กิจกรรมตัวดำเนินการและฟังก์ชันทางคณิตศาสตร์

1. ตัวดำเนินการ

นิพจน์ (Expression) คือ การดำเนินการที่ประกอบด้วยตัวดำเนินการ (Operator) และตัวถูกดำเนินการ (Operand)

ตัวดำเนินการ (Operator) คือ สัญลักษณ์ที่ใช้ในการแทนการกระทำอย่างใดอย่างหนึ่งกับข้อมูล เช่น +, -, *, **, /, //, %, =, >, <, !=, ==, <> เป็นต้น

ตัวถูกดำเนินการ (Operand) คือ ข้อมูลที่ถูกกระทำโดยตัวดำเนินการ ซึ่งตัวถูกดำเนินการอาจอยู่ในรูปของตัวแปร (Variable) ค่าคงที่ (Constant) ค่าที่ได้รับจากฟังก์ชัน (Return function) หรือแม้กระทั่งนิพจน์ (Expression) เอง ก็เป็นตัวถูกดำเนินการได้ เช่น a+b, x++, a != b, x = (a + b) / 2 เป็นต้น

จากตัวอย่าง y = x2 + 2x + 1

y, x, 1 คือ ตัวถูกดำเนินการ (Operand)

=, +, ยกกำลังสอง คือ ตัวดำเนินการ (Operator)

y = x2 + 2x + 1 ทั้งหมด คือ นิพจน์ (Expression)

ภาษาไพทอนสนับสนุนตัวดำเนินการหลายแบบ คือ Arithmetic Operators, Comparison Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

เพื่อให้เข้าใจถึงการทำงานของตัวดำเนินการทางคณิตศาสตร์ สมมุติให้ ตัวแปร a = 10, b = 5, c = 9.0, d = 2.0, e = -3.5 สำหรับตัวอย่างการดำเนินการทางคณิตศาสตร์แสดงในตาราง

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
( )	จัดกลุ่ม	(2 + 9) * 5	55
**	ยกกำลัง	5 ** 2	25
*	คูณ	10 * 3	30
/	หารจำนวนจริง	25 / 5	5.0
//	หารปัดเศษทิ้ง	8 // 3	2

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
%	หารเอาเศษ	28 % 24	4
+	บวก	5 + 7	12
-	ลบ	10 – 2	8

ตัวดำเนินการเปรียบเทียบ (Comparison Operation)

นิพจน์เปรียบเทียบ (relational expression) เป็นนิพจน์ที่ใช้ตัวดำเนินการเปรียบเทียบ (relational operation) ในการเปรียบเทียบค่าของนิพจน์คณิตศาสตร์สองนิพจน์ ซึ่งในภาษาไพทอนจะได้ผลลัพธ์เป็นคลาส bool ซึ่งมีค่าเป็น True หรือ False เท่านั้น

ตัวดำเนินการทางเปรียบเทียบที่ใช้ในภาษาไพทอน โดยกำหนด x = 5 และ y = 7

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
==	เท่ากัน	x == y	False
!=	ไม่เท่ากัน	x != y	True
>=	มากกว่าหรือเท่ากับ	x >= y	False
<=	น้อยกว่าหรือเท่ากับ	x <= y	True
>	มากกว่า	x > y	False
<	น้อยกว่า	x < y	True

ตัวดำเนินการตรรกะ

นิพจน์ตรรกะ (logical expression) เป็นนิพจน์ที่ใช้ตัวดำเนินการตรรกะ (logical operation) ในการประมวลผลตัวถูกดำเนินการที่อยู่ในคลาส bool

ตัวดำเนินการตรรกะที่ใช้ในภาษาไพทอน มีดังนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	ผลลัพธ์
not	นิเสธ	not 5 > 6	True
and	และ	3 < 4 and 5 < 6	False
or	หรือ	3 < 4 or 5 < 6	True

ตัวดำเนินการ and และ or ต้องการตัวถูกดำเนินการสองตัว แต่ตัวดำเนินการ not ต้องการตัวถูกดำเนินการเพียงตัวเดียว
โดยตัวดำเนินการตรรกะมีลำดับความสำคัญเรียงจากลำดับความสำคัญสูงไปยังลำดับความสำคัญต่ำ คือ not and or

