



## ชนิดข้อมูล ตัวแปร ค่าคงที่ และตัวดำเนินการ

การเขียนโปรแกรมภาษาใดก็ตามจะต้องมีการใช้งานข้อมูล (Data) ซึ่งอยู่ในรูปแบบต่างๆ เช่น ข้อความ , ตัวเลข, วันเวลา ฯลฯ โดยผู้ใช้งานจะนำข้อมูลเหล่านั้นมาคำนวณหรือประมวลผลในรูปแบบต่างๆ ได้ก็ต่อเมื่อนำมาจัดเก็บไว้ในหน่วยความจำภายในโปรแกรมที่เรียกว่าตัวแปร (Variable)

### 2.1 ชนิดข้อมูล (Data Type) ในภาษา C

ในภาษา C กำหนดชนิดของข้อมูลไว้หลากหลายชนิดเพื่อรองรับการจัดเก็บข้อมูลหลายๆ ประเภท ดังนี้

ชนิด	ขนาดความกว้าง	ช่วงของค่า	การใช้งาน
Char	8 บิต	ASCII character (-128 ถึง 127)	เก็บข้อมูลชนิดอักขระ
Unsigned char	8 บิต	0-255	เก็บข้อมูลอักขระแบบไม่คิดเครื่องหมาย
Int	16 บิต	-32768 ถึง 32767	เก็บข้อมูลชนิดจำนวนเต็ม
long	32 บิต	-2147483648 ถึง 2147483649	เก็บข้อมูลชนิดจำนวนเต็มแบบยาว
Float	32 บิต	3.4E-38 ถึง 3.4E+38 หรือ ทศนิยม 6	เก็บข้อมูลชนิดเลขทศนิยม
Double	64 บิต	1.7E-308 ถึง 1.7E+308 หรือ ทศนิยม 12	เก็บข้อมูลชนิดเลขทศนิยม
Unsigned int	16 บิต	0 ถึง 65535	เก็บข้อมูลชนิดจำนวนเต็ม ไม่คิดเครื่องหมาย
Unsigned long	32 บิต	0 ถึง 4294967296	เก็บข้อมูลชนิดจำนวนเต็มแบบยาว ไม่คิดเครื่องหมาย

### 2.2 ตัวแปรและค่าคงที่

1. **ตัวแปร (Variable)** เป็นการอ้างถึงข้อมูลโดยค่าของตัวแปรสามารถเปลี่ยนแปลงได้ตลอดเวลาที่โปรแกรมกำลังทำงานอยู่ ก่อนการใช้งานตัวแปรจะต้องถูกประกาศการใช้งานและต้องระบุชนิดของข้อมูลที่จะใช้กับตัวแปรนั้นๆ ไว้

- รูปแบบการประกาศตัวแปร

ชนิดข้อมูล ชื่อตัวแปร ;

### ตัวอย่างการประกาศตัวแปร

`int number ;`

หมายถึง ตัวแปรชื่อ number เป็นชนิดข้อมูลแบบจำนวนเต็ม (int)

`float width, high ;`

หมายถึง ตัวแปรชื่อ width และ high เป็นชนิดข้อมูลแบบทศนิยม(float)

นอกจากนี้แล้วยังสามารถประกาศตัวแปรแบบมีการกำหนดค่าเริ่มต้นให้กับตัวแปรได้อีกด้วย ดังตัวอย่าง

- รูปแบบการประกาศตัวแปรที่มีการกำหนดค่าเริ่มต้น

ชนิดข้อมูล ชื่อตัวแปร=ค่าเริ่มต้น ;
-------------------------------------

### ตัวอย่างการประกาศตัวแปร

`int number=100 ;` หมายถึง ตัวแปรชื่อ number เป็นชนิดข้อมูลแบบจำนวนเต็ม (int) ที่มีการกำหนดค่าเริ่มต้นเท่ากับ 100

`float answer=0;` หมายถึง ตัวแปรชื่อ answer เป็นชนิดข้อมูลแบบทศนิยม (float) ที่มีการกำหนดค่าเริ่มต้นเท่ากับ 0

