

ตัวแปรและชนิดของข้อมูล - Contents

ชนิดของข้อมูล (Data Types)	2
Scientific Notation.....	3
คำสงวน (Keywords).....	3
ตัวแปร (Variables)	4
การใช้งานคำสั่ง Debug.Print.....	7
การคำนวณทางคณิตศาสตร์.....	8
การใช้งานสตริง (String)	9
การกำหนดค่าให้ตัวแปร	9
การเขียนคำอธิบายโปรแกรม.....	10
ขอบเขตของตัวแปร	11
ตัวแปรแบบ Local	11
ตัวแปรแบบ Form-Level	11
ชนิดของข้อผิดพลาด	12
Syntax Errors	12
Run-time Errors.....	12
Logical Errors	12
การใช้งานคำสั่งสำเร็จรูป.....	13
คำสั่งเกี่ยวกับคณิตศาสตร์	13
Sqr Function	13
Int Function.....	13
Round Function.....	13
Val Function.....	14
คำสั่งเกี่ยวกับสตริง.....	14
Left Function.....	14
Mid Function	15
Right Function	15
Str Function	16
UCase Function	16
LTrim, RTrim, and Trim Functions.....	16
Len Function.....	17

ชนิดของข้อมูล (Data Types)

- การเขียนโปรแกรมเป็นการนำชุดคำสั่งต่าง ๆ มาเรียบเรียงเพื่อกำหนดวิธีการทำงานให้แก่คอมพิวเตอร์ คำสั่งที่นำมาใช้บางคำสั่งต้องทำการคำนวณ บางคำสั่งต้องรับข้อมูลจากผู้ใช้
- ข้อมูลพื้นฐานที่ใช้ในการเขียนโปรแกรมด้วย Visual Basic มีดังนี้

<i>ชนิดของข้อมูล</i>	<i>คำอธิบาย</i>
Boolean	ใช้เก็บค่าพิเศษคือ TRUE หรือ FALSE
Integer	ใช้เก็บค่าเลขจำนวนเต็มระหว่าง -32768 ถึง 32767
Long	ใช้เก็บค่าเลขจำนวนเต็มระหว่าง -2,147,483,648 ถึง 2,147,483,647
Single	ใช้เก็บค่าเลขจำนวนจริงระหว่าง $-3.402823E+38$ ถึง $3.402823E+38$
Double	ใช้เก็บค่าเลขจำนวนจริงระหว่าง $-1.79769313486232E+308$ ถึง $1.79769313486232E+308$
String	ใช้เก็บตัวอักษร ข้อความ หรือตัวเลข มีความยาวตั้งแต่ 0 ถึง 65,500 ตัวอักษร
Variant	เก็บข้อมูลได้ทุกชนิด

– ตัวอย่าง - integer

21

0

-9455

– ตัวอย่าง - long

32768

-95445

492848559

– ตัวอย่าง - single

0.01

565.32

-192.3424

6.5440E-24

– ตัวอย่าง - double
-5968.5765934211133
4.532112E+92
928374.344838273567899990

– ตัวอย่าง - string
"London Bridge"
"1932 Sycamore Street"
"^%#@#%\$%3939\$%^&^&"

– ตัวอย่าง - ข้อมูลที่ไม่สามารถใช้ได้
14,005.5
6.8%
33-
\$190.04
15 78
3.5&

Scientific Notation

- การใช้สัญลักษณ์ E แทนการแสดงข้อมูลตัวเลขในรูปของ 10 ยกกำลัง

Scientific	Value	Power of 10
1.4E+09	1400000000	$14 * 10^9$
3E-07	0.0000003	$3 * 10^{-7}$

คำสงวน (Keywords)

- คำสงวน หมายถึง คำศัพท์เฉพาะที่ใช้เป็นส่วนหนึ่งของภาษา Visual Basic
 - ตัวอย่าง - คำสงวน
Dim
As
Print
Cls
If
While

ตัวแปร (Variables)

- ตัวแปร หมายถึง สิ่งที่สามารถเปลี่ยนแปลงได้ในการเก็บข้อมูลต่าง ๆ ระหว่างที่โปรแกรมทำงาน
- คุณลักษณะที่สำคัญของตัวแปร คือ
 - ตัวแปรจะต้องมีชื่อ
 - ตัวแปรจะต้องมีชนิดของข้อมูล
- หลักการตั้งชื่อตัวแปร
 - ต้องขึ้นต้นด้วยตัวอักษรเท่านั้น
 - หลังจากนั้นจะเป็นตัวเลขหรือตัวอักษรก็ได้
 - ห้ามตั้งชื่อตัวแปรซ้ำกับคำสั่ง

