

Basic Python Programing

Chapter 1

Nachai Paramesthanakorn

What is Programing?

โปรแกรมคอมพิวเตอร์จะทำงานตามชุดคำสั่ง ซึ่งชุดคำสั่งต่างๆ คนต้องเขียนเพื่อสั่งการคอมพิวเตอร์ให้ทำงานตามวัตถุประสงค์

การเขียนหนังสือ	การเขียนโปรแกรม
ได้ผลลัพธ์หนังสือ	ได้ผลลัพธ์เป็นโปรแกรม
เขียนให้คนอ่าน	เขียนให้คอมพิวเตอร์อ่าน
มีหลายภาษา เช่น ภาษาไทย, ภาษาอังกฤษ	มีหลายภาษา เช่น Python, C, C#, Java
ถ้าเขียนผิดไวยากรณ์ หรือเขียนคำผิดเล็กน้อย ยังพอเข้าใจความหมาย	ถ้าเขียนผิดไวยากรณ์ โปรแกรมจะทำงานไม่ได้
หนังสือเล่มเดียวกัน บางครั้งคนอ่านอาจจะเข้าใจไปคนละอย่าง คนละความหมาย	โปรแกรมถูกเขียนขึ้นเพื่อให้คอมพิวเตอร์ทุกเครื่องอ่านแล้วเข้าใจเหมือนกันหมด

Level of programming language

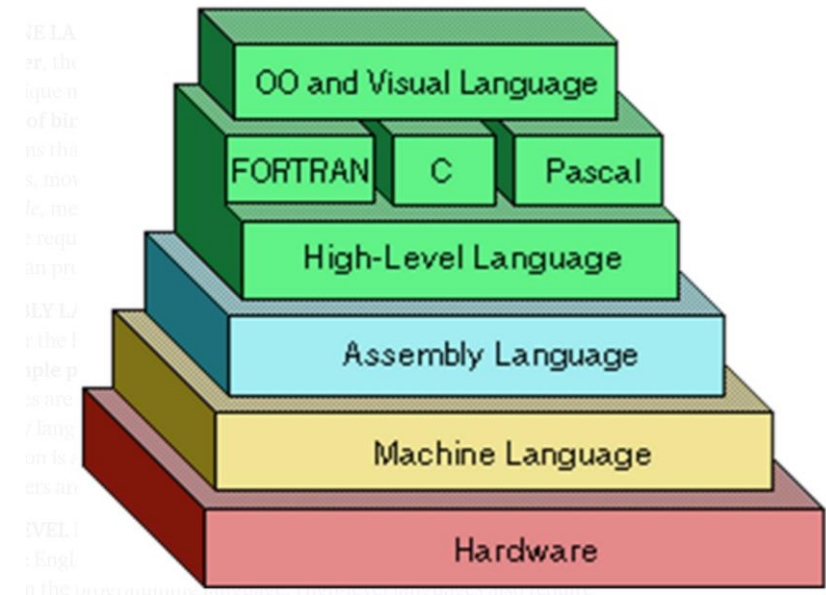
Low-level programming language

- Machine code คนเข้าใจยาก แต่คอมพิวเตอร์เข้าใจง่าย
- Assembly ใช้ภาษาอังกฤษเขียน แล้วแปลงเป็นภาษาเครื่องภายหลัง

High-level programming language

ใช้ภาษาอังกฤษ ตัวเลข อักขระ ตาม syntax ของแต่ละภาษา ทำให้คนทั่วไปสามารถอ่านคำสั่งได้ แล้วใช้ตัวแปลภาษาให้เครื่องเข้าใจอีกทีหนึ่ง เช่น