- กฎเกณฑ์การตั้งชื่อตัวแปร มีดังนี้

- 1) ตัวอักษรตัวแรกของชื่อต้องเป็นตัวอักษรภาษาอังกฤษ หรือตัวขีดเส้นใต้ ( \_ )
- 2) ต้องประกอบไปด้วยตัวอักษรภาษาอังกฤษ (A-Z, a-z) ตัวเลข (0-9) หรือ เครื่องหมายขีดเส้นใต้ ( \_ ) เท่านั้น
- 3) ต้องไม่ซ้ำกับคำสงวน (Reserved word) 32 คำ โดยมีคำต่างๆ ดังตารางต่อไปนี้

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	far	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				

2. ค่าคงที่ (Constants) ค่าคงที่เป็นการอ้างอิงถึงข้อมูลเช่นเดียวกับตัวแปร สิ่งที่แตกต่างกันคือ ค่าของมันไม่สามารถเปลี่ยนแปลงได้อีกหลังจากการประกาศ ในภาษา C ค่าคงที่จะต้องถูกประกาศโดยไม่ต้องระบุชนิดของข้อมูล แต่ต้องระบุคีย์เวิร์ด #define ในการประกาศค่าคงที่ก่อนถูกนำมาใช้งานเสมอ

- รูปแบบการประกาศค่าคงที่

**#define ชื่อค่าคงที่ ค่าที่ต้องการเก็บ**

ตัวอย่างการประกาศค่าคงที่

```
#define pi 3.14
```

หมายถึง ค่าคงที่ชื่อ pi ที่มีค่าเท่ากับ 3.14

```
#define x 3
```

หมายถึง ค่าคงที่ชื่อ x ที่มีค่าเท่ากับ 3

### 2.2.3 ตัวดำเนินการ

ในการเขียนโปรแกรมเราจะพบว่าตัวดำเนินการเป็นส่วนหนึ่งที่มีความสำคัญต่อการเขียนคำสั่งต่างๆ ให้โปรแกรมสามารถทำงานได้ตามที่เราต้องการ เช่น การคำนวณทางคณิตศาสตร์ หรือการหาผลลัพธ์ต่างๆ โดยปกติแล้วในคำสั่งที่มีตัวดำเนินการนั้นจะเห็นว่าประกอบไปด้วยข้อมูล 2 ส่วน และจะมีตัวดำเนินการคั่นอยู่ตรงกลาง ซึ่งข้อมูลที่นำมาใช้กับตัวดำเนินการเราเรียกว่า Operand

ในการแบ่งประเภทของตัวดำเนินการนั้นมีการแบ่งอยู่หลายแบบ แต่ในที่นี้จะขอแบ่งประเภทของตัวดำเนินการตามรูปแบบการใช้งาน และขออธิบายเฉพาะตัวดำเนินการที่ใช้งานอยู่เป็นประจำ ดังนี้

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)
- ตัวดำเนินการเปรียบเทียบ (Comparison Operator)
- ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)
- ตัวดำเนินการในการกำหนดค่า (Assignment Operator)

## 1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

ตัวดำเนินการชนิดนี้จะพบบ่อยที่สุด และยังถูกนำไปใช้งานบ่อยอีกด้วย ซึ่งจะถูกนำไปใช้ในการคำนวณที่เกิดจากข้อมูลทั้ง 2 ฝั่งของตัวดำเนินการ ซึ่งมีให้เลือกใช้งาน ดังต่อไปนี้

ตารางที่ 2.2.3 ตารางแสดงตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการใช้งาน
+	บวก	$c = a + b$ ; c คือผลลัพธ์จาก a บวก b
-	ลบ	$c = a - b$ ; c คือผลลัพธ์จาก a ลบ b
*	คูณ	$c = a * b$ ; c คือผลลัพธ์จาก a คูณ b
/	หาร	$c = a / b$ ; c คือผลลัพธ์จาก a หาร b
%	หาเศษจากการหาร	$c = a \% b$ ; c คือเศษที่เกิดจาก a หารด้วย b

ถ้ามีการใช้ตัวดำเนินการมากกว่าหนึ่งตัว ภาษา C# จะคำนวณค่าโดยเรียงลำดับความสำคัญของเครื่องหมายดังนี้

- ( )
- \* , / และ %
- + และ -
- หากตัวดำเนินการมีลำดับเท่าเทียมกันจะคำนวณจากซ้ายไปขวา

