Input / Output / Process

Variables (ตัวแปร)

↳ เป็น ที่ เก็บข้อมูล

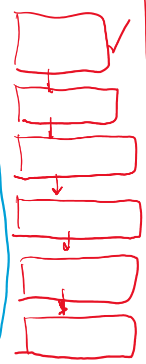
↳ Syntax: var_name = value (เก็บไว้)

↳ เมื่อ เจอ variable ที่ เก็บไว้ มาแทน

Ex

```
x = 5
x = x + 1        5 + 1 = 6
print(x)         → 6
x + 2            6 + 2 = 8
x = 3 + x        3 + 6 = 9
print(x)         → 9
```

ไม่ได้ ไปเก็บค่า หรือแสดงผล ไม่เกิดอะไรขึ้น (← x + 2)



Variable Types

1) string

'Hello', "my_name"

my_name = "Nut"

2) int

จำนวนเต็ม

1, 0, 2

3) float

จำนวนจริง

1.01, -3.01

Math Operators

↳ ใช้ได้กับ int, float

↳ +, -, *, /, //, %, **

Order of math operators

1. ( )
2. **
3. *, /, //, %

* ทำ order แรกก่อน

2.

3. *, /, //, %

4. +, -

EX1 $12 + 6*3 = 30$

12 + 18

30

EX2 $10 + 6//3 - 1*2 = 10$

10 + 2 - 2

10

EX3 $1-(10+2)*(10+3)**2$

1 − 12 * 13 ** 2

1 − 12 * 169

1 − X

Math Equations:

1) $y = \frac{3X}{2}$ → y = (3*X)/2

2) $z = abc + 4$ → z = a*b*c + 4

3) $a = \frac{x+2}{b+1}$ → a = (x+2)/(b + 1)

Input function

↳ Syntax: var = input("optional")

↳ ส่งค่าจาก user กลับมา เป็น string

↓ ถ้าจะใช้ต้อง เก็บไว้ หรือใช้ทันที

↳ ถ้าต้องการ ตัวเลข ต้องใช้ int() หรือ float()

เพื่อ เปลี่ยน string เป็น ตัวเลข

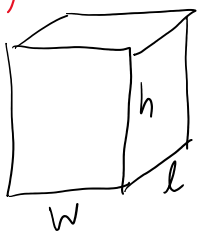
↳ เพื่อ เปลี่ยน string เป็น ตัวเลข

ex `age = int(input("Enter your age: "))`

string

EX คำนวณ Vol ของ ลูกบาศก์

- Vol = w * h * l ← process



Input:
- w, h, l : เป็นจำนวนจริง

Output:
- Vol

```
weight = float(input("Enter weight: "))   ✓
height = float(input("Enter height: "))   ✓
length = float(input("Enter length: "))   ✓
volumn = weight * height * length         ✓
print("Vol = ", volumn)                   ✓
```

input → weight, height, length

process → volumn

Formatting

"\n"

. print() : แสดง ผล แล้ว ขึ้น บรรทัดใหม่

↳ ใช้ commas คั่น ถ้าต้อง การ แสดง หลายค่า
↳ จะ มี ช่อง ว่าง ระหว่างค่า

↳ อีก end = "string" → จะ เปลี่ยน การจบประโยค จาก newline เป็น "string"

ex1
```
print("A")
print("B")
print("C")
```
→
A
B
C

ex2 `print("A", end=".")` → A.B,Cabc

ex2
```
print("A", end=".")
print("B", end=",")
print("C", end="abc")
```
→ A.B,Cabc

↳ เพิ่ม sep = "string" → เปลี่ยนจากช่องว่างระหว่างค่า เป็น "string" (sep → separate)

Ex
```
print("A", "B", "C")
```
→ A B C
```
print("A", "B", "C", sep="")
```
→ ABC

(sep="" ← ไม่มีค่า / empty string)
```
print("A", "B", "C", sep=",")
```
→ A,B,C

f-string

↳ formatted string → ไปทำงาน/รันให้เรา แล้วเอาค่ามาใส่ตรงนั้น

↳ Syntax: f" {variable/expression} "

(f ← เติม f หน้า string)

ex1
```
name = "Nut"
print("Hello", name, "!")
```
→ Hello Nut !
```
print(f"Hello {name}!")
```
→ Hello Nut!

({name} → run → "Nut")

ex2
```
name = "Nut"
gender = "Male"
age = 33
print(f"Name: {name}. G= {gender}. Age= {age}")
```
↳ Name: Nut. G= Male. Age= 33

ex3

```python
x=5
print(f"x+5 = {x+5}")
```

5 + 5 = 10

→ x+5 = 10

↳ format จุดทศนิยม ด้วย f-string

↳ Syntax: f" {value: .2f } "

- ตำแหน่งทศนิยม
  ↳ 2 คือ 2 ตำแหน่ง

ex:

```python
print(f"{1.234: .2f}")   # → 1.23
print(f"{1: .3f}")       # → 1.000
x= 1.579
print(f"{x: .1f}")       # → 1.6
```

* เติม commas ในตัวเลข เช่น: 1,000,000.00

f"{1000000 : ,.2f}"

Named Constant

- การตั้งชื่อตัวแปร สำหรับ ค่าที่ไม่เปลี่ยนแปลง (constant)
- ตั้งชื่อตัวแปร เป็น ตัวใหญ่หมด : DONUT_PRICE = 20
  ↳ วางด้านบนสุด ของ โปรแกรม

Ex1

```python
amount = balance * 0.069
```

0.069 ?