- Python
- Java
- C



ตัวแปลภาษา

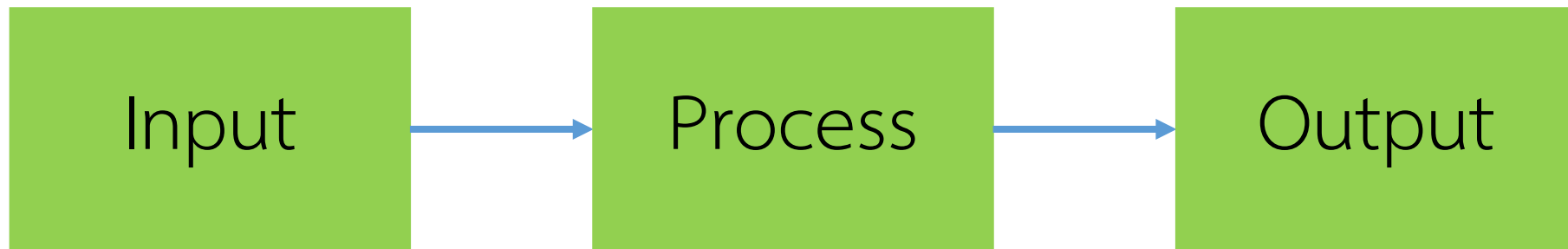
- Compiler

แปลภาษาก่อนแล้วค่อยสั่งให้ทำงาน เช่น ไฟล์ .exe

- Interpreter

แปลภาษาพร้อมกับสั่งให้ทำงานไปด้วย เช่น การรันคำสั่งบนเว็บไซต์

กระบวนการเขียนโปรแกรม



- การพิมพ์ข้อความบน keyboard
- การคลิกเมาส์
- การพูดผ่านไมค์โครโฟน
- การถ่ายรูปจากกล้อง

- การแสดงผลข้อความบนหน้าจอ
- การส่งเสียงจากลำโพง
- การส่งภาพ/วิดีโอบนหน้าจอ

Install Python & IDE

- ใช้ Python version 3.9 +
- ใช้ IDE ชื่อ Visual Studio Code

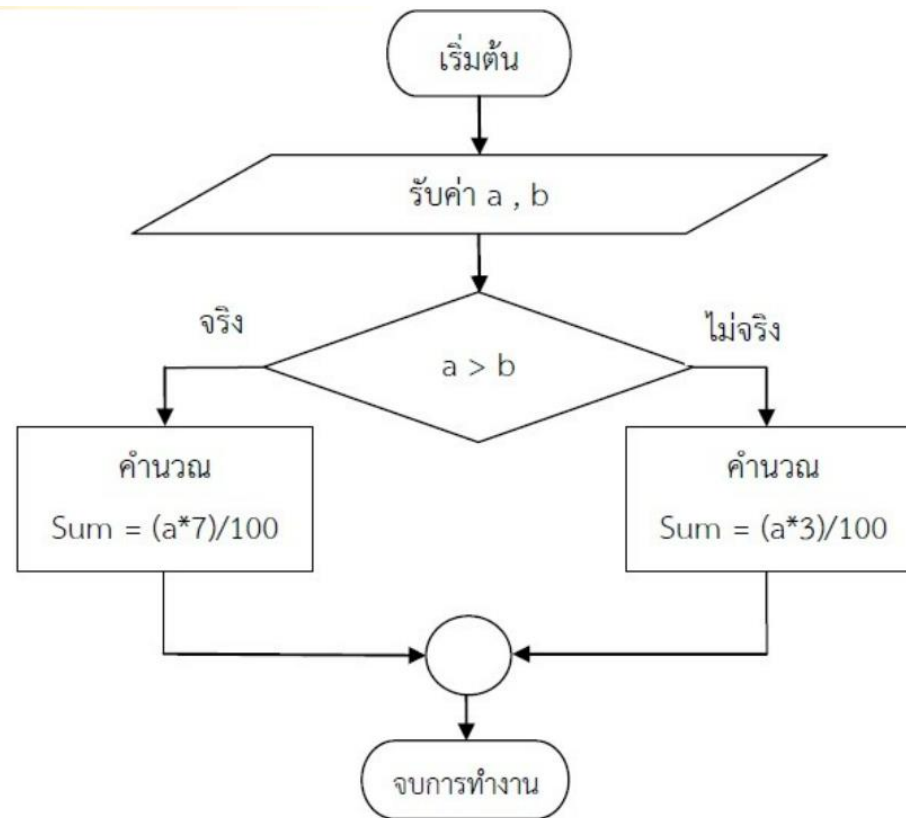
โหมดการใช้งาน Python มี 2 โหมด

- โหมดสคริปต์ (Script mode) ต้องเขียนชุดคำสั่งทั้งหมดลงในไฟล์แล้วบันทึกไว้ เวลาสั่งรันโปรแกรมก็ให้ทำจากไฟล์ที่เดียวเลย ซึ่งข้อดีเราสามารถรันโปรแกรมได้ซ้ำๆ และแก้ไขเพิ่มเติมได้ภายหลัง
- อินเตอร์แอคทีฟโหมด (interactive mode) เป็นการเขียนชุดคำสั่งทีละคำสั่ง แล้วรันทันทีที่ละคำสั่งด้วยมือเราเอง ไม่มีการบันทึกเป็นไฟล์ ข้อดีคือ เราสามารถตรวจสอบผลลัพธ์ได้ที่ละ block

