

# พื้นฐานการเขียนโปรแกรม ภาษาไพรอน

บทที่ 1 บทนำสู่การเขียนโปรแกรมโดยใช้ภาษาไพรอน

# วัตถุประสงค์ของบทเรียน

เมื่อสิ้นสุดบทเรียน ผู้เรียนจะสามารถ

- อธิบายถึงองค์ประกอบพื้นฐานของคอมพิวเตอร์
- อธิบายถึงหลักการเขียนโปรแกรมเบื้องต้นได้
- อธิบายถึงประวัติของโปรแกรมภาษาโปรแกรม

# อะไรคือเครื่องคอมพิวเตอร์

*A calculating-machine; esp. an automatic electronic device for performing mathematical or logical operations; freq. with defining word prefixed, as analogue, digital, electronic computer*

Oxford English Dictionary

*An electronic machine that is used for storing, organizing, and finding words, numbers, and pictures, for doing calculations, and for controlling other machines*

Cambridge Dictionary

# อะไรคือเครื่องคอมพิวเตอร์

เครื่องคอมพิวเตอร์ คือ เครื่องมือที่ทำหน้าที่จัดการกับข้อมูล เช่น การจัดเก็บข้อมูล การค้นหาข้อมูล และการประมวลผลข้อมูล ตามบุคคลากรและทำการแสดงผลลัพธ์จากบุคคลากร

# โครงสร้างของเครื่องคอมพิวเตอร์

โครงสร้างของเครื่องคอมพิวเตอร์ประกอบด้วยองค์ประกอบดังต่อไปนี้

- อุปกรณ์นำเข้าข้อมูล (Input unit)
- อุปกรณ์แสดงผลข้อมูล (Output unit)
- หน่วยความจำ (Memory)
- หน่วยประมวลผลกลาง (Central processing unit)
- หน่วยเก็บข้อมูลสำรอง (Secondary storage unit)

# อะไรคือซอฟต์แวร์ (Software)

- ซอฟต์แวร์ คือ ชุดคำสั่งที่เขียนขึ้นมา เพื่อใช้ในการสั่งการทำงานให้กับคอมพิวเตอร์ ให้ดำเนินการให้บรรลุตามวัตถุประสงค์ของงาน
- ซอฟต์แวร์มีกึ่งหมวด 2 ประเภท
  - ซอฟต์แวร์สำหรับงานทั่วไป (Application software)
  - ซอฟต์แวร์ระบบ (System software)

# การเขียนชุดคำสั่ง (Programming)

## คำศัพท์ที่ควรทราบ

- ชุดคำสั่ง โปรแกรม (Program) ซอฟต์แวร์ (Software) โปรแกรมประยุกต์ (Application)
- การเขียนชุดคำสั่ง การเขียนโปรแกรม (Programming) โค้ดดิ้ง (Coding)
- โปรแกรมเมอร์ (Programmer) นักพัฒนาซอฟต์แวร์ (Software Developer)
- รหัสต้นฉบับ (Source Code) โค้ด (Code)

# การเขียนชุดคำสั่ง (Programming)

ปัญหา / งาน



โปรแกรม

- การเขียนชุดคำสั่ง คือ การออกแบบ และการพัฒนา ขั้นตอนการ ทำงานที่เครื่องคอมพิวเตอร์เข้าใจ และดำเนินการเพื่อให้เกิดผลลัพธ์ที่ ต้องการได้

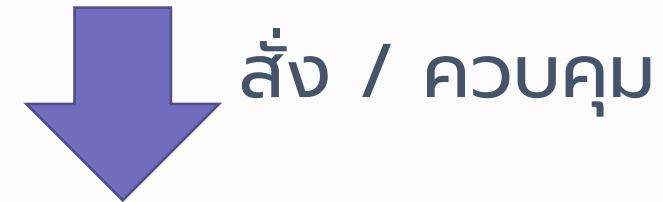
# การเขียนชุดคำสั่ง (Programming)

ปัญหา / งาน



โปรแกรม

การรัน (Run) คือ การสั่งให้คอมพิวเตอร์ทำงานตามโปรแกรม  
(คำสั่ง) ที่ถูกเรียกใช้

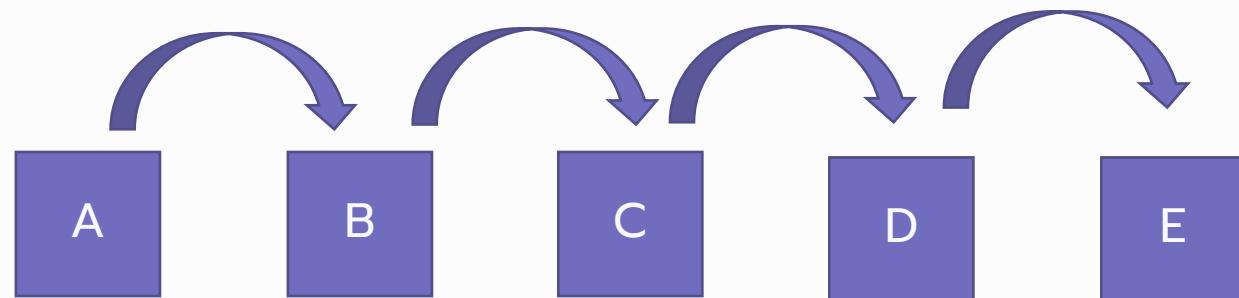


คอมพิวเตอร์

# การเขียนชุดคำสั่ง (Programming)

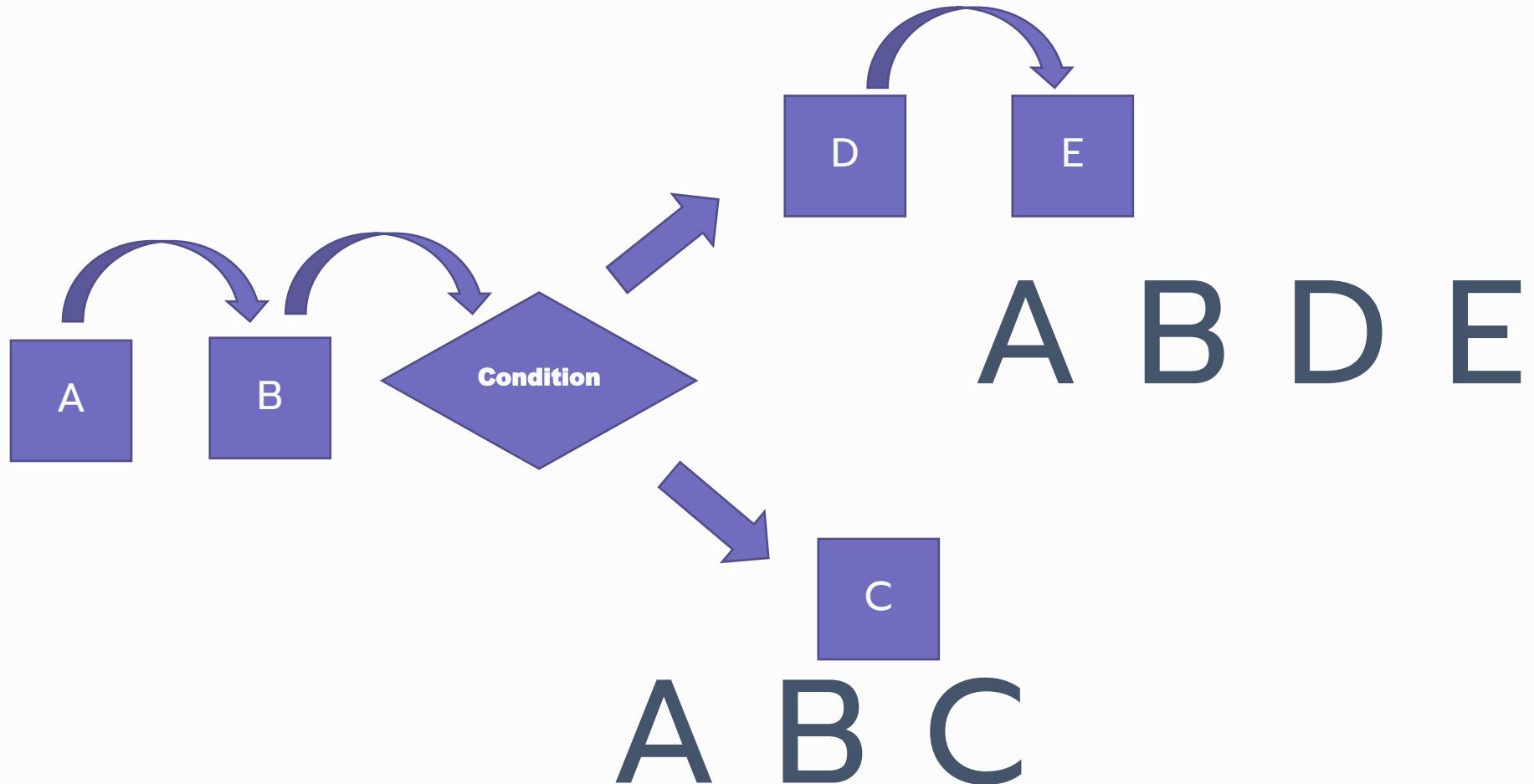
- **โครงสร้างเชิงเส้น (Sequential structure)**
  - ดำเนินการคำสั่งตามลำดับ ไม่มีการข้ามขั้นตอน
- **โครงสร้างทางเลือก (Selection structure)**
  - เลือกเส้นทางระหว่าง 2 เส้นทาง โดยใช้เงื่อนไขเป็นตัวตัดสินใจ
- **โครงสร้างการทำซ้ำ (Repetitive structure)**
  - ทำซ้ำชุดของคำสั่ง โดยใช้เงื่อนไขเป็นตัวตัดสินใจ

# โครงสร้างเชิงเส้น (Sequential structure)

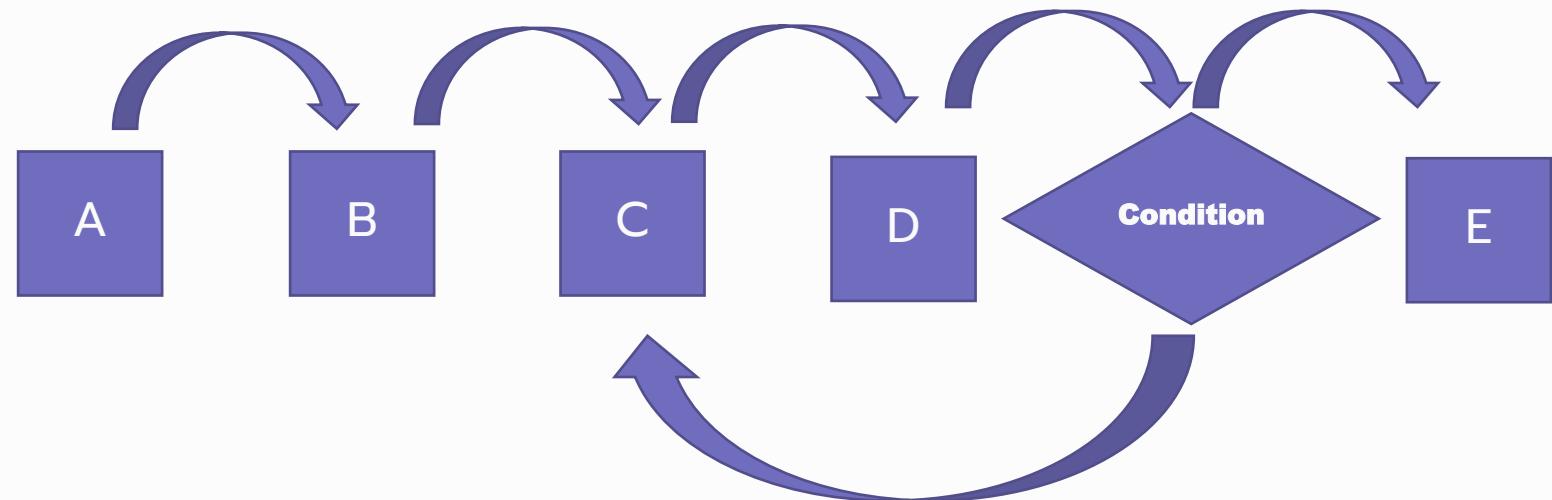


A B C D E

# โครงสร้างทางเลือก (Selection structure)



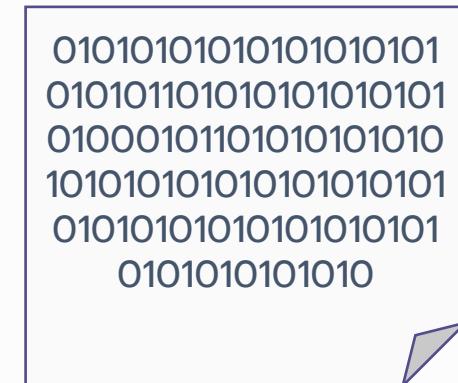
# โครงสร้างการทำซ้ำ (Repetitive structure)



