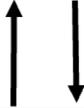
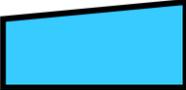


# อัลกอริทึม (Algorithm)

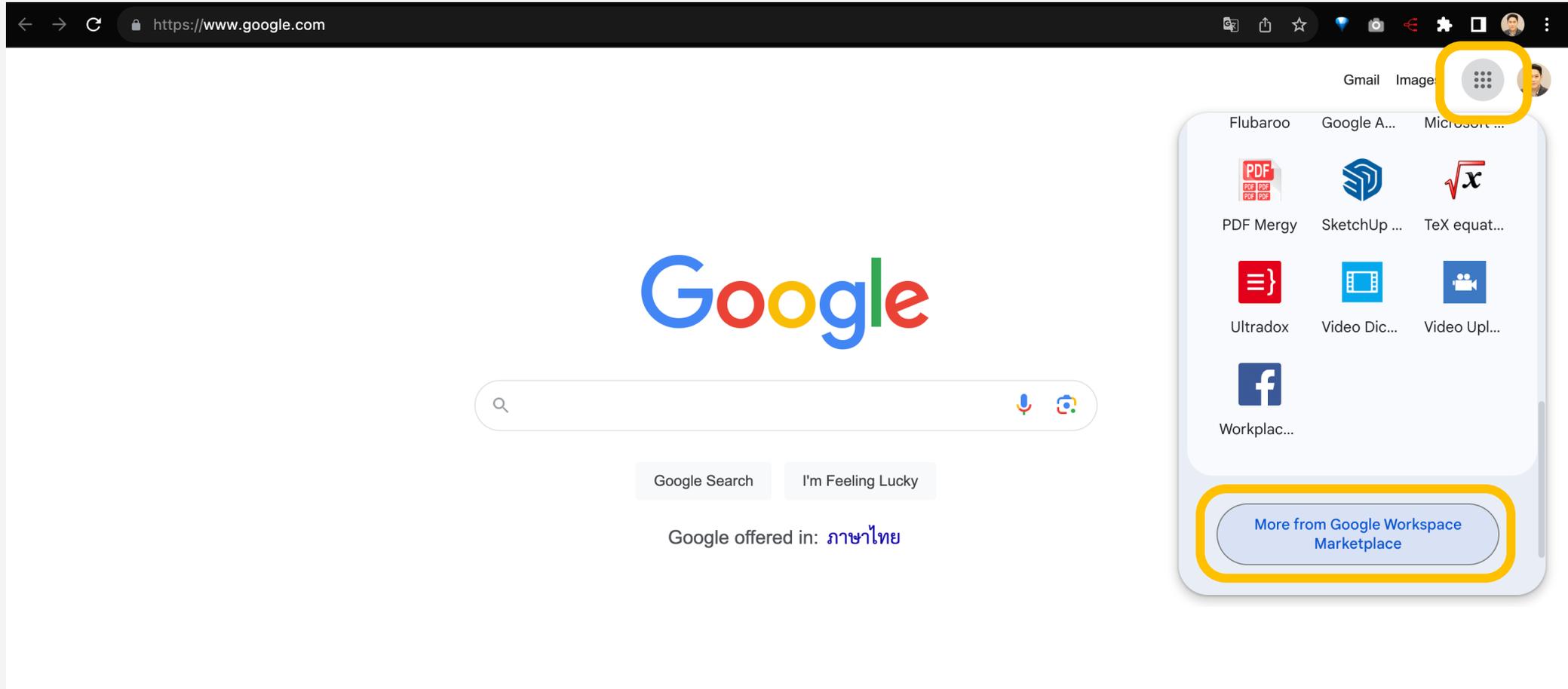
การแสดงอัลกอริทึมด้วยผังงาน หรือ โฟลวชาร์ต (Flowchart)

**โฟลว (Flow) แปลว่า การไหล**  
**ชาร์ต (Chart) แปลว่า แผนผัง**

เมื่อรวมกันแล้ว จึงหมายถึง แผนผังที่แสดงการไหลของงาน หรือ แผนผังที่แสดงขั้นตอนการทำงานอย่างชัดเจน นั่นเอง ซึ่งสามารถนำผังงานมาใช้แสดงขั้นตอนการทำงานของโปรแกรมได้

สัญลักษณ์	ความหมาย	ภาษาอังกฤษ
	หมายถึง จุดเริ่มต้น หรือ จุดสิ้นสุดการทำงาน	Begin และ End
	หมายถึง ทิศทางการไหลของงาน	Flow line
	หมายถึง การดำเนินการ หรือ การประมวลผล	Process
	หมายถึง การนำเข้าข้อมูลด้วยการป้อนค่าทางแป้นพิมพ์	Manual input
	หมายถึง การตัดสินใจตามเงื่อนไขที่กำหนดไว้	Decision
	หมายถึง การแสดงผลบนหน้าจอ	Display

# เครื่องมือสำหรับออกแบบอัลกอริทึม



# เครื่องมือสำหรับออกแบบอัลกอริทึม



draw.io

Uninstall

draw.io is the most tightly Google Drive integrated diagramming application available.

By: [JGraph Ltd](#)

Listing updated: June 15, 2023

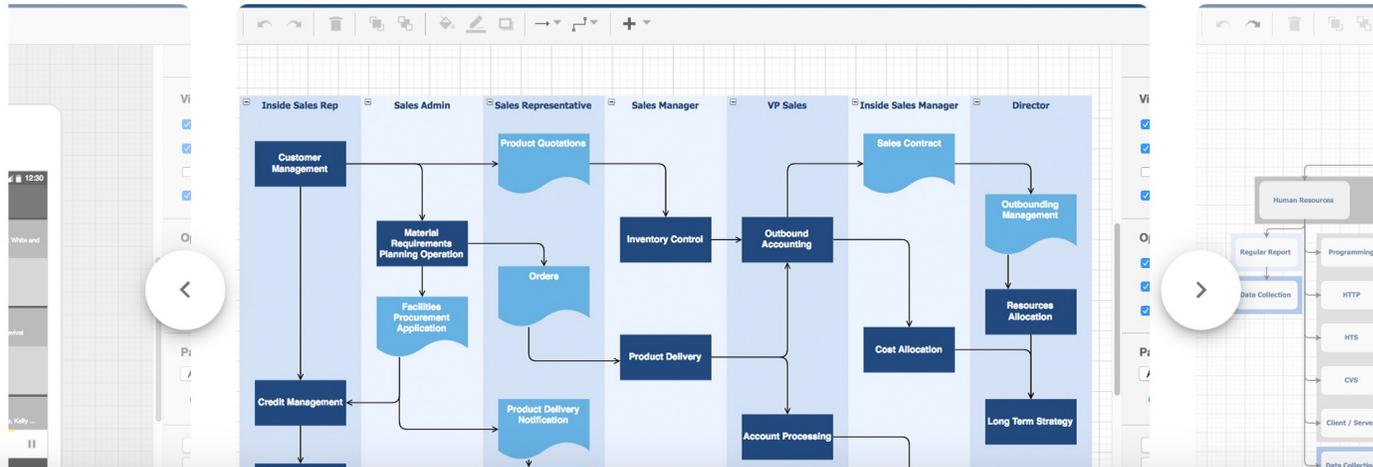
Works with:

★★★★☆ 689 ⓘ ↓ 20M+

Overview

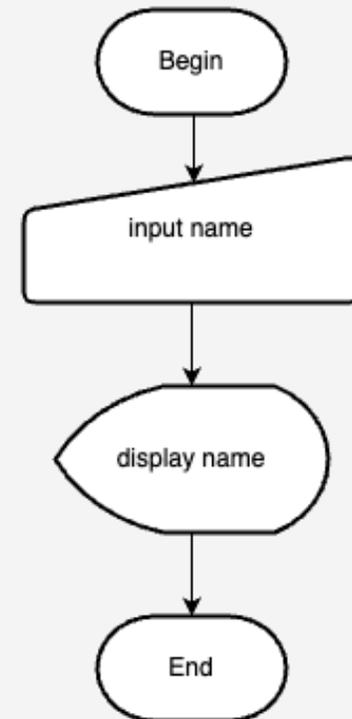
Permissions

Reviews



Flowchart: Show name

Objective: Display name from keyboard input



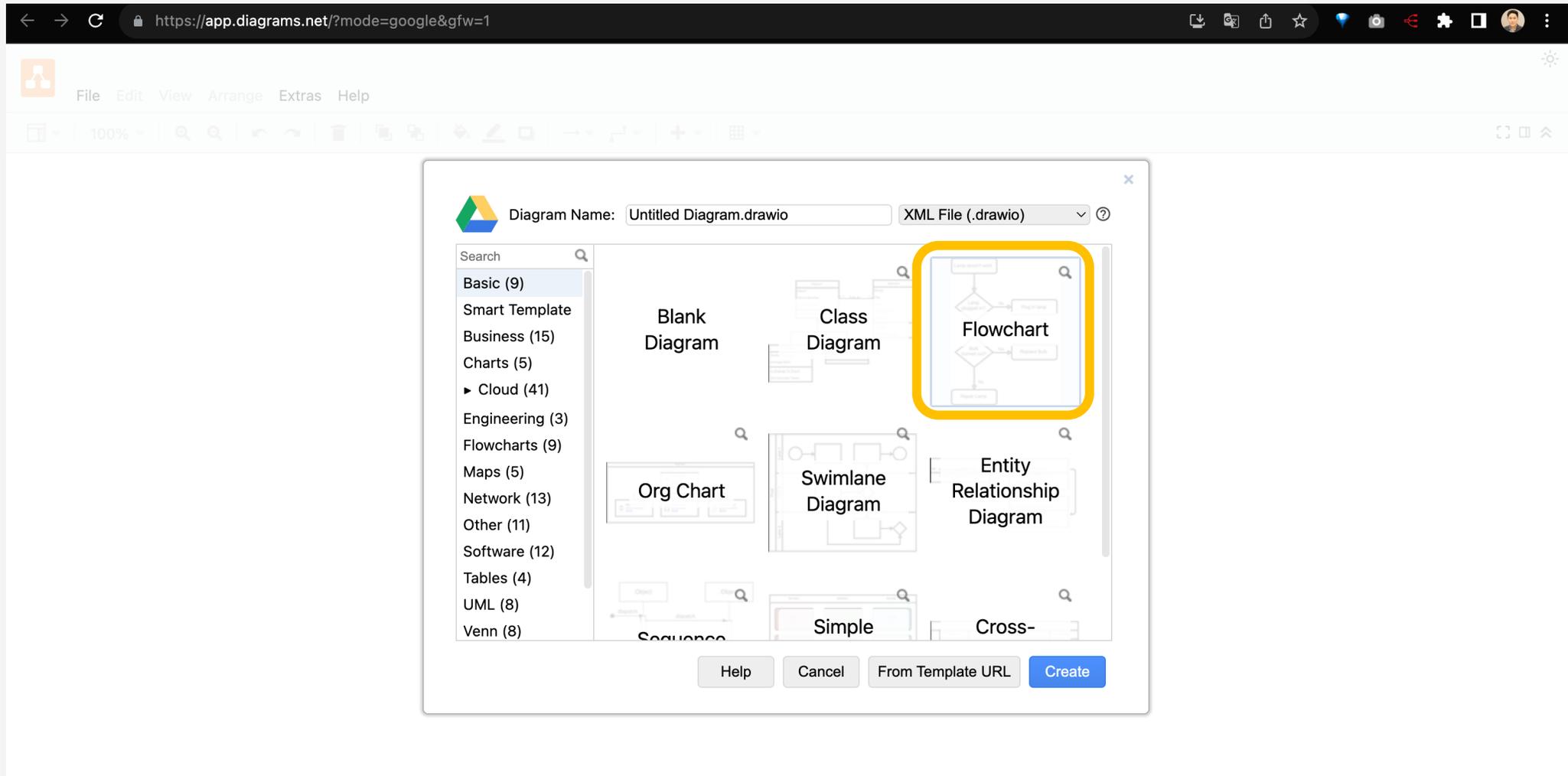
# เครื่องมือสำหรับออกแบบอัลกอริทึม

The screenshot shows the Google Workspace Marketplace interface. At the top, the search bar contains 'draw.io' and is highlighted with a yellow border. Below the search bar, there are filter buttons for 'All Filters', 'Works with', 'Price', and 'Internal apps'. The search results are displayed in a grid format. The first result, 'draw.io' by JGraph Ltd, is highlighted with a yellow border. It shows a rating of 4.6 stars and over 20 million downloads. Other results include 'Lucidchart' by Lucid Software (4.0 stars, 50M+ downloads), 'Shared Contact Dashboard for Google Workspace' by GAPPs Experts (4.9 stars, 2M+ downloads), and 'diagrams.net for D...' by JGraph (3.7 stars, 4M+ downloads). The left sidebar contains navigation options like 'Home', 'Recommended for...', 'Editor's choice', 'Intelligent apps', 'Work from everywhere', 'Business essentials', 'Apps to discover', 'Built by Google', 'Top charts', 'Most popular', 'Top rated', and 'Categories'.