– ตัวอย่าง - ชื่อตัวแปรที่ใช้ได้

MyNum	I	strInputValue
Number2	timeElapsed	taxRate

– ตัวอย่าง - ชื่อตัวแปรที่ใช้ไม่ได้

1Week	Earnings.To	Number One
maximum/average	1stChoice	square yard

- ก่อนที่จะนำตัวแปรมาใช้งานจะต้องมีการประกาศตัวแปรให้โปรแกรมรู้จัก โดยมีรูปแบบคำสั่งดังนี้

รูปแบบการใช้งาน

DIM ชื่อตัวแปร AS ชนิดของข้อมูล

– ตัวอย่าง - การประกาศตัวแปร

Dim Length As Integer

Dim Price As Single

Dim Structure As Double

Dim FirstName As String

- การเขียนโปรแกรมที่ดีนิยมระบุชนิดของข้อมูลของตัวแปรไว้ในชื่อตัวแปร โดยการ
ใช้ Prefix ซึ่งช่วยให้ทราบได้ทันทีว่าตัวแปรใช้เก็บข้อมูลชนิดใด

Prefix	Data Type	Example
bln	Boolean	blnIsOverTime
dbl	Double	dblMicroMeasurement
int	Integer	intCount
lng	Long	lngStarDistance
sng	Single	sngYearSales
str	String	strLastName
var	Variant	varControlValue

- ตัวอย่าง - การประกาศตัวแปรโดยใช้ Prefix

```
Dim intLength As Integer
Dim dblStructure As Double
Dim strFirstName As String
Dim varValue As Variant
```

- ตัวอย่าง - การประกาศตัวแปรหลายตัวในบรรทัดเดียวกัน

```
Dim A As Integer
Dim B As Double
Dim C As Integer
Dim D As String
Dim E As String
```

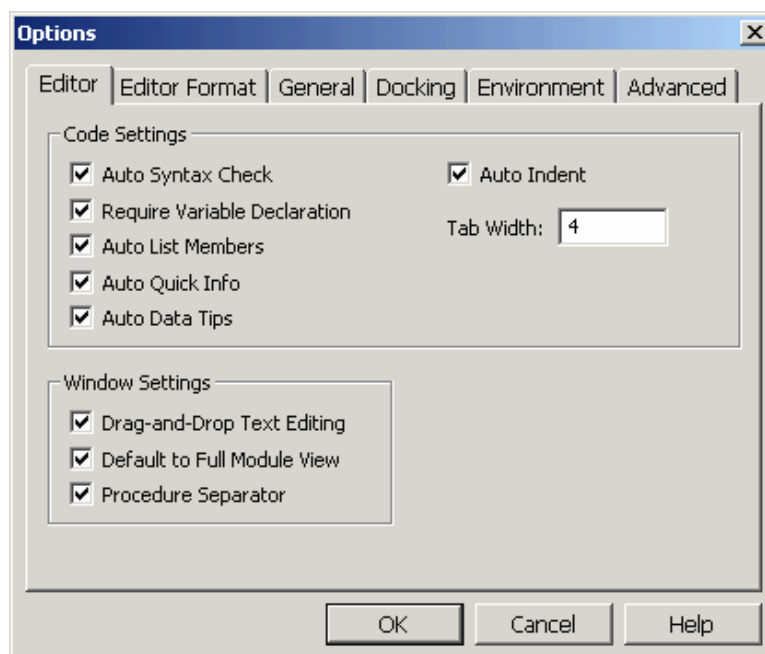
สามารถเขียนใหม่ได้ดังนี้

```
Dim A As Integer, C As Integer
Dim B As Double
Dim D As String, E As String
```

- หลังจากการประกาศตัวแปร จะทำให้ตัวแปรแต่ละตัวมีค่าเริ่มต้นตามชนิดของข้อมูล ดังนี้

Data Type	Value
Integer	0
Long	0
Single	0
Double	0
String	"" (blank)
Boolean	False
Variant	EMPTY

- การประกาศตัวแปรใน Visual Basic สามารถทำได้ 2 แบบ คือ
 - **Implicit Declaration** หมายถึง Visual Basic ยอมให้ใช้งานตัวแปรได้โดยไม่ต้องมีการประกาศตัวแปร ชนิดของข้อมูลที่ได้จะเป็นแบบ Variant
 - **Explicit Declaration** หมายถึง จะต้องมีการประกาศตัวแปรก่อนการใช้งานทุกครั้ง จึงสามารถนำตัวแปรนั้นๆไปใช้งานได้ การประกาศตัวแปรแบบ Explicit ให้เลือกเมนู Tools => Option เลือกหัวข้อ Require Variable Declaration หลังจากกำหนด Option จะปรากฏคำว่า **Option Explicit** ในส่วนต้นของโปรแกรม