A B C D C D ... C D E

# ภาษาคอมพิวเตอร์

คอมพิวเตอร์เข้าใจ / ดำเนินงานตาม



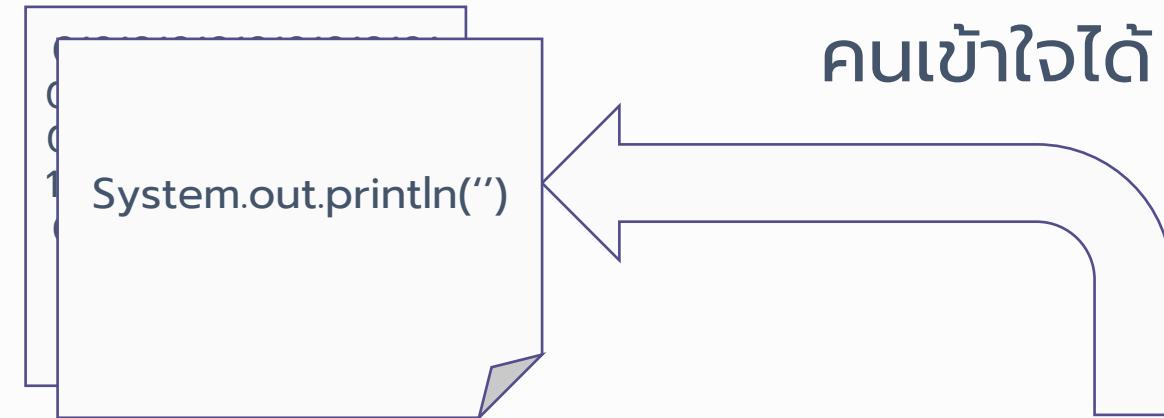
ชุดคำสั่ง  
(ภาษาเครื่อง)

คนไม่เข้าใจ



# ภาษาคอมพิวเตอร์

คอมพิวเตอร์ไม่เข้าใจ



บุดคำสั่ง  
(ภาษาระดับสูง)



# ภาษาคอมพิวเตอร์

คอมพิวเตอร์เข้าใจ



ชุดคำสั่ง  
(ภาษาระดับสูง)

ชุดคำสั่ง  
(ภาษาเครื่อง)

```
010101010101010101  
0101011010101010101  
010001011010101010  
1010101010101010101  
010101010101010101  
0101010101010101010
```

ตัว  
แปลภาษา

```
System.out.println("")
```

คนเข้าใจได้



# ตัวแปลงภาษา

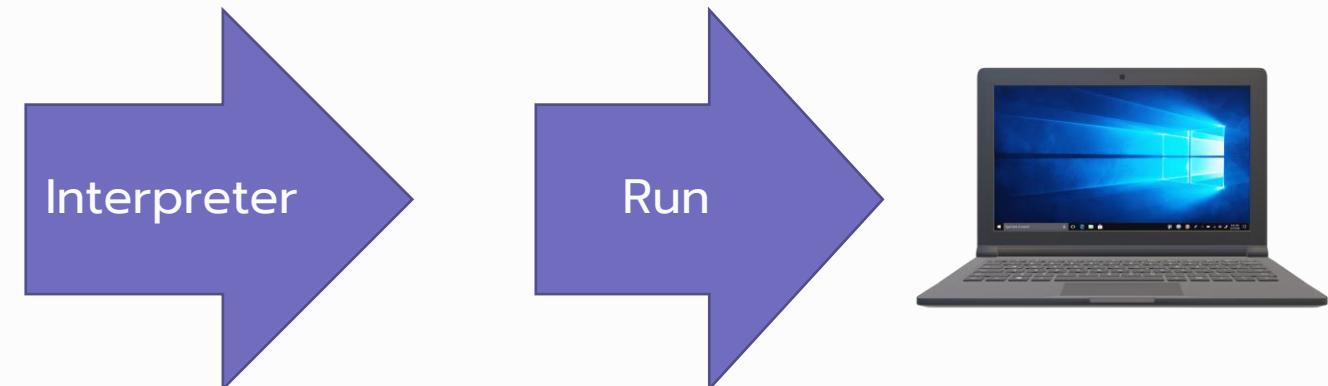
## 1. Complier

`System.out.println("")`  
e.g. C, C++, C#,  
Scala, Java



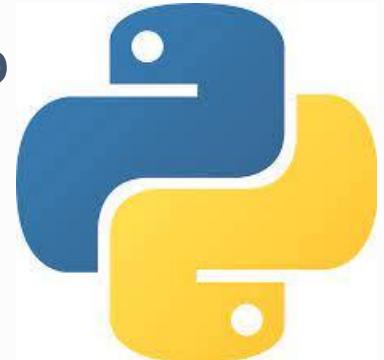
## 2. Interpreter

`print("")`  
e.g. PHP, Perl,  
Ruby, Python



# ภาษาโปรแกรม (Python)

- ภาษาโปรแกรมเป็นภาษาการเขียนโปรแกรมระดับสูง (High-level programming language)
- ภาษาโปรแกรมเป็นภาษาที่ใช้ในการเขียนโปรแกรมที่ได้รับความนิยมสูง
  - ชุดรหัสแบบเปิด (Open source)
  - สามารถใช้งานได้ง่าย (User friendly)
  - ไม่ขึ้นกับระบบปฏิบัติการใด (Platform independent)
- รหัสต้นฉบับของภาษาโปรแกรมจะมีนามสกุล py
- ตัวอย่างการใช้งาน ภาษาโปรแกรม เช่น การวิเคราะห์ข้อมูล (Data analytics) การทำเว็บแอปพลิเคชัน

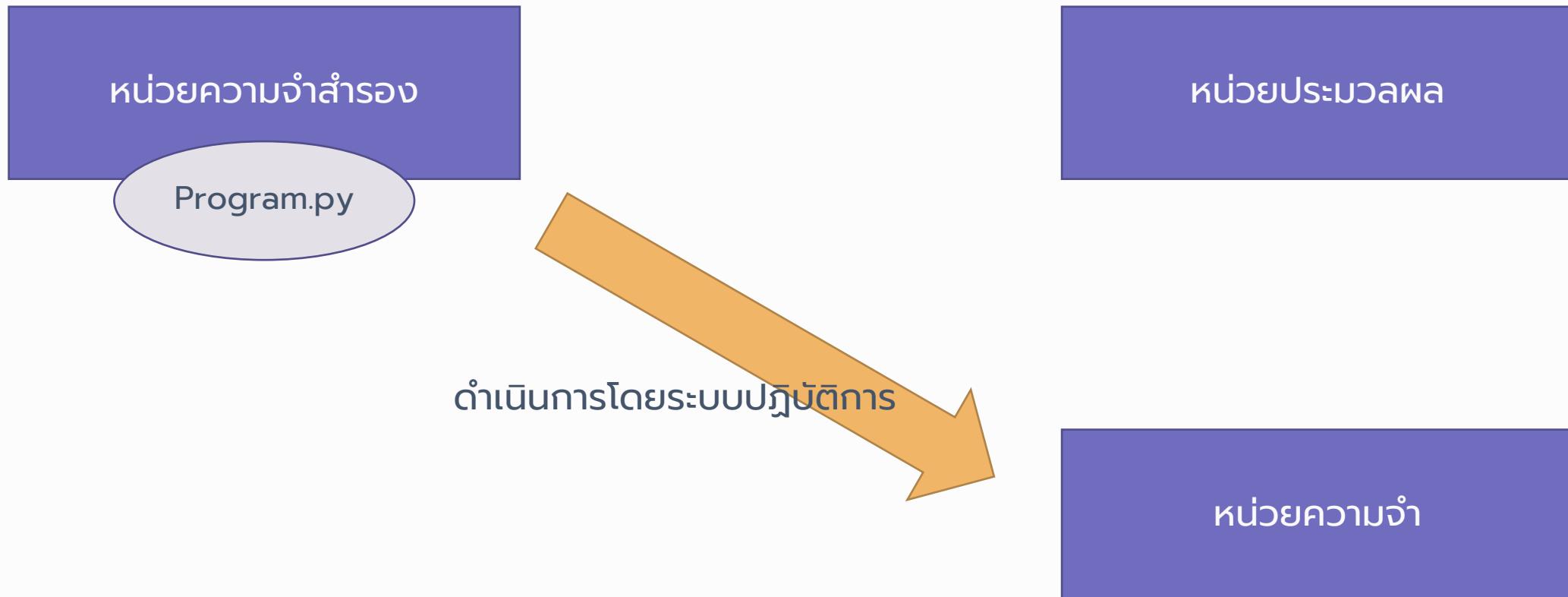


# ประวัติของภาษาไพธอน (Python)

- ภาษาไพธอนถูกพัฒนาใน ปี พ.ศ. 2532 (ค.ศ. 1989) โดย Guido van Rossum ซึ่งทำงานอยู่ที่ Research Institute for Mathematics and Computer Science (CWI)
- ภาษาไพธอน เป็นการพัฒนาข้อดีของภาษาโปรแกรมหลายภาษาเข้าด้วยกัน เช่น All Basic Code (ABC) และ Modula-3
- ภาษาไพธอนถูกเผยแพร่ครั้งแรกในปี พ.ศ. 2534 (ค.ศ. 1991)