2. ฟังก์ชันทางคณิตศาสตร์

ฟังก์ชัน (Function) คือ โปรแกรมย่อยหรืองานย่อย ๆ (Sub-program) ภายในโปรแกรมขนาดใหญ่ หรือบางครั้งเรียกว่า เมท็อด (Method) หรือรูทีน (Routine) ก็ได้ โดยปกติโปรแกรมที่มีขนาดใหญ่ จะประกอบด้วยคำสั่งต่างๆ มากมายและในโปรแกรมขนาดใหญ่นั้นจะมีคำสั่งที่ทำงานเหมือนกัน ซ้ำกันหรือถูกเรียกใช้งานอยู่เป็นประจำปรากฏอยู่ ดังนั้นเพื่อลดคำสั่งให้โปรแกรมสั้นและกระชับลง จึงได้รวบรวมคำสั่งที่ทำหน้าที่เหมือนกันเหล่านั้นเข้าไว้ด้วยกันเป็นโปรแกรมย่อย เมื่อต้องการเรียกใช้โปรแกรมย่อยเหล่านั้นตรงจุดใดๆ ในโปรแกรม สามารถเรียกใช้โดยผ่านทางชื่อของฟังก์ชันได้ทันที (Function call) กรณีการเขียนโปรแกรมเชิงโครงสร้าง (Structural programming) นิยมเรียกว่า“ฟังก์ชัน” แต่ในการเขียนโปรแกรมเชิงวัตถุ (Object oriented programming) เรียกว่า “เมธอด” ประเภทของฟังก์ชันแบ่งออกเป็น 2 ชนิดคือ ฟังก์ชันมาตรฐาน (Standard functions) ที่มีอยู่ในไลบรารีของภาษาไพทอน และฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้นเอง (User defined functions) ฟังก์ชันที่มีอยู่แล้วในภาษาไพทอนเมื่อเรียกใช้งานจะต้องทำการ import แฟ้มเข้ามาก่อนเสมอ ในบทนี้จะกล่าวถึงการสร้างฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้นใช้งานเอง

เกร็ดความรู้ : ฟังก์ชัน Vs เมธอด แม้ว่าฟังก์ชันและเมธอดจะมีลักษณะการทำงานที่คล้ายกัน แต่ในการทำงานจริงนั้นจะมีความแตกต่างกันอยู่พอสมควร เนื่องจากแนวคิดโปรแกรมเชิงวัตถุมองว่า เมธอดคือพฤติกรรมต่างๆ ของวัตถุ (Object) ที่แสดงออก ซึ่งพฤติกรรมอาจจะไม่ใช่งานประจำเหมือนที่ฟังก์ชันทำก็ได้

ประโยชน์ของฟังก์ชัน

1. ช่วยลดคำสั่งที่ซ้ำซ้อนกันในโปรแกรม
2. ช่วยให้ผู้พัฒนาโปรแกรมสามารถปรับปรุงและแก้ไขโปรแกรมได้อย่างรวดเร็ว

เนื่องจากฟังก์ชันแต่ละฟังก์ชันนั้นมีหน้าที่ที่ชัดเจนในตัวเอง เช่น read_input คือฟังก์ชันสำหรับอ่านค่าอินพุตเข้ามาทำงานในโปรแกรม หรือ print ทำหน้าที่พิมพ์ข้อความ

3. ช่วยทำให้โปรแกรมมีความกะทัดรัด ทำให้เข้าใจง่ายและรวดเร็ว เพราะโปรแกรมถูกแบ่งออกตามหน้าที่ของงานชัดเจน

4. ฟังก์ชันสามารถนำกลับมาใช้ได้อีกหลายครั้ง (reusable code)

5. ป้องกันข้อผิดพลาดได้ดีเพราะงานจะถูกแบ่งตามหน้าที่ชัดเจน การเขียนโปรแกรมจะไม่ก้าวก่ายงานในฟังก์ชันอื่นๆ ที่ไม่เกี่ยวข้อง

6. ช่วยให้หาข้อผิดพลาดของโปรแกรมได้รวดเร็วและเป็นระบบ กรณีถ้าโปรแกรมเกิดข้อผิดพลาดเกิดขึ้นในขณะทำงาน การทดสอบจะทดสอบตามฟังก์ชัน

7. ฟังก์ชันมีการท างานเป็นอิสระ สามารถนำฟังก์ชันที่ถูกสร้างไว้และมีประสิทธิภาพเก็บไว้เป็นโมดูล คลาส หรือไลบรารี เพื่อนำไปใช้งานต่อได้ในอนาคตได

ไพทอนได้เตรียมฟังก์ชันทางคณิตศาสตร์ไว้ใช้สำหรับในการแก้ปัญหาหลายฟังก์ชัน โดยสามารถศึกษาและดูตัวอย่างการใช้งานได้จากลิงก์ต่อไปนี้

https://docs.python.org/3/library/math.html หรือ

https://www.w3schools.com/python/module_math.asp

โดยในการใช้งานฟังก์ชันทางคณิตศาสตร์ (หรือฟังก์ชันอื่น ๆ) จะต้องทำการเรียกใช้โมดูลนั้น ๆ ก่อน ตัวอย่างเช่น