Ex2

```python
INTEREST_RATE = 0.069
amount = balance * INTEREST_RATE
```

Ex ร้านขายขนม ขาย donut กับ cupcake

Ex ร้านขายขนม ขาย donut กับ cupcake
donut ราคา 10 บาท, cupcake 20 บาท
โปรแกรมรับจำนวน donut กับ cupcake
แล้วสั่งราคาทั้งหมดออกมา

* ห้ามใช้ Named Constant
* ราคาทั้งหมดห้ามคูณทศนิยมลงตำแหน่งหน่วย

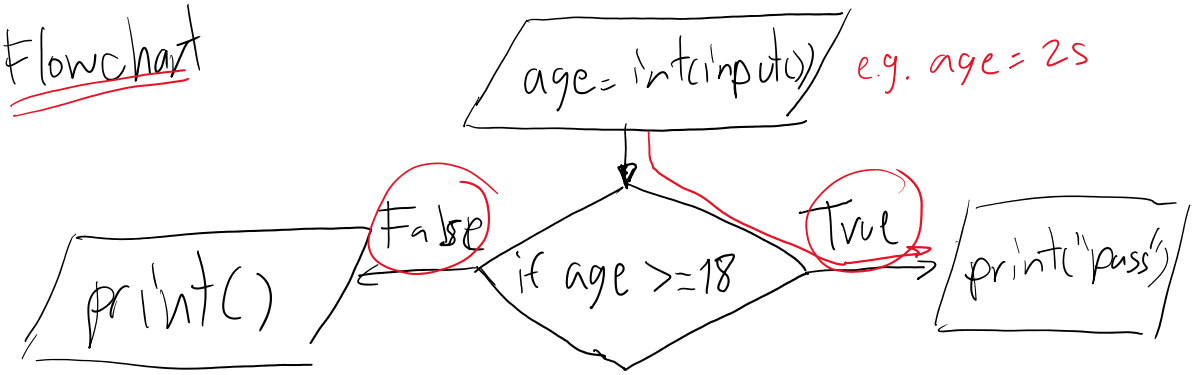
```
P_DONUT = 10
P_CUPCAKE = 20
```
} Named Constant

input | `n_donut = int(input("Enter # donut: "))`
`n_cupcake = int(input("Enter # cupcake: "))`
process | `total = (n_donut * P_DONUT) + (n_cupcake * P_DONUT)`
output | `print(f"Total price is {total:.2f}")`

Chapter 2: Decision Structures

* Code บางส่วนจะทำก็ต่อเมื่อเงื่อนไขเป็นจริง
  - ตัวอย่าง: รับค่าอายุจาก user
    แล้วแสดงผล "pass" ถ้า user อายุมากกว่า 18

Flowchart



Boolean Type : ค่าความจริง

Boolean Type : ค่าความจริง

↳ มี 2 ค่า True กับ False

ใน python T, F ตัวใหญ่

Comparison Operators

- เปรียบเทียบ ค่า สอง ค่า ว่า มีความสัมพันธ์ เป็นจริงไหม?
- เช่น $10 > 5$ → 10 มากกว่า 5 ไหม?
  ↳ จริง ⇒ True
  $5 > 10$ → 5 มากกว่า 10 ไหม?
  ↳ False
- คำสั่ง

`>`	มากกว่า ?
`<`	น้อยกว่า ?
`>=`	มากกว่า เท่ากับ ?
`<=`	น้อยกว่า เท่ากับ ?
`==`	เท่ากันไหม ? (*)
`!=`	ไม่เท่ากันไหม ? (*)

↳ ใช้ได้ กับ ทั้ง string กับ ตัวเลข

Ex
- `10 >= 5` → True
- `5 == 5` → True
- `6 != 7` → True
- `7 == (4+1)` → False

- 6 != 1 → True
- 7 == (4+1) → False
  (4+1 = 5)
- (10+5) > 4+5*6 → False
  15 > 34 (4+30)

* คำนวณ เลข ก่อน เปรียบ เทียบ

- **if statement**
  ↳ ถ้า เงื่อนไข เป็นจริง จะมา ทำ ใน if-block

Syntax:

```
if ค่าความจริง/condition :   → True
    statement      ← if-block
    statement
    statement
```

↑ ช่องว่าง (indentation) เพื่อ บอกว่า code อยู่ ใน if structure

Ex1

```
age = 25 ✓
           25 >= 18 → True
if (age >= 18) :  → True
    print("pass") ✓ if-block ✓  →  pass
print("Done")                   →  Done
```

Output:

pass
Done

Ex2

```
age = 10 ✓
          10 >= 18 → False
if age >= 18 :
    print("pass") ✗
print("Done")                  →  Done
```

print("Done") → Done

EX3

```
age = 10
if age >= 18:
    print("Hello")
    print("you're in")
    print("***")
print("Done")
```

10 >= 18 → False

True

False

→ Done

EX4

```
score = 45
if score >= 40:
    print("Good")
if score >= 50:
    print("***")
    print("Congrat")
print("Done")
```

45 >= 40 → True

→ Good

45 >= 50 → False

False

→ Done