# การทำงานของภาษาโปรแกรม

1. ระบบปฏิบัติการทำการย้ายโปรแกรมจากหน่วยความจำสำรองมายังหน่วยความจำ



# การทำงานของภาษาโปรแกรม

2. ระบบปฏิบัติการทำการย้ายโปรแกรมจากหน่วยความจำสำรองมายังหน่วยความจำ

หน่วยความจำสำรอง

หน่วยประมวลผล

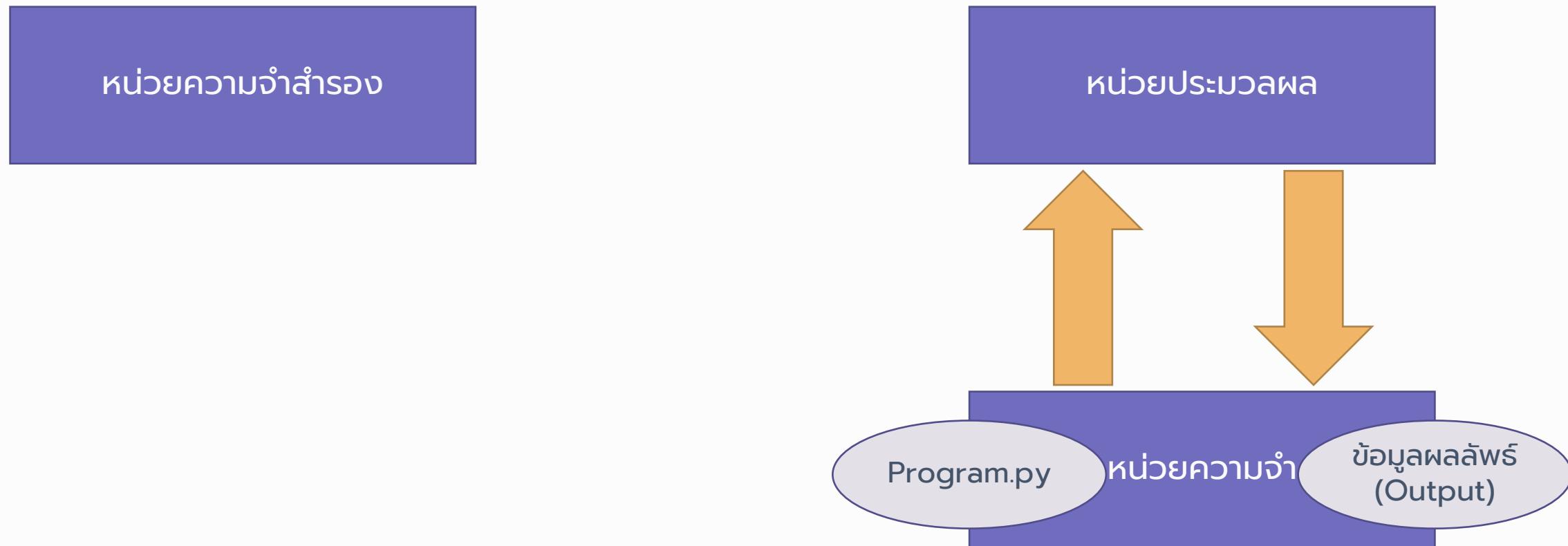
Program.py

หน่วยความจำ

ข้อมูลนำเข้า  
(Input)

# การทำงานของภาษาโปรแกรม

3. หน่วยประมวลผลกลางทำการประมวลผลโปรแกรมและข้อมูลบนหน่วยความจำ จนได้ผลลัพธ์



# พื้นฐานการเขียนโปรแกรม ภาษา Python

บทที่ 2 การใช้ข้อมูล และตัวแปร

# วัตถุประสงค์ของบทเรียน

เมื่อสิ้นสุดบทเรียน ผู้เรียนจะสามารถ

- อธิบายถึงหลักการของข้อมูล และตัวแปรได้
- ใช้ข้อมูล และตัวแปรในการเขียนโปรแกรมได้

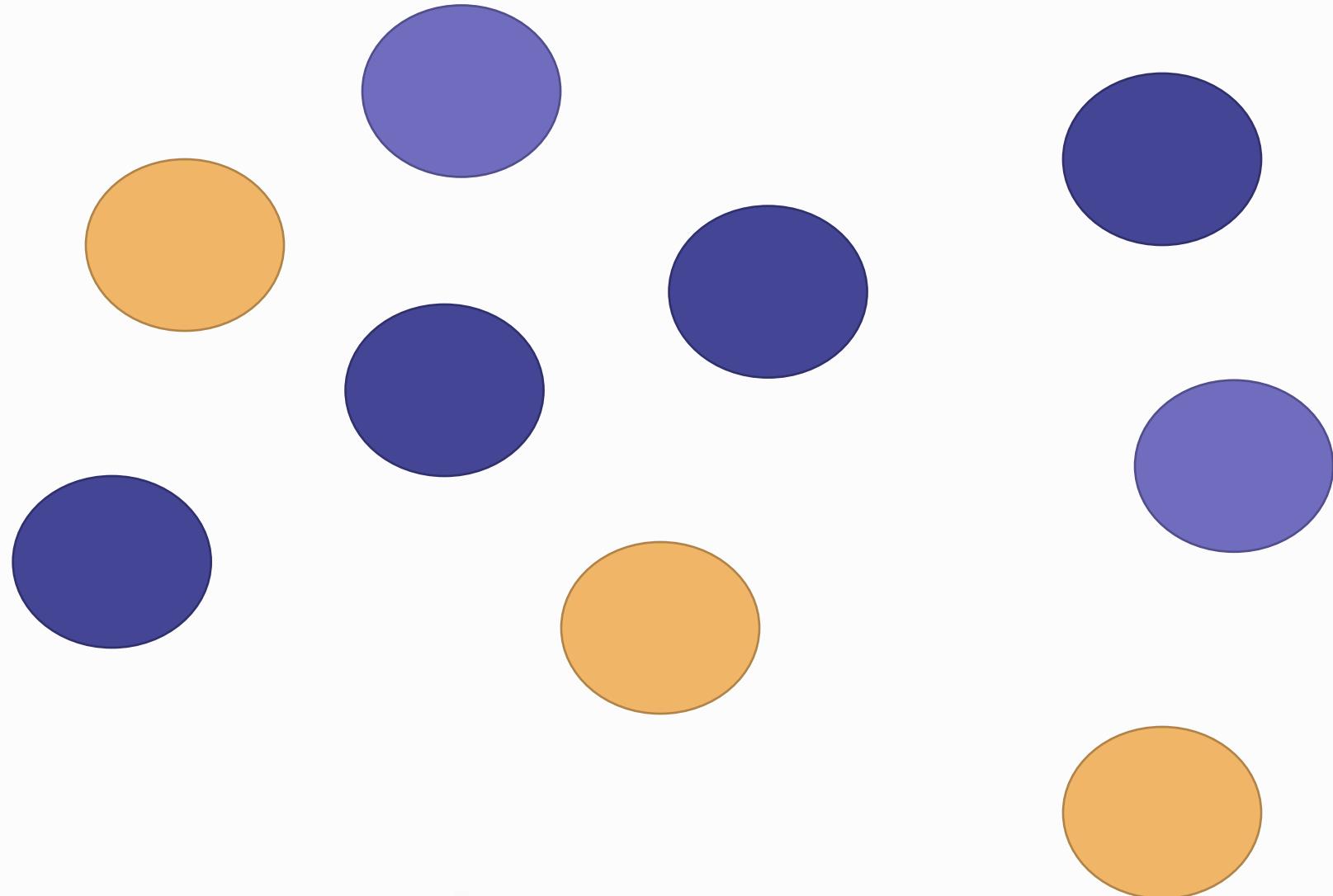
# ข้อมูล (Data)

- นิยาม

ข้อเท็จจริง ที่เกิดจากการวัด สังเกต หรือ ประเมินผลทางสถิติ ที่ถูกใช้  
เพื่อการอนุมาน dakdeying หรือ การคำนวณ

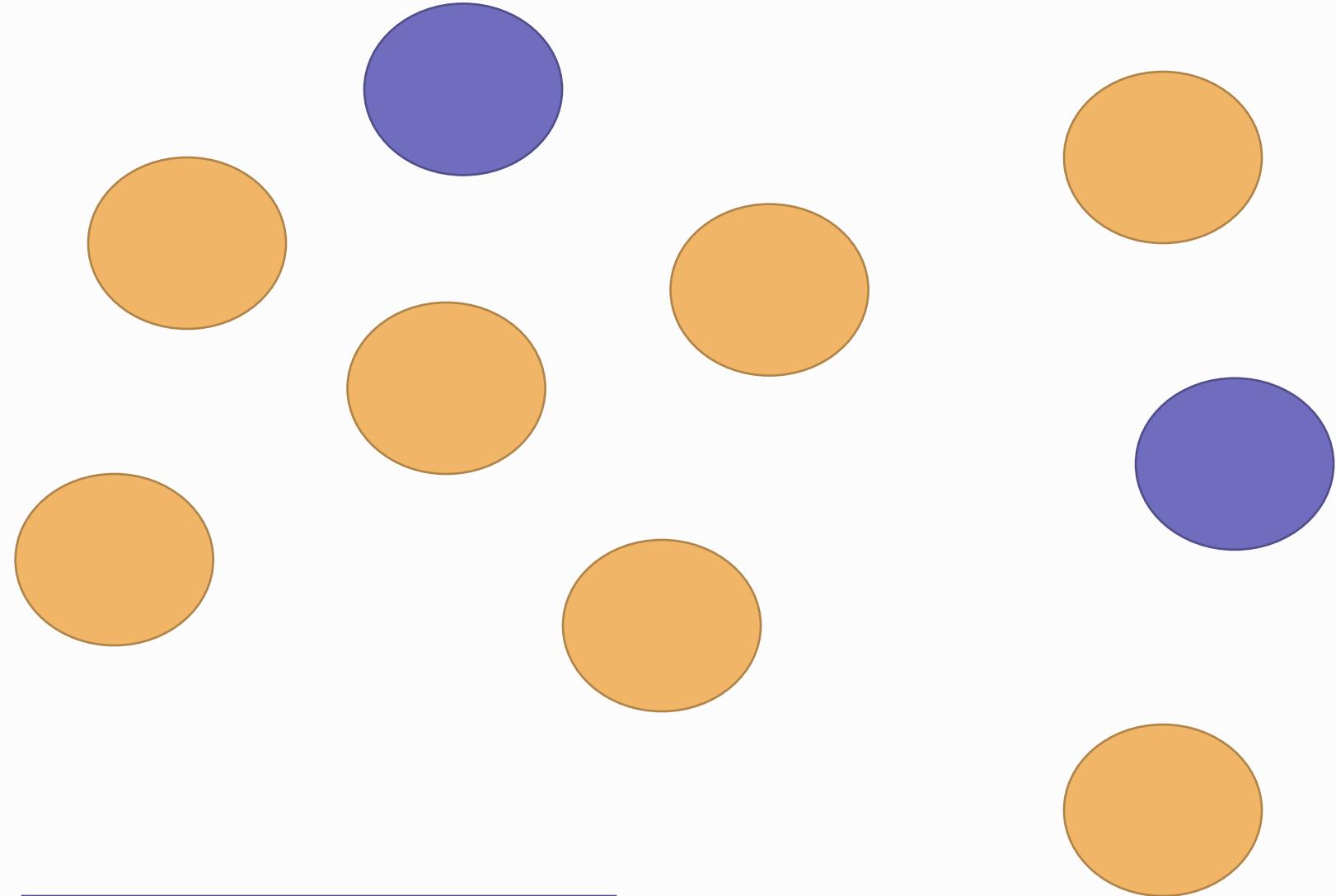
เช่น จำนวนผู้เรียนในห้อง ส่วนสูงของตัวไม้ น้ำหนักของรถยนต์

ให้บับจำนวนจุดสี่เหลี่องจาก  
ภาพต่อไปนี้



กสณที่ 1

จำนวนจุดสี่เหลี่อง คือ 3



กสณที่ 2

จำนวนจุดสี่เหลี่อง คือ 7

# ข้อมูล

- ข้อมูลตัวเดียว กัน แต่ต่างสถานการณ์อาจจะมีค่าที่ไม่เท่ากัน
  - บางครั้งในโปรแกรมเดียว กัน ข้อมูลอาจจะมีการเปลี่ยนแปลงตลอดเวลา
- ข้อมูลเป็นสิ่งที่แปรเปลี่ยนได้
- ข้อมูลที่เปลี่ยนแปลงไม่ได้ จะ เรียกว่า ค่าคงที่ (Constant) เช่น
  - $\pi$
  - $e$
  - $g$

