



# Mastering Flowcharts: ศิลปะการออกแบบผังงาน

จากแนวคิดนามธรรม สู่สถาปัตยกรรมตรรกะที่จับต้องได้

[INPUT]  
ข้อมูลเข้า



[PROCESS]  
ประมวลผล



[OUTPUT]  
ผลลัพธ์



คุณสมบัติของอัลกอริทึมที่ดี

- มีลำดับก่อน-หลังที่ชัดเจน
- เข้าใจง่าย ไม่กำกวม
- กระบวนการต้องมีจุดสิ้นสุดเสมอ

อัลกอริทึม: วิธีการเป็นขั้นๆ  
และมีการวนซ้ำจนกว่าจะเสร็จสิ้น

# 3 ภาษาสื่อสารความคิด

ภาษาพูด  
(Natural Language)



ตื่นนอน อาบน้ำ แต่งตัว  
กินข้าว ไปโรงเรียน



รหัสจำลอง  
(Pseudocode)



1. รับค่าความยาวฐาน  
2. รับค่าความสูง

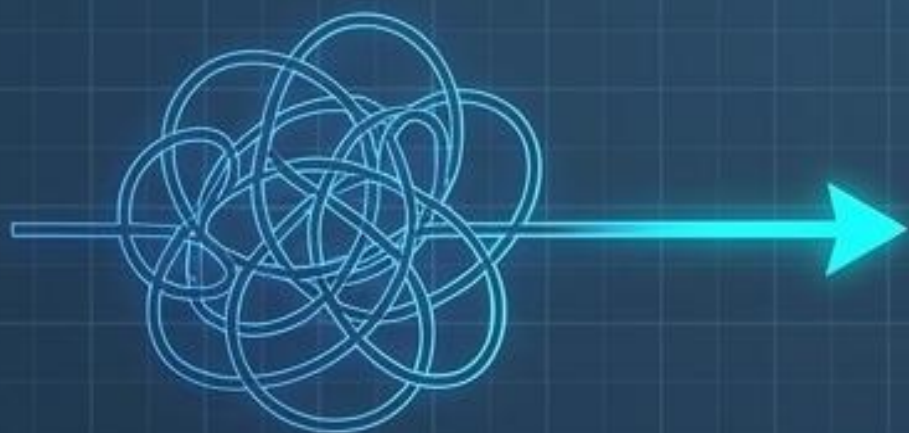


ผังงาน  
(Flowchart)



แผนภาพแสดงลำดับขั้นตอน  
ที่มองเห็นภาพรวมได้ทันที

จากคำพูด สู่รหัสจำลอง และจบด้วยแผนผัง ทุกขั้นตอนคือการย่อยความคิดให้เป็นระบบก่อนลงมือเขียนโค้ดจริง



### คลี่คลายปัญหา (Problem Define)

ทำให้เข้าใจและแยกแยะปัญหาที่ซับซ้อนได้ง่ายขึ้น



### เห็นภาพรวม (Step Flowing)

แสดงลำดับการทำงานตั้งแต่ต้นจนจบอย่างชัดเจน ไม่สับสน



### ตรวจสอบบักง่าย (Easy to Debug)

เมื่อเกิดข้อผิดพลาด สามารถไล่ดูตามเส้นทางเพื่อแก้ไขได้ตรงจุด

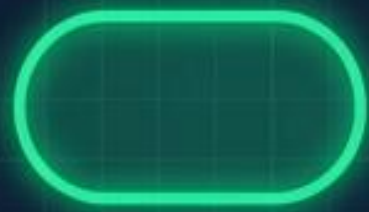


### สื่อสารไ้ร่อยต่อ (Flexible & Easy to Read)

ไม่ขึ้นกับภาษาโปรแกรมใดๆ ผู้อื่นสามารถทำความเข้าใจได้อย่างรวดเร็ว

# พจนานุกรมภาพ: อุปกรณ์มาตรฐาน ANSI

## Basic Symbols



Start/Terminator - จุดเริ่มต้น  
และ จุดสิ้นสุด ของการทำงาน



Process - การประมวลผล /  
การดำเนินการ



Input/Output -  
การรับและแสดงข้อมูล



Flow Line / Direction -  
ทิศทาง การไหลของข้อมูล  
(ลูกศรด้านเดียว)

## Control & Connectors



Decision - การตัดสินใจตามเงื่อนไข  
(ต้องมีออก 'จริง' และ 'เท็จ')