Flowchart คือ

แผนภาพที่ใช้ สำหรับเรียงลำดับขั้นตอนและวิธีการทำงานของโปรแกรม

โดยการทำงานจะเริ่มจากจุดเริ่มต้น (Start) ไปยัง จุดสิ้นสุด (Stop) ของโปรแกรม



Symbols



จุดเริ่มต้น (Start) , จุดสิ้นสุด (Stop)

ใช้เป็นจุดเริ่มต้นและจุดสิ้นสุดของโปรแกรม



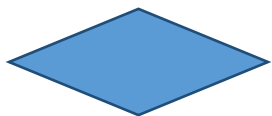
การแสดงผล (Output) , การรับค่า (Input)

ใช้เป็นสัญลักษณ์สำหรับการแสดงผลและการรับค่า



การประมวลผล (Process)

ใช้เป็นสัญลักษณ์สำหรับการประมวลผลข้อมูล



การตัดสินใจ (Decision)

ใช้เป็นสัญลักษณ์สำหรับการตัดสินใจ



จุดเชื่อมต่อ (Node)

ใช้เป็นสัญลักษณ์สำหรับเชื่อมต่อการทำงานต่าง ๆ

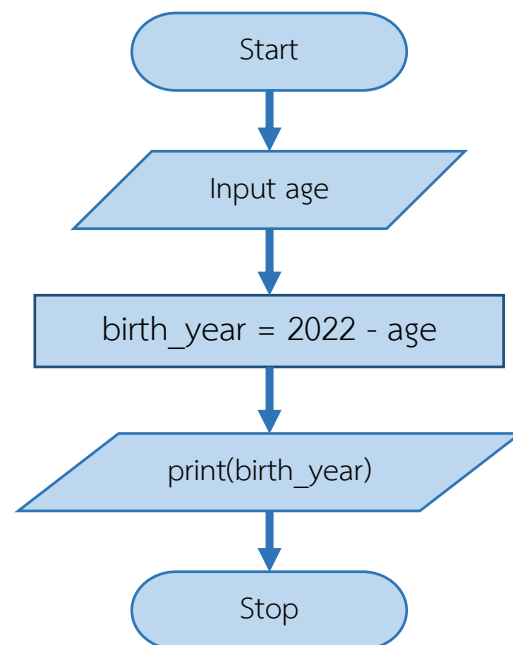


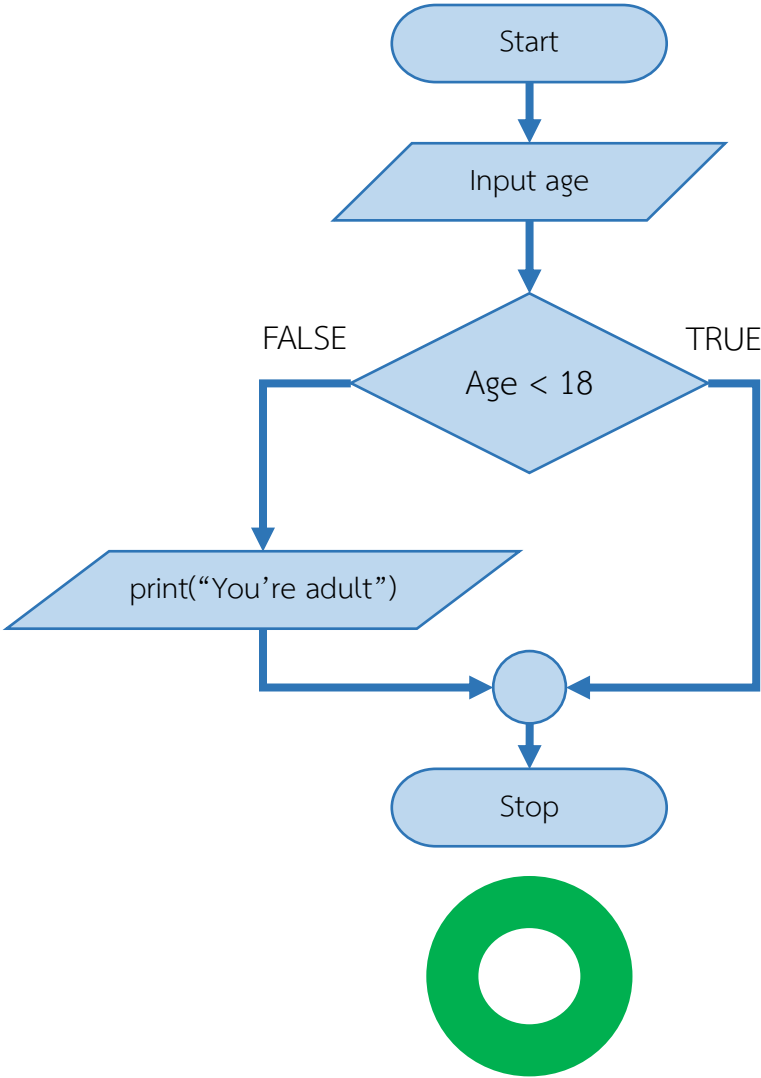
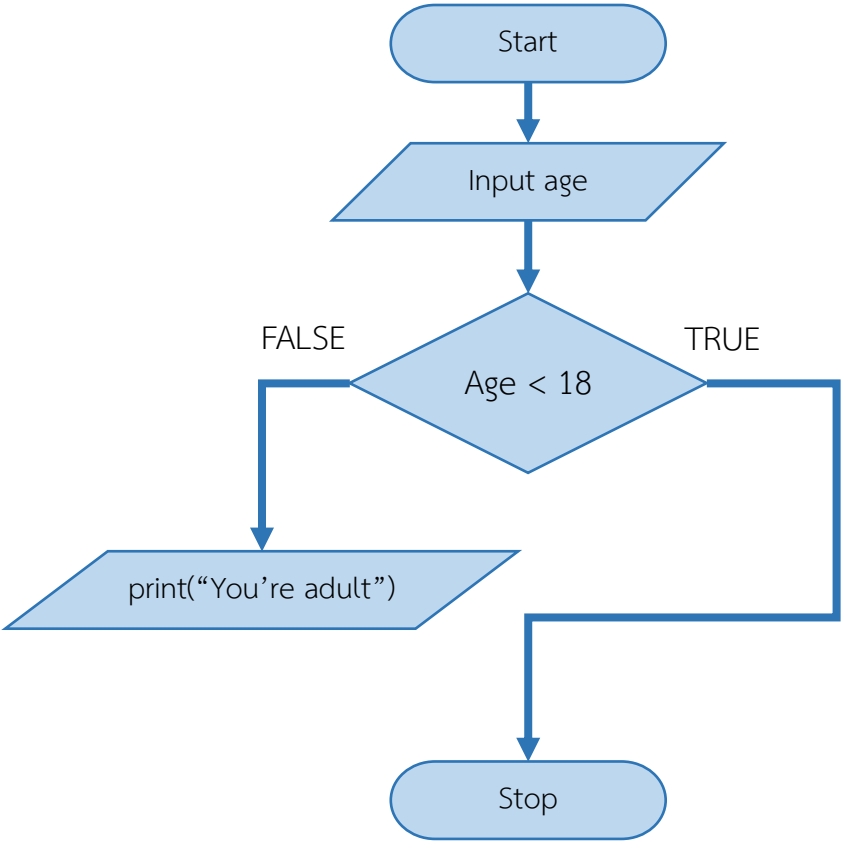
เส้นเชื่อม (Flow Line)

ใช้เป็นเส้นเชื่อมลำดับการทำงาน

หลักการเขียน Flowchart

- ต้องมี จุดเริ่มต้น (Start) และ จุดสิ้นสุด (Stop) เสมอ
- แต่ละสัญลักษณ์เชื่อมต่อกันด้วย เส้นเชื่อม (Flow Line) เสมอ เพื่อบอกทิศทางการทำงาน ไปยังคำสั่งต่อไป
- เมื่อทำการ ไล่ Flowchart ต้องเริ่มจากจุดเริ่มต้น (Start) และ สามารถไปยังจุดสิ้นสุด (Stop) ได้เสมอ