# ข้อมูล

- ภาษาโปรแกรมมีข้อมูลพื้นฐานที่เป็นค่าอยู่ 4 ชนิด
  - จำนวนเต็ม (Integer) เช่น 1, 100, 200
  - 浮点数 (Float) เช่น 3.14, 1.99, 100.5
  - ข้อความ (String) เช่น 'Hello Word', "Sawasdee"
  - ค่าความจริง (Boolean) เช่น True, False

# ข้อมูล

- การเลือกใช้ชนิดข้อมูลเพื่อเก็บข้อมูล

ชนิดข้อมูล	ข้อมูลที่เหมาะสม
จำนวนเต็ม (Integer)	<ul style="list-style-type: none"> <li>ข้อมูลที่เกิดจากการนับ</li> </ul>
浮点数 (Float)	<ul style="list-style-type: none"> <li>ข้อมูลที่เกิดจากการบวก ลบ 乖 และหาร</li> </ul>
ข้อความ (String)	<ul style="list-style-type: none"> <li>ข้อมูลที่ใช้ในการสื่อความหมาย และไม่ต้องการที่ทำการดำเนินการทางคณิตศาสตร์</li> </ul>
ค่าความจริง (Boolean)	<ul style="list-style-type: none"> <li>ข้อมูลที่เป็นผลลัพธ์ของการดำเนินการตรรกะ</li> <li>ข้อมูลที่ต้องการให้มีเพียง 2 ค่า</li> </ul>

# ข้อมูล

- การมีตัวที่ซึ้งไปกับข้อมูลจึงง่ายกว่า ที่จะอ้างถึงข้อมูลโดยตรง
  - ทำให้สามารถที่จะมีการเรียกใช้ เปลี่ยนแปลงค่าได้
- ตัวซึ่งถึงข้อมูล เรียกว่า ตัวแปร (Variable)

แก้วกาแฟ = ตัวแปร

น้ำในแก้ว = ข้อมูล



แก้วกาแฟเดียวกัน แต่น้ำต่างกัน

# ตัวแปร

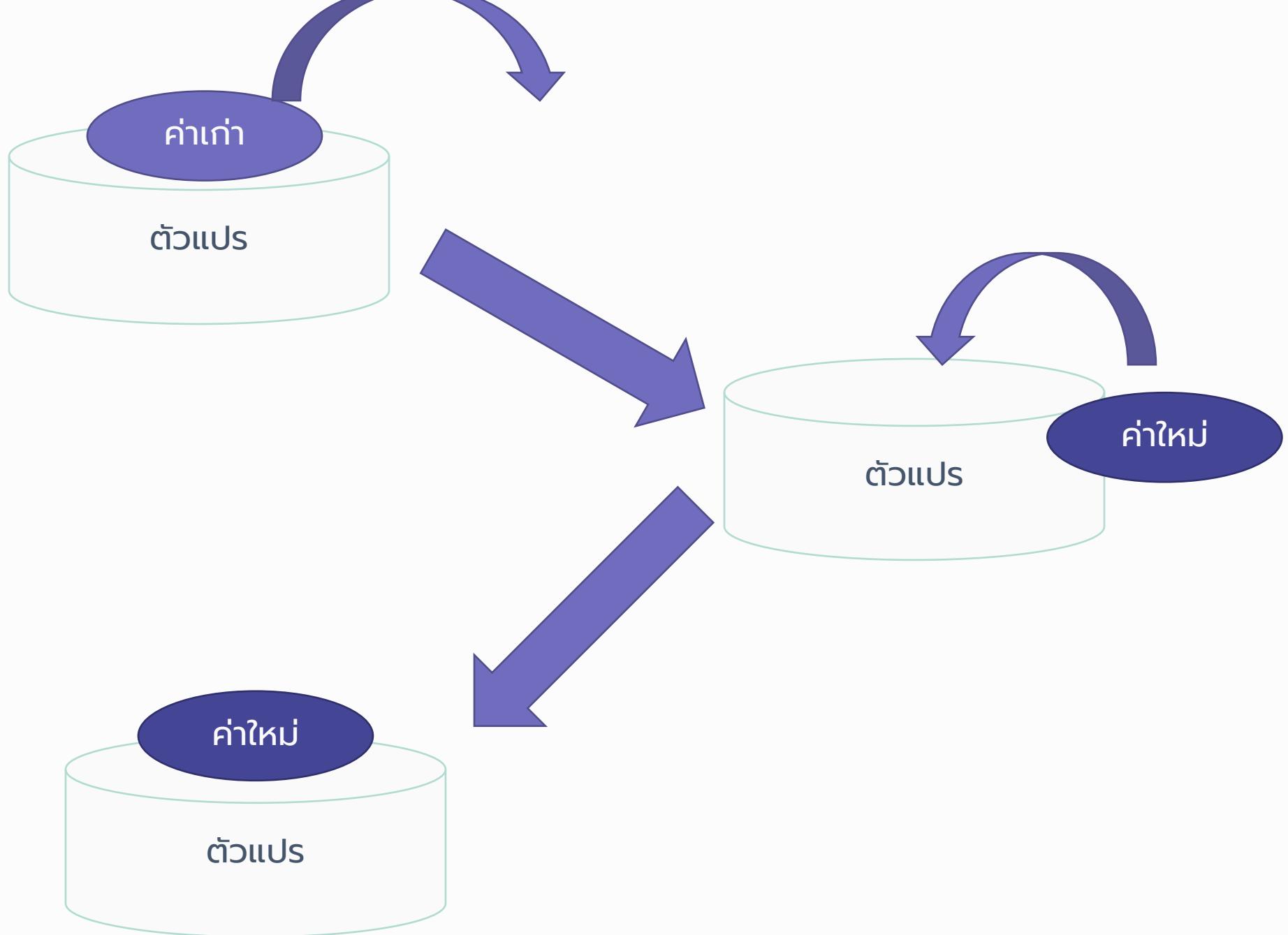
- ตัวแปร คือ พื้นที่ในหน่วยความจำ ที่มีการกำหนดชื่อ ทำให้สามารถ อ้างอิง และ เปลี่ยนแปลงข้อมูลได้
- กิจกรรมที่สามารถใช้งานกับตัวแปรได้คือ
  - สามารถใช้ตัวแปรเพื่อเข้าถึงข้อมูลได้
  - สามารถกำหนดข้อมูลให้ตัวแปร

# การอ้างอิงถึงข้อมูล

- การอ้างอิงถึงข้อมูล
  - เป็นกระบวนการเข้าถึงค่าในหน่วยความจำที่ตัวแปรซึ่งถึงค่าที่อ้างอิงจะสามารถถูกใช้เพื่อการแสดงผล และการคำนวณได้

# การกำหนดค่าให้ตัวแปร (Assignment)

- การกำหนดค่าให้ตัวแปร
  - เป็น กระบวนการกำหนดค่าใหม่ให้ตัวแปร
  - ค่าเก่าจะไม่สามารถถูกเข้าถึงได้อีก



# บทนำสู่การเขียนโปรแกรม ภาษา Python

บทที่ 4 การเขียนโปรแกรมเชิงเส้น

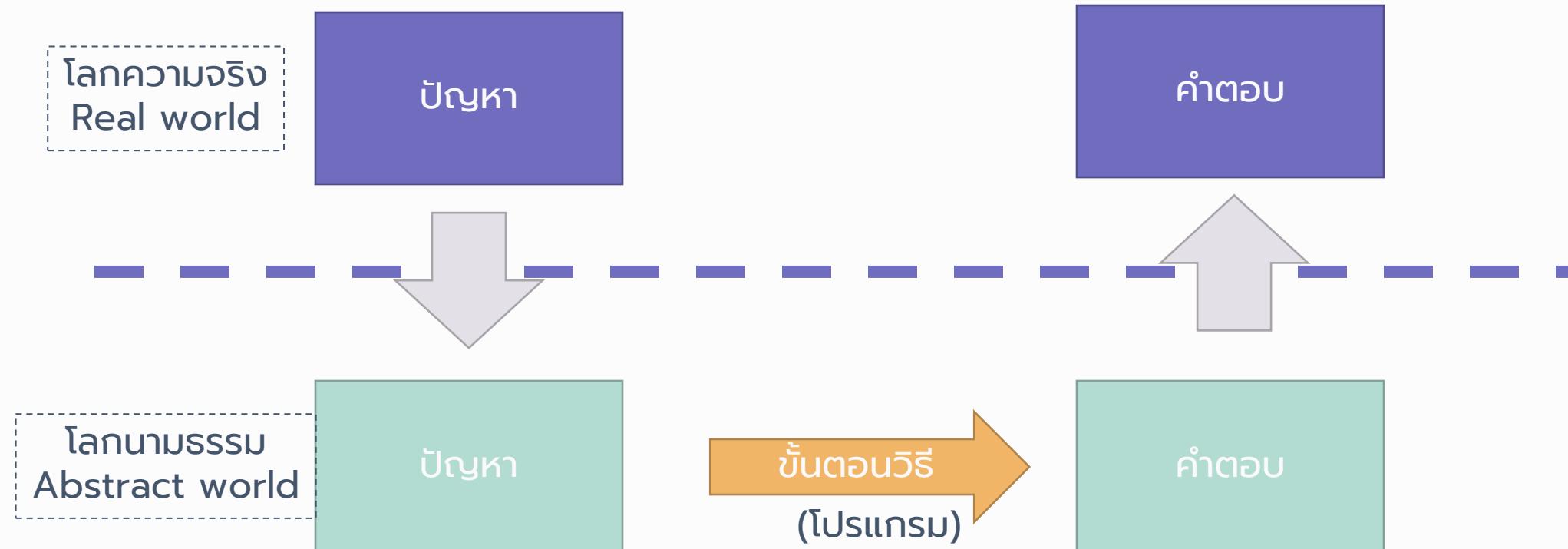
# วัตถุประสงค์ของบทเรียน

เมื่อสิ้นสุดบทเรียน ผู้เรียนจะสามารถ

- อธิบายถึงหลักการการเขียนโปรแกรมเชิงเส้นได้
- ใช้การเขียนผังไหลในการออกแบบโปรแกรมได้

# โปรแกรม

- โปรแกรมคือชุดคำสั่งที่ถูกออกแบบมาให้ปฏิบัติงานเพื่อดำเนินงาน (task) บางประการ



# การอ่านแบบโปรแกรม: IPO Pattern

- วิธีการอ่านแบบโปรแกรมที่ง่ายที่สุด
- แบ่งชุดกระบวนการอ่านเป็น 3 ชุด
  - กระบวนการเอาข้อมูลเข้า (Input)
  - **กระบวนการแก้ไขปัญหา (Process)**
  - กระบวนการเอาข้อมูลออก (Output)



