

Operator

ຕັວດໍາເນື້ອກາຮ

Operator

ຕົວດຳເນີ້ນກາຣ

+ - * / %

Arithmetict operator

= += -=

*= /= %=

Assignment operator

== > < != >= <=

Comparison operator

&& || !

Logical operator

++ --

+ -

Unary operator

& | ^ >> << ~

Bitwise operator

Arithmetic operator

+

การบวก

-

การลบ

*

การคูณ

/

การหาร

%

การมือด

Arithmetict operator

+ การบวก	$9.0 + 5 = 14.000000$
- การลบ	$9 - 5.0 = 4.000000$
* การคูณ	$9.0 * 5.0 = 45.000000$
/ การหาร	$10.0 / 4 = 2.500000$



การทำงานของ + , - , * และ / ที่มีเลขทศนิยมอยู่ด้วย

จะได้ผลลัพธ์เป็นเลขทศนิยมเสมอ

Arithmetic operator

+ การบวก	$9 + 5 = 14$
- การลบ	$9 - 5 = 4$
* การคูณ	$9 * 5 = 45$
/ การหาร	$10 / 4 = 2$

why ???



การทำรายการ + , - , * แบบ / ที่มีเฉพาะผลหารจำนวนเต็ม จะได้

ผลลัพธ์เป็นจำนวนเต็ม

Arithmetic operator

การหารเหลือเศษ = %

% (mod)

$$13 / 5 = 2 \text{ เศษ } 3$$

$$13 \% 5 = 3$$

$$6 \% 2 = 0$$

$$10 \% 3 = 1$$

$$2 \% 5 = 2$$

$$-7 \% 3 = -1$$

การคำนวณ % ใช้ได้กับจำนวนเต็มเท่านั้น บวกลบ

จะได้ผลลัพธ์เป็นบวกเท่านั้น



Arithmetic operator

'a' + 3 = ??

Arithmetic Operator

$$'a' + 3 = ??$$

ការប្រើប្រាស់ + , - , * និង / កំណត់លទ្ធផល ព័ត៌មាននេះ

គឺជាការប្រើប្រាស់តែអក្រឡើងតែ ASCII Code

ការប្រើប្រាស់ប្រព័ន្ធលើក្រុមហ៊ុន

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	-	127	7F	[DEL]

Arithmetic operator

ເບີໂຈໜົມກົດໝາຍ

1. $(4 + 5) \times 3 = ?$

2. $30 \div (4 * 3) = ?$

3. $(-21 / 3) \times 4 = ?$

4. $'g' - 'a' + 'A' = ?$

5. $'S' - 'A' + 'a' = '?'$

6. $-14 \div 3 = ?$

7. $14 \div -3 = ?$

Assignment operator

=

ເທົກໝານ

+ =

ບວກເທົກໝານ

- =

ອບເທົກໝານ

* =

ຄູນເທົກໝານ

/ =

ຊາຮເທົກໝານ

% =

ມືອດເທົກໝານ

Assignment Operator

=

ເປົ້າການກຳຈະດົກຄ່າໃຫ້ຕົວແປຣດ້າໆຫ້າຍ

`x = 5`

x ມີຄ່າເທົ່າກັບ 5

`y = 'a'`

y ມີຄ່າເທົ່າກັບ 97 (ASCII Code)

Assignment Operator

`+=` บวกเพิ่มค่า

`X += Y` คือ `X = X + Y`

Assignment Operator

កំារអតិថិជ្ជ

$X = 12, Y = 3$

$+ =$ បន្ទប់ភាគខ្ពស់	$X += Y$ គឺជា $X = X + Y$	$X = 12 + 3 = 15$
$- =$ បន្ទប់ភាគខ្ពស់	$X -= Y$ គឺជា $X = X - Y$	$X = 12 - 3 = 9$
$* =$ គ្លួយបង្ហាញភាគខ្ពស់	$X *= Y$ គឺជា $X = X * Y$	$X = 12 * 3 = 36$
$/ =$ ចារបង្ហាញភាគខ្ពស់	$X /= Y$ គឺជា $X = X / Y$	$X = 12 / 3 = 4$
$\% =$ ផូតបង្ហាញភាគខ្ពស់	$X \% Y$ គឺជា $X = X \% Y$	$X = 12 \% 3 = 0$

Arithmetic Operator

ໂຄດຂ້າງຂ່າຍທີ່ຈະໃຊ້ຜົນລົບພຽບໄຮອກມາຫາສ console

```
#include<stdio.h>
int main() {
    int x = 15;
    x -= 7;
    x %= 3;
    x *= 4;
    printf("%d", x);
    return 0;
}
```