## 2. ตัวดำเนินการเปรียบเทียบ (Comparison Operator)

ตัวดำเนินการเปรียบเทียบใช้ในการเปรียบเทียบข้อมูลสองข้างของตัวดำเนินการ ซึ่งผลที่ได้จากการเปรียบเทียบจะได้ผลลัพธ์ที่เป็นจริง (True) หรือ เท็จ (False) เท่านั้น ตัวดำเนินการเปรียบเทียบมีดังนี้

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการใช้งาน
==	เท่ากับ	$a == b$ คำตอบเป็นจริง เมื่อ a เท่ากับ b คำตอบเป็นเท็จ เมื่อ a ไม่เท่ากับ b
!=	ไม่เท่ากับ	$a != b$ คำตอบเป็นจริง เมื่อ a ไม่เท่ากับ b คำตอบเป็นเท็จ เมื่อ a เท่ากับ b
<	น้อยกว่า	$a < b$ คำตอบเป็นจริง เมื่อ a น้อยกว่า b คำตอบเป็นเท็จ เมื่อ a มากกว่าหรือเท่ากับ b
>	มากกว่า	$a > b$ คำตอบเป็นจริง เมื่อ a มากกว่า b คำตอบเป็นเท็จ เมื่อ a น้อยกว่าหรือเท่ากับ b

<=	น้อยกว่าหรือเท่ากับ	$a \leq b$ คำตอบเป็นจริง เมื่อ a น้อยกว่าหรือเท่ากับ b คำตอบเป็นเท็จ เมื่อ a มากกว่า b
>=	มากกว่าหรือเท่ากับ	$a \geq b$ คำตอบเป็นจริง เมื่อ a มากกว่าหรือเท่ากับ b คำตอบเป็นเท็จ เมื่อ a น้อยกว่า b

### 3. ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)

ตัวดำเนินการทางตรรกศาสตร์ใช้เปรียบเทียบข้อมูลด้านตรรกศาสตร์ซึ่งผลลัพธ์ที่ได้จะเป็นจริง (True) หรือเท็จ (False) เท่านั้น มีดังนี้

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการใช้งาน
!	นิเสธ(NOT)	$!p$ คำตอบเป็นจริง เมื่อค่าของ p เป็นเท็จ คำตอบเป็นเท็จ เมื่อค่าของ p เป็นจริง
&&	และ(AND)	$p \&\& q$ คำตอบเป็นจริง เมื่อ p และ q เป็นจริงทั้งคู่
	หรือ(OR)	$p \ \  q$ คำตอบเป็นจริง เมื่อ p หรือ q อย่างใดอย่างหนึ่งเป็นจริง
^	Exclusive OR	$p \wedge q$ คำตอบเป็นจริง เมื่อ p และ q มีค่าที่ต่างกัน

### 4. ตัวดำเนินการในการกำหนดค่า (Assignment Operator)

ตัวดำเนินการกำหนดค่าสามารถใช้กำหนดค่าให้กับตัวแปร ดังตารางต่อไปนี้

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการใช้งาน
=	กำหนดค่า	$a=b$ ; หมายถึง นำค่า b ให้กับ a
+=	เพิ่มค่าแล้วกำหนดค่า	$a+=b$ ; หมายถึง $a = a + b$ ;
-=	ลดค่าแล้วกำหนดค่า	$a-=b$ ; หมายถึง $a = a - b$ ;
*=	คูณแล้วกำหนดค่า	$a*=b$ ; หมายถึง $a = a * b$ ;
/=	หารแล้วกำหนดค่า	$a/=b$ ; หมายถึง $a = a / b$ ;
%=	หาเศษจากการหารแล้วกำหนดค่า	$a\%=b$ ; หมายถึง $a = a \% b$ ;

ตัวดำเนินการอื่นๆ

- + สำหรับ string คือ การนำเอา string 2 ค่ามาต่อกัน (concatenate)
- ++ เป็นการเพิ่มค่า โดยที่  $a++$  เทียบเท่ากับ  $a=a+1$
- -- เป็นการลดค่า โดยที่  $a--$  เทียบเท่ากับ  $a=a-1$