Predefined Process - โปรแกรมย่อย  
หรือฟังก์ชันที่มีการกำหนดไว้แล้ว



Connector (In-page) -  
จุดเชื่อมต่อบนหน้ากระดาษเดียวกัน

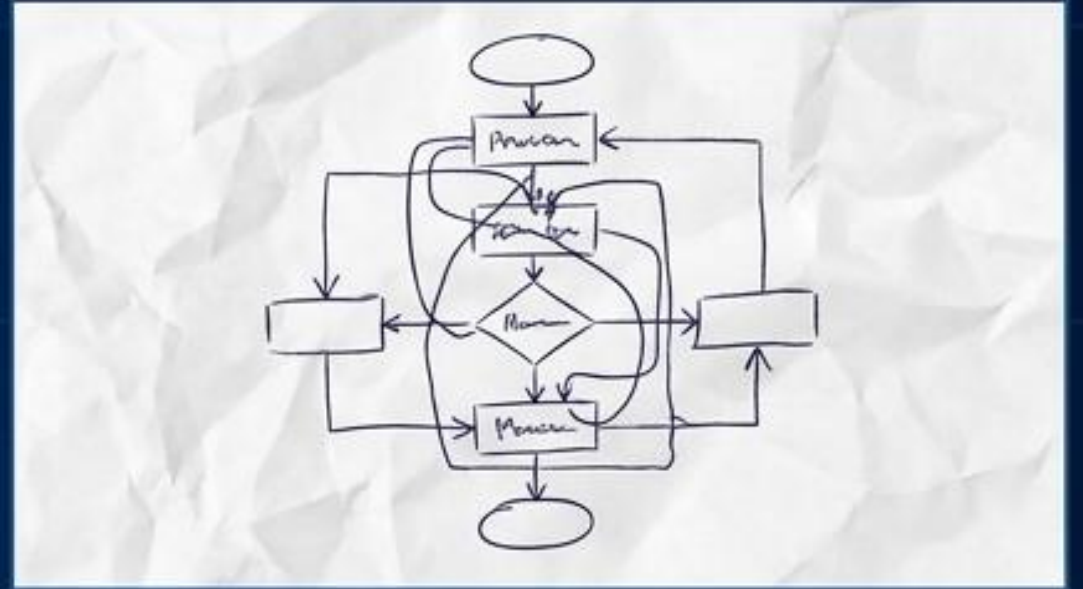


Between-page Connector -  
จุดเชื่อมต่อข้ามหน้ากระดาษ

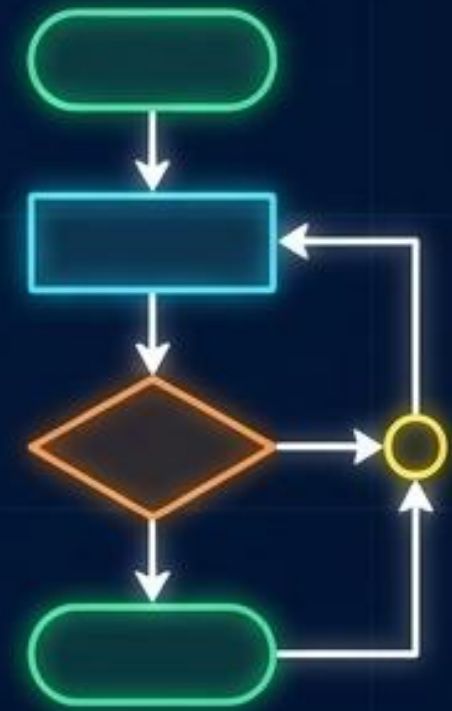
# กฎทองคำแห่งการร่างแบบ

- ✓ - ไหลจาก 'บนลงล่าง' (Top-to-Bottom) หรือ 'ซ้ายไปขวา' (Left-to-Right) เสมอ
- ✓ - ทุกสัญลักษณ์ต้องมีทางเข้าและออก (ยกเว้น Start/End)
- ✓ - ข้อความในกล่องต้องสั้น กระชับรัด เข้าใจง่าย
- ✓ - ห้ามลากเส้นตัดกันโยงไปมาอย่างไร้ระเบียบ ให้ใช้ 'จุดเชื่อมต่อ (Connectors)' แทน

⊗ DON'T



✓ DO



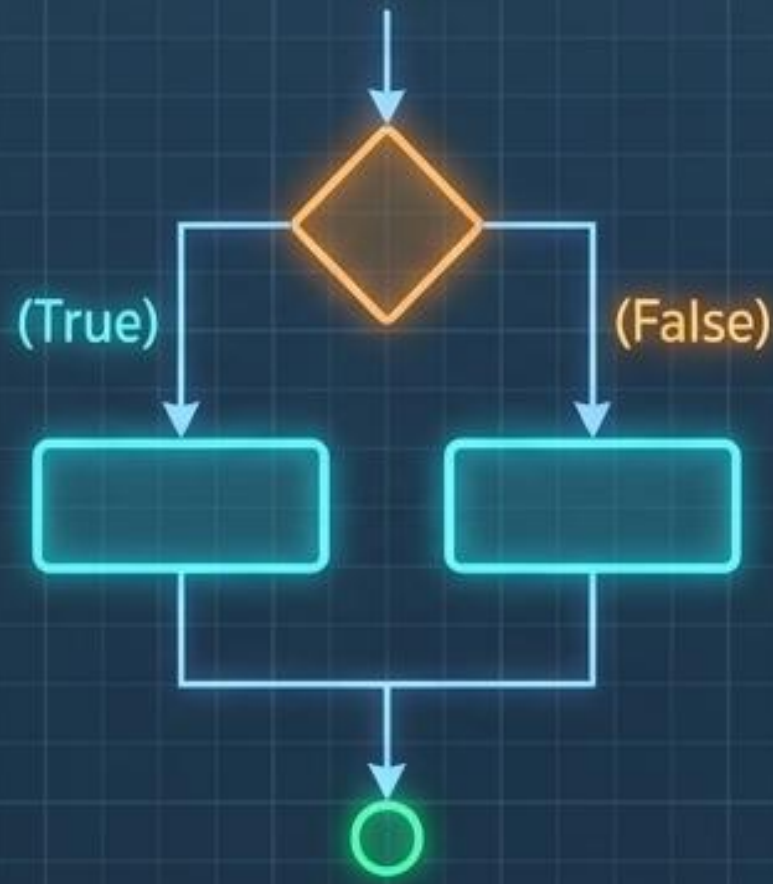
# โครงสร้างตรรกะทั้ง 3 รูปแบบ (Visual Control Flows)

## ลำดับ (Sequence)



โครงสร้างพื้นฐานที่สุด  
ไม่มีข้ามขั้น ไม่มีย้อนกลับ  
(ทำงานเรียงลำดับจากบนลงล่าง)

