01 : Data Type, Variable and Expression

สรุปเนื้อหา

ตัวแปรเป็นที่เก็บข้อมูลในโปรแกรม ต้องมีชื่อกำกับ

- ชื่อตัวแปรประกอบด้วยตัวอักษร ตัวเลข หรือเครื่องหมายขีดเส้นใต้ _ ตัวอังกฤษใหญ่ไม่เหมือนตัวเล็ก ห้ามขึ้นต้นชื่อด้วยตัวเลข
- อย่าตั้งชื่อตัวแปรซ้ำกับชื่อฟังก์ชันใน Python เช่น `int, str, max, sum, abs, ...` (ไม่ห้ามถ้าจะตั้งซ้ำ แต่ไม่ควรทำอย่างยิ่ง)

ข้อมูลใน Python ที่นำมาประมวลผลมีหลายประเภท ที่เราจะศึกษาในบทนี้มีดังต่อไปนี้

`int`	จำนวนเต็ม	`-10` `5000011` *(Python ห้ามไม่ให้เขียน `0` นำหน้าจำนวนเต็ม เช่น `020`)*
`float`	จำนวนจริง	`10.0` `1.23e59` *มีค่าเท่ากับ 1.23×10^{59}*
`str`	ข้อความ	`'Programming is easy'` `"Let's go shopping"`

การให้ค่ากับตัวแปร

- `a = b = c = 0.0` *ให้ตัวแปร `a` `b` และ `c` เก็บจำนวนจริง `0.0`*
- `a = 5; b = 6; a,b = b,a` *ตัวแปร `a` กับ ตัวแปร `b` สลับค่ากัน ได้ `a` เก็บ `6` และ `b` เก็บ `5`*
- `a = x` *ถ้า `x` ไม่เคยมีการให้ค่ามาก่อน คำสั่งนี้จะผิด เพราะไม่รู้ว่า `x` มีค่าเท่าใด*

ตัวดำเนินการ ลำดับการทำงาน และการแปลงประเภทข้อมูล

- ตัวดำเนินการ บวก (`+`), ลบ (`-`), คูณ (`*`), ยกกำลัง (`**`), หาร (`/`), หารปัดเศษ (`//`), เศษจากการหาร (`%`)
- การดำเนินการระหว่างจำนวนเต็มกับจำนวนจริงจะได้ผลเป็นจำนวนจริง (เช่น `2 + 1.0` ได้ `3.0`)
- `//` กับจำนวนลบ : `1//2` ได้ `0`, `(-1)//2` ได้ `-1`, `11//10` ได้ `1`, `(-11)//10` เหมือน `11//-10` ได้ `-2`
- `a = 3+97//2**3%8` a มีค่า `3+97//8%8 = 3+12%8 = 3+4 = 7`
- `a = 12//3/2+2**3**2` a มีค่า `12//3/2+2**9 = 12//3/2+512 = 4/2+512 = 2.0+512 = 514.0`
- `a += 2` ก็คือ `a = a + 2`, `a //= 2` ก็คือ `a = a // 2`, `a *= -1` ก็คือ `a = a * -1`
- ถ้า `import math` จะมีค่าคงตัวและฟังก์ชันทางคณิตศาสตร์ให้ใช้มากมาย
  - `math.pi, math.e, math.sin(x), math.cos(x), math.sqrt(x), math.log(x,b), ...`
- นำสตริงบวกกัน คือนำสตริงมาต่อกัน เช่น `'12'+'23'` คือ `'1223'`
- สตริงคูณกับจำนวนเต็ม คือนำสตริงนั้นมาต่อกันเป็นจำนวนครั้งเท่ากับค่าของจำนวนเต็มนั้น เช่น `'12'*3` คือ `'121212'`
- ฟังก์ชัน `int`, `float` และ `str` มีไว้เปลี่ยนประเภทข้อมูล เช่น
  - `int('12')` ได้ `12`, `float('1.2')` ได้ `1.2`, `str(12//2)` ได้ `'6'`, `str('Aa')` ได้ `'Aa'`
- **ข้อควรระวัง** : รู้ความแตกต่างของ `/` กับ `//` และศึกษาลำดับการทำงานของ operator ให้ดี ถ้าไม่มั่นใจ ใส่วงเล็บ เช่น `a/2*b` เท่ากับ `(a/2)*b` แต่ `a/(2*b)` เท่ากับ `a/2/b`

คำสั่งการแสดงข้อมูลทางจอภาพ

- `print(a,b,c)` *นำค่าในตัวแปร* a b *และ* c *มาแสดงต่อกันคั่นด้วยช่องว่างบนบรรทัดเดียวกัน*
- `print(str(a)+str(b)+str(c))` *นำค่าในตัวแปร* a b *และ* c *มาเปลี่ยนเป็นสตริงต่อกัน แล้วแสดงบนบรรทัดเดียวกัน*