Unary operator

++

ເພີ່ມຄໍາ 1

+

ເຄື່ອງຂໍາຍບວກ

--

ຈອດຄໍາ 1

-

ເຄື່ອງຂໍາຍລົບ

Unary operator

កំឡុងតិច

$X = 2$

$X++, ++X$

$X = X + 1 = 2 + 1 = 3$

$X--, --X$

$X = X - 1 = 2 - 1 = 1$

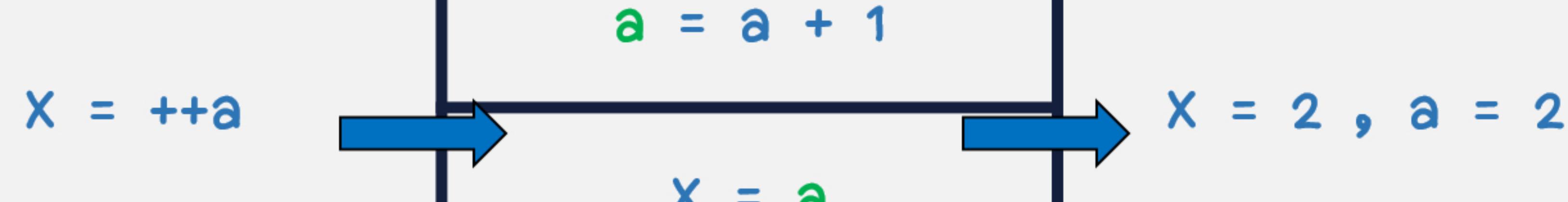
Unary Operator

ກຳລັງດິຈິ້ວ

a = 1

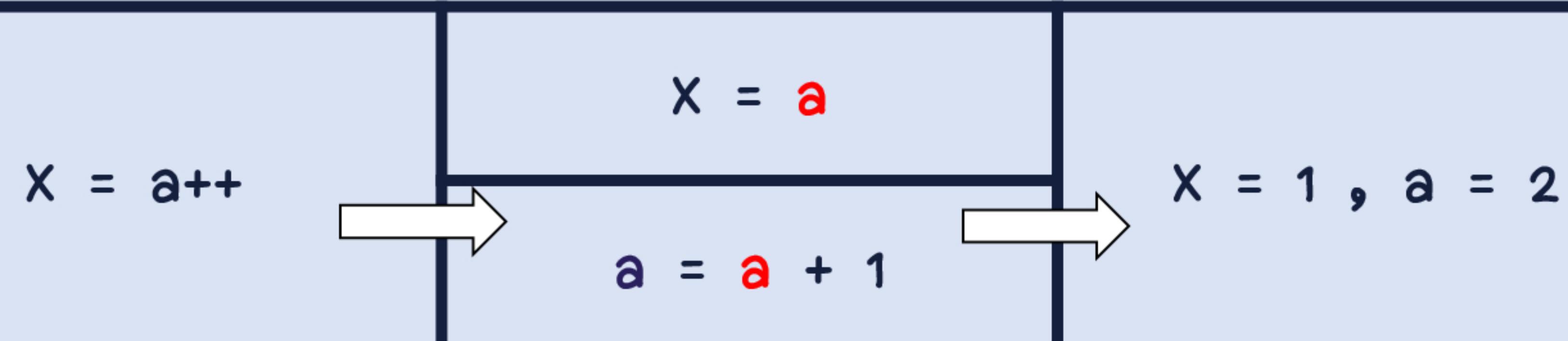
Prefix ແພື່ມ/ອດ ຄ່າໃຊ້ຕົວແປຣກ່ອນ ແລ້ວສິ່ງທີ່ໄປໃຫ້

(returns ຄ່າຂອງຕົວແປຣ ຂລື່ງຈາກ ຕົວແປຣຖຸກເພື່ມ/ອດຄ່າແລ້ວ)



Postfix ທີ່ມີຄ່າຢືນຢັນຂອງຕົວແປຣໄປໃຫ້ຈໍາກ່ອນ ແລ້ວສິ່ງເພື່ມ/ອດຄ່າຕົວແປຣ

(returns ຄ່າຂອງຕົວແປຣ ກ່ອນ ຕົວແປຣຈະຖຸກເພື່ມ/ອດຄ່າ)



Unary operator

ໂຄດຂ້າງຂ່າຍທີ່ຈະໃຫ້ຜົນລົບພຽງໄຮອກມາຖານ console

```
#include <stdio.h>
int main() {
    int x = 3;
    printf("%d ", ++x);
    printf("%d ", x++);
    printf("%d ", -x);
    return 0;
}
```