# เครื่องมือสำหรับออกแบบอัลกอริทึม

The screenshot shows the Google Workspace Marketplace interface. At the top, the search bar contains 'draw.io' and is highlighted with a yellow border. Below the search bar, there are filter buttons for 'All Filters', 'Works with', 'Price', and 'Internal apps'. The search results are displayed in a grid format. The first result, 'draw.io' by JGraph Ltd, is highlighted with a yellow border. It shows a rating of 4.6 stars and over 20 million downloads. Other results include 'Lucidchart' by Lucid Software (4.0 stars, 50M+ downloads), 'Shared Contact Dashboard for Google Workspace' by GAPPs Experts (4.9 stars, 2M+ downloads), and 'diagrams.net for D...' by JGraph (3.7 stars, 4M+ downloads). The left sidebar contains navigation options like 'Home', 'Recommended for you', 'Editor's choice', 'Intelligent apps', 'Work from everywhere', 'Business essentials', 'Apps to discover', 'Built by Google', 'Top charts', 'Most popular', 'Top rated', and 'Categories'.

# เครื่องมือสำหรับออกแบบอัลกอริทึม



# เครื่องมือสำหรับออกแบบอัลกอริทึม

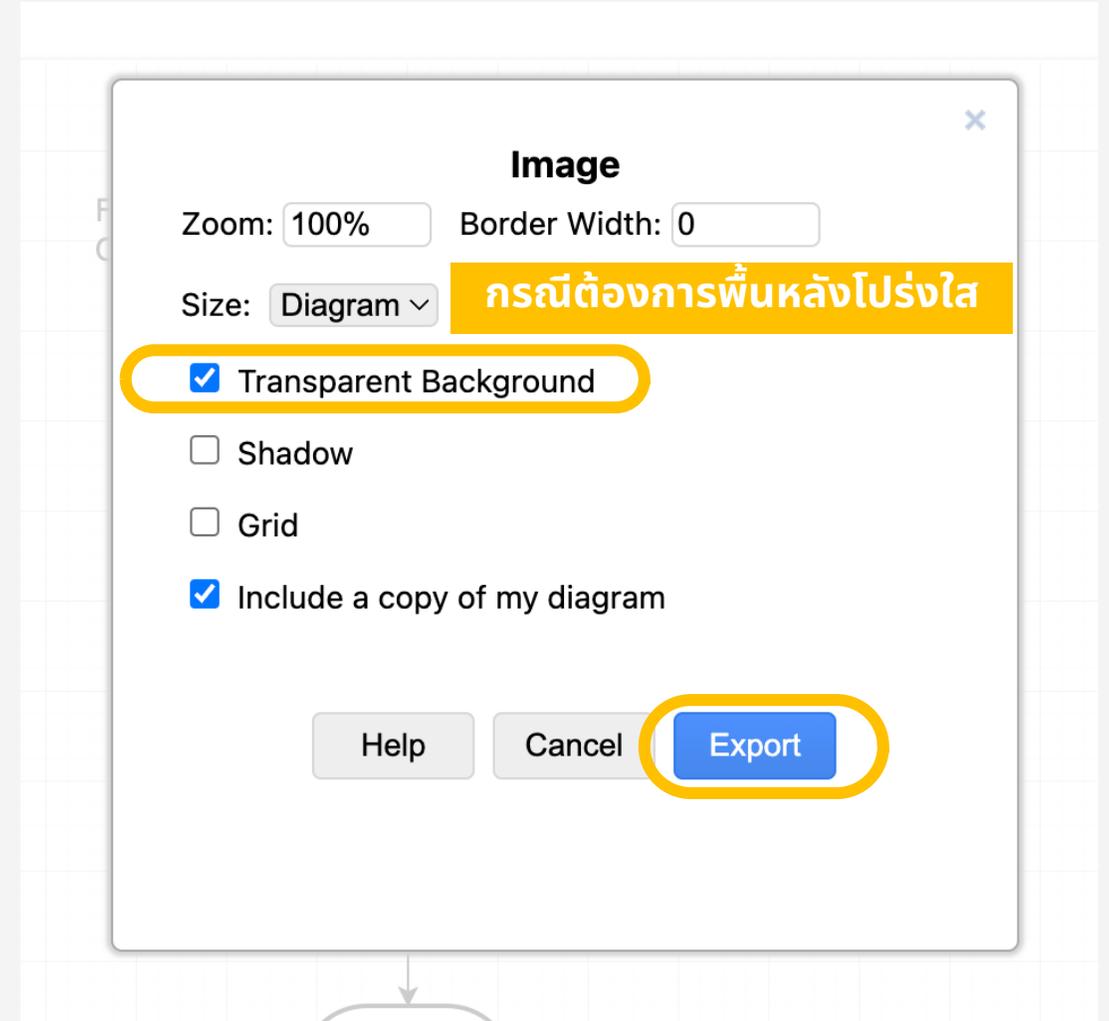
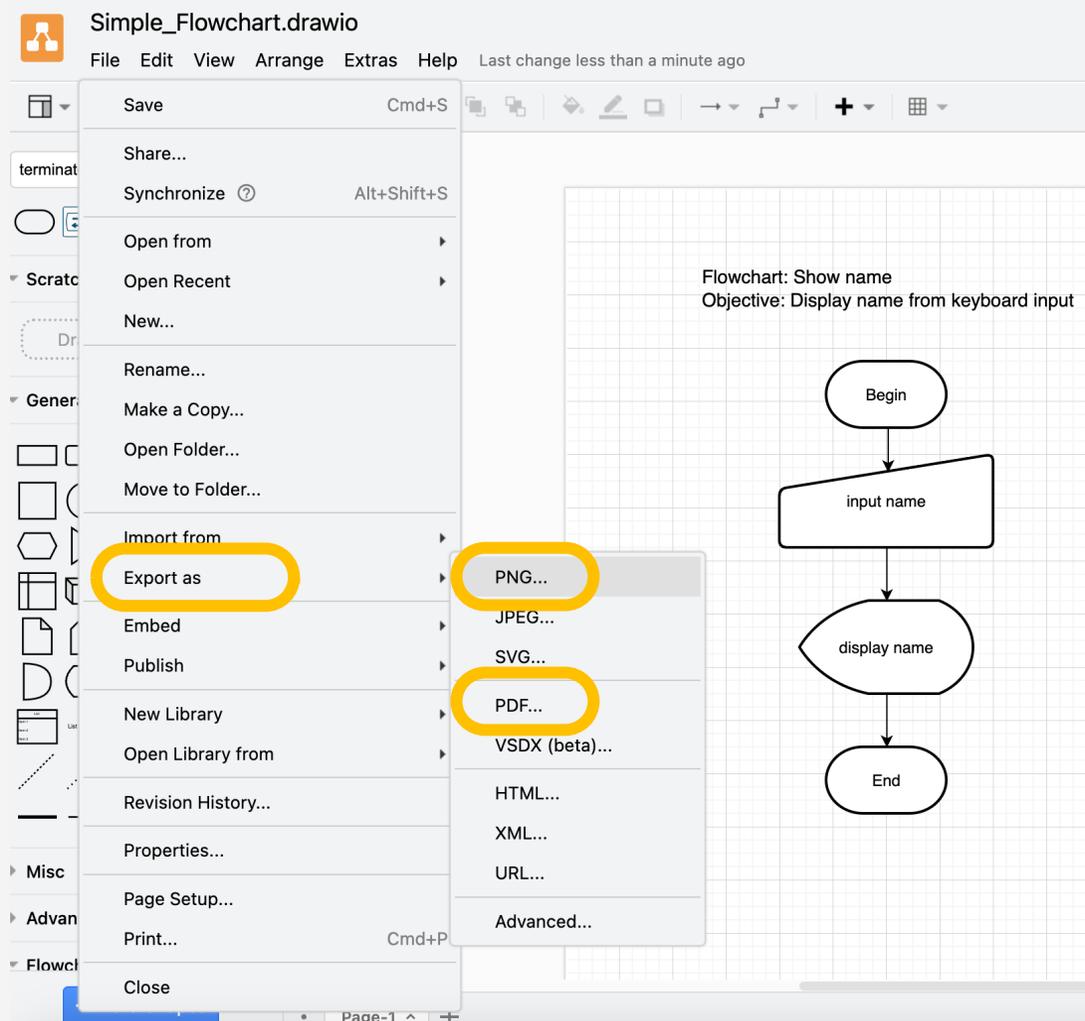
terminator

Scratchpad

Terminator

```
graph TD; A[Lamp doesn't work] --> B{Lamp plugged in?}; B -- No --> C[Plug in lamp]; B -- Yes --> D{Bulb burned out?}; D -- Yes --> E[Replace Bulb]; D -- No --> F[Repair Lamp];
```

# เครื่องมือสำหรับออกแบบอัลกอริทึม



# เครื่องมือสำหรับผู้เริ่มต้นการเขียนโปรแกรมจาก Flowchart

Recommended , Recommendation แปลว่า แนะนำ

- [www.flowgorithm.org](http://www.flowgorithm.org)

Welcome to the Flowgorithm Homepage!

Flowgorithm is a **free** beginner's programming language that is based on simple graphical flowcharts.

Typically, when a student first learns to program, they often use one of the text-based programming languages. Depending on the language, this can either be easy or frustratingly difficult. Many languages require you to write lines of confusing code just to display the text "Hello, world!".

By using flowcharts, you can concentrate on programming concepts rather than all the nuances of a typical programming language. You can also run your programs directly in Flowgorithm.

Once you understand programming logic, it is easy for you to learn one of the major languages. Flowgorithm can interactively convert your flowchart to over 18 languages. These include: C#, C++, Java, JavaScript, Lua, Perl, Python, Ruby, Swift, Visual Basic .NET, and VBA (used in Office).

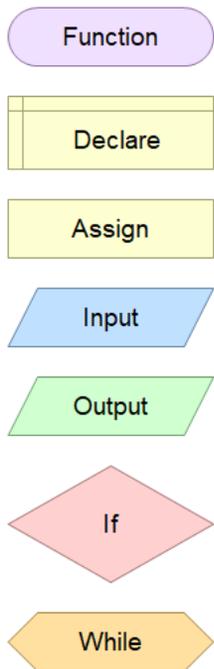
[More information](#)

```
graph TD
    Start(( )) --> A[secret = random(100) + 1]
    A --> B[/Output "Guess 1 to 100"/]
    B --> C{guess != secret}
    C -- True --> D[/Input guess/]
    C -- False --> C
    D --> E{guess > secret}
    E -- True --> F[/Output "Too HIGH"/]
    E -- False --> C
```

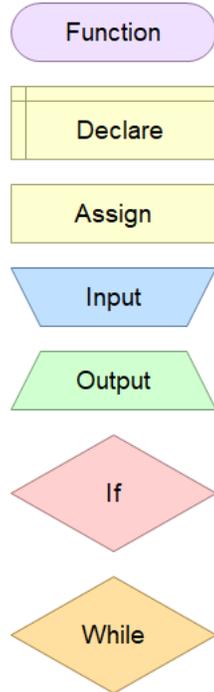
Click for a full size image

# มาตรฐานสัญลักษณ์และความหมายของ Flowchart

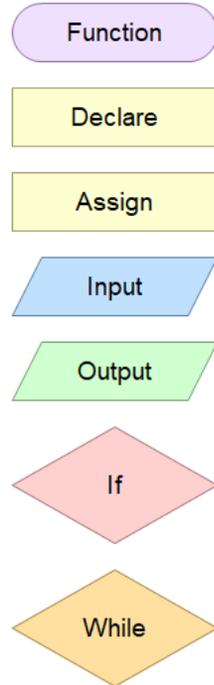
## Descriptive



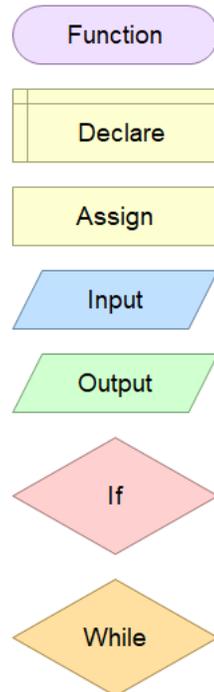
## Trapezoid



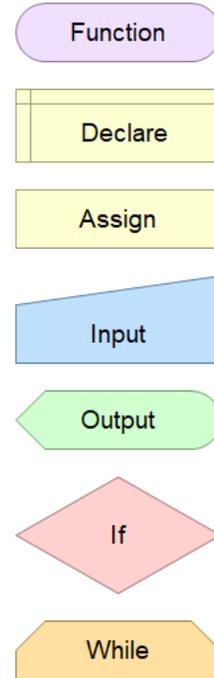
## Gaddis



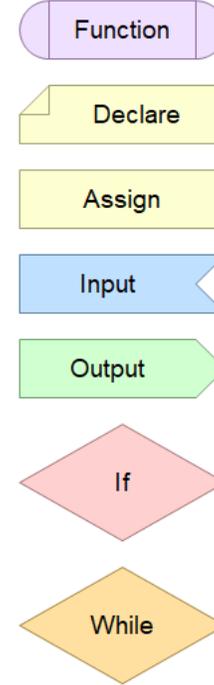
## GOST



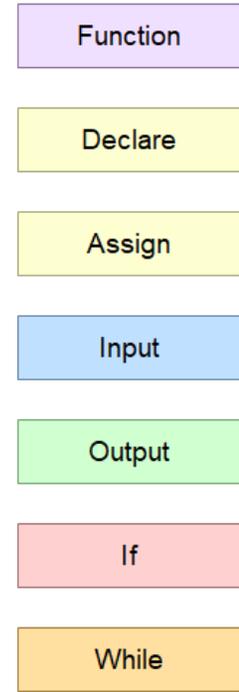
## IBM



## SDL



## Block



# มาตรฐาน Flowchart จาก ANSI: American National Standards Institute

## สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา เป็นองค์กรอิสระที่ไม่หวังผลกำไร

### Flowcharting With the ANSI Standard: A Tutorial

NED CHAPIN

infoSci Inc., Menlo Park, California

The ISO and ANSI X3.5 standard flowchart symbols and their usage in information processing are explained and examples given. The two main categories of flowchart—the system chart or run diagram, and the flow diagram or block diagram—are stressed. For each, the outline symbols and their manner of use are presented, as well as guidelines and conventions, such as cross-referencing. In the case of flow diagrams, notation is presented for use within the outline symbols.