## ทางเลือก (Selection)



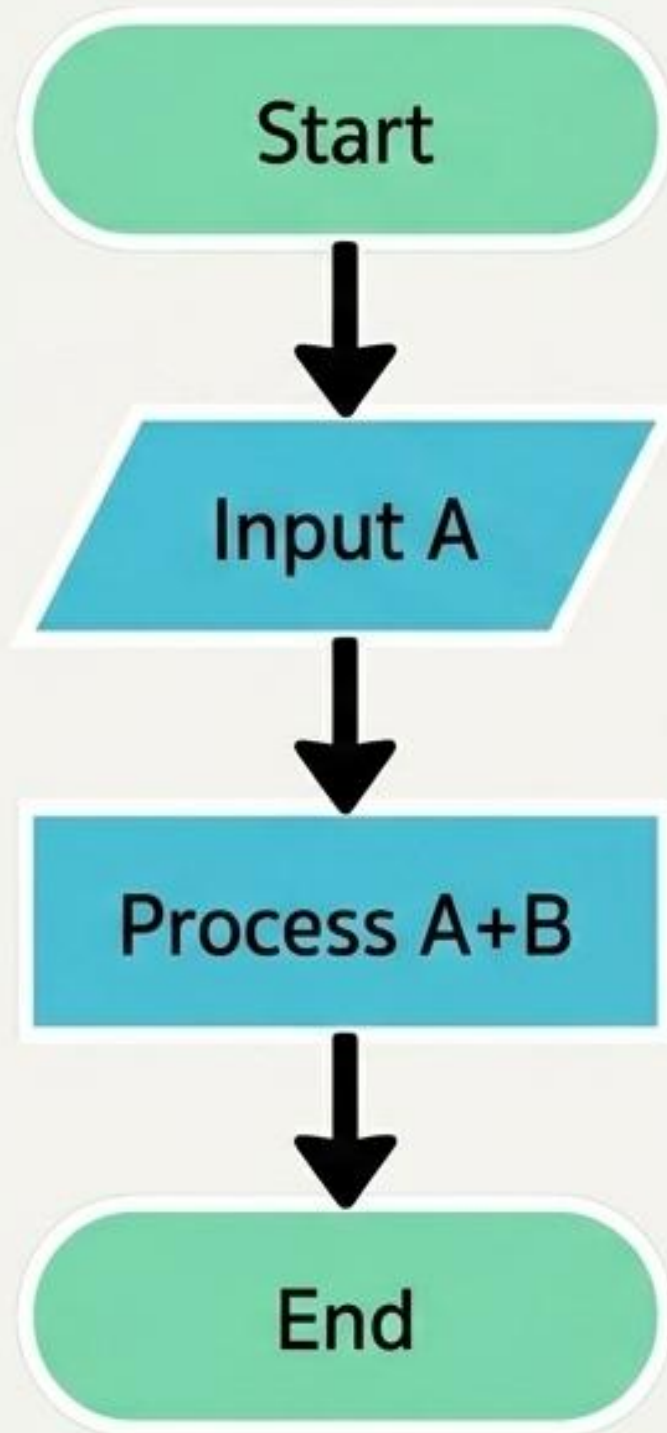
สร้างความฉลาดให้โปรแกรม  
บังคับให้เลือกทำอย่างใดอย่าง  
หนึ่งตามผลลัพธ์ (จริง/เท็จ)

## ทำซ้ำ (Iteration)



เครื่องมือลดทอนความซ้ำซ้อน  
ทำงานแบบเดิมซ้ำไปเรื่อยๆ  
จนกว่าเงื่อนไขจะกลายเป็น "เท็จ"

# โครงสร้างที่ 1: แบบลำดับ (Sequence)



- โครงสร้างพื้นฐานที่สุดของการคิด
- แสดงขั้นตอนการทำงานที่เรียงลำดับกันไปเป็นเส้นตรง

**\*กฎเหล็ก:** ไม่มีการข้ามขั้น และไม่มีการย้อนกลับไปทำคำสั่งที่ได้ทำไปแล้วโดยเด็ดขาด

# เคล็ดลับทอดไข่ดาวขอบกรอบ ไข่แดงเยิ้ม สไตล์เด็กหอ



1

เปิดไฟแรงสุดจนน้ำมันเดือดจัด



2

ตอกไข่ใส่กระทะทันที



3

รอ 30 วินาทีโดยไม่พลิกด้าน  
เพื่อให้ขอบกรอบแต่ไข่แดงยังความเยิ้ม



4

ปิดเตาและจัดเสิร์ฟ  
ตักไข่ชิ้นใส่จานทันทีไม่ให้ไข่สุกเกินไป

# สถาปัตยกรรมการทอดไข่ (The Logical Blueprint of an Egg)



# สร้างผลงานจากไอเดียสุดครีเอทีฟ

## (Idea Dashboard: Digital & Life)



### หมวดคอนเทนต์ครีเอเตอร์

- ขึ้นตอนตัดต่อคลิปเต้น TikTok ให้ขอลิขิต
- จัดเสียงทำพอดแคสต์ (Podcast) เล่าเรื่องผี
- ตั้งชื่อและทำปก YouTube ให้น่าดึงดูด



### หมวดเกมเมอร์ & อีสปอร์ต

- พิธีกรรมบูเตลูก่อนกด 'เปิดเกม' ให้ได้ตัว 5 ดาว
- วอร์มนิ้วและสายตาก่อนแข่ง FPS
- สร้างทีมดักแด้เพื่อนใน Minecraft



### หมวดบันเทิง & คอสเพลย์

- เตรียมตัวและแต่งหน้าไปงานคอสเพลย์
- แทะท่าเต้น K-POP จากคลิป Mirror
- จัดเพลย์ลิสต์เพลงเศร้าฟังตอนฝนตก



### หมวดทาสสัตว์เลี้ยง

- อบน้ำแมวโดยที่ทาสไม่โดนข่วน
- ฝึกน้องหมาให้ยอมขอมือ
- ประงอาหารเปียกสูตรพิเศษให้สัตว์เลี้ยง

# ตรรกะในชีวิตประจำวัน (Idea Dashboard: Everyday Life)



## หมวดเชฟวัยรุ่น

- ทำแซนวิชชีสเยิ้มด้วยไมโครเวฟ
- ผสมน้ำปั่นสูตร 'มั่วแต่อร่อย' จากของในตู้เย็น
- จัดจานมาฆ่าคัพให้ดูหรู 5 ดาว



## หมวดแฟชั่น

- ลงสกินแคร์ 5 เลเยอร์ก่อนนอนเพื่อหน้าใส
- จัดกระเป๋านักเรียนไม่ให้หลังหัก
- แมทช์ชุดไปเที่ยวสยามให้ดูดีที่สุด