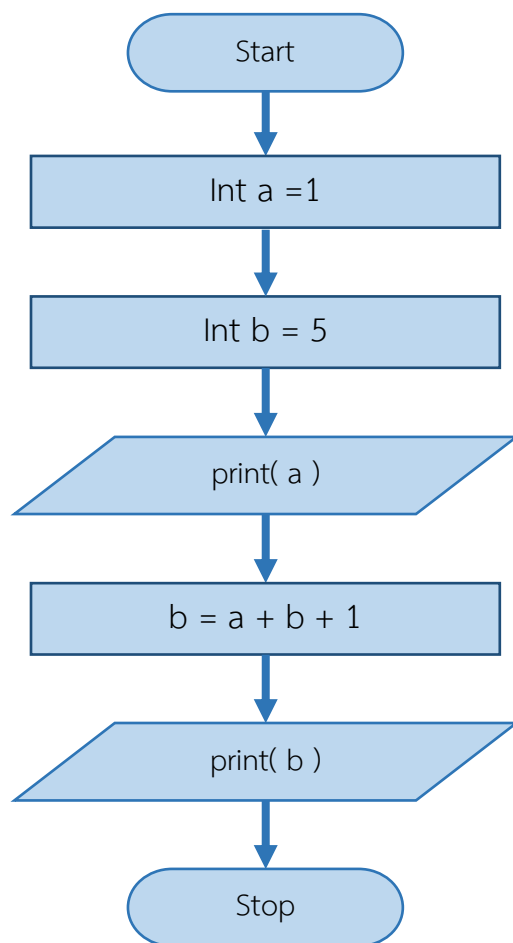




Control Statement

- โครงสร้างควบคุม การทำงานแบบ **เป็นลำดับ** (Sequence Control Statement)
- โครงสร้างควบคุม การทำงานแบบ **ทางเลือก** (Selection Control Statement)
- โครงสร้างควบคุม การทำงานแบบ **วนซ้ำ** (Repetition Control Statement)

Sequence Control Statement



1. Start เริ่มต้นโปรแกรม
2. ประกาศตัวแปร ชื่อ a เป็น ประเภทจำนวนเต็ม (int) และ กำหนดค่าเริ่มต้นเป็น 1
3. ประกาศตัวแปร ชื่อ b เป็น ประเภทจำนวนเต็ม (int) และ กำหนดค่าเริ่มต้นเป็น 5
4. แสดงผล ข้อมูลที่เก็บไว้ใน ตัวแปร a
5. คำนวณ ค่าในตัวแปร a + ค่าในตัวแปร b + 1 แล้วเก็บผลลัพธ์ลงใน ตัวแปร b
6. แสดงผล ข้อมูลที่เก็บไว้ใน ตัวแปร b
7. Stop จบการทำงานของโปรแกรม

Exercises ให้เขียน flowchart ของโปรแกรม

เขียนโปรแกรมสำหรับ แปลงอุณหภูมิ

กำหนดค่าข้อมูลอุณหภูมิ (องศาเซลเซียส)

คำนวณหาค่า อุณหภูมิ (องศาฟาเรนไฮต์)

คำนวณหาค่า อุณหภูมิ (เคลวิน)

ด้วยสมการ

$$F = 9/5 * C + 32$$

$$K = C + 273.15$$

กำหนดให้

C คือ องศาเซลเซียส

F คือ องศาฟาเรนไฮต์

K คือ เคลวิน

ตัวอย่าง

กำหนด

C = 39.85 # อุณหภูมิหน่วยองศาเซลเซียส

จะได้ Output คือ

103.73 F

313.0 K

Output บรรทัดแรกคือค่า F บรรทัดที่สองคือค่า K

เฉลี่ย

เขียนโปรแกรมสำหรับ แปลงอุณหภูมิ

กำหนดค่าข้อมูลอุณหภูมิ (องศาเซลเซียส)
 คำนวณหาค่า อุณหภูมิ (องศาฟาเรนไฮต์)
 คำนวณหาค่า อุณหภูมิ (เคลวิน)