**Key words and phrases:** standards, flowchart, flow diagram, system chart, run diagram, block diagram, program flowchart, documentation, outlines, boxes, notation, symbols, program description, system description, algorithm statement, communication

**CR categories:** 1.3, 2.2, 2.43, 4.0

#### HISTORICAL DEVELOPMENT

Flowcharting is a means of graphically stating ways of solving information handling problems. Flowcharting, as people use the term in working with computers, must be distinguished from other graphic aids. For stating clerical procedures, such as those used in systems and procedures work, people use a graphic means which has also been the subject of a standard [4]. But it is quite different from the standard under discussion here. Logic designers also use graphic aids for stating the character of the machines they design for handling information. These too have been the subject of a standard [3, 19]. The emphasis in this tutorial paper is on stating information handling problems where the information handling is done at least in major part with the aid of the automatic computer [7].

The intellectual father of flowcharting is John von Neumann. He and his associates at Princeton University's Institute for Advanced Study were the first to use

Copyright © 1970, by Ned Chapin

graphic aids systematically for this purpose and publish their use [11]. Even though the details of the flowcharting as the standard specifies it today differ considerably from what they advocated, the spirit, the philosophy, and the rationale remain much as they presented them.

For their own internal purposes and for dealing with customers, each of the major computer manufacturers has over the course of the years developed, adopted, published, modified, and advocated flowcharting conventions (see, for example, [13, 15]). These have differed from vendor to vendor, in part deliberately as an attempt to distinguish one vendor from the competing vendors, and in part out of a sincere attempt to reflect what each has felt to be unique differences in their philosophy and approach to information processing problems.

Users of computers have individually and collectively made decisions on flowcharting conventions. Most small and medium and many large computer users have adopted the conventions presented to them by the vendor of the computer they

Computing Surveys, Vol 2, No. 2, June 1970

122

Ned Chapin

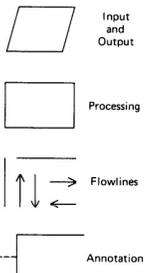


Fig. 1. Basic outlines

trained in the English language: from top to bottom and from left to right. Where the flow follows this normal pattern, no open arrowheads are needed to remind the reader. In the event of any significant deviation from this pattern, arrowheads are required to signal the deviation to the reader's attention. Whenever the direction of flow might be ambiguous to a reader, arrowheads should be used to provide clarification. Bidirectional flow may be indicated by dual arrows each with open arrowheads, or less preferably by open arrowheads in both directions on single flowlines.

The annotation outline provides a way to supply descriptive information, comments, and explanatory notes. Its dashed line indicates the outline to which this explanation or clarification applies.

#### Additional Outlines

The additional outlines are for the convenience of the reader, and not for the purpose of describing data-processing action. These symbols provide for handling the limitations of pages of various sizes, and make it more convenient to show connections in the sequences of flow. These outlines are shown in Figure 2.

The connector outline, a circle, must in practice be used at least in pairs. To that end, the standard advances two varieties, the inconnector or entry connector, and the outconnector or exit connector. An inconnector or entrance has a flowline leaving it but none entering it; an outconnector or exit has a flowline entering it but none leaving it. Each inconnector may have from zero through any number of outconnectors associated with it. However, each outconnector must have exactly one inconnector associated with it. One function of the connector outline is to enable a long sequence of outlines (a "flow") to be broken into pieces to fit conveniently on a page. The connector outline also provides ways of joining together convergent lines of flow that fan in to some particular point. And, it provides a way of identifying divergent lines of flow.

The terminal connector outline serves to indicate a beginning, an end, or a break in the usual line of flow. In the first two uses, it substitutes for an ordinary connector at the beginning and the end of major portions of a sequence of outlines (a "flow"), particularly when these portions are identified by a name, as, for example, for a closed subroutine. In its

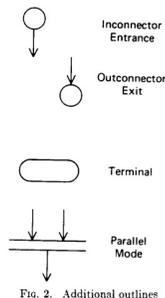


Fig. 2. Additional outlines

Computing Surveys, Vol. 2, No. 2, June 1970

128

Ned Chapin

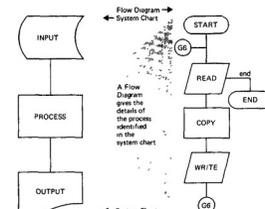


Fig. 10. Relationship between flow diagram and system chart

#### USE OF THE STANDARD

##### Situations

The ANSI standard flowchart symbols for information processing cover two major situations. One situation is for representing algorithms, especially those for execution by a computer. The other is for representing systems without indicating the character of the component algorithms. Some other situations are noted briefly later in this paper. The term "flowchart," as used in the standard, may therefore refer to either of these situations.<sup>2</sup> Hereafter in this paper a clear distinction is necessary between flowcharts of systems and flowcharts of algorithms. Hereafter, "flow diagram" designates a flowchart of an algorithm, and "system chart" designates a flowchart of a system.

Other terms are also current in the field for these two situations. Thus other terms sometimes used for flow diagram are block diagram, logic chart, and process chart, as well as flowchart, i'or system chart, <sup>3</sup>The standard's definition of the term "flowchart" is a subversion, well supported by popular usage, of a far older definition. The term "flowchart" has a history predating the use of computers. In the field of systems analysis, it historically has designated a graphic aid to analysis quite different from that contemplated in the standard. This older use of the term is illustrated in P. 1963 ed., pp. 227-239, and three forms of flowcharts following this older definition are illustrated in [6, Ch. 5].

Computing Surveys, Vol. 2, No. 2, June 1970

other terms are run diagram, procedure chart, and flowchart.

The distinction between the flow diagram and the system chart is vital because the use of the standard differs considerably for these two. In the case of the system chart, the focus is upon the inputs and the outputs produced by the sequences of runs, programs, or procedures. In contrast, the focus in the flow diagram is upon the sequences of data transformations needed to produce an output data structure from an input data structure. The flow diagram tells "how," whereas a system chart identifies programs, runs, or procedures by name and data structures by name, the flow diagram identifies individual operations on portions of data structures. The flow diagram is usually an elaboration of what is indicated by a single process outline in a system chart (see Figure 10).

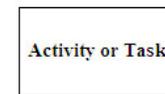
In the remainder of this paper, system chart conventions and system chart guidelines are considered first. These use a greater variety of outlines, but the logical complexity is relatively low. Then flow diagram conventions and flow diagram guidelines are discussed. Flow diagrams can become logically complex even though the number of different outlines utilized is typically fewer.

##### System Chart Conventions

**Basic format.** The basic format of the system chart follows a sandwich rule—that is, it is composed of alternating layers of data identifications and process identifications. The data identifications are equivalent to the bread of the sandwich, and the process identifications are equivalent to the filling in the sandwich. Just as sandwiches may be of the Dagwood type, so the output produced from one process operation may serve as the input for a following process operation (a compound system chart). But a system chart must always begin with inputs (data identifications) and must always end with outputs (data identifications).

To see this sandwich rule in use, con-

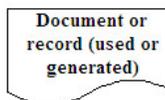
Figure 1: FLOWCHARTING SYMBOLS AND USE



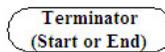
Depict time consuming steps in a process.  
*Examples: Inspect a product, machine a part, etc.*



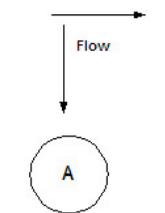
Depict questions/checks and show alternative outcomes depending on the result.  
*Examples: Is the part nonconforming? If yes, do one thing. If no, do another.*



Depict documents used to perform tasks or records generated by the activity.  
*Examples: Work Instructions (in) Scrap Record (out)*



Depict the start or end of a procedure or work instruction.



Depict the flow of logic, decisions, or steps of documentation.

Depict transfer of flow to a location or show the identity of a location.

# มาตรฐาน ISO 5807:1985 Flowchart



International Standard  5807

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ / ORGANISATION INTERNATIONALE DE NORMALISATION

**Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts**

Traitement de l'information — Symboles de documentation et conventions applicables aux données, aux organigrammes de programmation et d'analyse, aux schémas des réseaux de programmes et des ressources de système

First edition — 1985-02-15

<https://standards.iteh.ai/catalog/standards/sist/e1a4e558-0311-46cd-91f7-a9abc479294a/iso-5807-1985>

UDC 681.3 : 003.62 / 63 Ref. No. ISO 5807:1985 (E)

Descriptors : data processing, information interchange, computer programs, symbols, graphic methods, charts, flowcharts.

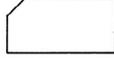
ISO 5807:1985 (E)

This symbol represents data, the medium being internal storage.



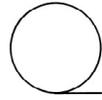
9.1.2.2 Sequential access storage

This symbol represents data that is only sequentially accessible, the medium being, for example, magnetic tape, tape cartridge, tape cassette.



9.1.2.3 Direct access storage

This symbol represents data directly accessible, the medium being, for example, magnetic disk, drum, flexible disk.



9.1.2.4 Document

This symbol represents human readable data, the medium being, for example, printed output, an OCR or MICR document, microfilm, tally roll, data entry forms.



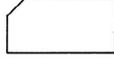
9.1.2.5 Manual input

This symbol represents data, the medium being of any type where the information is entered manually at the time of processing, for example, on-line keyboard, switch settings, push buttons, light pen, bar-code wand.



9.1.2.6 Card

This symbol represents data, the medium being cards, for example, punched cards, magnetic cards, mark sense cards, stub cards, mark scan cards.



9.1.2.7 Punched tape

This symbol represents data, the medium being paper tape.



9.1.2.8 Display

This symbol represents data, the medium being of any type where the information is displayed for human use, for example, video screens, on-line indicators.



9.2 Process symbols

9.2.1 Basic process symbol

Process

This symbol represents any kind of processing function, for example, executing a defined operation or group of operations resulting in a change in value, form or location of information, or in the determination of which one of several flow directions is to be followed.



