Mahidol University

Wisdom of the Land

Mahidol University

Lecture 04: Review

Lecturers:

Aj. Jidapa Kraisangka

Aj. Akara Supratak

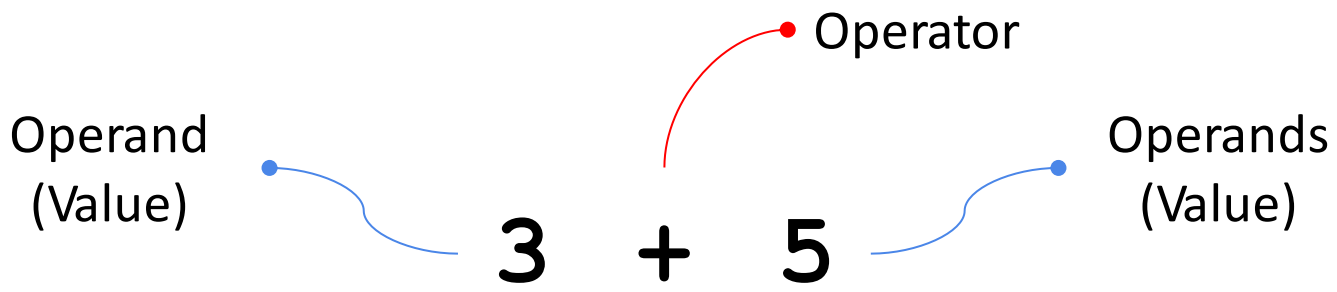
Aj. Tipajin Thaipisutikul

- Python Basics
- Flow Controls
- Boolean, Relational Expression and Logical Expression
- **`if`** statement
- **`else`** and **`elif`**

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Expression

- Expression เป็นกลุ่มของค่าต่าง ๆ (values, e.g 3, 5) มากระทำต่อกันโดยใช้ตัวดำเนินการคำนวณ (operators, e.g. +)



- ผลจากการคำนวณจาก Expression จะเป็นค่าผลลัพธ์เพียงค่าเดียว

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Arithmetic Op. Precedence

Table 1-1: Math Operators from Highest to Lowest Precedence

Operator	Operation	Example	Evaluates to . . .
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

Table from https://automatetheboringstuff.com/2e/chapter1/

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Example: Expression evaluation

Exercise: `5 * 0 + 4 - 3 // 2 ** 2 + 7`

Step 1:

Expression: `5 * 0 + 4 - 3 // 2 ** 2 + 7`

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า ** มีลำดับความสำคัญสูงสุด;

Result: `5 * 0 + 4 - 3 // 4 + 7` (Highest precedence)

Step 2:

Expression: `5 * 0 + 4 - 3 // 4 + 7`

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า // มีลำดับความสำคัญสูงสุด

Result: `5 * 0 + 4 - 0 + 7`

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Example: Expression evaluation

Step 3:

Expression: 5 * 0 + 4 – 0 + 7

เมื่อทำจากซ้ายไปขวา จะเห็นได้ว่า * มีลำดับความสำคัญสูงสุด;

Result: 0 + 4 – 0 + 7

Step 4: 0 + 4 – 0 + 7

Step 5: 0 + 4 + 7

Step 6: 4 + 7

Final result: 11

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Exercise: Expression evaluation

Exercise Q1: `3 // 2 * 3 ** 2 - 1`

Exercise Q2: `4 % 2 + 4 / 2 - 4 // 2`

Exercise Q3: `(20 % 2 ** 2) - 14 % 10`

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Logical Expression

Logical expression คือ expressions ที่ใช้ตัวดำเนินการด้านตรรกศาสตร์(Boolean operators ได้แก่ **`and or`** และ **`not`**) โดยจะเปรียบเทียบค่า Boolean และประมวลผลลัพธ์เป็นค่า Boolean

- *ลำดับการประมวลผลจะเรียง Operator Precedence ของ Logical expression ใน Python จะไล่ลำดับการประมวลจากซ้ายไปขวา และ not → and → or*

Table 2-2: The and Operator's Truth Table

Expression	Evaluates to . . .
True and True	True
True and False	False
False and True	False
False and False	False

Table 2-3: The or Operator's Truth Table

Expression	Evaluates to . . .
True or True	True
True or False	True
False or True	True
False or False	False