หากต้องการหารากที่สองของตัวเลขใด ๆ

```
# Import math Library
import math

# Return the square root of different numbers
print (math.sqrt(9))
print (math.sqrt(25))
print (math.sqrt(16))
```

ค่า pi

```
# Import math Library
import math

# Print the value of pi
print (math.pi)
```

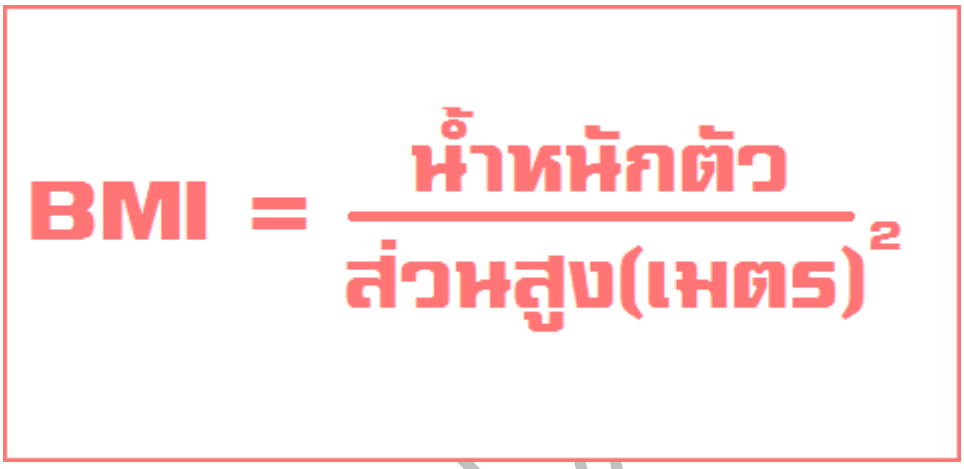
หา ห.ร.ม.

```
#Import math Library
import math

#find the  the greatest common divisor of the two integers
print (math.gcd(3, 6))
print (math.gcd(6, 12))
```

```
print (math.gcd(12, 36))
print (math.gcd(-12, -36))
print (math.gcd(5, 12))
print (math.gcd(10, 0))
print (math.gcd(0, 34))
print (math.gcd(0, 0))
```

ตัวอย่างโปรแกรมคำนวณหาค่า BMI




```
#Import math Library
import math
#Input
w = float(input("Input weight(kg.) : "))
h = float(input("Input height(cm.) : "))
#process
h_in_m = h/100
bmi = w/math.pow(h_in_m,2)
#output
print("BMI = %.2f"%bmi)
```

โจทย์ท้าประลอง!!!

1. รับเลขจำนวนเต็ม 2 จำนวน จากนั้น หา ห.ร.ม. และ ค.ร.น. ของเลข 2 จำนวนนั้น
2. จงเขียนโปรแกรมเพื่อรับค่าเลขจำนวนเต็ม 3 จำนวน

พร้อมคำนวณหาผลลัพธ์เพื่อแสดงผลดังต่อไปนี้

```
Input Number1 ==> 1
Input Number2 ==> 2
Input Number3 ==> 3
1 + 2 + 3 = 6
sqrt(6) = 2.45
pow(6,2) = 36
```

3. รับด้านทั้ง 3 ของรูปสามเหลี่ยม จากนั้นคำนวณหาพื้นที่สามเหลี่ยม โดยใช้สูตร

$$Area = \sqrt{s(s-a)(s-b)(s-c)}$$

โดยที่ a, b และ c คือ ความยาวของด้านทั้ง 3 ของสามเหลี่ยมและค่า s มีค่าดังสูตร

$$s = {(a+b+c)}/{2}$$

4. รถยนต์คันหนึ่งเคลื่อนที่ด้วยความเร็วต้น u m/s เมื่อเวลาผ่านไป t วินาที รถยนต์มีความเร็ว v m/s ให้นักเรียนเขียนโปรแกรมคำนวณหาค่าระยะทางที่รถยนต์เคลื่อนที่ได้ เมื่อทราบค่า u, t และ v

5. ให้นักเรียนเขียนโปรแกรมหาความยาวของด้านตรงข้ามมุมฉาก เมื่อทราบความยาวด้านประกอบมุมฉากคือ a และ b

ณัฐพล บัวอุไร