## หมวดท่องเที่ยว

- วางแผนทริปเที่ยวสวนสนุกกับเพื่อน
- เดินทางกลับบ้านด้วยรถสาธารณะ (ระบุดาย/สถานี)
- จัดกระเป๋าไปเข้าค่ายลูกเสือ

เริ่มคิด

ย่อข้อมูล

**ความคิดที่ซับซ้อน ต้องการสถาปัตยกรรมที่ชัดเจน**

จัดระเบียบตรรกะ

สถาปัตยกรรมที่สมบูรณ์

เริ่มต้นสร้างสรรค์ผลงานผังงานของคุณได้ที่:  
[www.draw.io](http://www.draw.io)

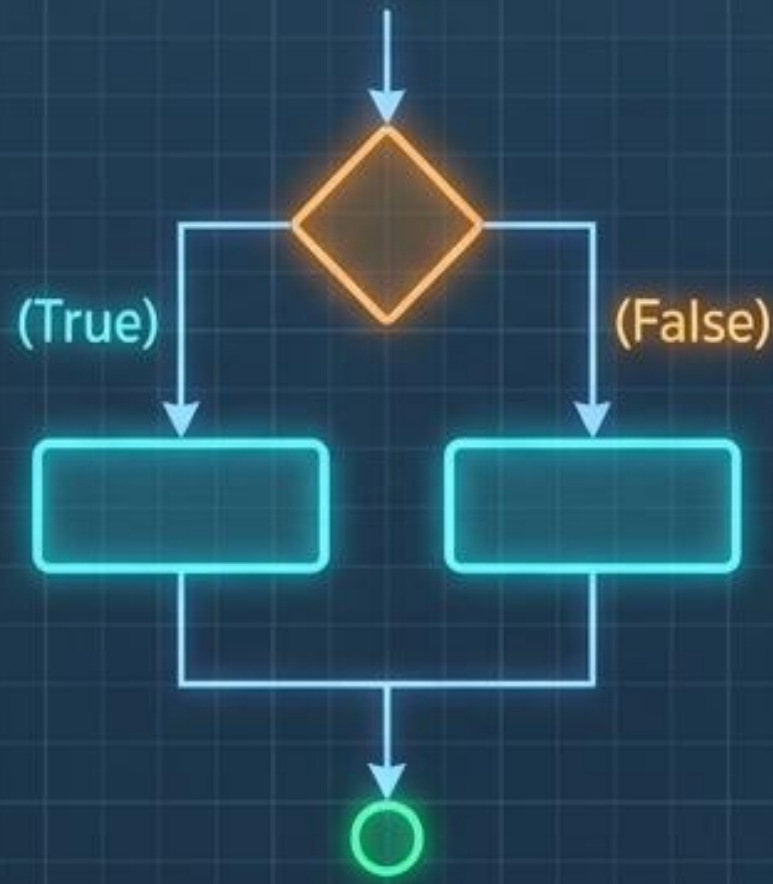
# โครงสร้างตรรกะทั้ง 3 รูปแบบ (Visual Control Flows)

## ลำดับ (Sequence)



โครงสร้างพื้นฐานที่สุด  
ไม่มีข้ามขั้น ไม่มีย้อนกลับ  
(ทำงานเรียงลำดับจากบนลงล่าง)

## ทางเลือก (Selection)



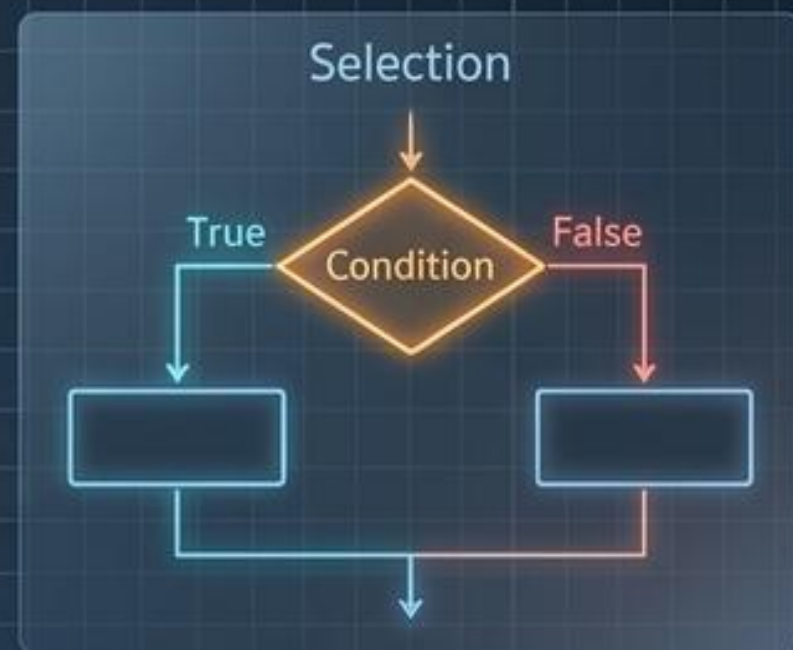
สร้างความฉลาดให้โปรแกรม  
บังคับให้เลือกทำอย่างใดอย่าง  
หนึ่งตามผลลัพธ์ (จริง/เท็จ)