ด้วยสมการ

$$F = 9/5 * C + 32$$

$$K = C + 273.15$$

กำหนดให้

C คือ องศาเซลเซียส

F คือ องศาฟาเรนไฮต์

K คือ เคลวิน

ตัวอย่าง

กำหนด

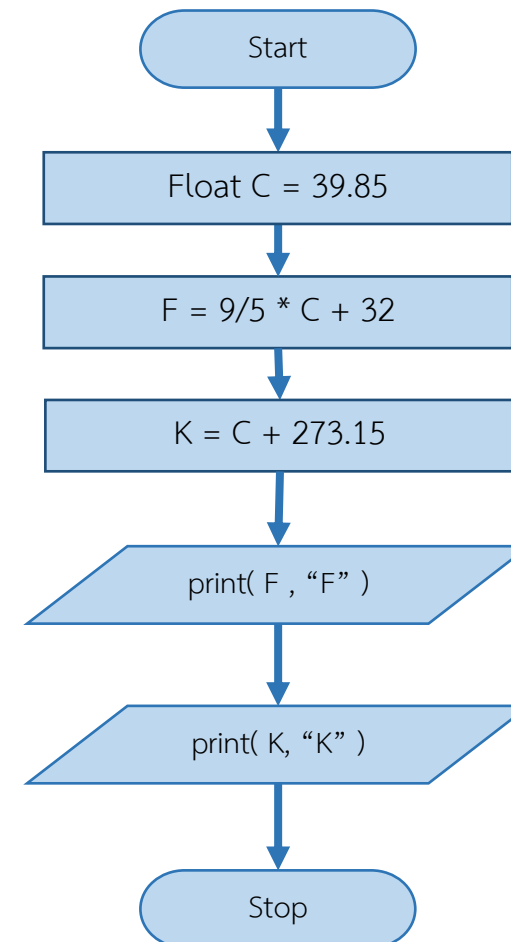
C = 39.85 # อุณหภูมิหน่วยองศาเซลเซียส

จะได้ Output คือ

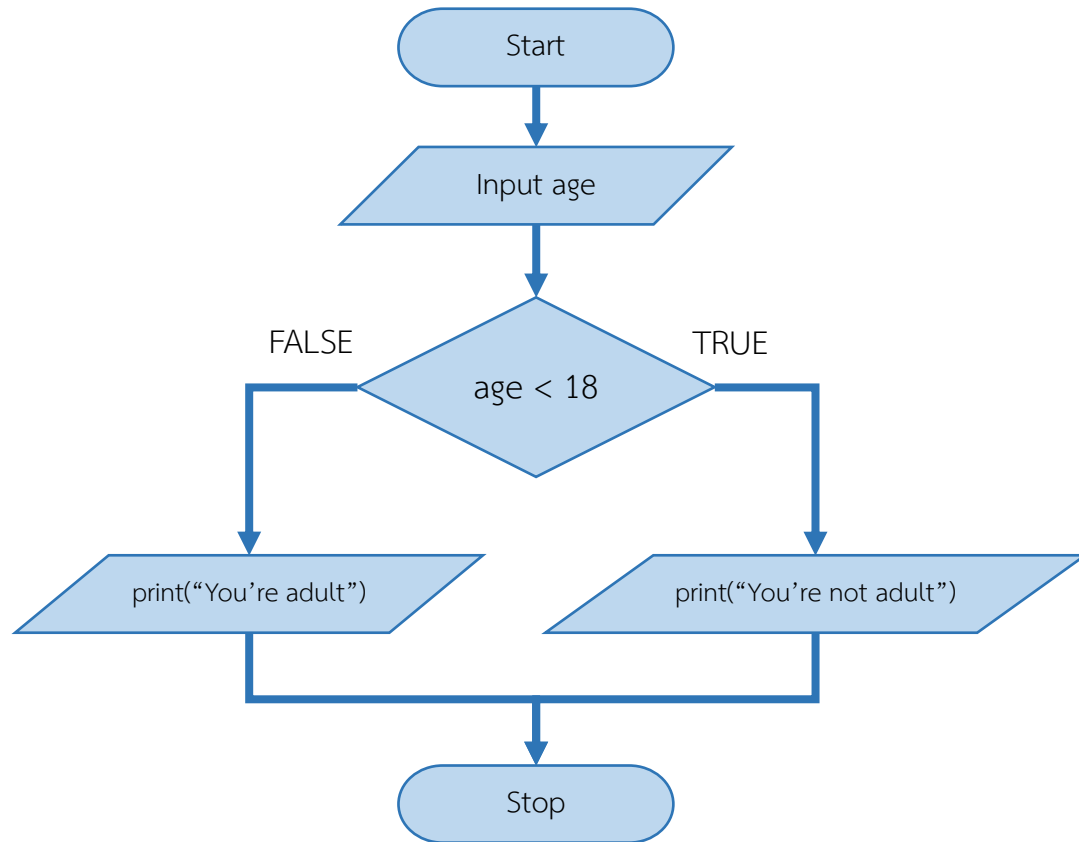
103.73 F

313.0 K

Output บรรทัดแรกคือค่า F บรรทัดที่สองคือค่า K



Selection Control Statement



หลักการทำงาน

ถ้าเงื่อนไขเป็นจริง (True) จะไปทำงานทาง ฝั่งจริง (True)
แต่ถ้าเงื่อนไขเป็นเท็จ (False) ก็จะไปทำงานทางเท็จ (False)