# กระบวนการเอาข้อมูลเข้า (Input)

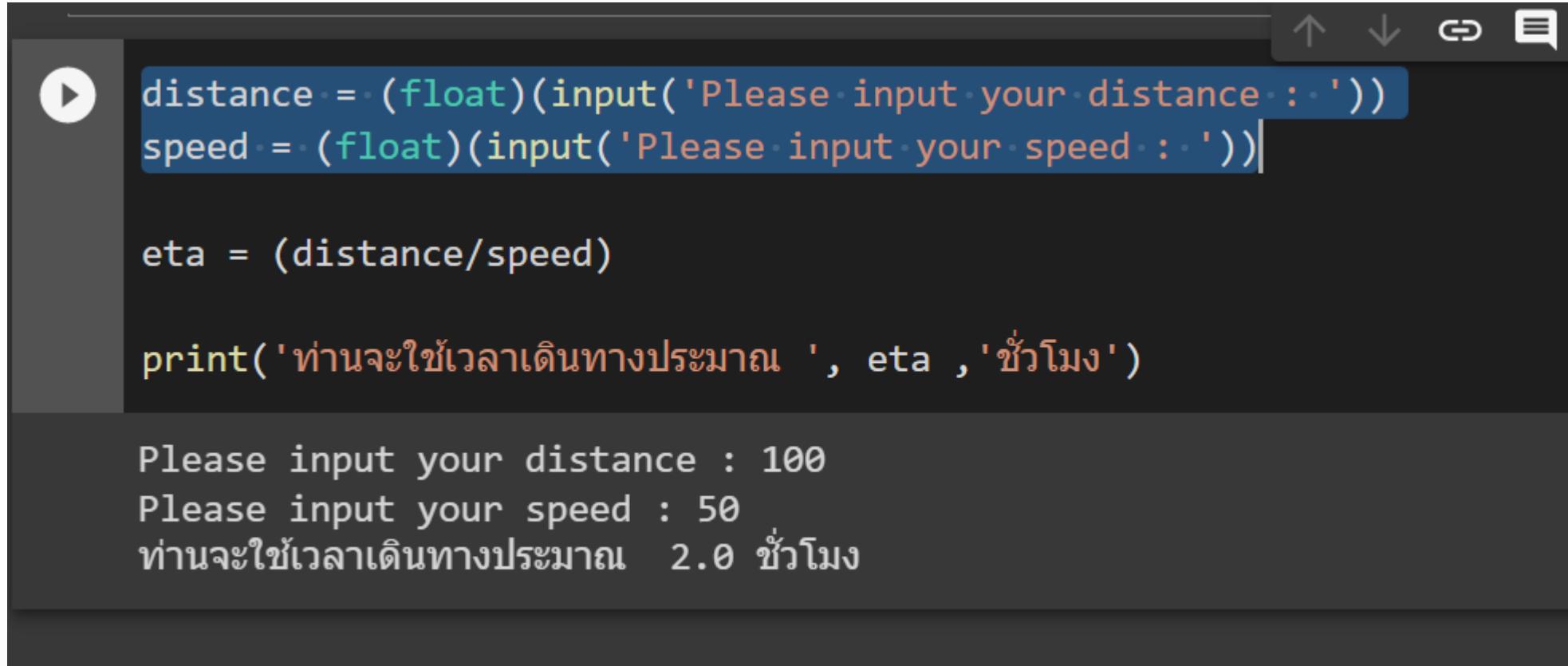
- กระบวนการรับข้อมูลเข้าสู่ระบบ
- สามารถเป็น：  
**การกำหนดค่าเริ่มต้นของโปรแกรม**  
**รับค่าจากผู้ใช้**  
**ดึงข้อมูลจากคลังเก็บข้อมูล**
- เม้วัดดุประสงค์เพื่อเตรียมข้อมูลสำหรับการประมวลผล

# กระบวนการแก้ไขปัญหา (Process)

- กระบวนการแก้ไขปัญหา แปลงข้อมูลนำเข้า เป็นผลลัพธ์ที่ต้องการ
  - การคำนวณ
- เป็นการผสมผสานโครงสร้างการเขียนโปรแกรม
  - Sequential** programming
  - Selection** programming
  - Iterative** programming
- การมีเอกสารเป็นเรื่องสำคัญ

# กระบวนการเอาข้อมูลออก (Output)

- กระบวนการรับข้อมูลออกสู่ระบบ
- สามารถเป็น：  
**การแสดงผลการคำนวณของโปรแกรม**
- มีวัตถุประสงค์เพื่อแสดงผลลัพธ์ของการคำนวณให้กับผู้ใช้
- อาจจะต้องมีการเพิ่มเติมข้อความให้เกิดความเข้าใจ



A screenshot of a Python code editor window. The code in the editor is:

```
distance = float(input('Please input your distance : '))
speed = float(input('Please input your speed : '))

eta = (distance/speed)

print('ท่านจะใช้เวลาเดินทางประมาณ ', eta , ' ชั่วโมง')
```

The output window below shows the execution results:

```
Please input your distance : 100
Please input your speed : 50
ท่านจะใช้เวลาเดินทางประมาณ 2.0 ชั่วโมง
```

# ทำไมต้องใช้หลักการ IPO

- แยกความสนใจแต่ละส่วน (Concern Separation)
- ทำให้โปรแกรมสามารถเข้าใจได้ง่าย
- ทำให้โปรแกรมสามารถปรับปรุง/แก้ไขได้ง่าย

# วิธีการเขียนโปรแกรม

- แยกปัญหาออกเป็นปัญหาย่อยๆ
  - ส่วนย่อย = 1 ขั้นตอน
  - เป็นกิจกรรมที่เล็กที่สุด
  - เป็นกิจกรรมที่แบ่งแยกไม่ได้
- แต่ละกิจกรรมย่อจะถูกเรียบเรียงให้สอดคล้องกับแนวการทำงานแก้ไขปัญหา

# กรณีศึกษา #1

- หากคุณทำการเป็นพนักงานชั่วคราว (part-time) คุณต้องการที่จะคำนวณเงินที่คุณได้จากการทำงาน โดยใช้ระยะเวลาทำงาน และรายได้ต่อชั่วโมง
- หมายเหตุ : โปรแกรมที่เขียนขึ้นมาจะต้องทำงานได้กับข้อมูลนำเข้าทุกรูปแบบ
- เช่น :
  - ถ้าทำงาน 2 ชั่วโมง และได้ค่าแรง 10 บาท คำนวบ :  $2 * 10 = 20$
  - ถ้าทำงาน 5 ชั่วโมง และได้ค่าแรง 20 บาท คำนวบ :  $5 * 20 = 100$

# กรณีศึกษา #1

- 1. การรับข้อมูลเข้า
  - ระยะเวลาทำงาน (ชั่วโมง)
  - ค่าแรง (บาท)
- 2. กระบวนการทำงาน
  - ค่าแรง = ระยะเวลาทำงาน \* ค่าแรง
- 3. การแสดงข้อมูล
  - ค่าแรง

มีรูปแบบอื่นก็เป็นไปได้ไหม????

# เอกสารที่ใช้ในการออกแบบ

- เป็นเอกสารที่ใช้ในการออกแบบขั้นตอนวิธีการทำงาน / อธิบายการทำงาน เพื่อประโยชน์ในการสื่อสารกับผู้ร่วมงาน
- เทคโนโลยี
  - ภาษาธรรมชาติ Natural language
  - Formatted description
  - Diagram

# ภาษาธรรมชาติ Natural Language

- เป็นการเขียนเอกสารบรรยายถึงขั้นตอนการทำงานแบบไร้รูปแบบ (Free hand)
- เขียนบรรยายโดยไม่มีกำหนดรูปแบบกี่ชัดเจน
- ผู้เขียนจะต้องใช้ทักษะในการแปลงคำอธิบายเป็นโปรแกรม

# เอกสารที่มีการกำหนดรูปแบบ Formatted Text

- รูปแบบที่ชัดเจนสำหรับการจัดทำเอกสาร
  - รูปแบบการเขียนที่ชัดเจน
  - เช่น
    - นิพจน์ทางคณิตศาสตร์
    - คำสั่งการคำนวณ
    - การแสดงผล
- เมาะสมกับโปรแกรมเมอร์ ก็จะที่ Easier for programmers to understand
- ตัวอย่าง
  - Pseudo code

# แผนผัง Diagram

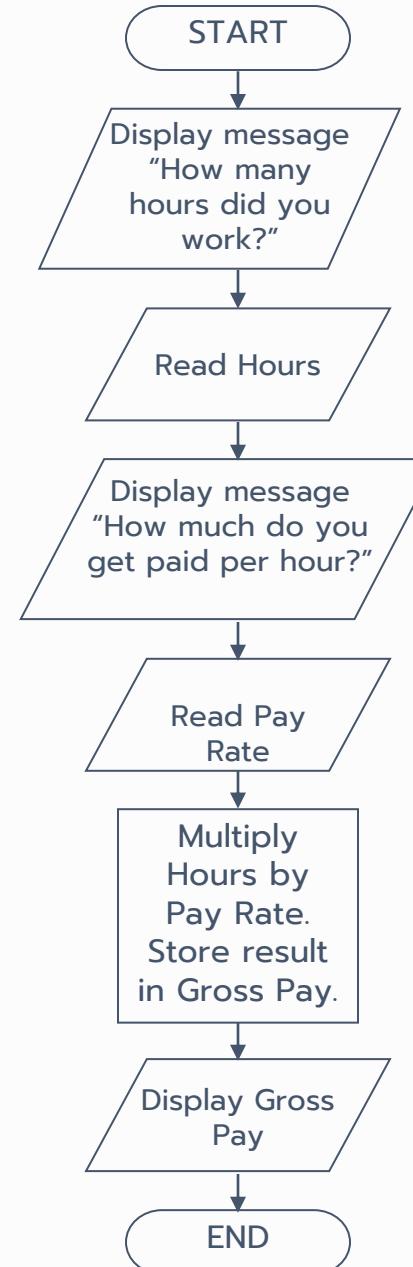
- ' One picture is equal to **thousands** word '
- ใช้สัญลักษณ์แทนกิจกรรมต่างๆ
- ตัวอย่าง
  - UML diagram
  - **Flow chart**

# ຜັງໃຫຍ່

# FLOWCHART

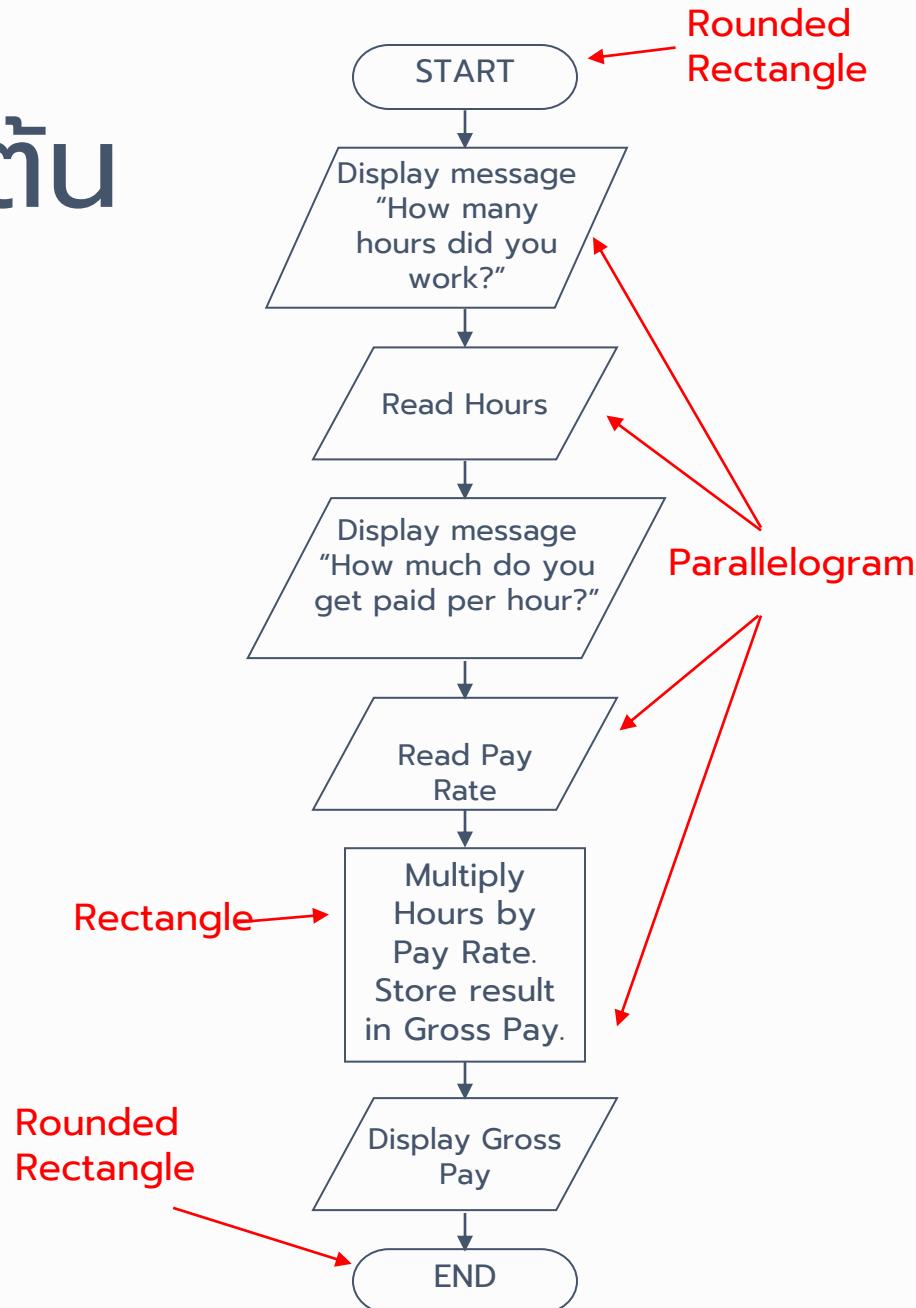
# อะไรคือผังไหล

- ผังไหลคือไวยแกรมที่แสดงถึงการไหลของการทำงานของโปรแกรม
- ยังใช้หลักการ IPO
- หนึ่งสัญลักษณ์ คือ หนึ่งความหมาย
- ผังໄລຍັງแสดงถึงลำดับของการประมวลผล



