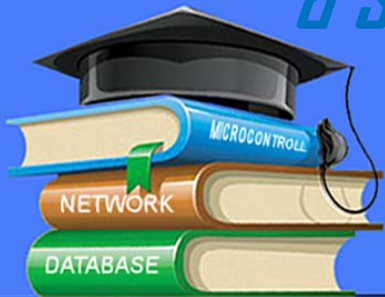


Chapter 4: Flow Chart & Selection Statement



Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Contents

Python
FlowChart &
Selection
Statement

Flow Chart

Review: Relational expression

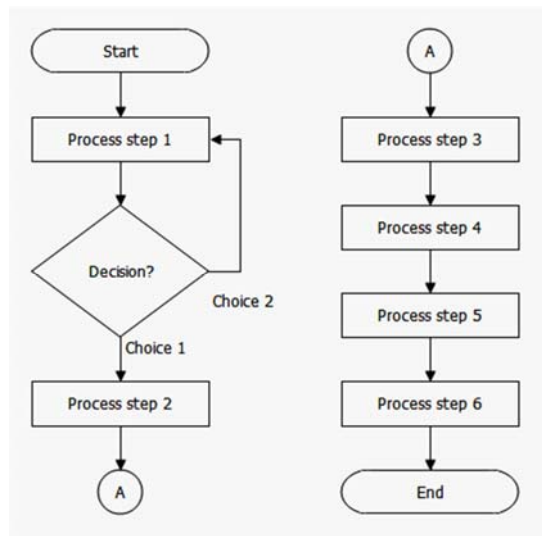
Introduce: Boolean data type

Selection Statement

2




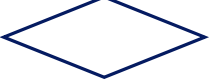
Flow Chart

◆ **ผังงาน (Flow Chart) คือ**
วิธีการออกแบบโปรแกรม
อย่างเป็นขั้นตอน โดยการ
ใช้สัญลักษณ์และตัวอักษร
ประกอบการอธิบายในแต่ละ
ขั้นตอน



3

Flow Chart

สัญลักษณ์	ความหมาย
	จุดเริ่มต้นหรือจุดจบของโปรแกรม (Terminal)
	การรับและแสดงผลของข้อมูล (Input/Output)
	การประมวลผล (Process)
	การตัดสินใจ (Decision /Selection)

4

Flow Chart

สัญลักษณ์	ความหมาย
	ข้อมูลออกทางจอภาพ (Display)
	ทิศทางของขั้นตอนการดำเนินงาน (Flow Line)
	จุดเชื่อมการทำงานที่อยู่หน้าเดียวกัน (Connector)
	จุดเชื่อมการทำงานที่อยู่คนละหน้า (Off Page Connector)

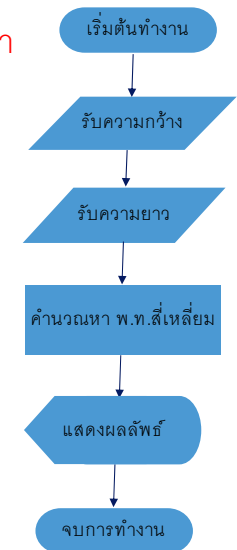
5

Flow Chart

ตัวอย่างคำนวณหาพื้นที่ของรูปสี่เหลี่ยมผืนผ้า

ขั้นตอนการทำงานของโปรแกรม :

- เริ่มต้นทำงาน
- รับความกว้างของรูปสี่เหลี่ยมผืนผ้า
- รับความยาวของรูปสี่เหลี่ยมผืนผ้า
- คำนวณหาพื้นที่ของรูปสี่เหลี่ยมผืนผ้า
- แสดงผลลัพธ์ที่ได้ออกทางจอภาพ
- จบการทำงาน