จากภาพ

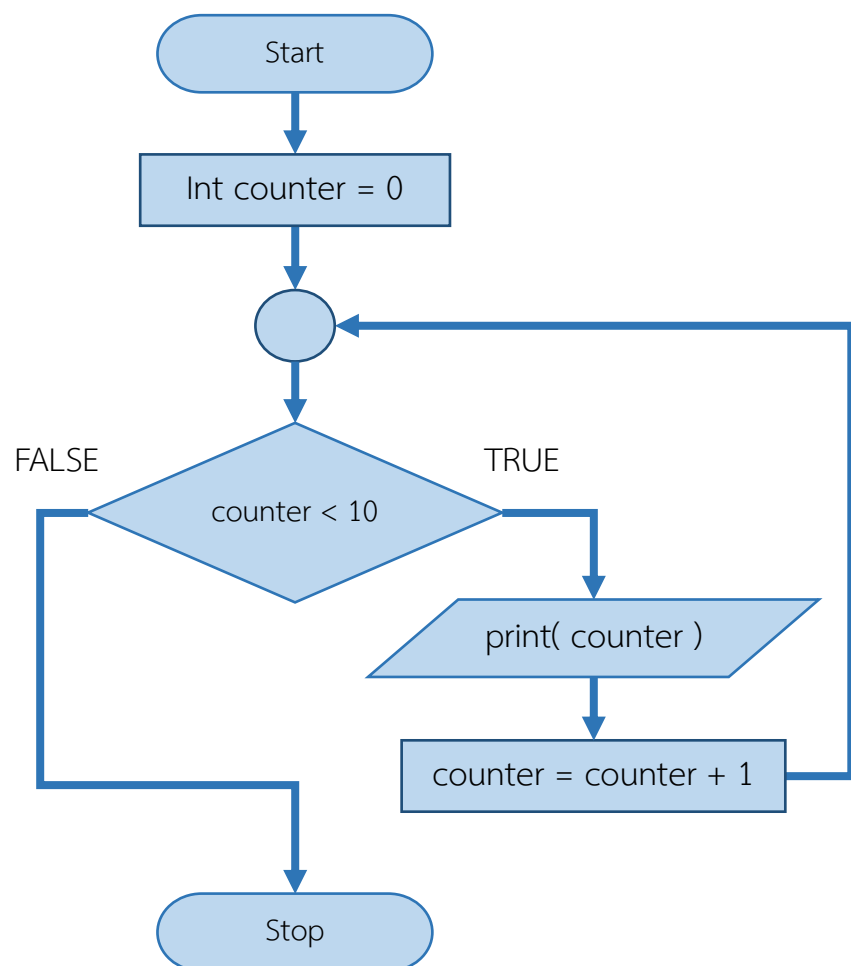
ถ้า age มีค่าน้อยกว่า 18

จะแสดงผลฝั่ง TRUE คือ “You’re not adult”

แต่ ถ้า age มีค่าตั้งแต่ 18 ขึ้นไป

จะแสดงผลฝั่ง FALSE คือ “You’re adult”

Repetition Control Statement



หลักการทำงาน

ทำงานวนซ้ำกระบวนจนกว่าเงื่อนไขจะเป็นเท็จ (False)

จากภาพ

ประกาศตัวแปร *counter* เป็นจำนวนจริง มีค่าเริ่มต้นเท่ากับ 0
ถ้า *counter* มีค่าน้อยกว่า 10

จะแสดงผลลัพธ์ คือ ค่าในตัวแปร *counter* (ฝั่ง TRUE)

แล้วนำค่าในตัวแปร *counter* + 1 เก็บผลลัพธ์ในตัวแปร

counter

แล้วทำซ้ำขั้นตอนการตรวจสอบค่า *counter*

แต่ ถ้า *counter* มีค่าตั้งแต่ 10 ขึ้นไป

จะจบการทำงาน

Basic Mathematical Logic

ประพจน์ (Proposition) คือ ประโยคหรือข้อความ ที่สามารถบอกค่าความจริงว่าเป็นจริงหรือเท็จอย่างใดอย่างหนึ่งเท่านั้น