คำสั่งการอ่านข้อมูลจากแป้นพิมพ์

- `a = input()` *อ่านข้อความจากแป้นพิมพ์หนึ่งบรรทัด เก็บใส่ตัวแปร* a *(เป็นสตริง)*
- `a = input().strip()` *อ่านข้อความจากแป้นพิมพ์หนึ่งบรรทัด ตัดช่องว่างทางซ้ายและขวาออก เก็บใส่ตัวแปร* a
- `a = int(input())` *อ่านจำนวนเต็มหนึ่งจำนวนจากแป้นพิมพ์ เก็บใส่ตัวแปร* a
- `a = float(input())` *อ่านจำนวนจริงหนึ่งจำนวนจากแป้นพิมพ์ เก็บใส่ตัวแปร* a
- ถ้าต้องการอ่านข้อมูลหลาย ๆ ตัวที่ผู้ใช้ป้อนเข้ามาในบรรทัดเดียวกัน โดยข้อมูลแต่ละตัวคั่นด้วยช่องว่าง
  - `a,b,c = [e for e in input().split()]` *หรือ*
    `a,b,c = input().split()` *อ่านสตริง 3 ตัว*
  - `x,y = [int(e) for e in input().split()]` *อ่านจำนวนเต็ม 2 จำนวน*
  - `a,b,c = [float(e) for e in input().split()]` *อ่านจำนวนจริง 3 จำนวน*
  - หากจะอ่านจำนวนจริงตามด้วยจำนวนเต็ม ก็อ่านเป็นสตริงก่อน โดยใช้คำสั่ง `f,n = input().split()` แล้วจึงค่อยแปลงเป็นจำนวนจริงกับจำนวนเต็ม โดยใช้คำสั่ง `f = float(f); n = int(n)`

*** ถ้าโจทย์บอกว่าข้อมูลที่รับมาคั่นด้วยช่องว่างในบรรทัดเดียวกัน อย่าใช้ `input().split(' ')` แต่ควรใช้ `input().split()` แทน

เรื่องผิดบ่อย

รับข้อมูลจากแป้นพิมพ์แล้วลืมแปลงเป็นจำนวน ก่อนนำไปคำนวณ	`x = input()` `y = x**2 + 7` *ผิดเพราะ* x *เป็นสตริง*

จำลำดับการทำงานของตัวดำเนินการ
`+ - * / // % **` ผิด
(`**` ทำก่อน `* / // %` ทำก่อน `+ -`)
	`y = x / 2*a` *จะได้* `y = (x/2)*a`
	ถ้าต้องการคำนวณ $y = \frac{x}{2a}$
	ต้องเขียน `y = x/(2*a)`
	`y = x**1/3` *จะได้* `(x**1)/3`
	ถ้าต้องการหารากที่สามของ x
	ต้องเขียน `y = x**(1/3)`

ลืมใส่ * สำหรับการคูณ	`y = 2x + 1` *ต้องเขียน* `y = 2*x + 1`
`10e7` มีค่าไม่เท่ากับ 10^7	`10e7` มีค่าเท่ากับ 10×10^7 อยากได้ 10^7 ต้องเขียน `1e7`
`1e3` ไม่ใช่จำนวนเต็ม `1000`	`1e3` มีค่าเท่ากับ `1000.0` ดังนั้น `2345 % 1e2` ได้ `45.0`

สำหรับผู้ที่เคยเรียนภาษา C อย่าเผลอเขียนคำสั่ง ++k หรือ --k	++k คือการติดบวกค่าใน k สองครั้ง จึงมีค่าเท่ากับ k ค่าใน k ไม่เปลี่ยน --k คือการติดลบค่าใน k สองครั้ง จึงมีค่าเท่ากับ k ค่าใน k ไม่เปลี่ยน
ลืม `import math` เมื่อใช้ฟังก์ชันของ `math`	`y = (-b+math.sqrt(b*b-4*a*c)) / (2*a)` จะฟ้องว่าไม่รู้จัก `math`
ใส่วงเล็บเปิดกับปิดไม่ครบ	`import math` `y = 2+(x*abs(y-z/2)` *วงเล็บปิดมีน้อยไป* `y = -b+math.sqrt(b*b-4*a*c)) / (2*a)` *ขาดวงเล็บเปิด*
สะกดชื่อตัวแปรผิด หรือผิดเรื่องการใช้ตัวอังกฤษเล็กกับใหญ่	`count = 0` `Count = count + 1` `Count` *กับ* `count` *เป็นคนละตัว*

ตั้งชื่อตัวแปรซ้ำกับชื่อฟังก์ชันมาตรฐาน
ใน IDLE ตัวแปรที่ถูกต้องมีสีดำ เป็นสีอื่นจะสร้างปัญหา
	`int = 27`
	`print( int )` *ได้* `27`
	แต่หลังจากนี้ ใช้คำสั่ง `a = int(input())`
	เพื่ออ่านข้อมูลจากแป้นพิมพ์ แล้วแปลงเป็นจำนวนเต็มไม่ได้แล้ว
	(IDLE แสดง `int` ด้วยสีม่วง)

นำข้อมูลที่ไม่ใช่สตริงมาบวกกับสตริง	`import math`
	`a = math.pi * r**2`
	`print('area = '+a)` *ผิด*
	`print('area = '+str(a))` *แปลง* `a` *เป็นสตริงก่อน*
	`print('area =',a)` *แบบนี้* `print` *แปลง* `a` *เป็นสตริงให้*