## ISO 5807:1985

Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts

- เป็นมาตรฐานที่เกี่ยวกับการนำเสนอเอกสารที่มีรูปแบบต่าง ๆ :
- 1.การนำเสนอเอกสาร:** ISO 5807 กำหนดหลักการในการนำเสนอเอกสารที่มีความซับซ้อน รวมถึงการนำเสนอเนื้อหาที่สอดคล้องกับเนื้อหาหลักที่เสนอในเอกสาร
  - 2.รูปแบบของหน้ากระดาษ:** มาตรฐานนี้กำหนดรูปแบบของหน้ากระดาษที่ถูกต้องเพื่อให้การนำเสนอเอกสารเป็นไปได้อย่างมีประสิทธิภาพและมีส่วนประกอบที่เหมาะสม
  - 3.ตัวหนังสือและการเขียน:** ISO 5807 ระบุวิธีการใช้ตัวหนังสือที่เหมาะสม รวมถึงการเขียนและการจัดเรียงข้อความในเอกสาร เพื่อให้ผู้อ่านสามารถเข้าใจและติดตามได้ง่าย
  - 4.การนำเสนอข้อมูล:** มาตรฐานนี้คำนึงถึงการนำเสนอข้อมูลให้เป็นระเบียบและชัดเจน โดยการใช้ตัวเลข เกมเพลต เซิงกราฟ หรือสัญลักษณ์ที่เหมาะสม
  - 5.ตารางและภาพ:** มาตรฐานนี้เน้นความชัดเจนในการนำเสนอตารางและภาพ รวมถึงการอธิบายและคำอธิบายของตารางและภาพเหล่านั้น
  - 6.เครื่องหมายและสัญลักษณ์:** มาตรฐานระบุการใช้เครื่องหมายและสัญลักษณ์ที่เหมาะสมเพื่อช่วยในการนำเสนอข้อมูลและแสดงความหมาย
  - 7.เนื้อหาเพิ่มเติม:** มาตรฐานนี้ระบุถึงข้อมูลเพิ่มเติมที่สามารถนำมาใช้ในการเพิ่มความเข้าใจในเอกสาร เช่น การสร้างดัชนี เรื่องย่อ เป็นต้น

สรุปคือ **ISO 5807:1985** เป็นมาตรฐานที่ระบุหลักการและแนวปฏิบัติในการนำเสนอเอกสารที่มีความซับซ้อนเพื่อให้ผู้อ่านเข้าใจและติดตามได้ง่าย โดยให้คำแนะนำเกี่ยวกับรูปแบบหน้ากระดาษ ตัวหนังสือ การเขียน การนำเสนอข้อมูล ตาราง ภาพ เครื่องหมาย และเนื้อหาเพิ่มเติมที่เกี่ยวข้อง.

# มาตรฐาน ISO 5807:1985 Flowchart



## International Standard 5807

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ / ORGANISATION INTERNATIONALE DE NORMALISATION

### Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts

Traitement de l'information — Symboles de documentation et conventions applicables aux données, aux organigrammes de programmation et d'analyse, aux schémas des réseaux de programmes et des ressources de système

First edition — 1985-02-15

(standards.iteh.ai)

ISO 5807:1985  
<https://standards.iteh.ai/catalog/standards/sist/e1a4e558-0311-46cd-91f7-a9abc479294a/iso-5807-1985>

UDC 681.3 : 003.62 / 63

Ref. No. ISO 5807:1985 (E)

Descriptors : data processing, information interchange, computer programs, symbols, graphic methods, charts, flowcharts.

ISO 5807:1985 (E)

This symbol represents data, the medium being internal storage.



9.1.2.2 Sequential access storage

This symbol represents data that is only sequentially accessible, the medium being, for example, magnetic tape, tape cartridge, tape cassette.



9.1.2.3 Direct access storage

This symbol represents data directly accessible, the medium being, for example, magnetic disk, drum, flexible disk.



9.1.2.4 Document

This symbol represents human readable data, the medium being, for example, printed output, an OCR or MICR document, microfilm, tally roll, data entry forms.



9.1.2.5 Manual input

This symbol represents data, the medium being of any type where the information is entered manually at the time of processing, for example, on-line keyboard, switch settings, push buttons, light pen, bar-code wand.



9.1.2.6 Card

This symbol represents data, the medium being cards, for example, punched cards, magnetic cards, mark sense cards, stub cards, mark scan cards.



9.1.2.7 Punched tape

This symbol represents data, the medium being paper tape.



9.1.2.8 Display

This symbol represents data, the medium being of any type where the information is displayed for human use, for example, video screens, on-line indicators.

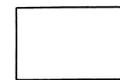


9.2 Process symbols

9.2.1 Basic process symbol

Process

This symbol represents any kind of processing function, for example, executing a defined operation or group of operations resulting in a change in value, form or location of information, or in the determination of which one of several flow directions is to be followed.



## ISO 5807:1985

Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts

เป็นมาตรฐานที่เกี่ยวข้องกับการนำเสนอเอกสารที่มีรูปแบบต่าง ๆ :

- 1.การนำเสนอเอกสาร: ISO 5807 กำหนดหลักการในการนำเสนอเอกสารที่มีความซับซ้อน รวมถึงการนำเสนอเนื้อหาที่สอดคล้องกับเนื้อหาหลักที่เสนอในเอกสาร
- 2.รูปแบบของหน้ากระดาษ: มาตรฐานนี้กำหนดรูปแบบของหน้ากระดาษที่ถูกต้องเพื่อให้การนำเสนอเอกสารเป็นไปได้อย่างมีประสิทธิภาพและมีส่วนประกอบที่เหมาะสม
- 3.ตัวหนังสือและการเขียน: ISO 5807 ระบุวิธีการใช้ตัวหนังสือที่เหมาะสม รวมถึงการเขียนและการจัดเรียงข้อความในเอกสาร เพื่อให้ผู้อ่านสามารถเข้าใจและติดตามได้ง่าย
- 4.การนำเสนอข้อมูล: มาตรฐานนี้คำนึงถึงการนำเสนอข้อมูลให้เป็นระเบียบและชัดเจน โดยการใช้ตัวเลข เกมเพลต เซิงกราฟ หรือสัญลักษณ์ที่เหมาะสม
- 5.ตารางและภาพ: มาตรฐานนี้เน้นความชัดเจนในการนำเสนอตารางและภาพ รวมถึงการอธิบายและคำอธิบายของตารางและภาพเหล่านั้น
- 6.เครื่องหมายและสัญลักษณ์: มาตรฐานระบุการใช้เครื่องหมายและสัญลักษณ์ที่เหมาะสมเพื่อช่วยในการนำเสนอข้อมูลและแสดงความหมาย
- 7.เนื้อหาเพิ่มเติม: มาตรฐานนี้ระบุถึงข้อมูลเพิ่มเติมที่สามารถนำมาใช้ในการเพิ่มความเข้าใจในเอกสาร เช่น การสร้างดัชนี เรื่องย่อ เป็นต้น

สรุปคือ ISO 5807:1985 เป็นมาตรฐานที่ระบุหลักการและแนวปฏิบัติในการนำเสนอเอกสารที่มีความซับซ้อนเพื่อให้ผู้อ่านเข้าใจและติดตามได้ง่าย โดยให้คำแนะนำเกี่ยวกับรูปแบบหน้ากระดาษ ตัวหนังสือ การเขียน การนำเสนอข้อมูล ตาราง ภาพ เครื่องหมาย และเนื้อหาเพิ่มเติมที่เกี่ยวข้อง.

# ประเภทโครงสร้างของผังงาน

**Sequence: เรียงตามลำดับ**    ชอบพูดกันว่า " เรียงตาม sequence "

ผังงานโครงสร้างแบบเรียงลำดับ  
Sequential Structure

มีขั้นตอนการทำงานชัดเจน

**Conditional Structure / Condition / Selection Structure**

ผังงานโครงสร้างแบบตัดสินใจ  
Decision Structure

มีทางเลือกในการทำงาน

**Looping Structure / Iterative / Repetition Structure**

ผังงานโครงสร้างแบบทำซ้ำ  
Iteration Structure

มีการทำงานอย่างอัตโนมัติ

# ผังงานโครงสร้างแบบเรียงลำดับ

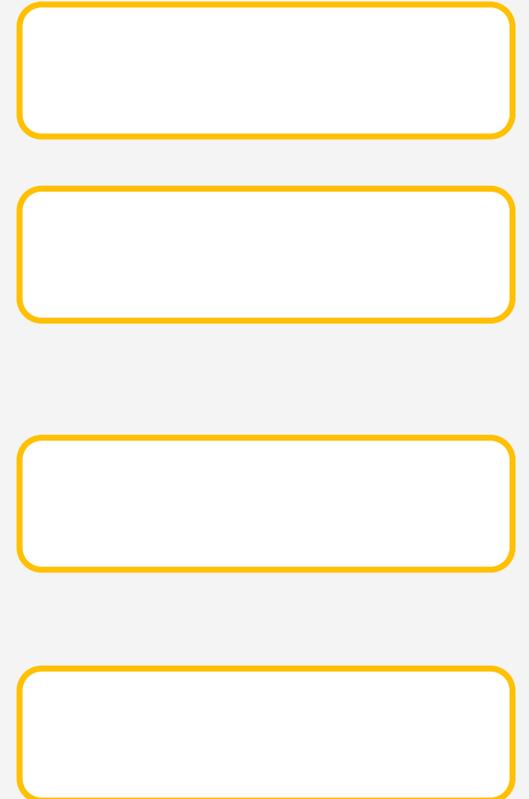
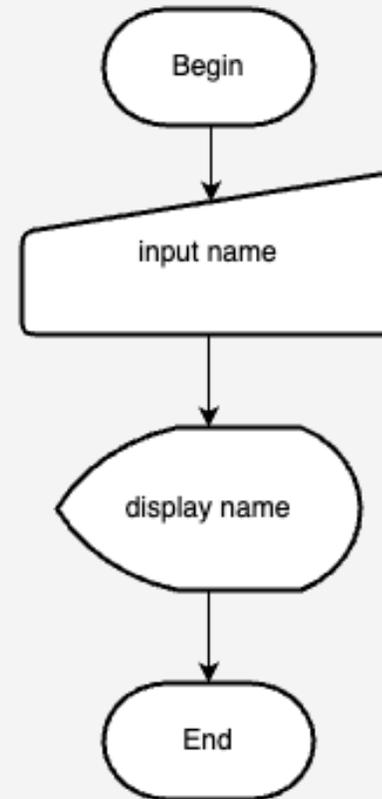
## Sequential Structure

ผังงานโครงสร้างแบบเรียงลำดับ (Sequential Structure) คือ

ผังงานที่แสดงขั้นตอนเรียงลำดับตั้งแต่เริ่มต้น  
ขั้นตอนแรก ขั้นตอนถัดไป จนถึง ขั้นตอนสุดท้าย  
และสิ้นสุดการทำงาน เป็นโครงสร้างที่เรียบง่ายที่สุด

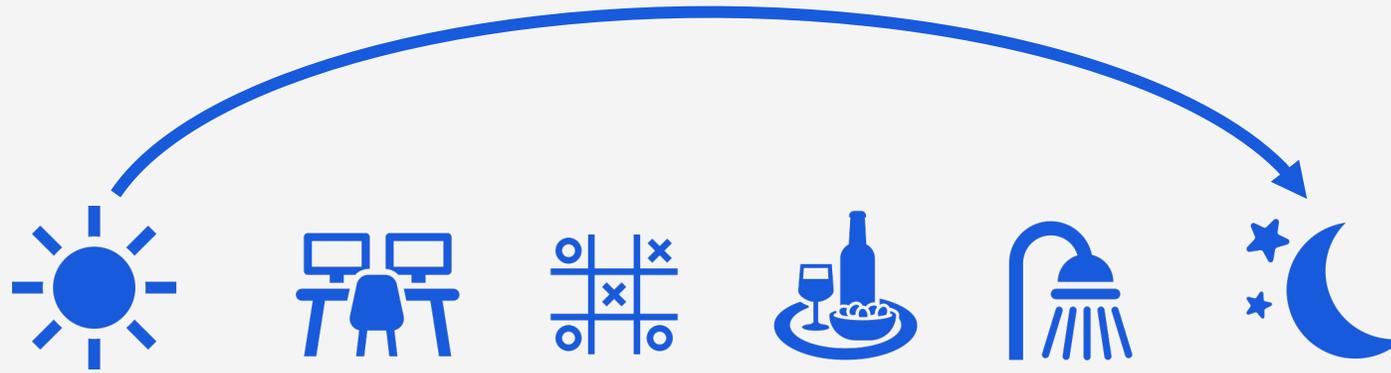


Flowchart: Show name  
Objective: Display name from keyboard input



# ผังงานโครงสร้างแบบเรียงลำดับ

## Sequential Structure



ในชีวิตประจำวัน มีกระบวนการ หรือ ขั้นตอนการใช้ชีวิตใดบ้าง  
ที่มีลักษณะเป็นโครงสร้างแบบเรียงลำดับ?