การใช้งานคำสั่ง Debug.Print

- คำสั่ง Debug.Print ใช้สำหรับพิมพ์ค่าของข้อมูลต่าง ๆ ที่เก็บอยู่ในตัวแปร หรือค่าที่ได้จากการคำนวณ
- ผลลัพธ์ที่ได้จากคำสั่ง Debug.Print จะแสดงที่ immediate window

– ตัวอย่าง

```
Private Sub cmdCompute_Click()
```

```
    Debug.Print 3 - 2
```

```
    Debug.Print 3 * 2
```

```
    Debug.Print 3 / 2
```

```
    Debug.Print 3 ^ 2
```

```
    Debug.Print 2 * (3 + 4)
```

```
End Sub
```

- สามารถใช้ ; (semicolon) สำหรับการพิมพ์ข้อมูลหลายๆ ค่าในบรรทัดเดียวกัน

– ตัวอย่าง

```
Debug.Print speed
```

```
Debug.Print taxRate
```

```
Debug.Print "Class average is"; total / 3
```

การคำนวณทางคณิตศาสตร์

- การคำนวณทางคณิตศาสตร์ใน Visual Basic มีการคำนวณเช่นเดียวกับการคำนวณตัวเลขทั่วไป

Mathematical	Visual Basic	Comment
$a + b$	$a + b$	Addition
$a - b$	$a - b$	Subtraction
a / b	a / b	Division
$a \times b$	$a * b$	Multiplication
a^r	$a ^ r$	Exponentiation
Div	$a \setminus b$	Div ($7 \setminus 2 \Rightarrow 3$)
Mod	$a \text{ mod } b$	Mod ($7 \text{ mod } 2 \Rightarrow 1$)

- การคำนวณค่าตัวเลขใน Visual Basic ใช้ลำดับในการคำนวณ ดังนี้

Operation	Operator	Comment
Parentheses	()	Left => Right
Exponentiation	^	Caret
Negation (unary)	-	
Multiplication, Division	*, /	/ Forward
Integer division	\	\ Backward
Integer Modulus	mod	
Addition, Subtraction	+, -	

- ตัวอย่าง - ลำดับในการคำนวณ

```
Private Sub cmdCompute_Click()
    Debug.Print 3 + 2
    Debug.Print 3 - 2
    Debug.Print 3 * 2
    Debug.Print 3 / 2
    Debug.Print 3 ^ 2
    Debug.Print 2 * (3 + 4)
    Debug.Print 3 \ 2
    Debug.Print 3 Mod 2
    Debug.Print 1.2 * 10 ^ 34
    Debug.Print 1.2 * 10 ^ 8
    Debug.Print 1.2 * 10 ^ 3
End Sub
```


การใช้งานสตริง (String)

- สตริงต้องเขียนอยู่ในเครื่องหมายคำพูด เช่น "Somchai"
- การนำ String 2 ชุดมาต่อกันให้ใช้เครื่องหมาย & เช่น "Good" & "Bye" ได้ผลลัพธ์เป็น "GoodBye"

– ตัวอย่าง - การใช้งาน String

```
Private Sub cmdCompute_Click()  
    Debug.Print "Good"  
    Debug.Print "Bye"  
    Debug.Print "Good" & " " & "Bye"  
End Sub
```

การกำหนดค่าให้ตัวแปร

- การกำหนดค่าให้แก่ตัวแปร ทำได้โดยใช้คำสั่ง **assignment**

รูปแบบการใช้งาน

ชื่อตัวแปร = Expression

- Expression หมายถึง ค่าที่ได้จากการคำนวณ หรือตัวแปรอื่น

– ตัวอย่าง

```
Dim intAge As Integer  
Dim dblTemp as Double  
Dim strName As String  
intAge = 18  
dblTemp = 42.1  
strName = "Samart"
```

– ตัวอย่าง

```
Dim sngSales As Single, sngNewSales As Single  
sngSales = 3945.42  
sngNewSales = sngSales
```

– ตัวอย่าง

```
Private Sub Command1_Click ()  
    Label1.Caption = "Brush your teeth daily!"  
    Label1.BorderStyle = 1  
End Sub
```

– ตัวอย่าง

```
Dim strVar1 as String
Dim strVar2 as String
Dim strVar3 as String
Dim strVar4 as String
strVar1 = "Hello"
strVar2 = "World"
strVar3 = strVar1 & strVar2           ' HelloWorld
strVar4 = strVar1 & " " & strVar2    ' Hello World
```