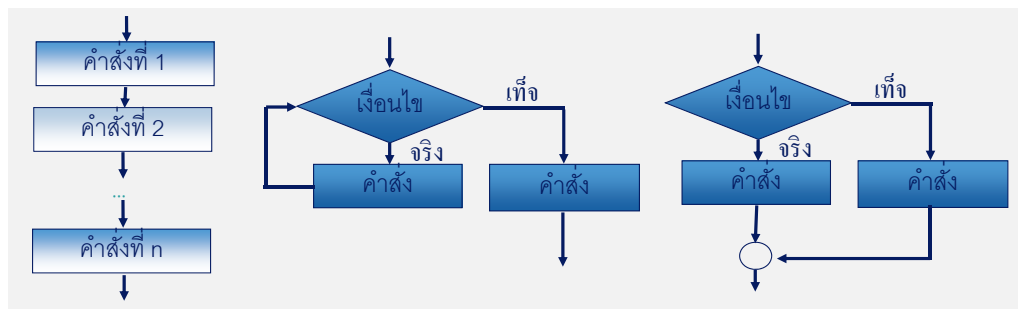


6

Flow Chart

◆ ผังงานโครงสร้างควบคุมหลักในการเขียนโปรแกรม

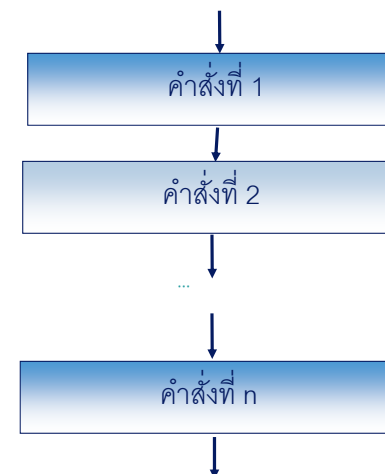
- โครงสร้างแบบลำดับ (Sequential structure)
- โครงสร้างแบบมีทางเลือก (Selection structure)
- โครงสร้างแบบทำซ้ำ (Repetition structure)



7

Flow Chart

◆ โครงสร้างแบบลำดับ (Sequential structure)



- โครงสร้างแสดงขั้นตอนการทำงานที่เป็นไปตามลำดับก่อนหลัง
- แต่ละขั้นตอนจะถูกประมวลผลเพียงครั้งเดียวเท่านั้น

8

Flow Chart

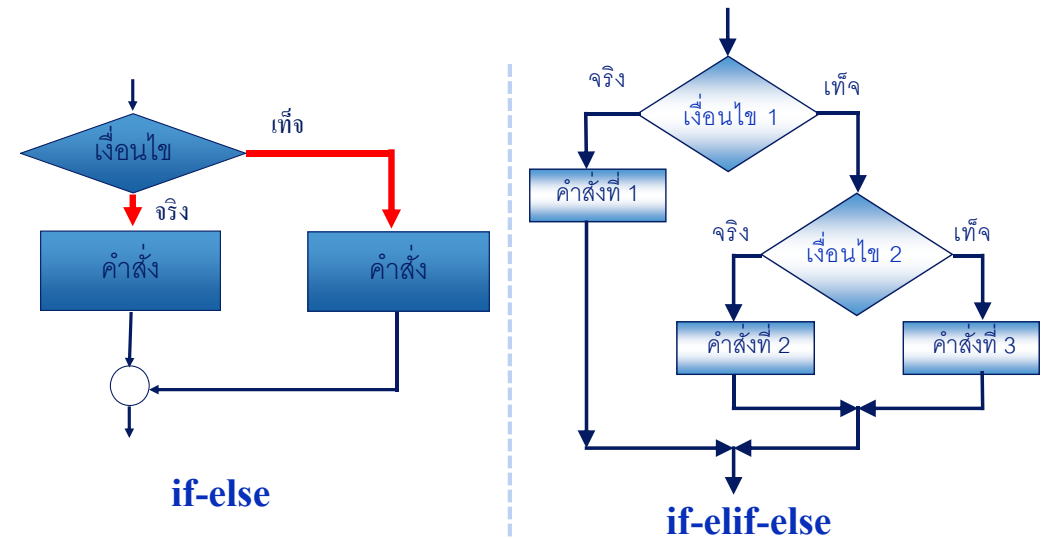
◆ โครงสร้างแบบมีทางเลือก (Selection structure)