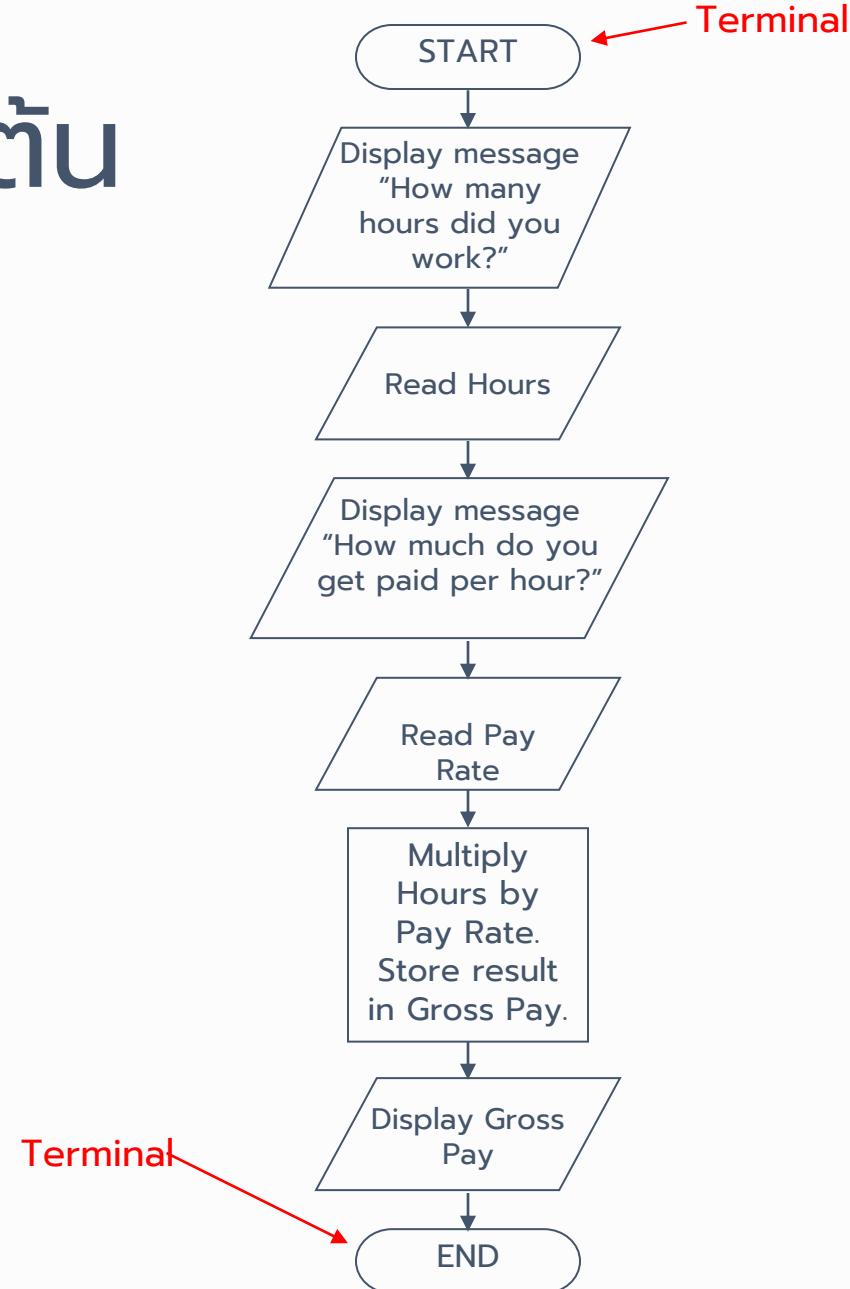
# ສ້າງລັກບະນິເບື້ອງຕົ້ນ

- ສ້າງລັກບະນິເບື້ອງຕົ້ນມີກັ່ງໜົດ 3 ປະເທດ:
  - rounded rectangles
  - parallelograms
  - a rectangle
- ແຕ່ລະສ້າງລັກບະນິແສດງດີ່ງກົງກຣມ 1 ກົງກຣມ



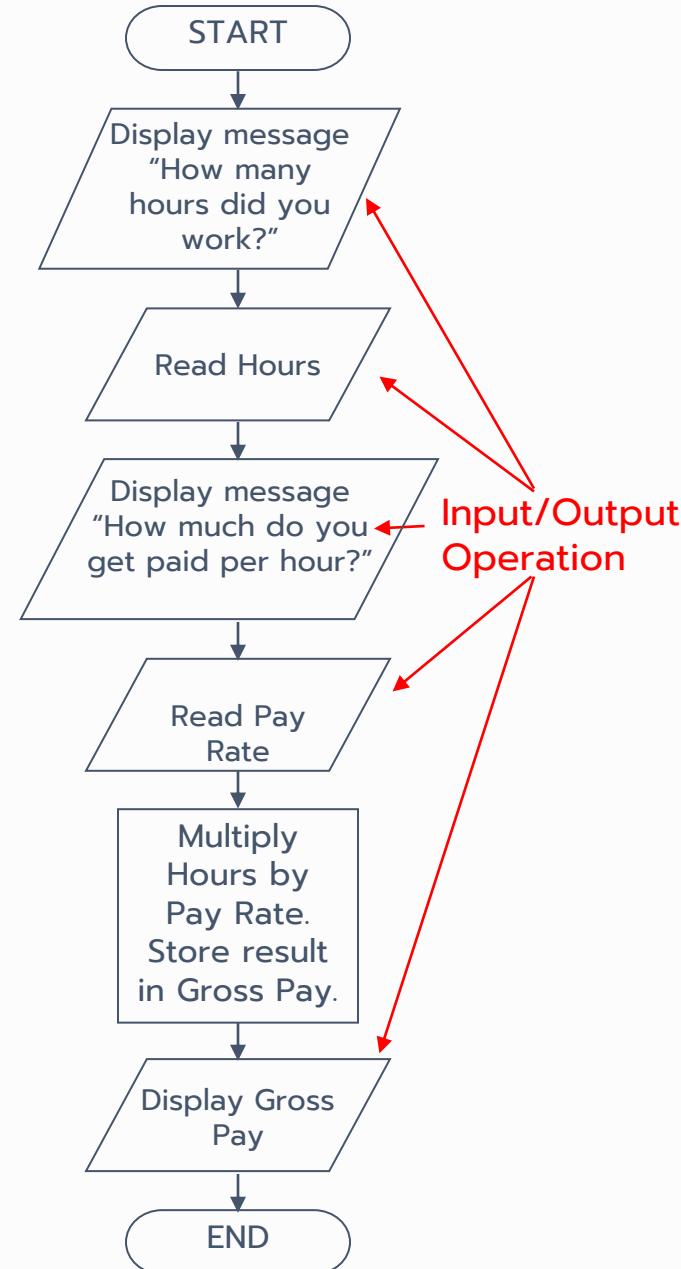
# ສ້າງລັກບະນິເປົ້ອງຕັນ

- ສ້າງລັກບະນິເຮັມຕັນ / ສິ້ນສຸດ  
Terminals
  - ແສດງໂດຍໃຫ້ **rounded rectangles**
  - ໃຫ້ແສດງຄົ່ງຈຸດເຮັມຕັນ ແລະ ຈຸດສິ້ນສຸດ



# ສ້າງລັກບະນິເປົ້ອງຕົ້ນ

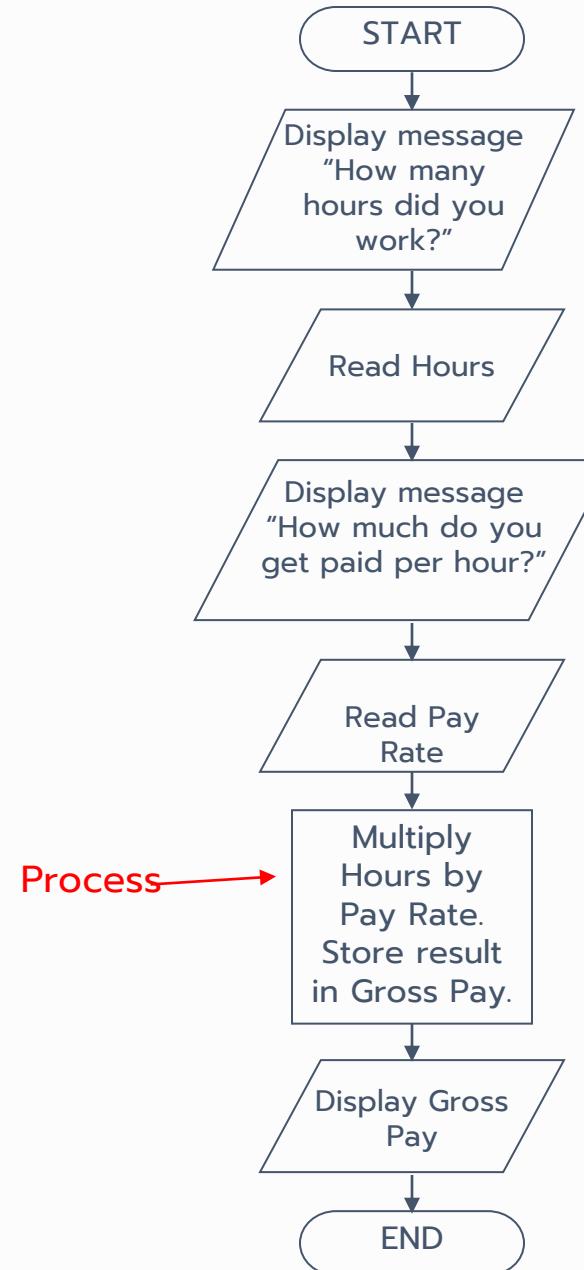
- กิจกรรมการรับข้อมูล / การแสดงผล
  - ใช້ສ້າງລັກບະນິ parallelograms