– ตัวอย่าง - ไม่ถูกต้อง

```
Dim sngTodayTemp As Single
sngTodayTemp = 42.1
sngTodayTemp = "Forty-Two point One"
```

– ตัวอย่าง - ไม่ถูกต้อง

```
10 = count
```

– ตัวอย่าง - ไม่ถูกต้อง

```
count + 1 = count
```

– ตัวอย่าง - ไม่ถูกต้อง

```
Dim intCount as integer
Dim strName as string
Dim intTest as integer
intTest = intCount + strName
```

การเขียนคำอธิบายโปรแกรม

- ในบางครั้งหากต้องการเขียนคำอธิบายการทำงานของโปรแกรมในแต่ละบรรทัด ให้ใช้เครื่องหมาย ' (*apostrophe*) นำหน้าประโยคที่ต้องการ เช่น

```
Private Sub cmdCompute_Click()
    'This program use for calculate salary
    Dim intHrs as Integer           'Hours worked
    Dim dblRate as Double           'Hours rate
    Dim dblSal as Double            'Salary
    dblSalary = intHrs * dblRate
End Sub
```

ขอบเขตของตัวแปร

ตัวแปรแบบ Local

- เมื่อมีการประกาศตัวแปรภายในโปรแกรมย่อย ตัวแปรที่ประกาศจะมองเห็นเฉพาะภายในโปรแกรมย่อยเท่านั้น เรียกตัวแปรแบบนี้ว่า *"Local Variable"*

- ตัวอย่าง - ตัวแปรแบบ Local

```
Private Sub cmdCompute_Click()  
    Dim intAge As Integer  
    Dim dblTemp As Double  
    Dim strName As String  
End Sub
```

ตัวแปรแบบ Form-Level

- ถ้าต้องการประกาศตัวแปรให้สามารถเรียกใช้ได้จากทุก ๆ ส่วนในโปรแกรมจะต้องประกาศตัวแปรในระดับ Form

- ตัวอย่าง - ตัวแปรระดับ Form

```
Option Explicit  
Dim num1 As Integer, num2 As Integer
```

```
Private Sub cmdCompute_Click()  
    num1 = num1 + 1  
    num2 = num2 + num1  
    Debug.Print num1; num2  
End Sub
```

- ตัวอย่าง - ตัวแปรระดับ Form และตัวแปรแบบ Local

```
Option Explicit  
Dim num1 As Integer, num2 As Integer, num3 As Integer
```

```
Private Sub cmdCompute_Click()  
    Dim num3 As Integer  
    num1 = num1 + 1  
    num2 = num2 + num1  
    num3 = num3 + num2 + num1  
    Debug.print num1; num2; num3  
End Sub
```

– ตัวอย่าง - การให้ค่าเริ่มต้นแก่ตัวแปรระดับ Form

```
Option Explicit  
Dim pi As Double
```

```
Private Sub Form_Load()  
    pi = 3.14159  
End Sub
```

```
Private Sub cmdCompute_Click()  
    Debug.Print "The area of a circle of radius 5 is "; pi * 5 * 5  
End Sub
```

ชนิดของข้อผิดพลาด

Syntax Errors

- เกิดขึ้นในขั้นตอนการ Compile โปรแกรมเนื่องจากรูปแบบของคำสั่งไม่ถูกต้องตามหลักไวยากรณ์ เช่น

<i>Statement</i>	<i>Syntax Errors</i>
picBox.Print 3	Keyword
picBox.Print 2 +	Arithmetic
9W = 5	Assignment

Run-time Errors

- เกิดขึ้นในขั้นตอนการ Run โปรแกรมเนื่องจากข้อผิดพลาดบางอย่าง เช่น
 $z = x / y$ $y = 0$

Logical Errors

- เกิดขึ้นจากการสั่งงานผิดพลาด ทำให้ได้ผลลัพธ์ไม่ตรงตามที่ต้องการ เช่น
 $average = x1 + x2 / 2$

การใช้งานคำสั่งสำเร็จรูป

คำสั่งเกี่ยวกับคณิตศาสตร์

Sqr Function

- Returns a Double specifying the square root of a number.

Sqr(number)

- The required number argument is a Double or any valid numeric expression greater than or equal to zero

Dim MyNumber as Double

```
MyNumber = Sqr(4)           ' Returns 2
MyNumber = Sqr(23)        ' Returns 4.79583152331272
MyNumber = Sqr(0)         ' Returns 0
MyNumber = Sqr(-4)       ' Generates a run-time error
```

Int Function

- Returns the integer portion of a number.