- โครงสร้างแบบมีทางเลือก คือ โครงสร้างที่มีเงื่อนไข ขั้นตอนการทำงานบางขั้นตอนที่ต้องมีการตัดสินใจ เพื่อเลือกวิธีการประมวลผลขั้นตอนต่อไป และอาจจะมีบางขั้นตอนที่ไม่ได้รับการประมวลผล
- การตัดสินใจอาจมีทางเลือก 2 ทางหรือมากกว่าก็ได้

9

Flow Chart

◆ โครงสร้างแบบมีทางเลือก (Selection structure)



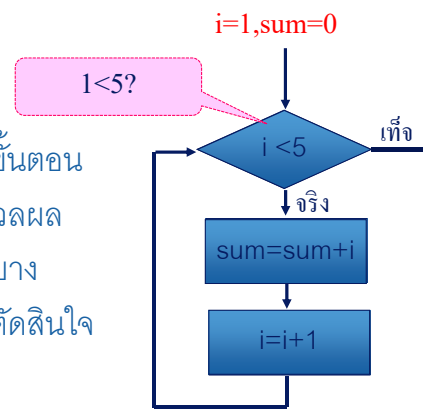
10

Flow Chart

โครงสร้างแบบทำซ้ำ

(Repetition structure)

โครงสร้างแบบทำซ้ำ คือ โครงสร้างที่ขั้นตอนการทำงานบางขั้นตอนได้รับการประมวลผลมากกว่า 1 ครั้ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขบางประการ โครงสร้างแบบนี้ต้องมีการตัดสินใจในการทำงานซ้ำ



11

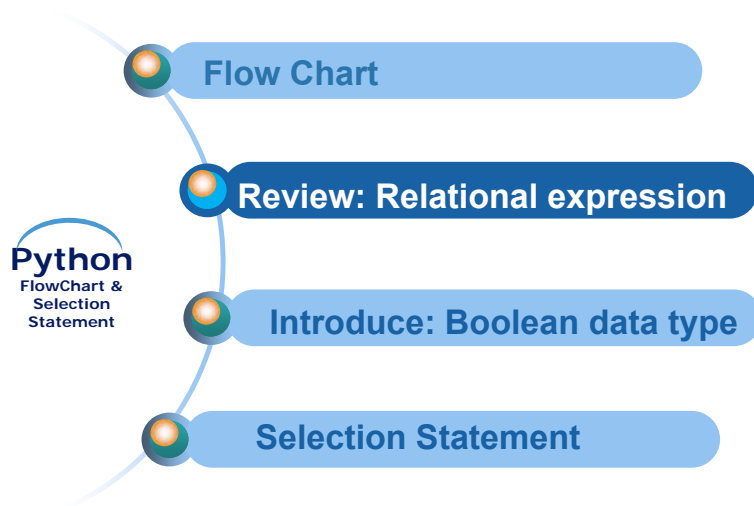
QuickCheck1

เขียนผังงาน (flow chart) และโปรแกรมสำหรับการแปลงอุณหภูมิองศาเซลเซียส(C)ให้เป็นองศาฟาเรนไฮต์(F) โดยรับค่า C จากคีย์บอร์ด ($F = C \times 1.8 + 32$)

Temperature in C: **34**
34.0 C = 93.2 F

12

Contents



13

Review: Relational expression

Python	รายละเอียด	ตัวอย่าง	ข้อสังเกต
>	มากกว่า	$x > 5$	
>=	มากกว่าหรือเท่ากับ	$x \geq 5$	ระวังที่จะเขียนผิดเป็น $=>$ วิธีการช่วยจำให้เขียนตามลำดับการพูด
<	น้อยกว่า	$x < 10$	
<=	น้อยกว่าหรือเท่ากับ	$x \leq 10$	ระวังที่จะเขียนผิดเป็น $=<$ วิธีการช่วยจำให้เขียนตามลำดับการพูด
==	เท่ากับ	$x == 5$	อย่าสับสนกับเครื่องหมาย = ในนิพจน์กำหนดค่า
!=	ไม่เท่ากับ	$x != 5$	