เช่น นาย A มีอายุ 20 ปี

ประพจน์นี้จะเป็นจริง ก็ต่อเมื่อนาย A มีอายุเท่ากับ 20 ปีเท่านั้น

ประพจน์นี้จะเป็นเท็จ ก็ต่อเมื่อนาย A มีอายุไม่เท่ากับ 20 ปี (มากกว่าหรือน้อยกว่า 20)

การเชื่อมประพจน์ จะกล่าวถึง 2 ตัวเชื่อม คือ “And” (และ), “Or” (หรือ)

AND Connection

เช่น A มีอายุ 20 ปี **และ** มีอาชีพโปรแกรมเมอร์

มี 2 ประพจน์ย่อย คือ

1. A มีอายุเท่ากับ 20 ปี

2. A เป็นโปรแกรมเมอร์

ประพจน์นี้จะเป็นจริง ก็ต่อเมื่อ ประพจน์ย่อยทั้งหมดเป็นจริง

ประพจน์นี้จะเป็นเท็จ ก็ต่อเมื่อ ประพจน์ใดประพจน์หนึ่งเป็นเท็จ

OR Connection

เช่น A มีอายุ 20 ปี **หรือ** มีอาชีพโปรแกรมเมอร์

มี 2 ประพจน์ย่อย คือ

1. A มีอายุเท่ากับ 20 ปี

2. A เป็นโปรแกรมเมอร์

ประพจน์นี้จะเป็นจริง ก็ต่อเมื่อ ประพจน์ใดประพจน์หนึ่งเป็นจริง

ประพจน์นี้จะเป็นเท็จ ก็ต่อเมื่อ ประพจน์ย่อยทั้งหมดเป็นเท็จ

Negation (นิเสธ)

นิเสธ คือ การเปลี่ยนค่าความจริงเป็นค่าตรงกันข้าม

เช่น นาย A มีอายุ 20 ปี

ถ้าประพจน์นี้เป็นจริง

นิเสธของประพจน์นี้จะเป็นเท็จ

ถ้าประพจน์นี้เป็นเท็จ

นิเสธของประพจน์นี้จะเป็นจริง

Expression (นิพจน์)

นิพจน์หมายถึงการนำตัวแปร ค่าคงที่มากระทำต่อกันโดยใช้ ตัวดำเนินการคำนวณ ตัวดำเนินการสัมพันธ์ หรือตัวดำเนินการตรรกะ มี 3 ชนิด คือ

- นิพจน์คำนวณ (Arithmetic Expressions)
- นิพจน์เชิงสัมพันธ์ (Relational Expressions)
- นิพจน์ตรรกะ (Logical Expressions)

ในการเขียนโปรแกรม เราจะใช้ *นิพจน์เชิงสัมพันธ์* และ *นิพจน์ตรรกะ* เป็นตัวกำหนดค่าความจริงแทน ประพจน์

Arithmetic Expressions (นิพจน์คำนวณ)

นิพจน์คำนวณเป็นนิพจน์ที่ใช้ตัวดำเนินการดำเนินการคณิตศาสตร์เป็นตัวกระทำ
ใน Python มีตัวดำเนินการพื้นฐาน คือ

- การบวก (+)
- การลบ (-)
- การคูณ (*)
- การยกกำลัง (**)
- การหาร (/)
- การหารพิเศษ (//)
- การหาเศษจากการหาร (%)

Relational Expressions (นิพจน์เชิงสัมพันธ์)

เป็นนิพจน์ที่ใช้ตัวดำเนินการเป็นตัวดำเนินการสัมพันธ์

ใน Python มีตัวดำเนินการพื้นฐาน คือ

- น้อยกว่า (<)
- มากกว่า (>)
- เท่ากับ (==)
- น้อยกว่าหรือเท่ากับ (<=)
- มากกว่าหรือเท่ากับ (>=)
- ไม่เท่ากับ (!=)

Logical Expressions (นิพจน์ตรรกะ)

เป็นนิพจน์ที่ใช้ตัวดำเนินการเป็นตัวดำเนินการตรรกะ

ใน Python มีตัวดำเนินการพื้นฐาน คือ

- and (และ)
- or (หรือ)
- ! (นิเสธ หรือ not)