Table 2-4: The not Operator's Truth Table

Expression	Evaluates to . . .
not True	False
not False	True

Tables from https://automatetheboringstuff.com/2e/chapter2/

Exercise: Logical Expression

```
(1!=-1) and not(5!='5')

(1==-1) or (5!=(5%2))

(1!=-1) and not(5!='5') or (5!=(5%2))

 not (10+10 > 1+1)
```

Writing an Expression

- การเขียนโปรแกรมจะอาศัยต้องแปลงโจทย์ให้อยู่ในรูปแบบของ Expression
- บางครั้งต้องมีการตรวจสอบเงื่อนไขของค่าตัวแปร เช่น

x ต้องมีค่าอย่างน้อย 10 → `x >= 10`

x ต้องเป็นเลขจำนวนบวก → `x > 0`

x ต้องเป็นเลขที่อยู่ระหว่าง 2 กับ 5 → `x>2 and x<5`

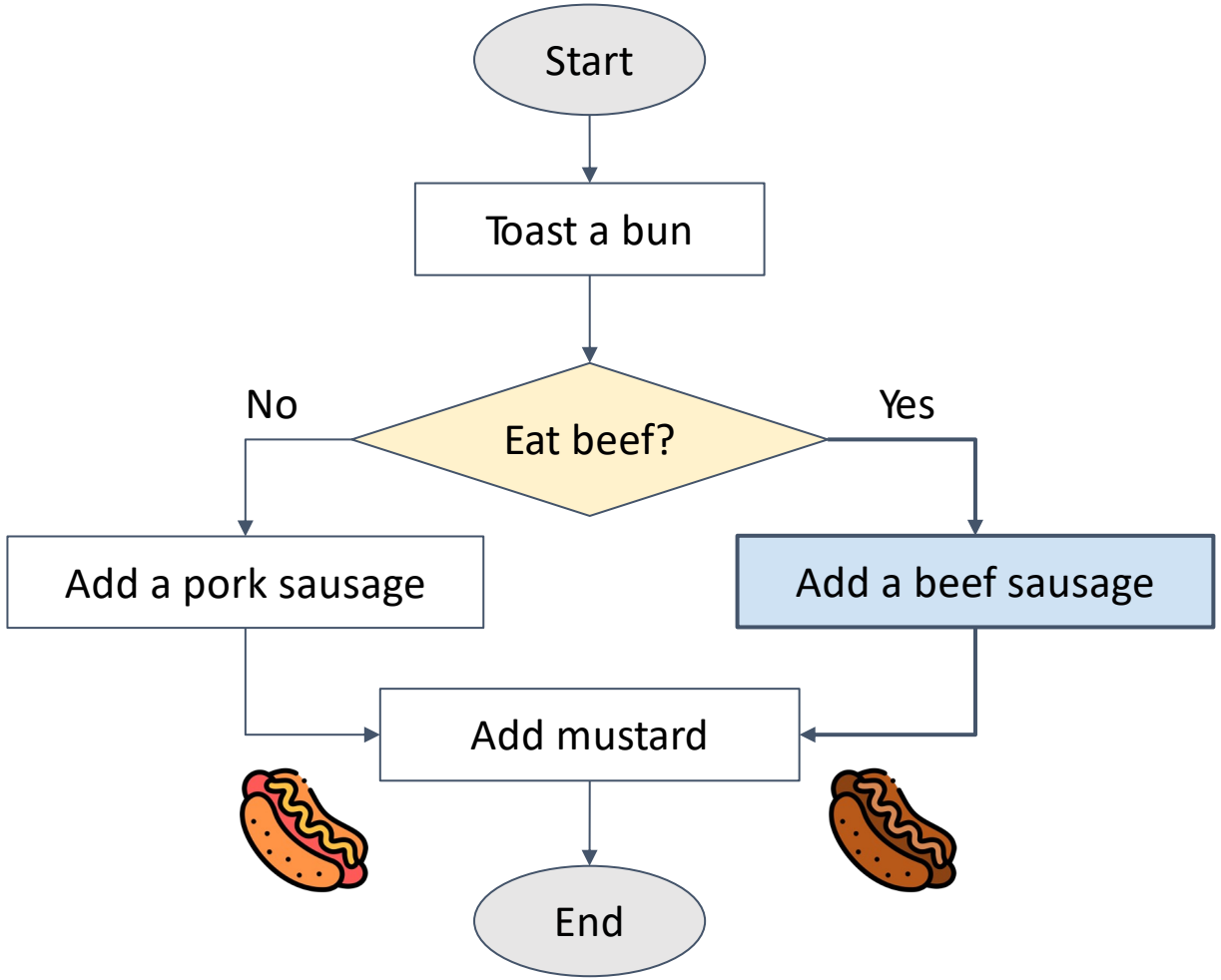
x ต้องเป็นเลขตั้งแต่ 2 ถึง 5 → `x>=2 and x<=5`