14

Review: Relational expression

การเปรียบเทียบ

ค่าคงที่กับค่าคงที่

$5 > 3$
 $20.5 < 30.25$

ตัวแปรกับค่าคงที่

$x != 10$
 $a \leq 150.5$

ตัวแปรกับตัวแปร

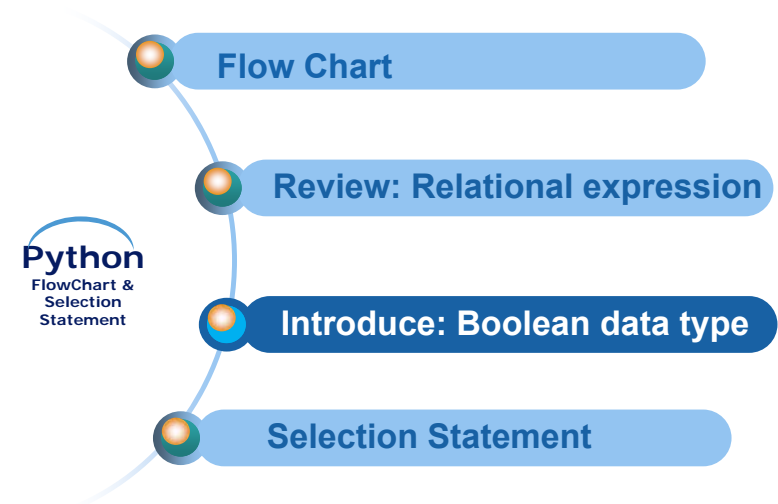
$\text{num1} == \text{num2}$
 $a \geq b$

นิพจน์กับค่าคงที่

$x + y > 5$
 $y / 2 != 0$

15

Contents



16

Boolean data type

- ◆ ข้อมูลชนิดบูลีนมีค่าได้สองค่าเท่านั้นคือ **จริง (True)** กับ **เท็จ (False)**
- ◆ ค่าจริงแทนด้วย "True" และค่าเท็จแทนด้วย "False"

ตัวอย่าง

```
print(5>6)
x='A'>'C'
print(x)
print(type(x))
```

17

Boolean data type

- ◆ สามารถแปลงค่าได้โดยใช้ bool() Function

bool(expression)

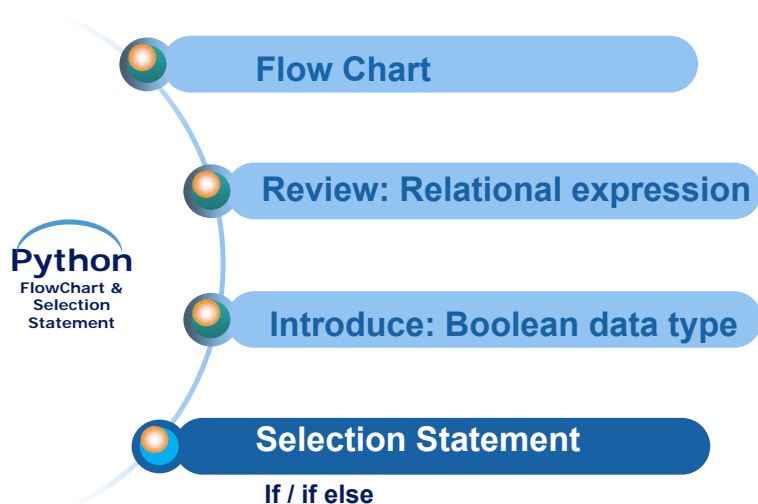
ตัวอย่าง

```
x=5
print(x)
print(bool(x))
print(type(bool(x)))

y=bool(0) and 101%2!=0
print(y)
```