Word Cloud

# ผังงานโครงสร้างแบบเรียงลำดับ

## Sequential Structure

**โจทย์:** ต้องการป้อนค่าด้วยแป้นพิมพ์และแสดงผลรวมของจำนวน 2 จำนวนออกทางหน้าจอ

### รหัสเทียม (Pseudocode)

เริ่มต้นการทำงาน

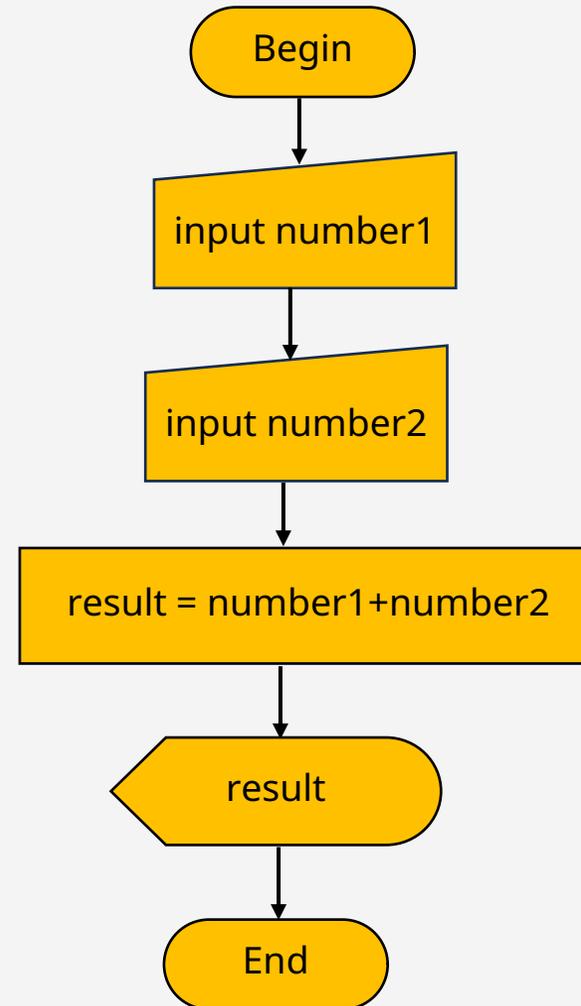
ป้อนค่าของจำนวนที่ 1

ป้อนค่าของจำนวนที่ 2

ผลรวม = จำนวนที่ 1 + จำนวนที่ 2

แสดง ผลรวม

สิ้นสุดการทำงาน



# ผังงานโครงสร้างแบบเรียงลำดับ

## Sequential Structure

**โจทย์:** ต้องการป้อนค่าด้วยแป้นพิมพ์และแสดงผลรวมของจำนวน 2 จำนวนออกทางหน้าจอ

### รหัสเทียม (Pseudocode)

เริ่มต้นการทำงาน

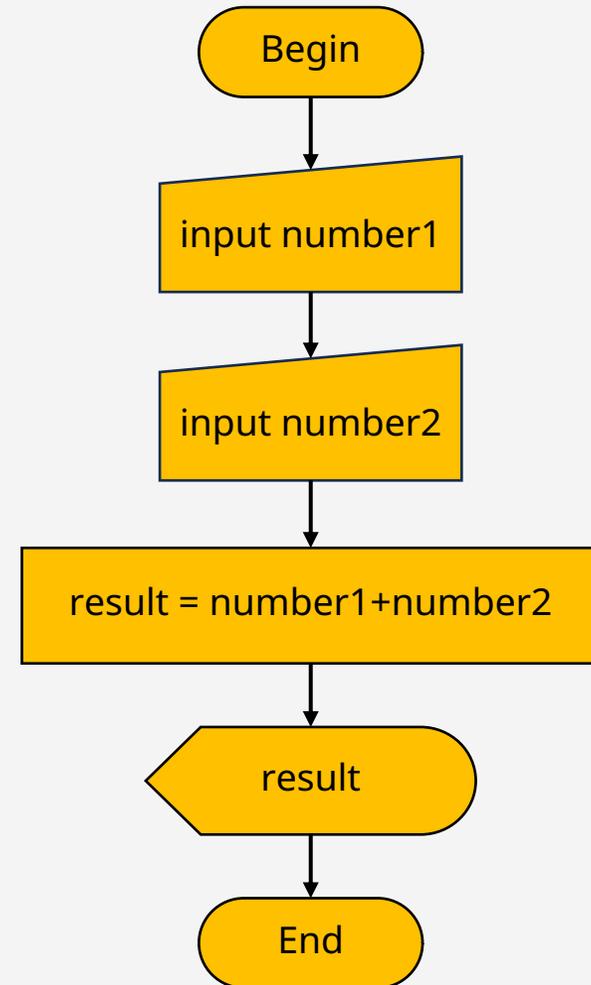
ป้อนค่าของจำนวนที่ 1

ป้อนค่าของจำนวนที่ 2

ผลรวม = จำนวนที่ 1 + จำนวนที่ 2

แสดง ผลรวม

สิ้นสุดการทำงาน



# ปัญหา: ต้องการคำนวณอายุ จาก ปี พ.ศ.เกิด

# ปัญหา: ต้องการหาพื้นที่สี่เหลี่ยมจัตุรัส?

สูตร คือ  $\text{area} = \text{side} \times \text{side}$

หรือ  $\text{area} = \text{side}^2$

# ปัญหา: ต้องการหาพื้นที่วงกลม?

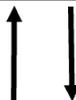
สูตร คือ  $area = Pi \times r \times r$

หรือ  $area = Pi \times r^2$

# ปัญหา: ต้องการหาเส้นรอบวงกลม?

สูตร คือ  $ring = 2 \times Pi \times r$

# ปัญหา: ต้องการทราบว่า ปี พ.ศ. ที่ปีก่อนเข้ามา คือ ค.ศ. อะไร?

สัญลักษณ์	ความหมาย	ภาษาอังกฤษ
	หมายถึง จุดเริ่มต้น หรือ จุดสิ้นสุดการทำงาน	Begin และ End
	หมายถึง ทิศทางการไหลของงาน	Flow line
	หมายถึง การดำเนินการ หรือ การประมวลผล	Process
	หมายถึง การนำเข้าข้อมูลด้วยการป้อนค่าทางแป้นพิมพ์	Manual input
	หมายถึง การตัดสินใจตามเงื่อนไขที่กำหนดไว้	Decision
	หมายถึง การแสดงผลบนหน้าจอ	Display

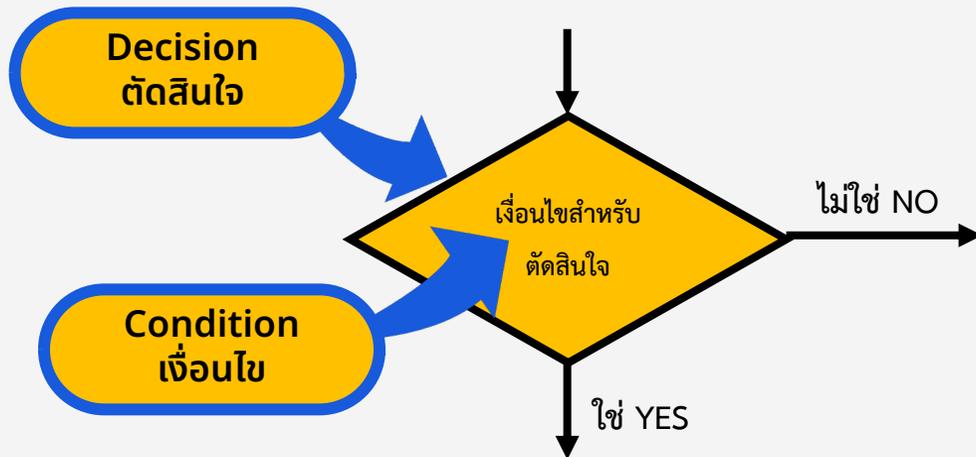
อัลกอริทึมที่เป็นข้อความ	อัลกอริทึมที่เป็นผังงาน
1. รับเงินจากคุณพ่อเป็นจำนวน 50 บาท	เริ่มต้น
2. จ่ายค่ารถเพื่อเดินทางไปโรงเรียน จำนวน 9 บาท	รายรับ = ..... บาท
3. จ่ายค่าข้าวกลางวัน จำนวน 15 บาท	รายจ่ายค่ารถขาไป = ..... บาท
4. จ่ายค่าน้ำเปล่า จำนวน 5 บาท	รายจ่ายค่าข้าวกลางวัน ..... 15 บาท
5. จ่ายค่าขนม จำนวน 10 บาท	รายจ่ายค่าน้ำเปล่า = ..... บาท
6. จ่ายค่าดินสอ จำนวน 4 บาท	รายจ่ายค่าขนม ..... 10 บาท
7. จ่ายค่ารถเพื่อเดินทางกลับบ้าน จำนวน 9 บาท	รายจ่ายค่าดินสอ = ..... บาท
8. หาผลรวมของรายจ่าย	รายจ่ายค่ารถขากลับ ..... 9 บาท
9. ยอดเงินคงเหลือมาจากรายรับหักลบด้วยรายจ่าย	หาผลรวมของ.....
10. แสดงยอดเงินคงเหลือออกทางหน้าจอ	..... = ..... - ผลรวมของรายจ่าย
	ยอดเงินคงเหลือ
	สิ้นสุด

# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure

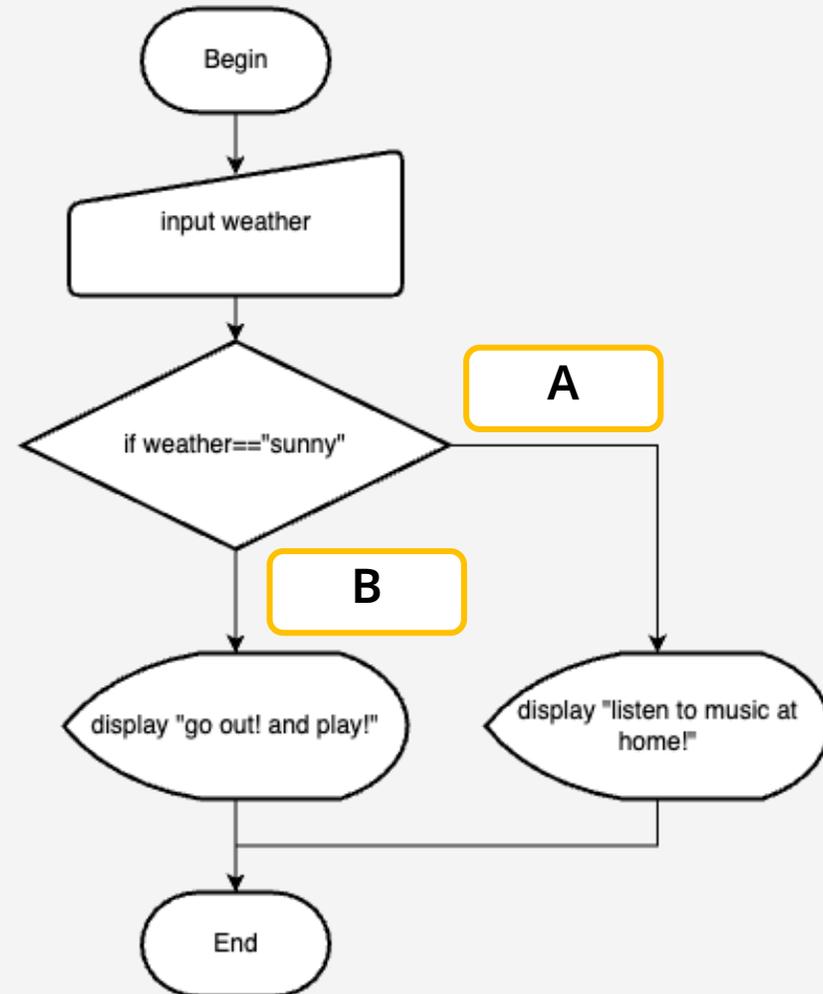
ผังงานโครงสร้างแบบตัดสินใจ (Decision Structure)

คือผังงานทางเลือกในการตัดสินใจ (Decision) โดยภายในบรรจุเงื่อนไข (Condition) เป็นสิ่งที่ใช้ในการตัดสินใจ



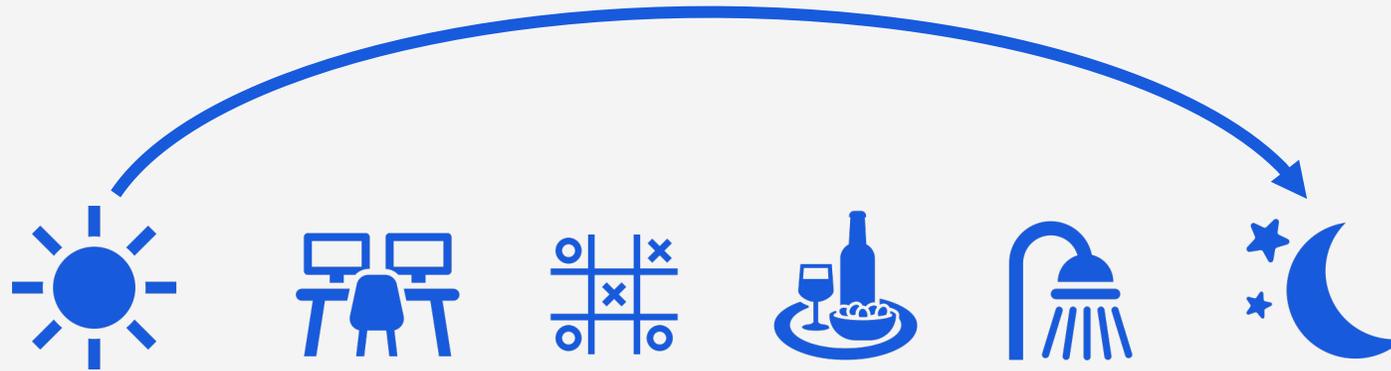
Flowchart: Play or Not?

