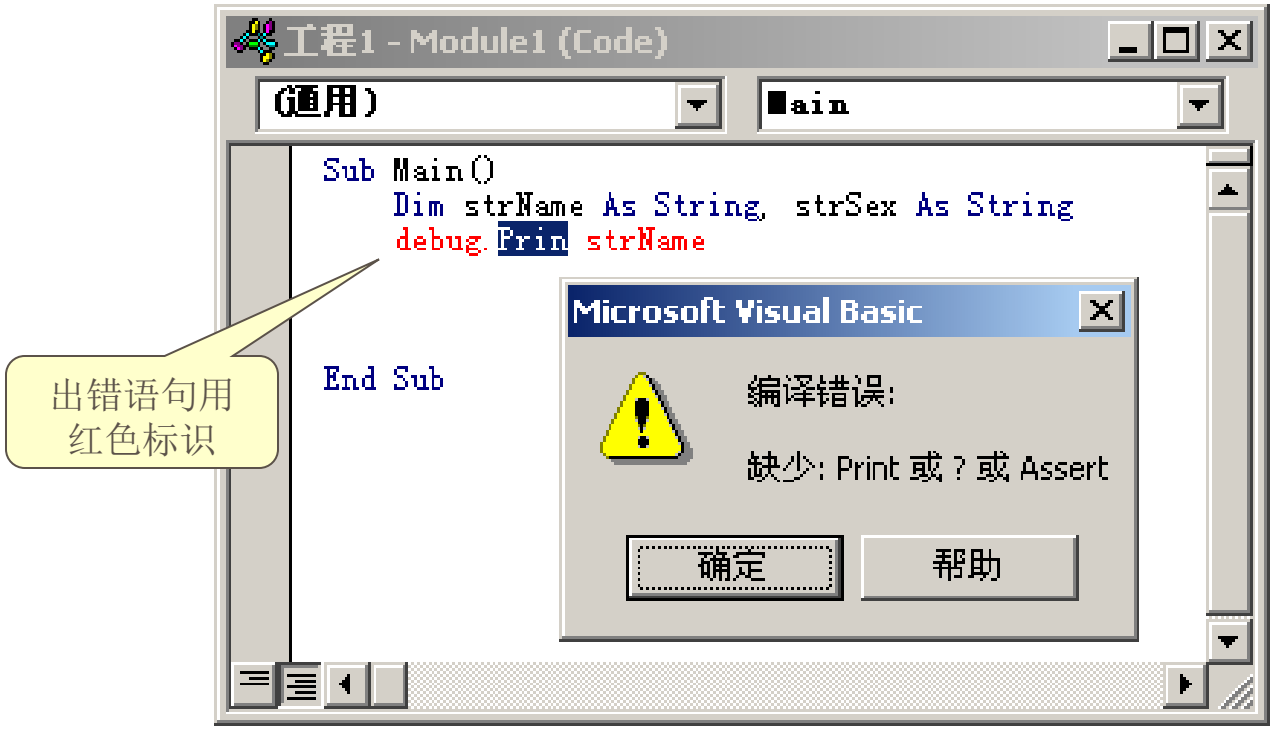
2.7 语句

- **语句** 是指程序中的一行代码，由VB关键字、函数、运算符以及VB可识别的指令组成。

- **语句的书写规则**
  - 通常是一条语句占一行，行尾不加任何终结符。
  - 将两个或多个语句放在同一行时，各语句之间要用冒号（:）隔开。
  - 在一行语句的末尾加续行符“_”（下划线），可以将一条语句分成数行书写。
  - 一行语句中以单撇号（'）开始的内容为注释信息。程序中的注释内容不会影响程序运行的结果。

2.7 语句

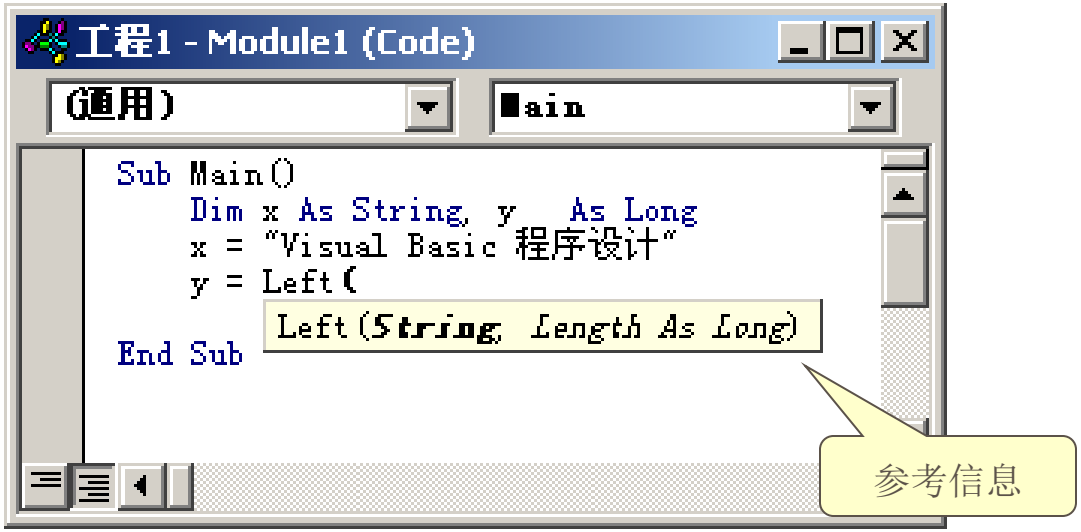
- 输入完一行代码并按回车键后，VB会自动对输入的内容进行语法检查；发现错误时，会显示一个提示框。



自动语法检查

2.7 语句

- VB会按约定对语句进行格式化处理，例如，关键字和函数的第1个字母自动变为大写。
- 在代码窗口中输入语句时，可以自动显示有关语句和函数语法格式的快速信息。



自动显示快速信息

编码小规则

- 定义
- 输入
- 计算（处理）
- 输出

小结

通过学习，掌握VB数据类型、变量或常量的命名规则、变量和常量的声明使用；运算符和表达式、常用函数、功能及使用方法；编码规则。