18

Contents



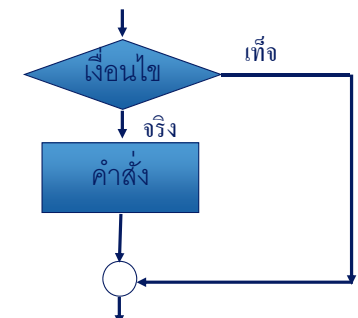
19

Selection Statement (if statement)

- ◆ คำสั่ง if ใช้สำหรับกรณีที่ต้องมีการตัดสินใจหรือมีเงื่อนไข
- ◆ คำสั่ง if **ต้องมี**เครื่องหมาย ":" (colon) ตามหลังส่วนนิพจน์หรือเงื่อนไข
- ◆ ประกอบด้วยสองส่วน คือ ส่วนของการตรวจสอบเงื่อนไข และส่วนของคำสั่งที่ต้องการให้ทำในกรณีที่เงื่อนไขนั้นเป็นจริง
- ◆ รูปแบบ

```
if (นิพจน์หรือเงื่อนไข) : คำสั่ง

if (นิพจน์หรือเงื่อนไข) :
    TAB → คำสั่ง
```



20

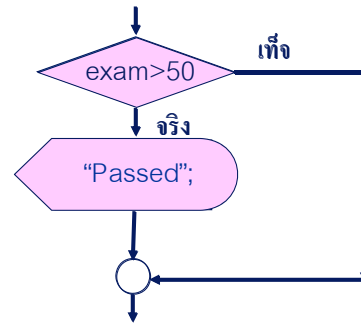
Selection Statement (if statement)

- ◆ ประกอบด้วยสองส่วน คือ ส่วนของการตรวจสอบเงื่อนไข และส่วนของคำสั่งที่ต้องการให้ทำในกรณีที่เงื่อนไขนั้นเป็นจริง

◆ ตัวอย่าง

```
if (exam>50): print("Passed")
```

```
if (exam>50):  
    print("Passed")
```



21

Selection Statement (if statement)

- ◆ สำหรับกรณีที่เงื่อนไขเป็นจริงแล้วต้องการจะทำหลายคำสั่ง **ให้ใช้** **ปีกกาคร่อม** คำสั่งทั้งหมดนั้น ซึ่งเรียกว่า "Block"

◆ รูปแบบ

```
if (นิพจน์หรือเงื่อนไข):  
    คำสั่ง  
    คำสั่ง
```

ตัวอย่าง

```
if (a > 0):  
    x = a * b  
    y = y + x
```

22

Selection Statement (if statement)

```
score = input()  
  
score = int(score)  
  
if(score >= 50):  
    print("You pass")
```

23

Selection Statement (if/else statement)

- ◆ คำสั่ง if/else ใช้สำหรับกรณีที่ ต้องมีการตัดสินใจ 2 ทางเลือก

รูปแบบ

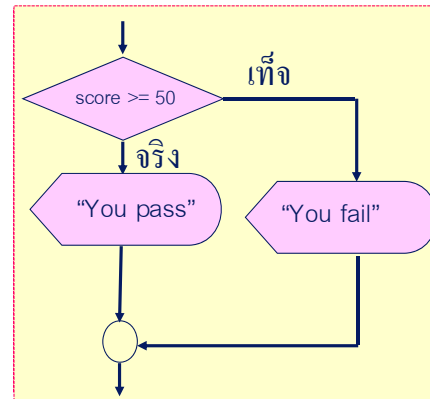
```
if (นิพจน์หรือเงื่อนไข): คำสั่งที่ 1  
else: คำสั่งที่ 2
```

```
if (นิพจน์หรือเงื่อนไข):  
    คำสั่งที่ 1  
else:  
    คำสั่งที่ 2
```

24

Selection Statement (if/else statement)

```
if (score >= 50):
    print("You pass")
else:
    print("You fail")
```



25

Selection Statement (if/else statement)

◆ กรณีที่แต่ละทางเลือกจะต้องทำงานมากกว่า 1 คำสั่ง

◆ รูปแบบ

if (นิพจน์หรือเงื่อนไข):