Int(number)

- The required number argument is a Double or any valid numeric expression.

Dim MyNumber as Double

```
MyNumber = Int(99.8)      ' Returns 99
MyNumber = Int(-99.8)    ' Returns -100
MyNumber = Int(-99.2)    ' Returns -100
```

Round Function

- Returns a number rounded to a specified number of decimal places.

Round(expression [,decimalplaces])

- expression - Required Numeric expression being rounded
- decimalplaces - Number indicating how many places to the right of the decimal are included in the rounding. If omitted, integers are returned by the Round function.

Dim MyNumber as Double

```
MyNumber = Round(2.7)     ' Returns 3
MyNumber = Round(2.317, 2) ' Returns 2.32
MyNumber = Round(2.317, 1) ' Returns 2.3
```

Val Function

- Returns the numbers contained in a string as a numeric value of appropriate type.

Val(string)

- The required string argument is any valid string expression.

```
Dim MyValue as Double
MyValue = Val("2457")           ' Returns 2457
MyValue = Val(" 2 45 7")       ' Returns 2457
MyValue = Val("24 and 57")     ' Returns 24
```

คำสั่งเกี่ยวกับสตริง

Left Function

- Returns a String containing a specified number of characters from the left side of a string

Left(string, length)

- string - String expression from which the leftmost characters are returned.
- length - Numeric expression indicating how many characters to return. If greater than or equal to the number of characters in string, the entire string is returned.

```
Dim AnyString, MyStr as
AnyString = "Hello World"
MyStr = Left(AnyString, 1) ' Returns "H"
MyStr = Left(AnyString, 7) ' Returns "Hello W"
MyStr = Left(AnyString, 20) ' Returns "Hello World"
```

Mid Function

- Returns a String containing a specified number of characters from a string.

Mid(string, start[, length])

- string - String expression from which characters are returned.
- start - Character position in string at which the part to be taken begins. If start is greater than the number of characters in string, Mid returns a zero-length string ("").
- length - Number of characters to return. If omitted or if there are fewer than length characters in the text, all characters from the start position to the end of the string are returned.

```
Dim MyString, FirstWord, LastWord, MidWords
MyString = "Mid Function Demo"
FirstWord = Mid(MyString, 1, 3)           ' Returns "Mid"
LastWord = Mid(MyString, 14, 4)         ' Returns "Demo"
MidWords = Mid(MyString, 5)             ' Returns "Function Demo"
```

Right Function

- Returns a Variant (String) containing a specified number of characters from the right side of a string.

Right(string, length)

- string - String expression from which the rightmost characters are returned.
- length - Numeric expression indicating how many characters to return. If 0, a zero-length string ("") is returned. If greater than or equal to the number of characters in string, the entire string is returned.

```
Dim AnyString, MyStr
AnyString = "Hello World"               ' Define string.
MyStr = Right(AnyString, 1)             ' Returns "d".
MyStr = Right(AnyString, 6)             ' Returns " World".
MyStr = Right(AnyString, 20)            ' Returns "Hello World".
```

Str Function

- Returns a String representation of a number.

Str(number)

- The required number argument is a Long containing any valid numeric expression.

```
Dim MyString
MyString = Str(459)           ' Returns " 459"
MyString = Str(-459.65)      ' Returns "-459.65"
MyString = Str(459.001)     ' Returns " 459.001"
```

UCase Function

- Returns a String containing the specified string, converted to uppercase.

UCase(string)

- The required string argument is any valid string expression. If string contains Null, Null is returned.

```
Dim LowerCase, UpperCase
LowerCase = "Hello World 1234"      ' String to convert
UpperCase = UCase(LowerCase)       ' "HELLO WORLD 1234"
```

LTrim, RTrim, and Trim Functions

- Returns a String containing a copy of a specified string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).

LTrim(string)

RTrim(string)

Trim(string)

- The required string argument is any valid string expression.

```
Dim MyString, TrimString
MyString = " <-Trim-> "
TrimString = LTrim(MyString)       ' "<-Trim-> "
TrimString = RTrim(MyString)       ' " <-Trim->"
TrimString = LTrim(RTrim(MyString)) ' "<-Trim->"
TrimString = Trim(MyString)         ' "<-Trim->"
```


Len Function

- Returns a Long containing the number of characters in a string or the number of bytes required to store a variable.

Len(string | varname)

- string Any valid string expression.

```
Dim MyString, MyLen
```

```
MyString = "Hello World"
```

```
MyLen = Len(MyString) ' Returns 11.
```