Objective: Evaluate the weather conditions in deciding to go out and play.



# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure



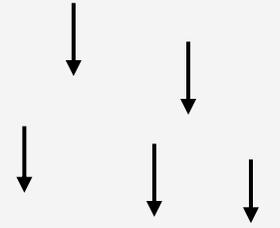
ในชีวิตประจำวัน มีกระบวนการ หรือ ขั้นตอนการใช้ชีวิตใดบ้าง  
ที่มีลักษณะเป็นโครงสร้างแบบตัดสินใจ?

 Short Answer

# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure

**โจทย์:** ต้องการตรวจสอบรหัสผ่านที่ Digital Door Lock



### รหัสเทียม (Pseudocode)

ถ้า รหัสผ่าน ไม่เหมือนกับ รหัสผ่านที่ตั้งไว้

แสดง "รหัสผ่านถูกต้อง"

ถ้า รหัสผ่าน เหมือนกับ รหัสผ่านที่ตั้งไว้

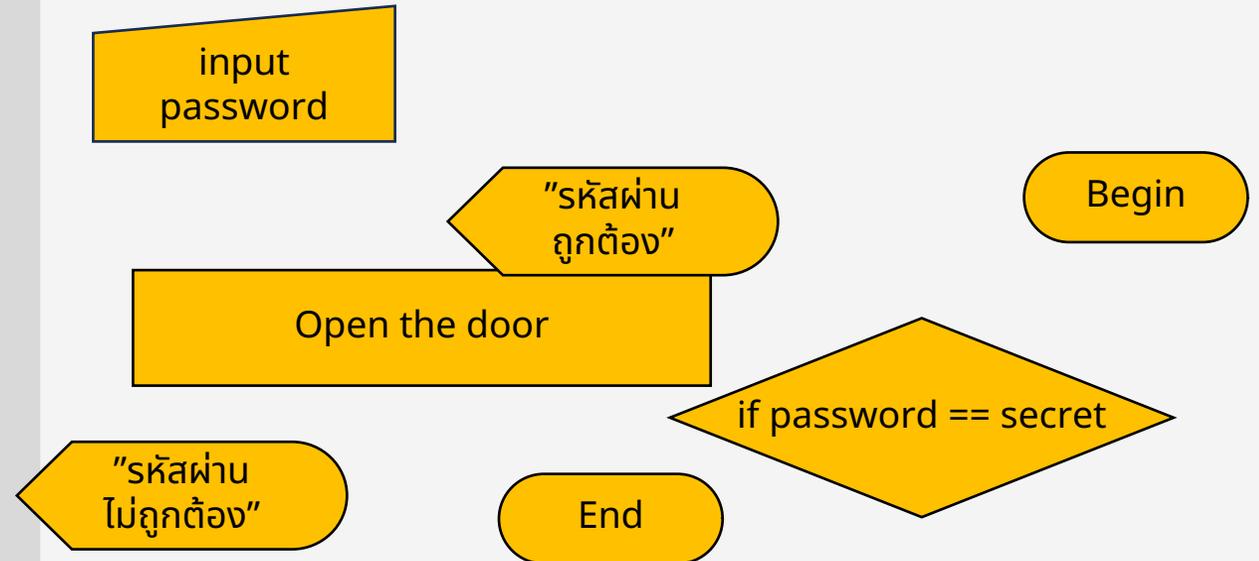
กลอนประตูเปิด

เริ่มต้นการทำงาน

ป้อนรหัสผ่าน

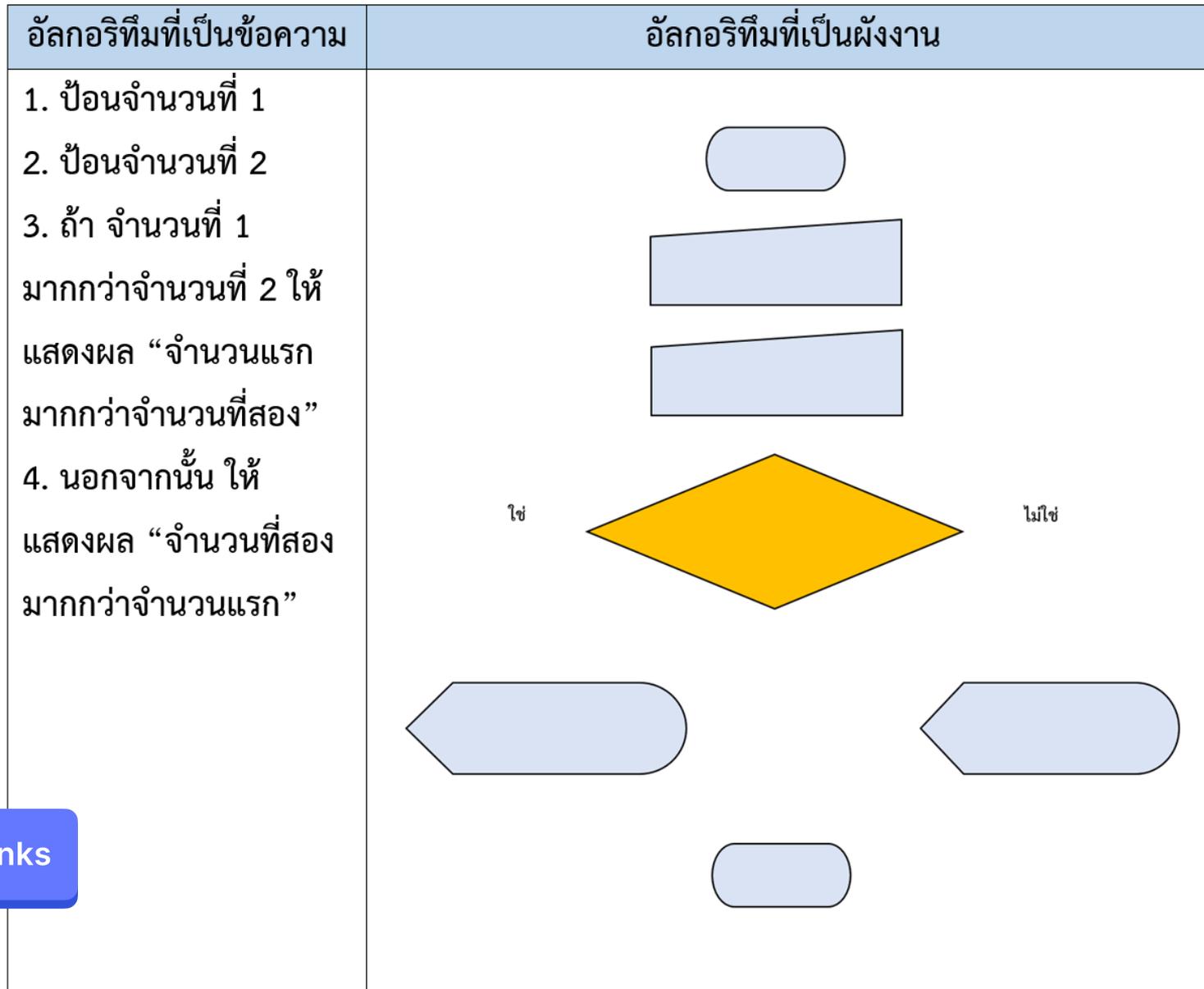
สิ้นสุดการทำงาน

แสดง "รหัสผ่านไม่ถูกต้อง"



# กิจกรรม: เติมสัญลักษณ์และข้อความภายในผังงานให้สมบูรณ์

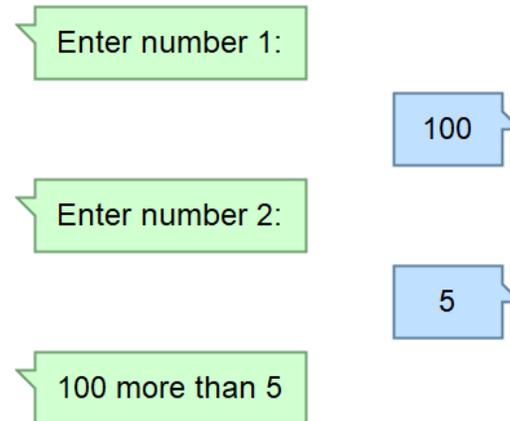
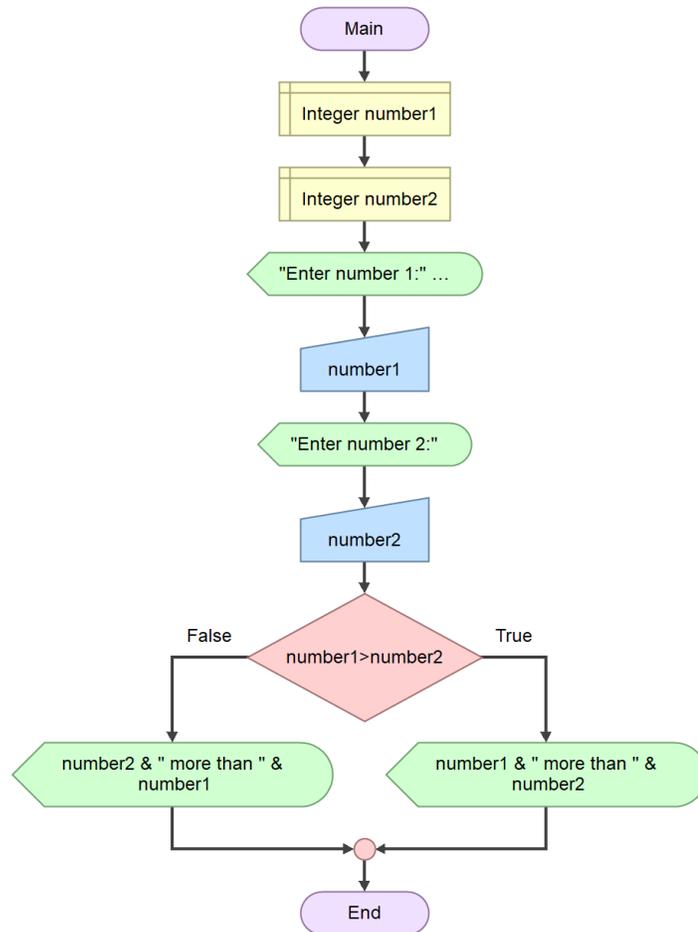
วัตถุประสงค์: ต้องการตรวจสอบจำนวนที่มากที่สุดจากจำนวน 2 จำนวน



 **Fill in the Blanks**

# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure



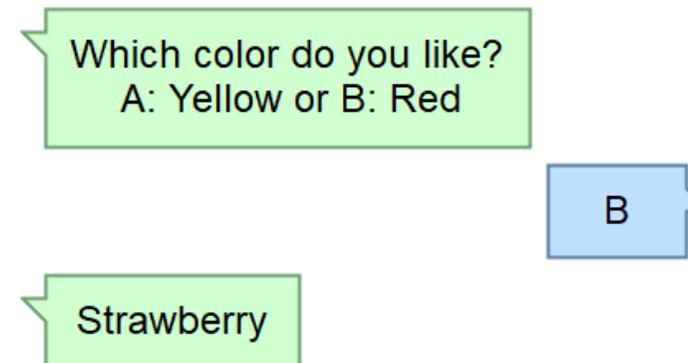
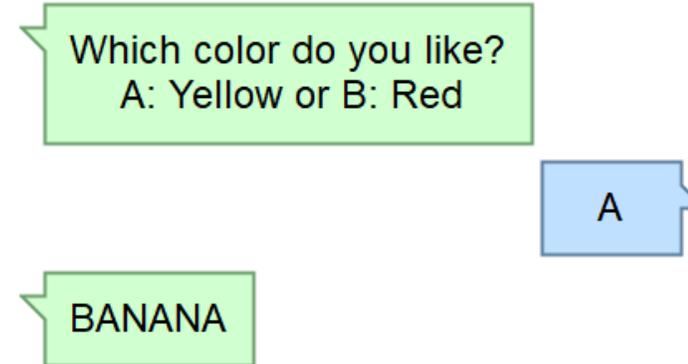
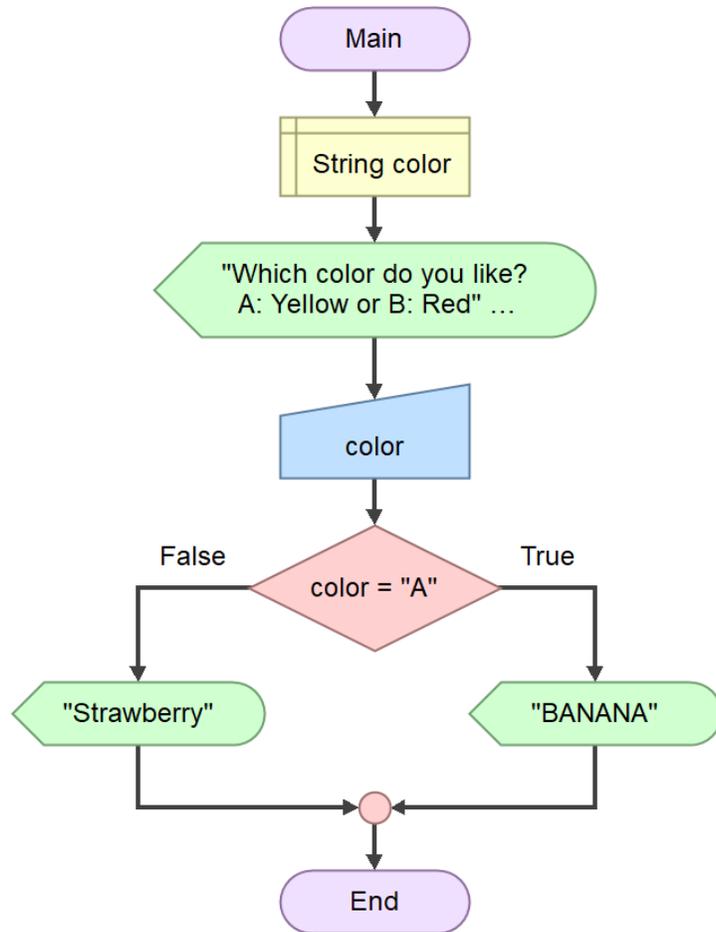
# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure

