ตัวดำเนินการ (Operator)

ตัวดำเนินการ (Operator)

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)
- ตัวดำเนินการเปรียบเทียบ (Comparison Operators)
- ตัวดำเนินการมอบหมายค่า (Assignment Operators)
- ตัวดำเนินการทางตรรกะ (Logical Operators)
- ตัวดำเนินการสำหรับข้อมูลชนิด BitWise
- ตัวดำเนินการเกี่ยวกับสมาชิก (Membership Operators)
- ตัวดำเนินการเกี่ยวกับบิตเล่น (Identity Operators)
- ตัวดำเนินการกับข้อความ (String Operators)
- ตัวดำเนินการกับรายการ (List Operators)

ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`+`	`2 + 3 = 5`	บวก
`-`	`5 - 2 = 3`	ลบ
`*`	`3 * 4 = 12`	คูณ
`/`	`10 / 2 = 5.0`	หาร
`//`	`11 // 3 = 3`	หารเอาส่วนลง
`%`	`11 % 3 = 2`	หารเอาเศษ
`**`	`2 ** 3 = 8`	ยกกำลัง

ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`==`	`5 == 5` เป็นจริง	เท่ากับ
`!=`	`5 != 3` เป็นจริง	ไม่เท่ากับ
`>`	`7 > 5` เป็นจริง	มากกว่า
`<`	`3 < 7` เป็นจริง	น้อยกว่า
`>=`	`7 >= 7` เป็นจริง	มากกว่าหรือเท่ากับ
`<=`	`3 <= 5` เป็นจริง	น้อยกว่าหรือเท่ากับ

ตัวดำเนินการทางตรรกะ

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`and`	`(5 > 3) and (3 > 1)` เป็นจริง	และ
`or`	`(5 < 3) or (3 < 1)` เป็นจริง	หรือ
`not`	`not(5 > 3)` เป็นเท็จ	ปฏิเสธ

ตัวดำเนินการสำหรับข้อมูลชนิด BitWise

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`&`	`0b1010 & 0b1100 = 0b1000`	AND บิต
`	`	`	`0b1010
`^`	`0b1010 ^ 0b1100 = 0b0110`	XOR บิต
`~`	`~0b1010 = -0b1011`	NOT บิต
`<<`	`0b1010 << 2 = 0b101000`	เลื่อนบิตไปทางซ้าย
`>>`	`0b1010 >> 2 = 0b0010`	เลื่อนบิตไปทางขวา

ตัวดำเนินการกับข้อความ(String)

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`+`	`'Hello' + 'World' = 'HelloWorld'`	รวมข้อความ
`*`	`'Hi' * 3 = 'HiHiHi'`	ทวนข้อความ

ตัวดำเนินการกับรายการ(List)

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
+	[1, 2] + [3, 4] = [1, 2, 3, 4]	รวมรายการ
*	[1, 2] * 3 = [1, 2, 1, 2, 1, 2]	ทวนรายการ

ตัวดำเนินการมอบหมายค่า

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`=`	`x = 5`	กำหนดค่า
`+=`	`x = 2; x += 3` ได้ `x = 5`	บวกแล้วกำหนดค่า
`-=`	`x = 5; x -= 3` ได้ `x = 2`	ลบแล้วกำหนดค่า
`*=`	`x = 2; x *= 3` ได้ `x = 6`	คูณแล้วกำหนดค่า
`/=`	`x = 6; x /= 3` ได้ `x = 2.0`	หารแล้วกำหนดค่า
`%=`	`x = 7; x %= 3` ได้ `x = 1`	หารเอาเศษแล้วกำหนดค่า
`**=`	`x = 2; x **= 3` ได้ `x = 8`	ยกกำลังแล้วกำหนดค่า
`//=`	`x = 7; x //= 3` ได้ `x = 2`	หารเอาส่วนลงแล้วกำหนดค่า

ตัวดำเนินการเกี่ยวกับสมาชิก

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
`in`	`'H' in 'Hello'` เป็นจริง	ตรวจสอบหาสมาชิกในวัตถุ
`not in`	`'x' not in 'Hello'` เป็นจริง	ตรวจสอบไม่พบสมาชิกในวัตถุ

ลำดับความสำคัญของ ตัวดำเนินการ

ลำดับความสำคัญ	ตัวดำเนินการ	คำอธิบาย
1	`**`	ยกกำลัง
2	`+x`, `-x`, `~x`	การกระทำครั้งเดียว (unary operation)
3	`*`, `/`, `//`, `%`	คูณ, หาร, หารเอาส่วนลง, หารเอาเศษ
4	`+`, `-`	บวก, ลบ
5	`<<`, `>>`	เลื่อนบิตไปทางซ้าย, เลื่อนบิตไปทางขวา
6	`&`	AND บิต
7	`^`	XOR บิต
8	`	`
9	`<`, `<=`, `>`, `>=`, `!=`, `==`	เปรียบเทียบ
10	`not x`	ปฏิเสธ
11	`and`	และ
12	`or`	หรือ

หมายเหตุ

- ตัวดำเนินการที่มีลำดับความสำคัญเดียวกันจะถูกประเมินจากซ้ายไปขวา
- การใช้วงเล็บ () สามารถเปลี่ยนลำดับความสำคัญของตัวดำเนินการได้ เช่น (2 + 3) * 4 จะคำนวณเป็น 20

นิพจน์ (expression)

ในภาษาโปรแกรม หมายถึง กลุ่มของค่าคงที่ (Literals) ตัวแปร (Variables) และตัวดำเนินการ (Operators) ที่ถูกนำมารวมกันเพื่อคำนวณหรือประมวลผลค่าใหม่ออกมา

องค์ประกอบของนิพจน์

- **ค่าคงที่** (Literals) เช่น ตัวเลข 42 สตริง "Hello" ค่าบูลีน True/False
- **ตัวแปร** (Variables) เช่น x, name, is_valid
- **ตัวดำเนินการ** (Operators) เช่น +, -, *, /, **, and, or
- **ฟังก์ชัน** (Functions) เช่น len(), sum(), print()
- **นิพจน์ย่อย** (Subexpressions) เช่น (x + y) * 2, not (x or y)

ตัวอย่างนิพจน์

- 2 + 3 ➔ เป็นนิพจน์ของค่าคงที่และตัวดำเนินการบวก ผลลัพธ์คือ 5
- x * y ➔ เป็นนิพจน์ของตัวแปรและตัวดำเนินการคูณ
- len("Hello") + 42 ➔ เป็นนิพจน์ที่ประกอบด้วยฟังก์ชัน ค่าคงที่ และตัวดำเนินการบวก
- (x + y) * z ➔ เป็นนิพจน์ที่มีนิพจน์ย่อย (x + y) และใช้วงเล็บเพื่อควบคุมลำดับการคำนวณ

แบบฝึกหัด

1. จงเขียนโปรแกรมคำนวณค่าของนิพจน์ต่อไปนี้ c * d – x / y

กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

c = 16

d = 10

x = 5

y = 2

แบบฝึกหัด

2.จงเขียนโปรแกรมคำนวณค่าของนิพจน์ต่อไปนี้

–(-5)* (x % y - x * (y+1)) กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

c = 16

d = 10

x = 5

y = 2

แบบฝึกหัด

3.จงเขียนโปรแกรมคำนวณค่าของนิพจน์ต่อไปนี้

c // d + y % x != 2*c - d กำหนดให้ตัวแปรแต่ละตัวมีค่าดังนี้

c = 16

d = 10

x = 5

y = 2