x ต้องไม่เป็นเลขคู่ → `x%2 != 0`

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Conditions

- Relational Expression และ Logical Expression เป็นพื้นฐานสำหรับการเขียนเงื่อนไข (Conditions)
- Conditions จะถูกประมวลผลเป็นค่า Boolean เสมอ
- ผลของ Conditions (ได้แก่ `True`, `False`) จะเป็นตัวกำหนดชุดคำสั่งที่จะกระทำ



Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

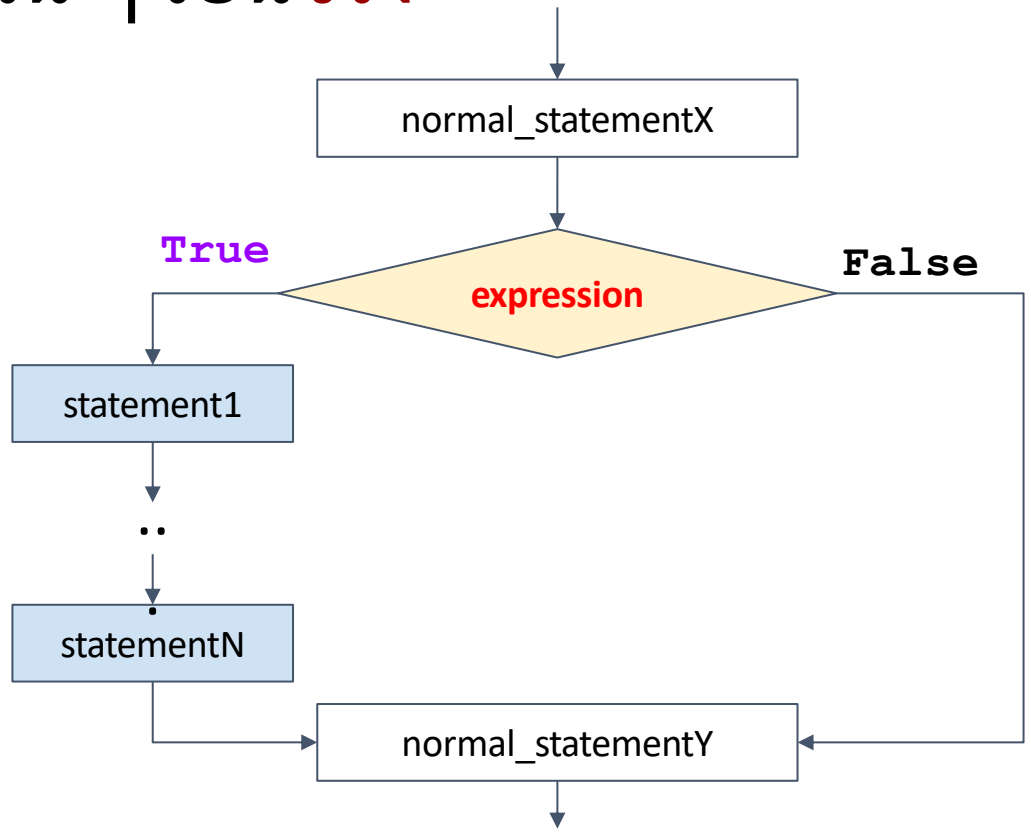
if statement

- **if** เป็นคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนดเมื่อเงื่อนไขนั้น ๆ เป็นจริง

```
normal_statementX

if expression:
    statement1
    ...
    statementN

normal_statementY
```



Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

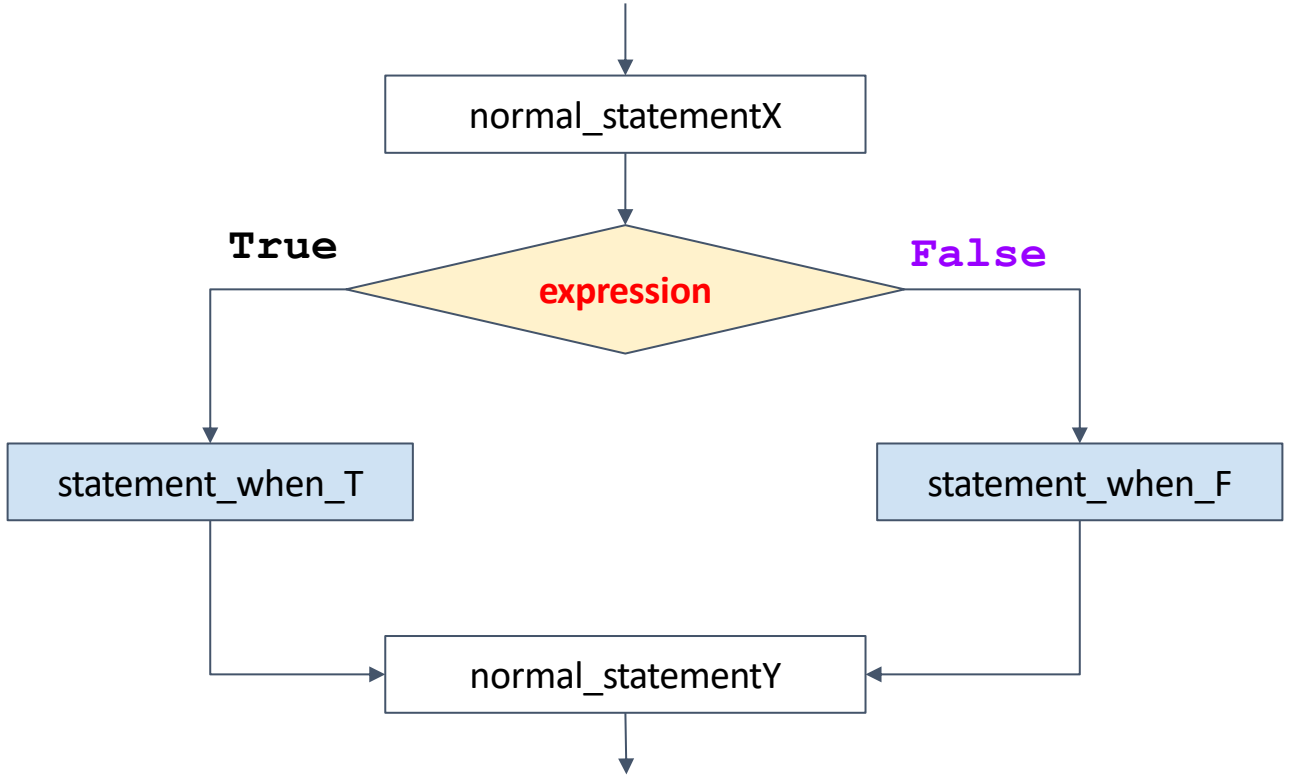
else statement

- **else** เป็นคำสั่งที่ใช้ต่อเนื่องจากคำสั่ง **if** เพื่อควบคุมการทำงานของโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนดเมื่อเงื่อนไขนั้น ๆ เป็นเท็จ

```
normal_statementX

if expression:
      statement_when_T
else :
    statement_when_F

normal_statementY
```



Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

if-else statement

เมื่อถึงคำสั่ง **if** ตัว **condition** จะถูกประมวลผล และแสดงผลลัพธ์ตามค่า Boolean หาก **condition** เป็นจริงจะทำใน Block **if** และ **condition** เป็นเท็จจะทำใน Block **else**

greeting-by-sec2.py

```
section=int(input());
if section==1: #True
    print('Hello, Aj.Pa')
    print('Nice to meet you')
else: #False
    print('Hello, Aj.Tip')
print('Have a good day!!')
```