Operator	C Family	BASIC Family	Mathematics (Unicode)
Equality	=	=	=
Inequality	≠	◇	≠
Less Than or Equal	≤	≤	≤
Greater Than Or Equal	≥	≥	≥
Logical Not	!	not	¬
Logical And	&&	and	∧
Logical Or		or	∨
Multiply	*	*	×
Divide	/	/	÷
Modulo	%	mod	

# ผังงานโครงสร้างแบบตัดสินใจ

## Decision Structure



**ปัญหา: ต้องการตรวจสอบรหัสผ่าน 4 หลัก ที่รับเข้ามา ถูกต้องหรือไม่  
หากถูกต้องให้แสดงคำว่า “ประตูเปิด” หากไม่ถูกต้อง “รหัสไม่ถูกต้อง”**

**ปัญหา: ต้องการรับจำนวน 3 จำนวนจากผู้ใช้ และตรวจสอบว่าจำนวนที่มากที่สุด คือ จำนวนใด**

**ปัญหา: ต้องการรับจำนวน 1 จำนวนจากผู้ใช้ และตรวจสอบว่าจำนวนที่รับมานั้น เป็นจำนวนคู่ (Even) หรือ จำนวนคี่ (Odd)**

# ปัญหา: ต้องการรับชื่อเครื่องดื่มกาแฟ และแสดงส่วนประกอบที่ต้องใส่ในเครื่องดื่มนั้นๆ โดยมีเงื่อนไขดังตารางต่อไปนี้

Input



Process

Please enter menu:  
Hot Espresso

Ingredient  
Hot Espresso  
Coffee: 50%  
Hot Water: 20%  
Milk: 30%

Output



Beverage	Coffee	Hot water	Milk	Milk foam
Hot Amaricano	50%	50%	-	-
Hot Capucino	30%	20%	40%	10%
Hot Espresso	50%	20%	30%	-

# ปัญหา: ต้องการรับค่า ส่วนสูง (cm) และ น้ำหนัก (kg) จากนั้นคำนวณค่า bmi เพื่อตรวจสอบและแสดงผลว่า ผอม หุ่นปกติ หรือ อ้วน ตามตาราง

สูตรการคำนวณ Body Mass Index (BMI)

= น้ำหนักตัวกิโลกรัม / (ส่วนสูงเมตร ยกกำลังสอง)

หรือ

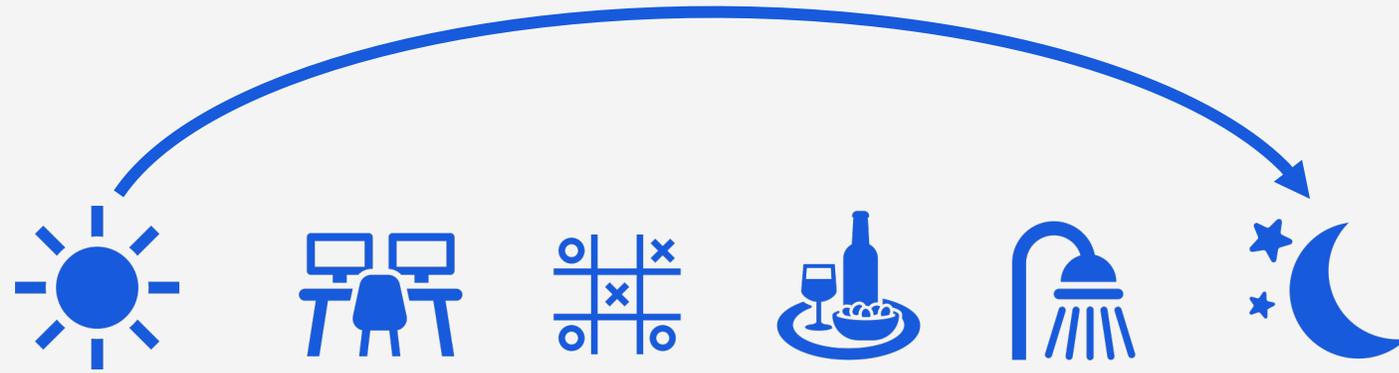
$bmi = weight\_kg / (height\_m * height\_m)$

## คำนวณค่าดัชนีมวลกาย BMI

ค่าอยู่ในเกณฑ์	ค่า BMI	ภาวะเสี่ยงต่อโรค
ผอมเกินไป	น้อยกว่า 18.5	เสี่ยงต่อการได้รับสารอาหารไม่เพียงพอ
น้ำหนักปกติ เหมาะสม	18.6 - 22.9	มีความเสี่ยงต่อโรคต่าง ๆ น้อยที่สุด
น้ำหนักเกิน	23.0 - 24.9	ถือว่ายังมีความเสี่ยงมากกว่าคนปกติ
อ้วน	25.0 - 29.9	ยังมีความเสี่ยงต่อการเกิดโรค
อ้วนมาก	30.0 ขึ้นไป	เสี่ยงต่อการเกิดโรค

# ผังงานโครงสร้างแบบวนทำงานซ้ำ

## Iteration Structure หรือ Looping



ในชีวิตประจำวัน มีกระบวนการ หรือ ขั้นตอนการใช้ชีวิตใดบ้าง  
ที่มีลักษณะเป็นโครงสร้างแบบวนทำงานซ้ำ?

 Short Answer



Untitled Project

Saved a few seconds ago

Rename

Share

Remix

Create ▼

Nutthapat ▼

# MINECRAFT



Blocks

Workspace

Start Over

</> S

move forward

turn left ↶

turn right ↷

destroy block

shear

place bedrock ▼

place torch

plant crop

place bedrock ▼ ahead

repeat 3 times

do

when run

move forward

move forward

move forward

Run



Untitled Project

Saved a minute ago

Rename

Share

Remix

Create ▼

Nutthapat ▼

# MINECRAFT



Run

Blocks

move forward

turn left ↶

turn right ↷

destroy block

shear

place bedrock ▼

place torch

plant crop

place bedrock ▼ ahead

repeat 3 ▼ times

do

Workspace

↶ Start Over

</> S

when run

repeat 3 ▼ times

do move forward



Untitled Project

Saved a few seconds ago

Rename

Share

Remix

Create ▼

Nutthapat ▼

# MINECRAFT



Blocks

move forward

turn left ↶

turn right ↷

destroy block

shear

place bedrock ▼

place torch

plant crop

place bedrock ▼ ahead

repeat 3 ▼ times

do

Workspace

Start Over

</> S

when run

move forward

move forward

move forward

when run

repeat 3 ▼ times

do move forward

Run



# MINECRAFT



Blocks

move forward

turn left

turn right

destroy block

shear

place bedrock

place torch

plant crop

place bedrock ahead

repeat 3 times

do

when run

move forward

move forward

move forward

when run

repeat 3 times

do move forward

ถ้าเดิน 100 ก้าว



Run

# ผังงานโครงสร้างแบบวนทำงานซ้ำ

## Iteration Structure หรือ Looping

**โจทย์:** ต้องการแสดงผลจำนวนที่มีค่าตั้งแต่ 1 ถึง 10

### รหัสเทียม (Pseudocode)

เริ่มต้นการทำงาน

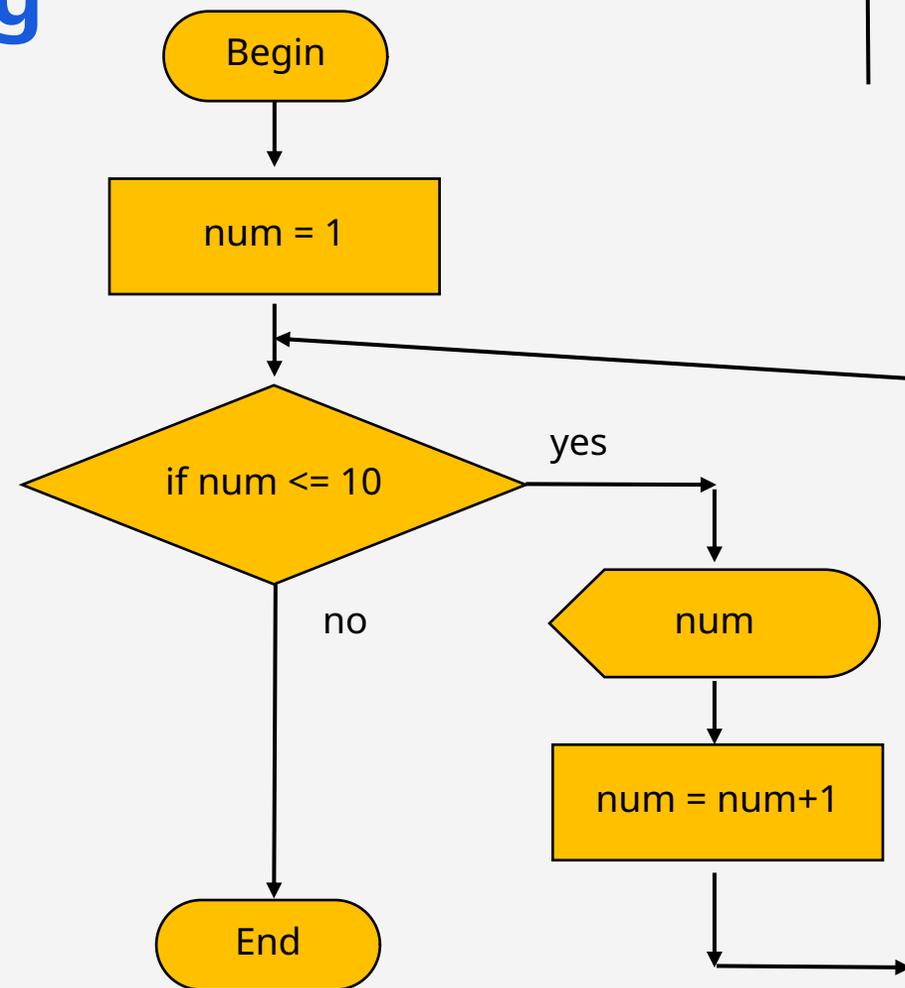
กำหนดตัวแปร num ให้เท่ากับ 1

ถ้าค่าในตัวแปร num ไม่เกิน 10

แสดงค่า num ล่าสุด

เพิ่มค่า num ไปอีก 1

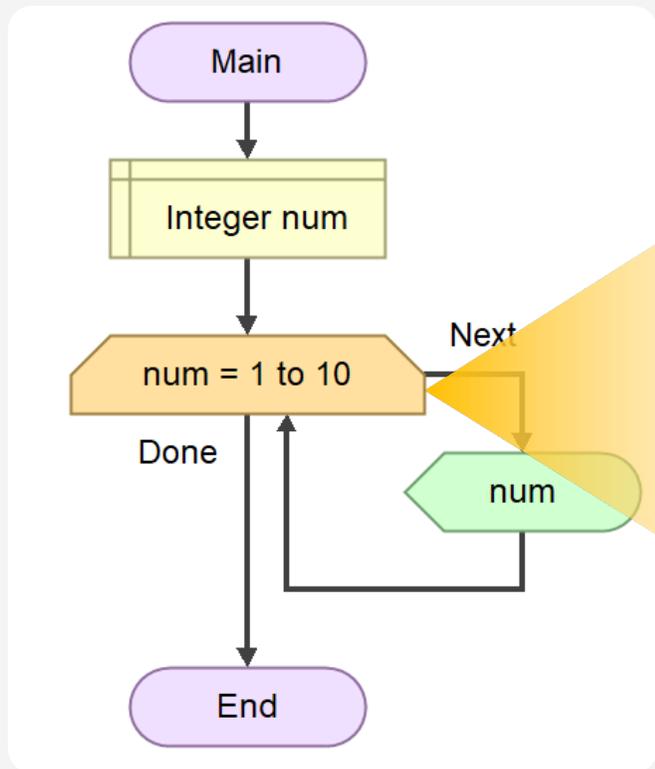
สิ้นสุดการทำงาน



# ผังงานโครงสร้างแบบวนทำงานซ้ำ

## Iteration Structure หรือ Looping

**โจทย์:** ต้องการแสดงผลจำนวนที่มีค่าตั้งแต่ 1 ถึง 10



For Properties

**For** A For Loop increments or decrements a variable through a range of values. This is a common and useful replacement for a While Loop.