## ทำซ้ำ (Iteration)



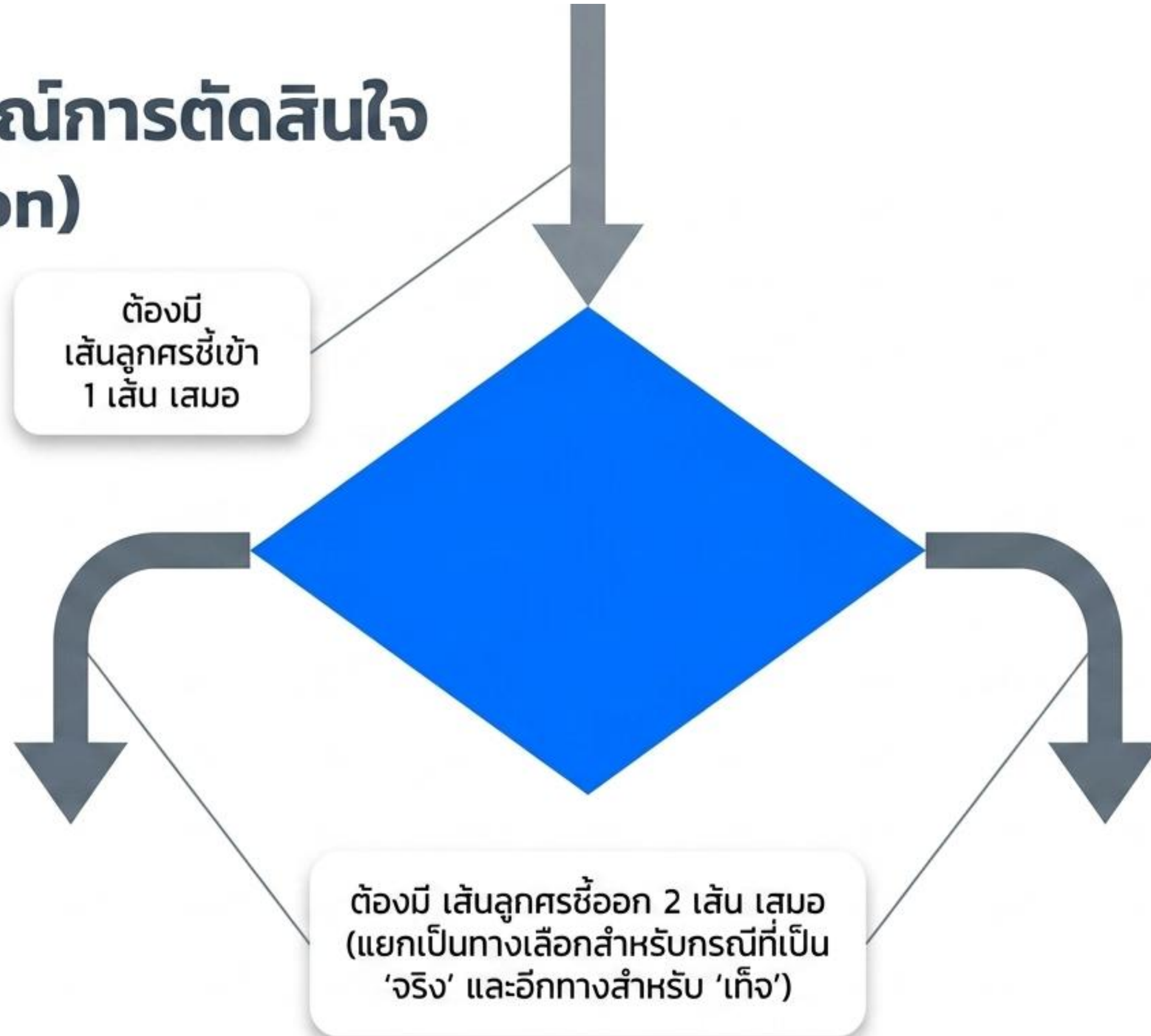
เครื่องมือลดทอนความซ้ำซ้อน  
ทำงานแบบเดิมซ้ำไปเรื่อยๆ  
จนกว่าเงื่อนไขจะกลายเป็น "เท็จ"

# ชำแหละจุดตัดสินใจ (Anatomy of a Decision)



เงื่อนไขคือ 'สวิตช์สับราง' ที่บังคับให้ข้อมูลต้องเลือกเดินไปเพียงทางเดียวเท่านั้น

# สัญลักษณ์การตัดสินใจ (Decision)



คะแนน  $\geq 50$   
หรือไม่?