เมื่อ `section` มีค่าเป็น 1 ทำให้ `section==1` เป็น **True**

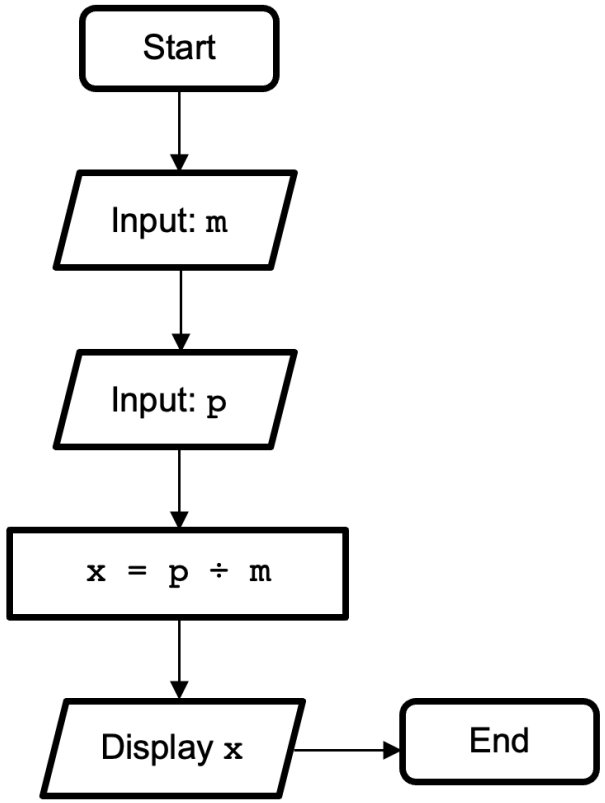
```
Hello, Aj.Pa
Nice to meet you
Have a good day!!
```

เมื่อ `section` มีค่าเป็นอื่นที่ไม่ใช่ 1 ทำให้ `section==1` เป็น **False**

```
Hello, Aj.Tip
Have a good day!!
```


Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Exercise: Flowchart -> Python code



Python Code

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Exercise: Student Discount

จงเขียนโปรแกรมเพื่อบอกราคาสุดท้ายที่รวมส่วนลดแล้วแก่ผู้ใช้ โดยรับ input เป็นจำนวนสินค้าที่ซื้อ `(qt: int)` ราคาสินค้าต่อหน่วยคือ 100 บาท หากผู้ใช้ซื้อยอดรวมอย่างน้อย 400 บาท จะได้รับส่วนลด 10 % ถ้าซื้อยอดรวมน้อยกว่า 400 บาท จะไม่ได้รับส่วนลด

Source code: `productDisc.py`

```
qt = int(input())
total = 100.0 * qt
if total >=400:
   print(total*0.9)
else:
   print(total)
```

Run a program:

```
4
360.0
```

```
3
300.0
```

```
5
450.0
```

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Exercise: Age Range

จงเขียนโปรแกรมเพื่อรับอายุ หากอายุอยู่ตั้งแต่ 13 ถึง 19 ปี โปรแกรมจะแสดงผลลัพธ์เป็นข้อความ "Teenager" ก่อนจะแสดงข้อความ "Bye bye" เพื่อจบการทำงาน

Source code: `age.py`

```
age = float(input())
if age >= 13 and age <=19:
   print("Teenager")
print("Bye bye")
```

Run a program:

```
10
Bye bye
```

```
13
Teenager
Bye bye
```

```
20
Bye bye
```


Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Exercise: AQI

จงเขียนโปรแกรมเพื่อรับค่าดัชนีคุณภาพอากาศ (AQI) เป็น int และแสดงผลลัพธ์คุณภาพอากาศตามตารางต่อไปนี้

AQI	ข้อความที่แสดงผล
0-50	Good
51-100	Moderate
101-200	Unhealthy
มากกว่า 200	Very unhealthy

Source code: `aqi.py`

```
aqi = int(input())
if aqi<=50:
   print("Good")
elif aqi>=51 and aqi<=100:
   print("Moderate")
elif aqi>=101 and aqi<=200:
   print("Unhealthy")
else:
   print("Very Unhealthy")
```

Run a program:

```
50
Good
```

```
150
Unhealthy
```

```
240
Very Unhealthy
```

Contact: [jidapa.kra, akara.sup, tipajin.tha]@mahidol.edu

Thanks to: https://keep-calm.net/keep-calm-and-code-on.html

Good Luck for CP1