คำสั่งที่ 1

คำสั่งที่ 2

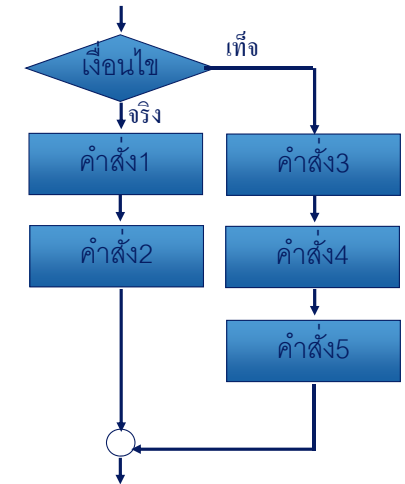
else:

คำสั่งที่ 3

คำสั่งที่ 4

คำสั่งที่ 5

•ผังงาน



26

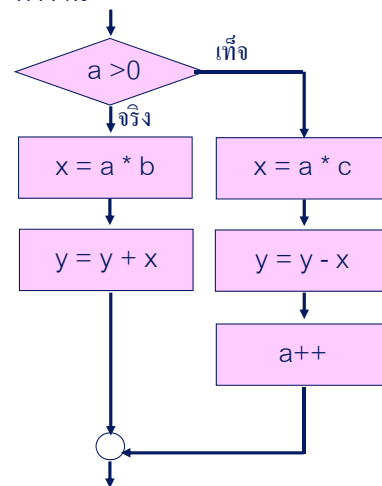
Selection Statement (if/else statement)

◆ กรณีที่แต่ละทางเลือกจะต้องทำงานมากกว่า 1 คำสั่ง

◆ ตัวอย่าง

```
if (a > 0)
    x = a * b
    y = y + x
else
    x = a * c
    y = y - x
    a++
```

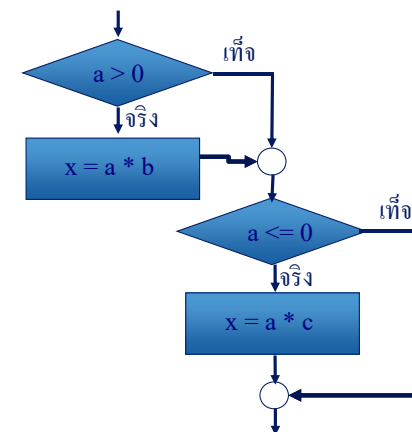
•ผังงาน



27

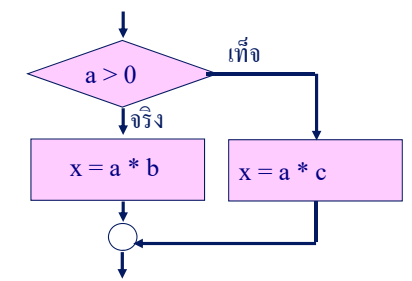
if VS if/else statement

```
if (a > 0):
    x = a * b
if (a <= 0):
    x = a * c
```



28

```
if (a > 0):
    x = a * b
else:
    x = a * c
```



QuickCheck2

- ◆ กำหนดให้โปรแกรมมีขั้นตอนการทำงานดังนี้
เริ่มต้น
รับค่า x และ y
ถ้า $x > y$ และ $y > 0$ ให้ y มีค่าเป็น 0
จบ

จงเขียน flow chart ของโปรแกรมนี้

จงหาค่า y เมื่อคอมพิวเตอร์ทำโปรแกรมนี้นจบ และผู้ใช้ใส่ค่า 5 และ 3



29

QuickCheck2

30

QuickCheck3

- ◆ จงหาผลลัพธ์ของโปรแกรมนี เมื่อผู้ใช้ใส่ค่าต่อไปนี้

```
n,m = input("Enter n,m: ").split()
n = int(n)
m = int(m)
if (n > 0):
    print("Positive number")
if (m > 0):
    print("Positive number")
```

n	m	Output
3	5	
3	0	
0	-2	

31