**1. เงื่อนไข (Condition):**  
สิ่งที่เราต้องการให้โปรแกรม  
ตั้งคำถามหรือตรวจสอบ

True

False

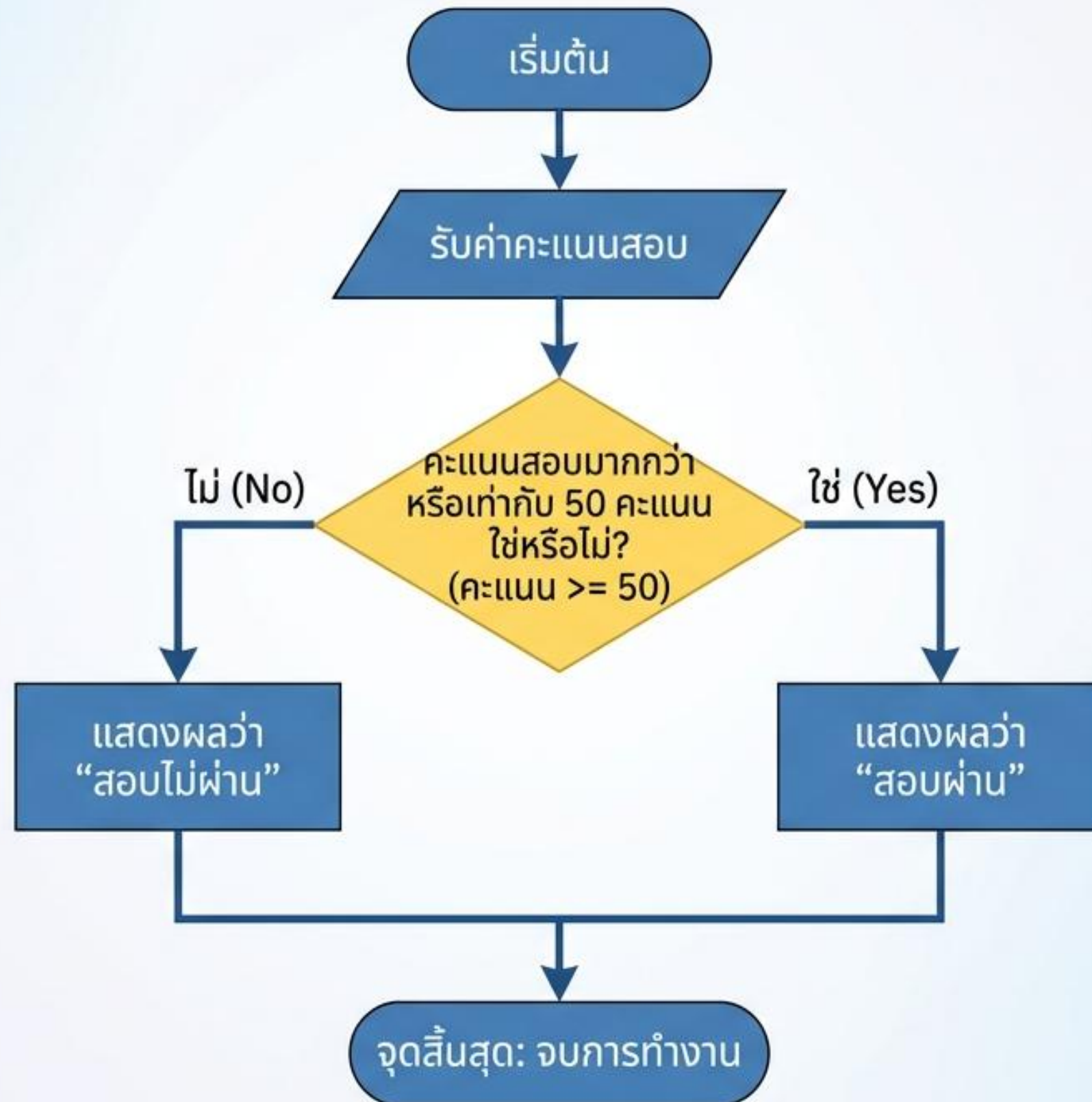
**2. ทางเลือก 'จริง' (True):**

ชุดคำสั่งที่จะให้โปรแกรมทำ  
หากผลการตรวจสอบถูกต้อง

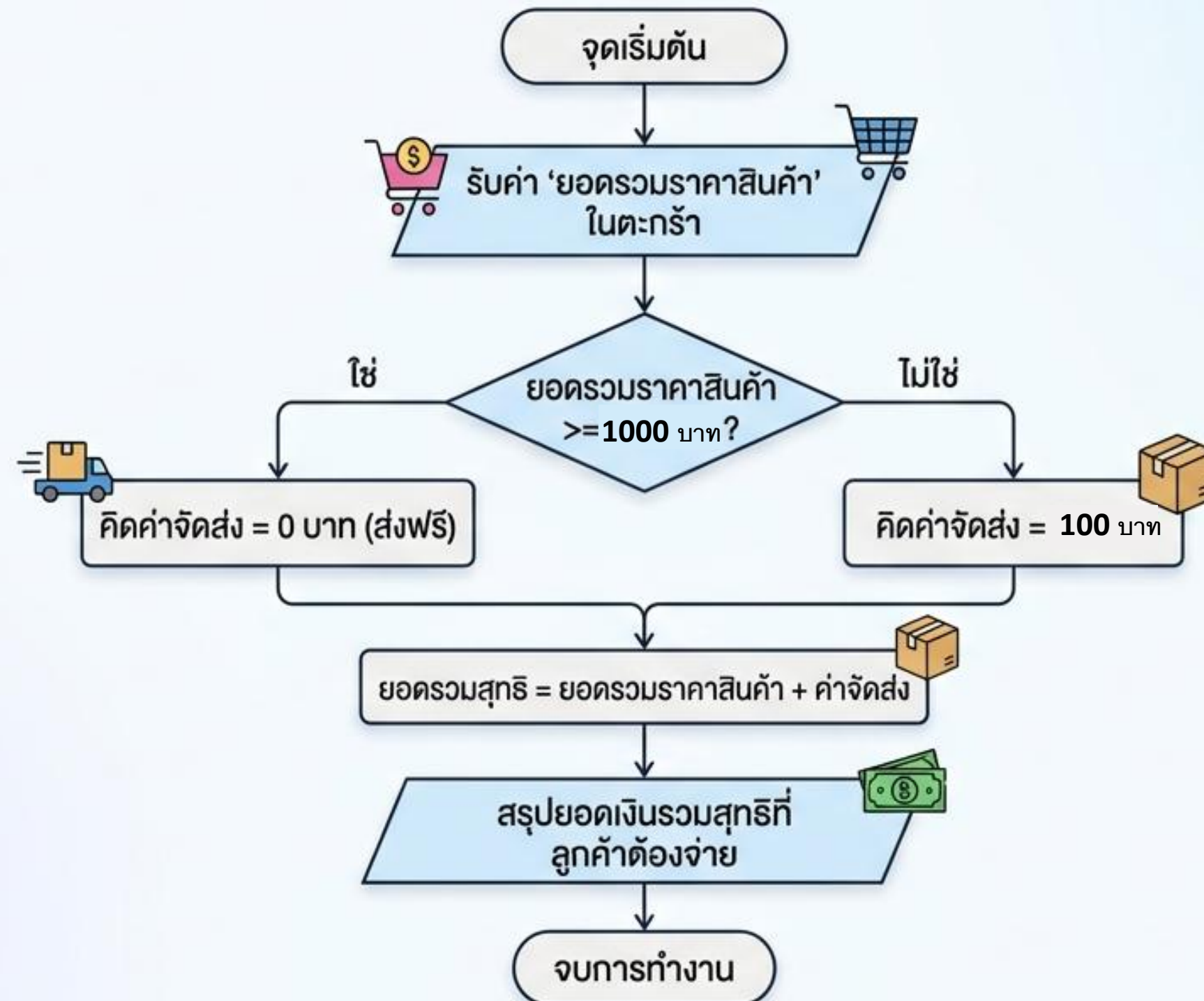
**3. ทางเลือก 'เท็จ' (False):**

ชุดคำสั่งที่จะให้โปรแกรมทำ  
หากผลการตรวจสอบไม่ถูกต้อง

# ผังงานการประเมินผลสอบ (ผ่าน / ไม่ผ่าน)



# ผังงานโปรแกรมจัดส่งฟรีของร้านค้าออนไลน์ (E-commerce)





# ภารกิจท้าทายความคิด!

ถึงเวลาที่นักเรียนจะได้ลองฝึกเขียนผังงานแบบเงื่อนไขด้วยตัวเองแล้ว  
หยิบกระดานและปากกาขึ้นมาเตรียมพร้อมได้เลย!