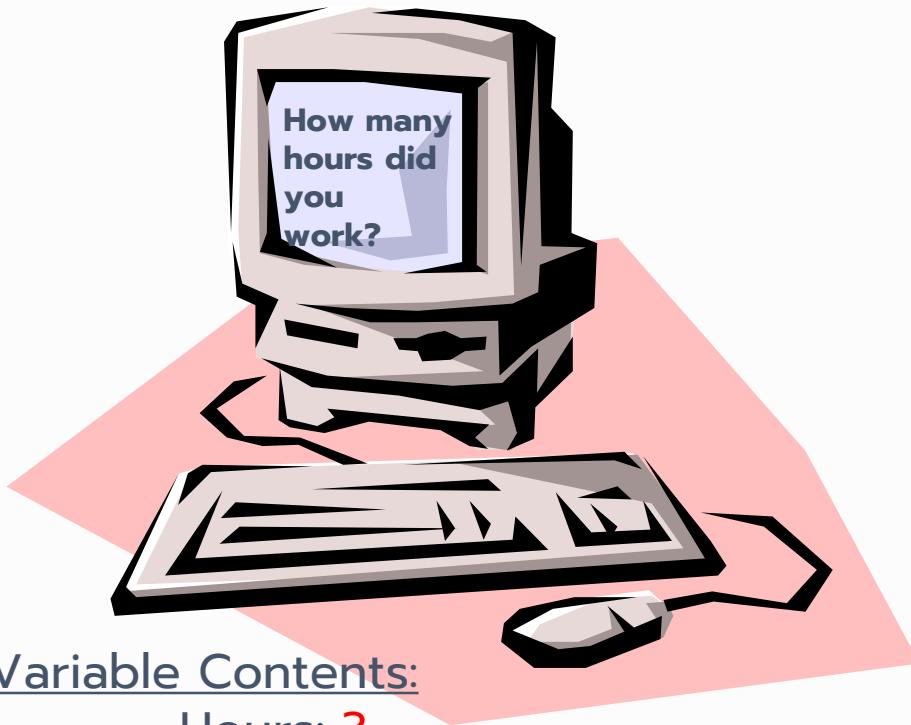
# ສັນລັກບະນົບເບື້ອງຕັນ

- ກົງກຽມກາຮປະນວລພດ
  - ແສດງພລໄດຍ rectangles
  - ຮະບູກົງກຽມກີ່ແສດງດິັງ
    - ກາຮຄຳນວນກາງຄណົຕຄາສຕຣ
    - ກາຮກຳຫຼັດຄໍາໃຫ້ຕັວແປຣ

Multiply Hours by  
Pay Rate. Store  
result in Gross Pay.



# ขั้นตอนการประมวลผล

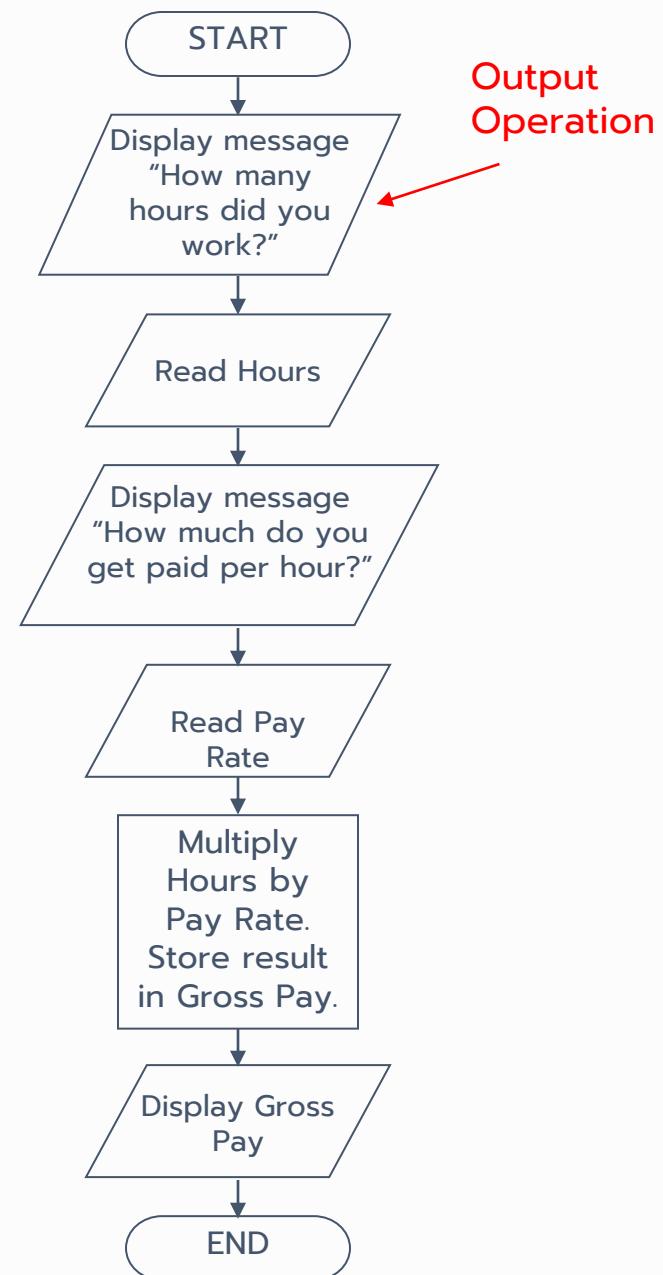


## Variable Contents:

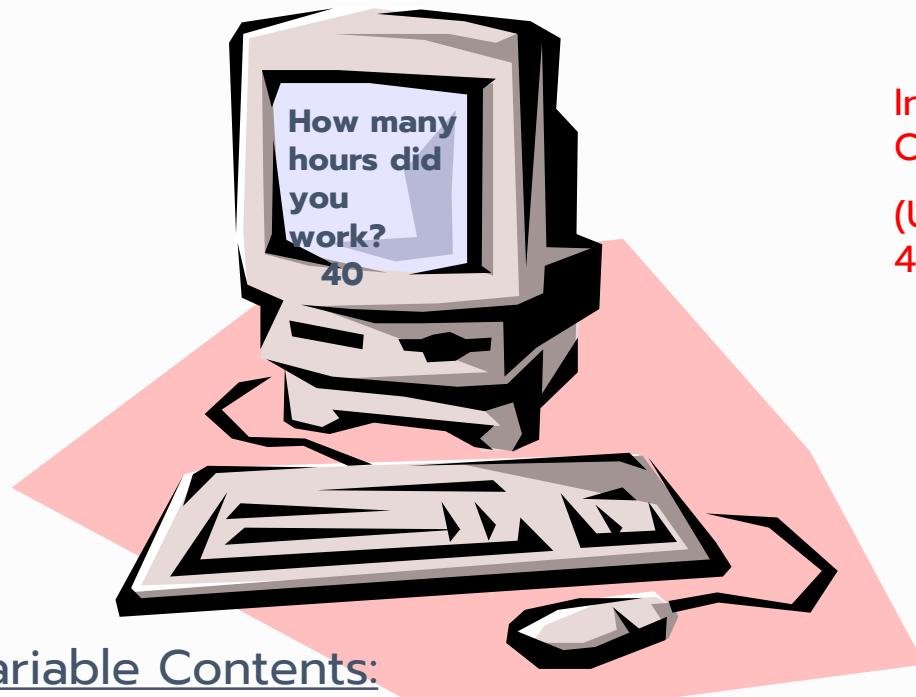
Hours: ?

Pay Rate: ?

Gross Pay: ?



# ขั้นตอนการประมวลผล

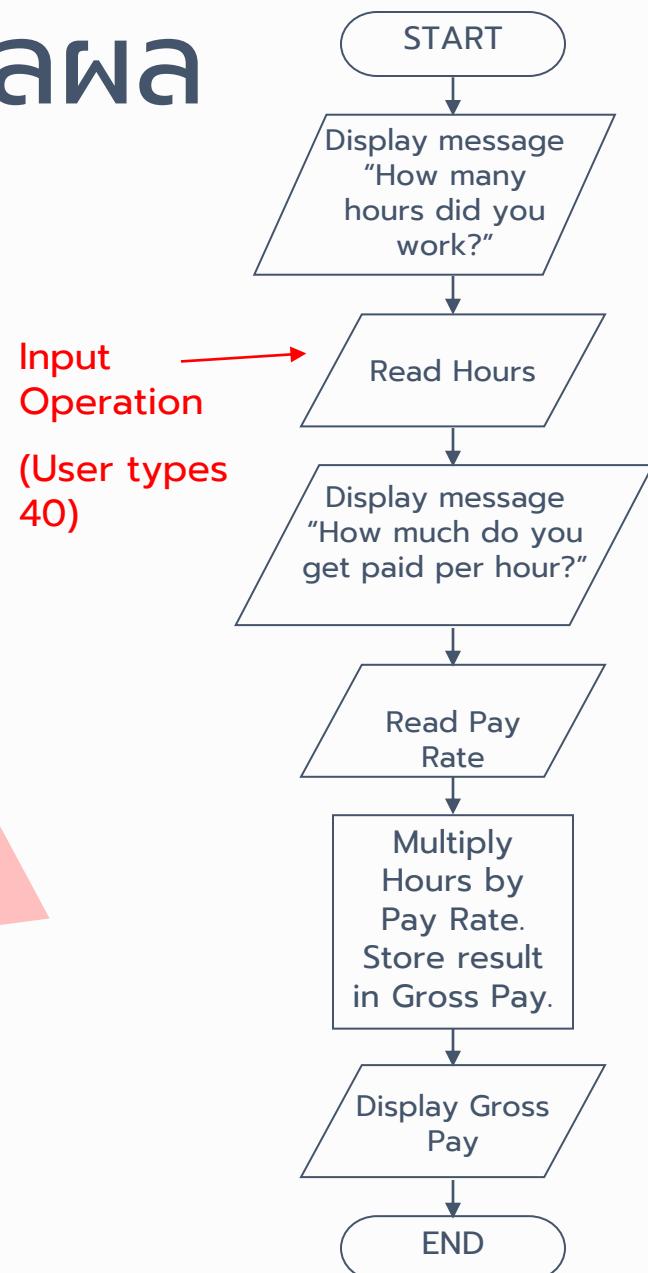


Variable Contents:

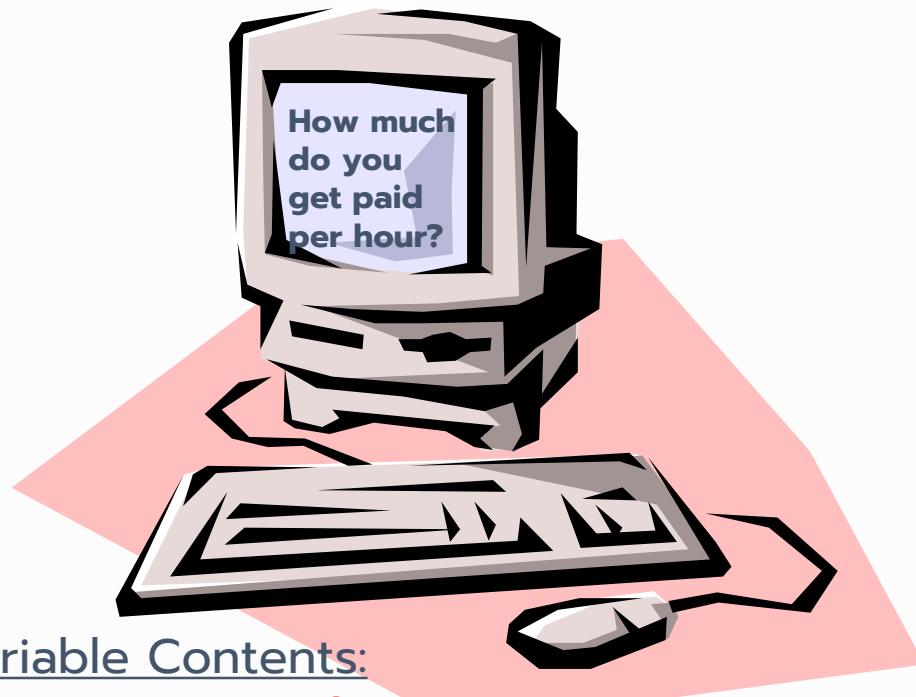
Hours: 40

Pay Rate: ?