Comparison operator

`==`

ເທົກກົບ

`>`

ມາກກວ່າ

`<`

ຊ້ອຍກວ່າ

`!=`

ໄສ່ເທົກກົບ

`>=`

ມາກກວ່າ

`<=`

ຊ້ອຍກວ່າ

ເທົກກົບ

ເທົກກົບ

Comparison Operator

ເບີຣີຍບະທີຍບໍ່ອໝູລດ້ານຫຍໍາຍກົບດ້ານຂວາ

`X == Y`

`X != Y`

ຄ້າຜລລົບຮົບເປົ້າມີຄ່າເປົ້າ 1

`X > Y`

ຄ້າຜລລົບຮົບເປົ້າມີຄ່າເປົ້າ 0

`X >= Y`

`X < Y`

`X <= Y`

Comparison Operator

ໂຄດຂ້າງຂ່າຍທີ່ຈະໃຊ້ຜົນລົບພຽບອະໄຮອກມາຫາສ console

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 5;
    bool isGreater = 10 > 20;
    printf("%d ", a != a);
    printf("%d ", b <= b);
    printf("%d ", a < b);
    printf("%d ", isGreater);
    return 0;
}
```

Logical Operator

&&

ပြခဲ့

||

သိခဲ့

!

မျိုးဆက်

Logical operator

ဇនତକମ୍ବାଣଟର୍

& &

||

!

^

v

?

ବ୍ୟବସ୍ଥା

ଜୀବିତ

ନିଃଶ୍ଵର

Logical Operator

ໃຫ້ກຳນົດເລື່ອໜີ້ມາກວ່າ 1 ເລື່ອໜີ້

$(X == Y) \&\& (Y > Z)$

ຕີ່ແຜລລົ້ມຮົງຈະມີຄ່າບັງຍິ່ນ 1 (True) ຊີ້ອ 0 (False)

ເນັ້ນຊື່!!

Logical Operator

ໂຄດຂ້າງຂ່າຍທີ່ຈະໃຊ້ຜົນລົບພຽບໄຮອກມາຫາສ console

```
#include<stdio.h>
int main() {
    int x = 2;
    int y = 3;
    printf("%d", x == y || x < y );
    return 0;
}
```

Bitwise Operator

ตัวดำเนินการระดับบิต(Bit)

เป็นตัวดำเนินการเพื่อใช้ในการจัดการหรือ

คำนวณระดับบิต(Bit)

&

Bitwise AND

|

Bitwise OR

^

Bitwise Exclusive OR

หรือ XOR

>>

Bitwise shift

Right

<<

Bitwise

shift Left

~

Bitwise One's

Complement

Bitwise Operator

បិត (Bit) គឺជាឯំរូលមុនក្នុងកិច្ចការប្រព័ន្ធដែលបានបង្កើតឡើងដោយសារតម្លៃទីផ្សារ 2
ស្ថាតមុនគឺ 0 ឬ 1 ហើយកើតឡើងវា លេខត្រាងសោរ

1 Byte = 8 Bit

0101 0111

Bitwise Operator

ବିଟ୍ସି ଓପରେଟର୍ସ

&

|

~

^

∨

~

ବିଟ୍ସି

ଅର୍ଥ

ଫିଲ୍ସଟ

Bitwise operator

P	$\sim P$	Q	$\sim Q$	$P \wedge Q$	$P \vee Q$
T	F	T	F	T	T
T	F	F	T	F	T
F	T	T	F	F	T
*	F	T	F	T	F

$0 = \text{False}$

$1 = \text{True}$

Bitwise operator

P	$\sim P$	Q	$\sim Q$	P & Q	P Q
1	0	1	0	1	1
1	0	0	1	0	1
0	1	1	0	0	1
0	1	0	1	0	0



In C: int A = 10

10 : 0000 1010

Bitwise operator

X = 1010₂

Y = X & 1100₂

Y = ?

Bitwise operator

$X = 1010_2$

$Y = X \mid 1100_2$

$Y = ?$

Bitwise operator

0 = False

1 = True

&

$$0 \& 0 = 0$$

$$0 \& 1 = 0$$

$$1 \& 0 = 0$$

$$1 \& 1 = 1$$

$$0 | 1 = 1$$

$$1 | 0 = 1$$

$$1 | 1 = 1$$

$$0 | 0 = 0$$



Bitwise Operator

1100 1010 → 1100 0000

Bitwise Operator

1100 **1010** → 1100 **0000**

Bitwise Operator

1100 1010

Bitwise Operator

1100 1010

&

1111 0101

1100 0000

1100 1010

|

0000 0101

1100 0101

Bitwise Operator

0011

>> 1

0001

Bitwise shift Right

1010

<< 1

0100

Bitwise shift Left

Bitwise Operator

ໂຄດຂ້າງຂ່າຍໃຊ້ຜົນລົບພຽບໄຮອອກມາຖາວ Serial Monitor

```
void setup() {
    Serial.begin(9600);
}

void loop() {
    byte data1 = B11001111;
    data1 = data1 & B11111100;
    data1 = data1 >> 1;
    Serial.println(data1, BIN);
}
```

ลำดับความสำคัญของตัวดำเนินการ (Precedence)

OPERATOR	TYPE	ASSOCIABILITY
() [] . ->		left-to-right
++ -- + - ! ~ (type) * & sizeof	Unary Operator	right-to-left
* / %	Arithmetic Operator	left-to-right
+ -	Arithmetic Operator	left-to-right
<< >>	Shift Operator	left-to-right
< <= > >=	Relational Operator	left-to-right
== !=	Relational Operator	left-to-right
&	Bitwise AND Operator	left-to-right
^	Bitwise EX-OR Operator	left-to-right
	Bitwise OR Operator	left-to-right
&&	Logical AND Operator	left-to-right
	Logical OR Operator	left-to-right
? :	Ternary Conditional Operator	right-to-left
= += -= *= /= %= &= ^= = <<= >>=	Assignment Operator	right-to-left
,	Comma	left-to-right