# | โจทย์ฝึกทำ (ภารกิจที่ 1)

## 01 ตรวจสอบคำนำหน้าชื่อ

**สถานการณ์:** ให้เขียนฟังก์ชันรับค่า 'อายุ' เพื่อแบ่งหมวดหมู่คำนำหน้าชื่อตามกฎหมายไทย

**เงื่อนไข:** อายุตั้งแต่ 15 ปีขึ้นไป (อายุ  $\geq 15$ )

✓ หากเป็น **จริง**: ให้แสดงคำว่า 'นาย/นางสาว'

✗ หากเป็น **เท็จ**: ให้แสดงคำว่า 'เด็กชาย/เด็กหญิง'

# | โจทย์ฝึกทำ (ภารกิจที่ 2)

## 02 ร้อนนี้ต้องพึ่งใคร?

สถานการณ์:

สร้างระบบสมาร์ทโฮม โดยให้เขียนฟังก์ชันรับค่า  
'อุณหภูมิห้อง' (องศาเซลเซียส)

**เงื่อนไข:** อุณหภูมิ มากกว่า 30 องศา  
(อุณหภูมิ > 30)

✓ หากเป็น **จริง**: ให้สั่ง 'เปิดเครื่องปรับอากาศ'

✗ หากเป็น **เท็จ**: ให้สั่ง 'เปิดพัดลม'

# | โจทย์ฝึกทำ (ภารกิจที่ 3)

## 03 โปรโมชั่นส่งฟรี!

**สถานการณ์:** ร้านค้าออนไลน์มีโปรโมชั่นพิเศษ  
ให้เขียนฟังก์ชันรับค่า 'ราคาสินค้ารวม'  
เพื่อคำนวณค่าส่ง

**เงื่อนไข:** ชื้อสินค้าตั้งแต่ 500 บาทขึ้นไป  
(ราคารวม  $\geq 500$ )

 **หากเป็น จริง:** ให้แสดงข้อความ 'จัดส่งฟรี'

 **หากเป็น เท็จ:** ให้แสดงข้อความ 'ค่าจัดส่ง 50 บาท'

# สรุปสิ่งที่ได้เรียนรู้ในวันนี้

## ผังงานแบบทางเลือก

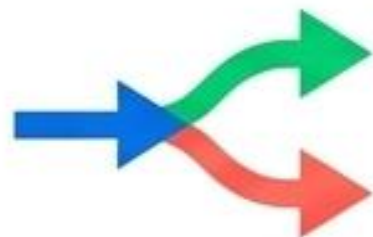
ใช้เมื่อโปรแกรมต้องมีการตัดสินใจเลือกทำอย่างใดอย่างหนึ่ง ตามสถานการณ์ที่กำหนด