Gross Pay: ?



# ขั้นตอนการประมวลผล

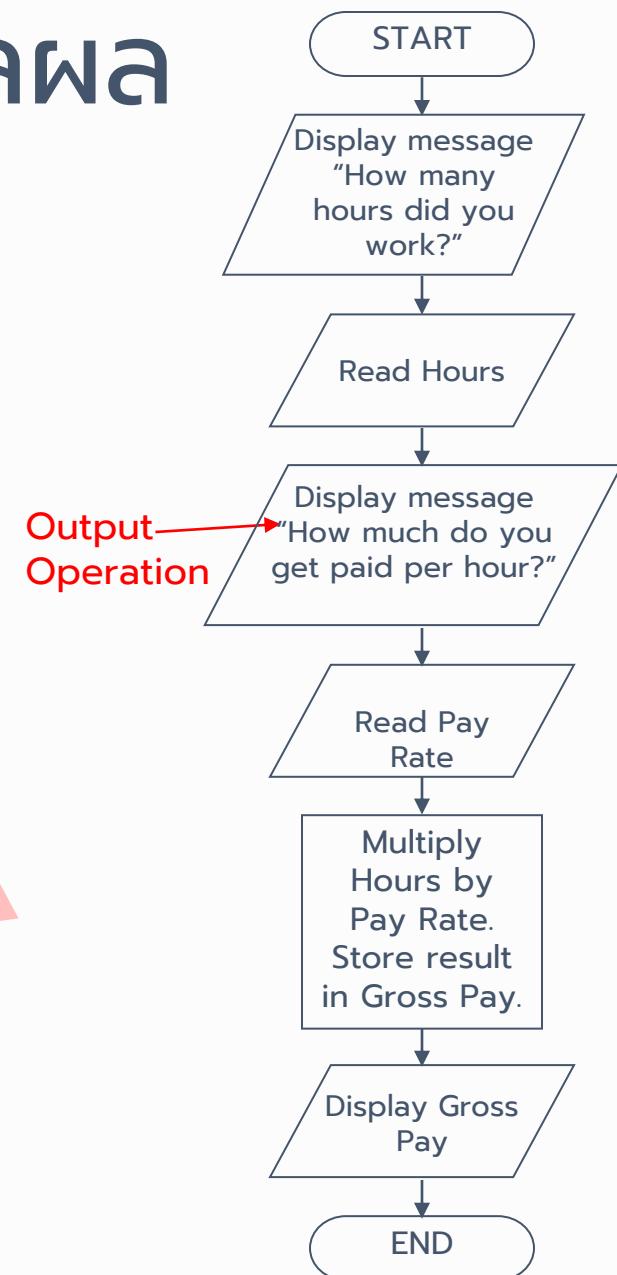


Variable Contents:

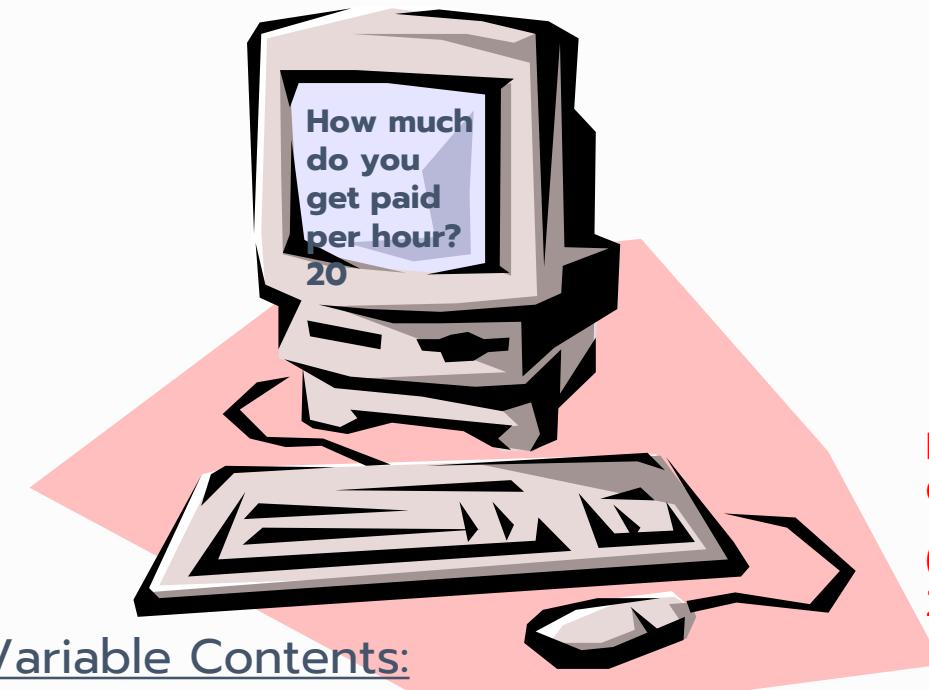
Hours: 40

Pay Rate: ?

Gross Pay: ?



# ขั้นตอนการประมวลผล



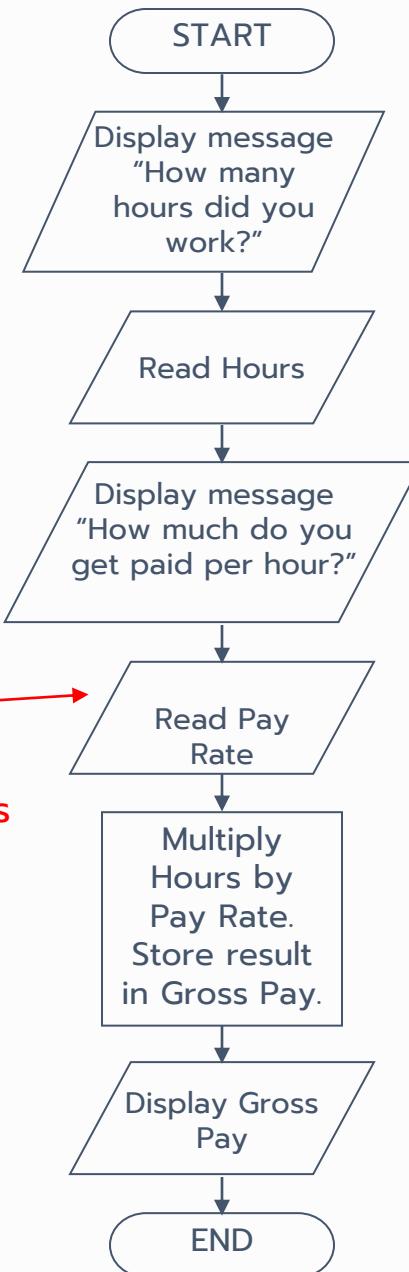
Variable Contents:

Hours: 40

Pay Rate: 20

Gross Pay: ?

Input Operation  
(User types 20)



START

Display message  
"How many  
hours did you  
work?"

Read Hours

Display message  
"How much do you  
get paid per hour?"

Read Pay  
Rate

Multiply  
Hours by  
Pay Rate.  
Store result  
in Gross Pay.

Display Gross  
Pay

END

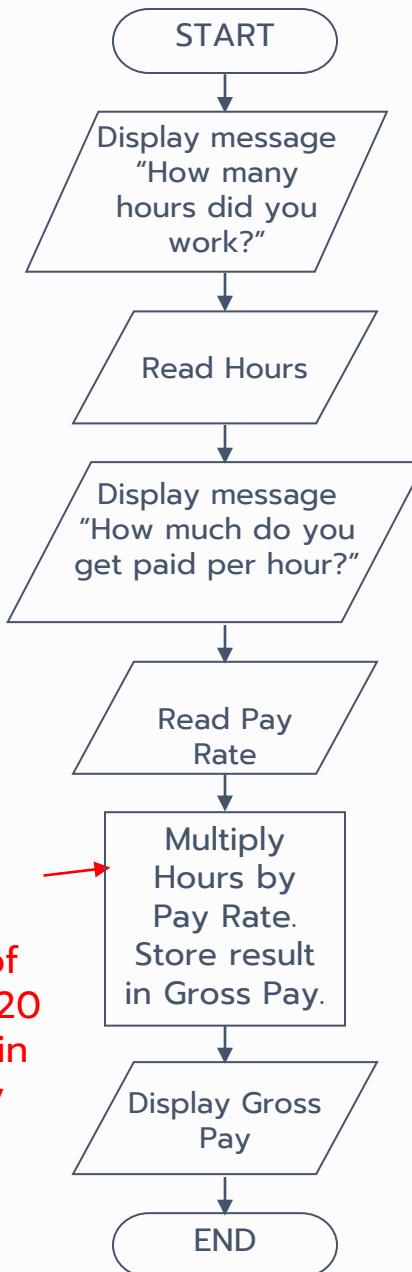
# ขั้นตอนการประมวลผล



Variable Contents:

Hours: 40  
 Pay Rate: 20  
 Gross Pay: 800

Process:  
 The product of 40 times 20 is stored in Gross Pay



# ขั้นตอนการประมวลผล



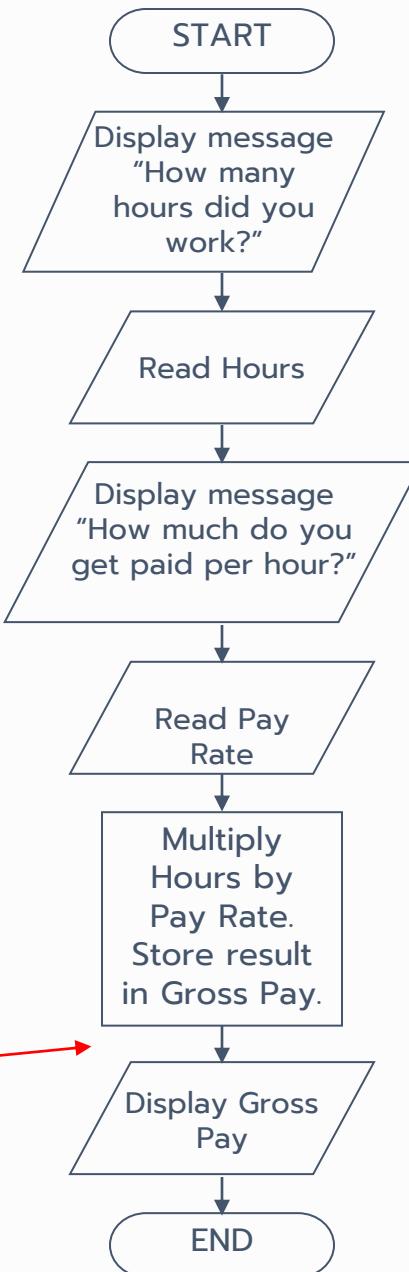
Variable Contents:

Hours: 40

Pay Rate: 20

Gross Pay: 800

Output Operation →



# การเชื่อมต่อระหว่างแผนผังย่อย

- หลายครั้ง แผนผังไม่สามารถบรรจุในหนึ่งหน้ากระดาษได้
- ตัวเชื่อมต่อ (Connector) จะใช้สัญลักษณ์วงกลม และมีอักษรระบุ  
ในสัญลักษณ์ เพื่อระบุทางเข้า และทางออก



# การเชื่อมต่อระหว่างแผนผังย่อย

- ตัวอย่างการเชื่อมต่อระหว่างแผนผังย่อย