## สัญลักษณ์ที่ใช้



ใช้สัญลักษณ์ 'สี่เหลี่ยมขนมเปียกปูน' สำหรับจุดที่มีการตรวจสอบเงื่อนไข

## ทิศทางของข้อมูล

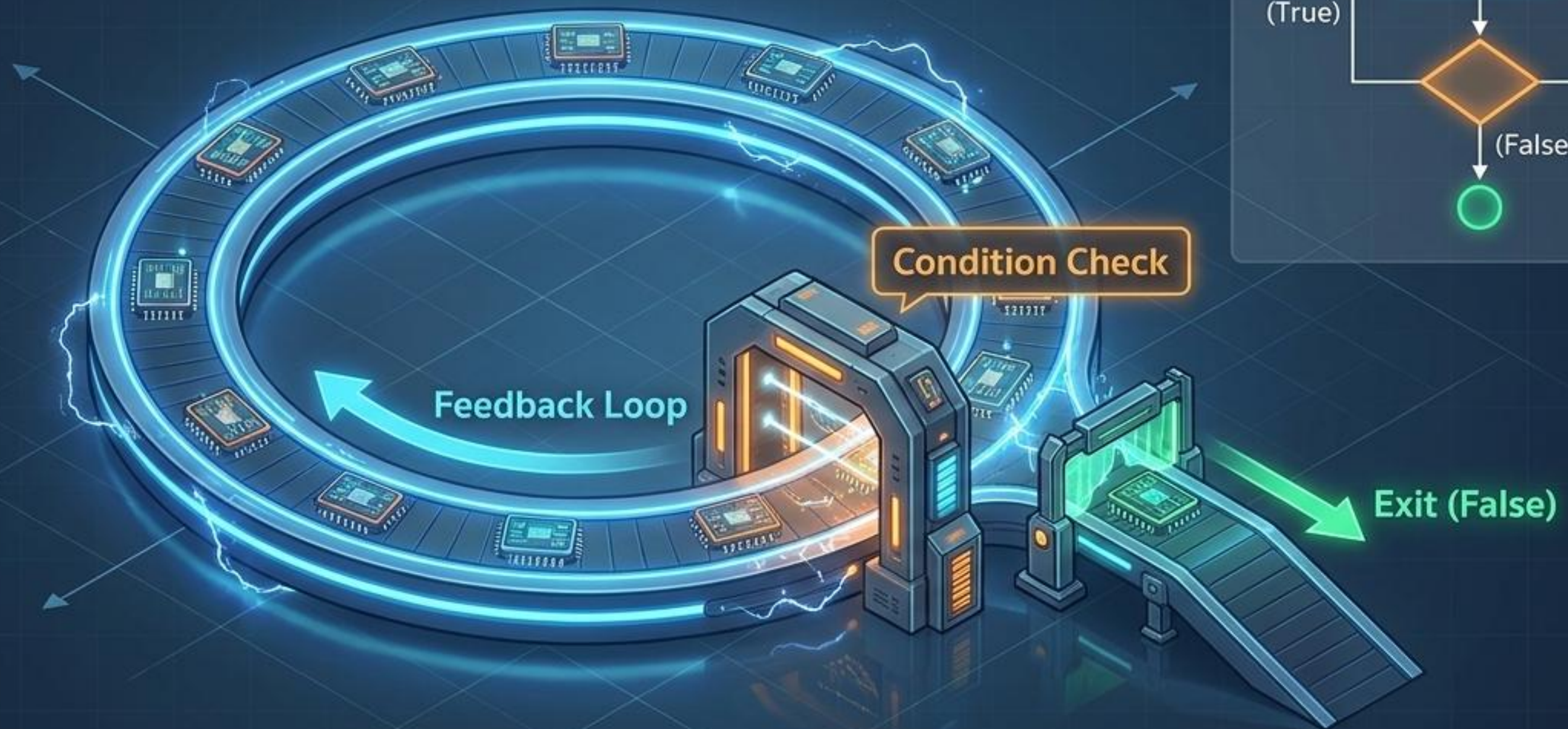


การตรวจสอบจะได้ทางแยกผลลัพธ์ออกมาเป็น 2 ทางเสมอ คือ '**จริง**' (True) และ '**เท็จ**' (False)

## ความสำคัญ

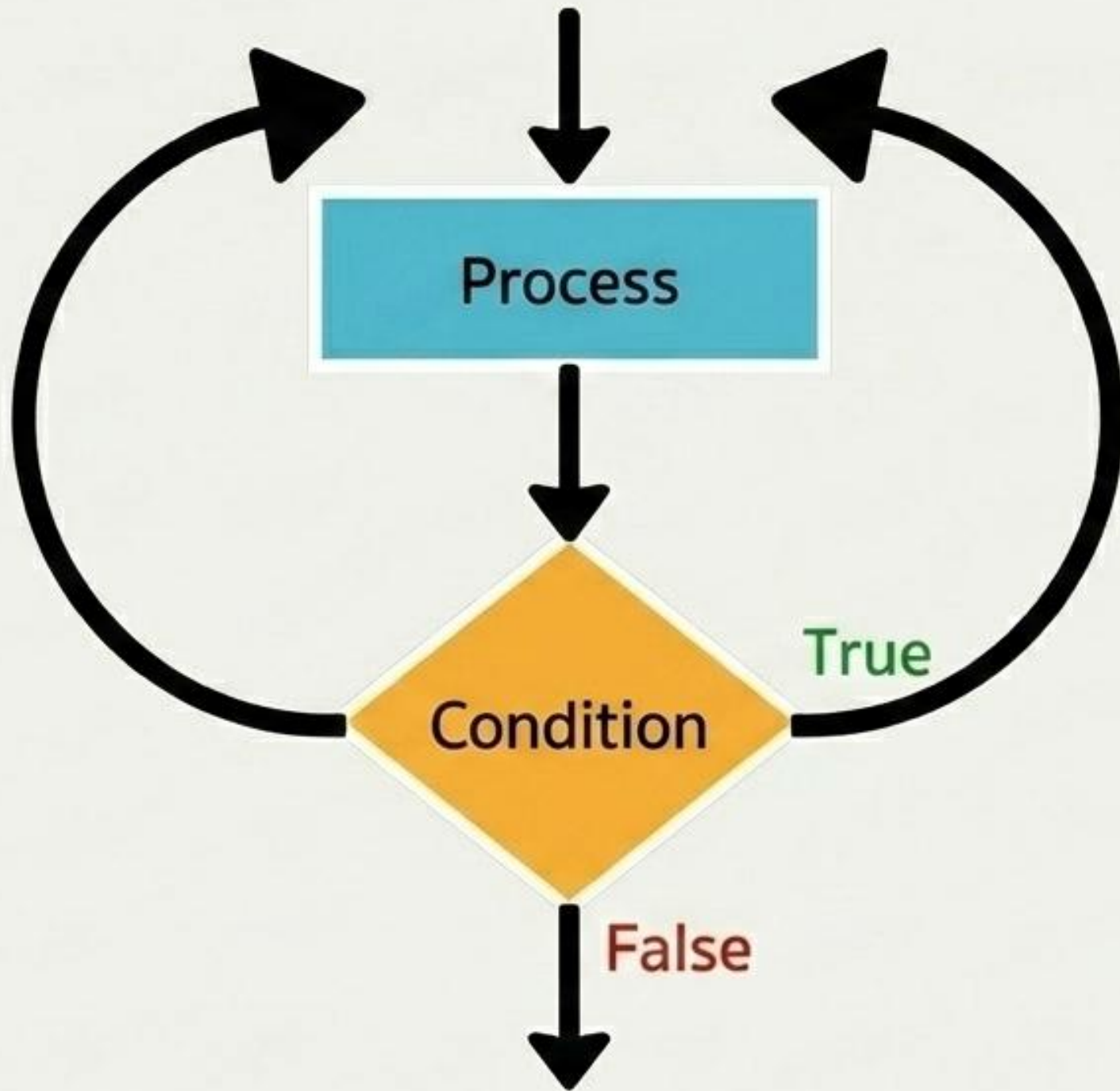
การตั้งเงื่อนไขที่ชัดเจน จะช่วยให้โปรแกรมทำงานและแก้ปัญหาได้อย่างถูกต้องแม่นยำ

# วัฏจักรการทำซ้ำ (The Iteration Loop)



‘Feedback Loop’ ดึงข้อมูลกลับขึ้นไปประมวลผลใหม่จนกว่าเงื่อนไขจะถูกทำลาย (เป็นเท็จ)

# โครงสร้างที่ 3: แบบทำซ้ำ (Iteration / Loop)



- เครื่องมือลดทอนความซ้ำซ้อนของการเขียนโค้ด
- ทำงานแบบเดียวกันซ้ำไปเรื่อยๆ ในขณะที่เงื่อนไขยังคงเป็น จริง

วงจรนี้จะถูกทำลายและทำงานอื่นต่อไป  
ก็ต่อเมื่อเงื่อนไขกลายเป็น เท็จ