Variable: ตัวแปรที่ต้องการใช้นับเพื่อวนทำงานซ้ำ

num

Start Value: ค่าเริ่มต้น

1

End Value: ค่าสิ้นสุด

10

Direction:

Increasing **เพิ่ม**

Decreasing **ลด**

Step By: ทีละ หรือ รอบละ

1

OK Cancel

# ผังงานโครงสร้างแบบวนทำงานซ้ำ

## Iteration Structure หรือ Looping

**โจทย์:** ต้องการแสดงผลจำนวนที่มีค่าตั้งแต่ 1 ถึง 10

For Properties

**For** A For Loop increments or decrements a variable through a range of values. This is a common and useful replacement for a While Loop.

Variable: ตัวแปรที่ต้องการใช้นับเพื่อวนทำงานซ้ำ

num

Start Value: ค่าเริ่มต้น 1

End Value: ค่าสิ้นสุด 10

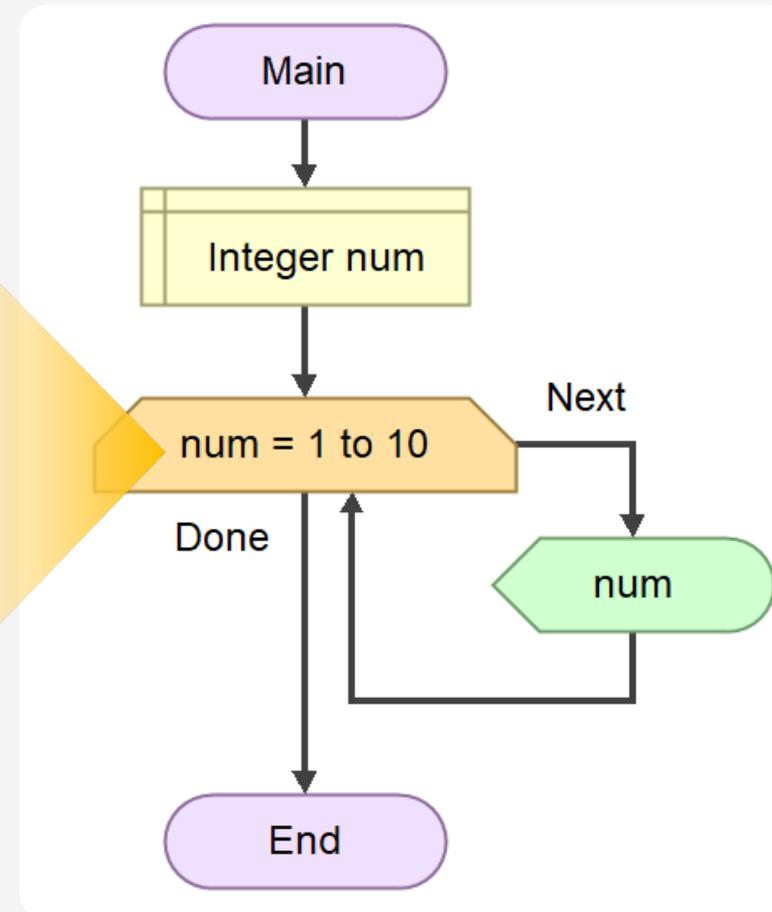
Direction:

Increasing **เพิ่ม**

Decreasing **ลด**

Step By: ทีละ หรือ รอบละ 1

OK Cancel



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

**ปัญหา: ต้องการแสดงผลจำนวนที่มีค่าตั้งแต่ 1 ถึง จำนวนที่ผู้ใช้ป้อนเข้ามา  
เช่น ผู้ใช้ป้อน 50 ต้องแสดง 1,2,3,...,48,49,50**

**ปัญหา: ต้องการแสดงผลจำนวนที่มีค่าตั้งแต่จำนวนที่ผู้ใช้ป้อนเข้ามาและ  
ลดไปถึง 0 เช่น ผู้ใช้ป้อน 100 ต้องแสดง 100,99,98,...,2,1,0**

# ปัญหา: ต้องการตรวจสอบ Password

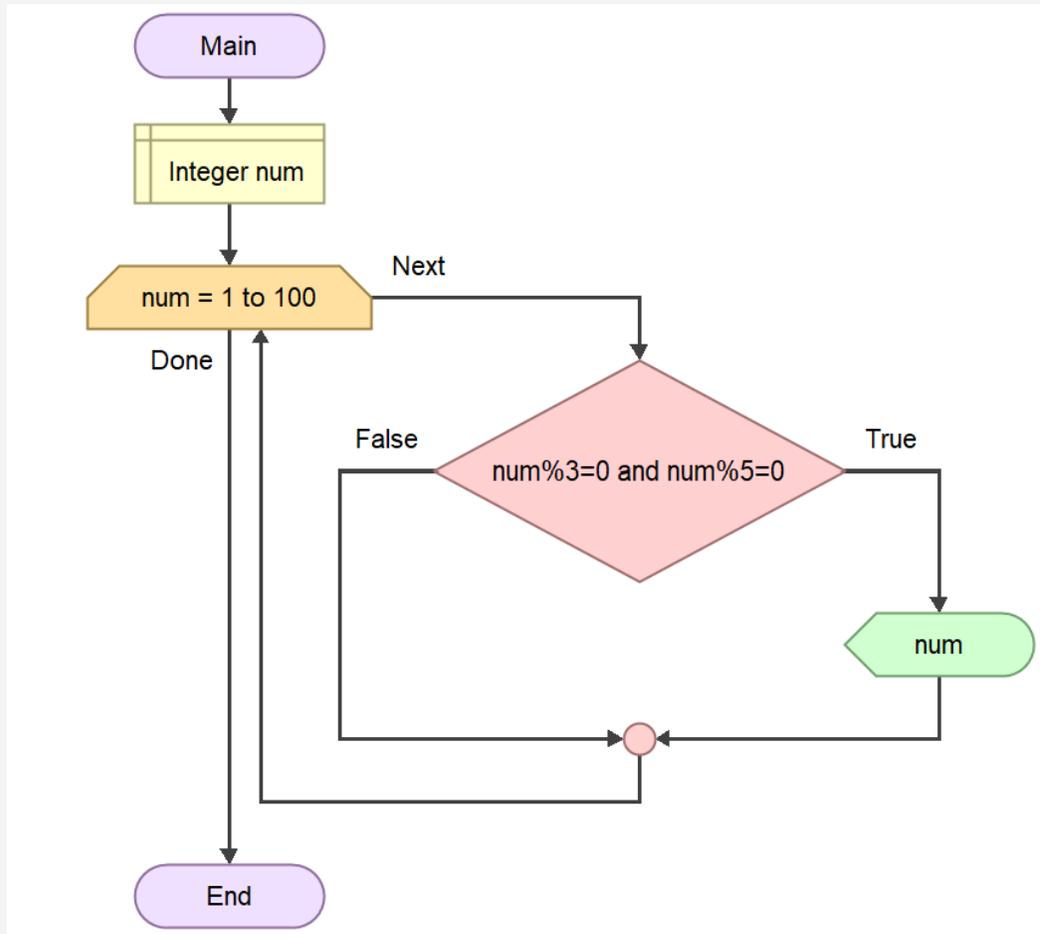
- โดยหาก Password มีค่าเท่ากับ “onepiece” แสดงคำว่า “You are login”
- แต่หาก Password ไม่ตรงตามที่ตั้งไว้ ให้แจ้งว่า “Wrong Password!, Please try again”
- และวนรับค่า Password ไปเรื่อย ๆ จนกว่าจะป้อน Password ถูกต้อง

**ปัญหา: ต้องการแสดงผลจำนวนที่หาร 5 ลงตัว ตั้งแต่ 1 ถึง 50**  
**ตัวอย่างแสดงผล: 5,10,15,20,...,45,50**

**ปัญหา: ต้องการแสดงผลจำนวนที่หาร 5 ลงตัว ตั้งแต่ 1 ถึง 50**  
**ตัวอย่างแสดงผล: 5,10,15,20,...,45,50 และนับด้วยว่า มีกี่จำนวน?**

**ปัญหา: ต้องการหาจำนวนที่หาร 3 และ หาร 5 ลงตัว ตั้งแต่ 1 ถึง 100**

# ปัญหา: ต้องการหาจำนวนที่หาร 3 และ หาร 5 ลงตัว ตั้งแต่ 1 ถึง 100 และนับด้วยว่ามีกี่จำนวน?



# ปัญหา: ต้องการรับค่าจากผู้ใช้เพื่อแสดงจำนวน factorial ดังตัวอย่างต่อไปนี้

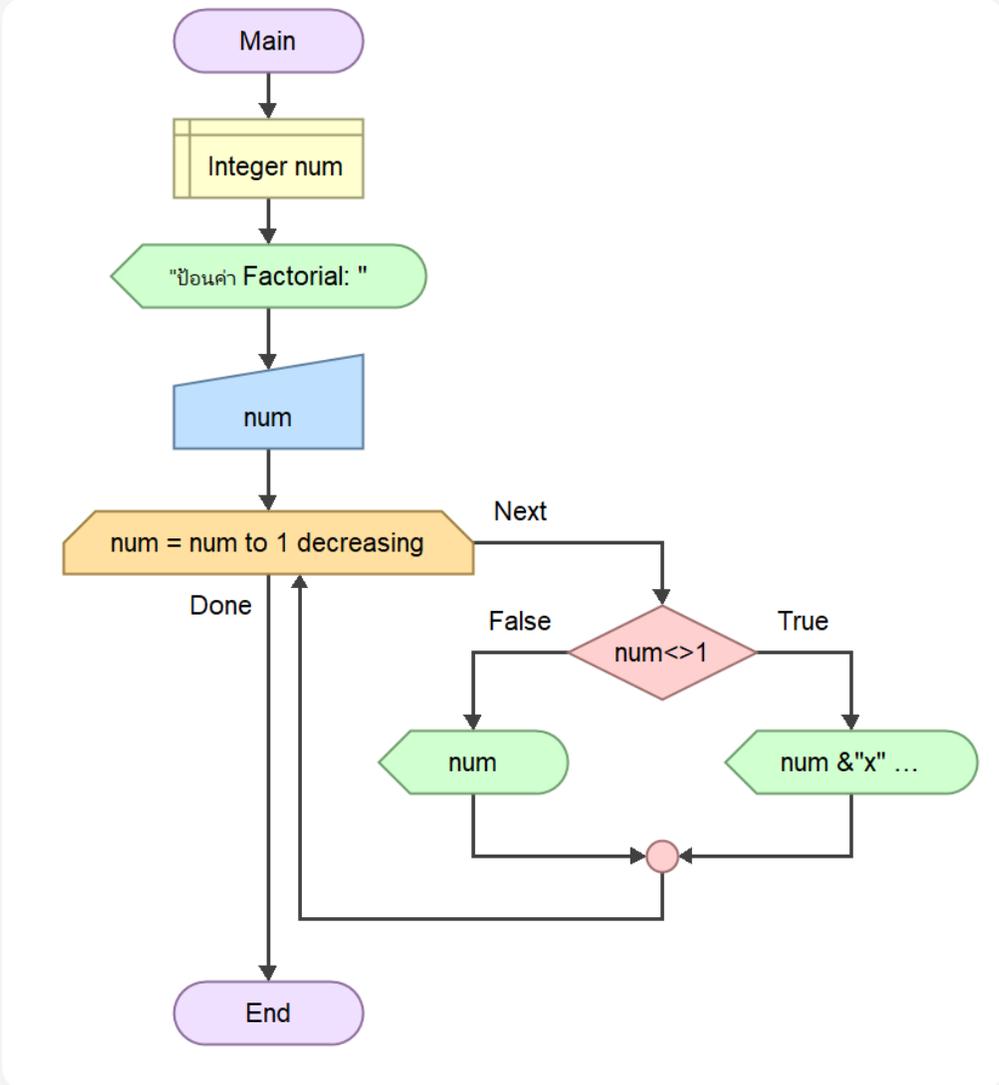
รับค่า: 5

แสดงผล: **5 x 4 x 3 x 2 x 1**

ป้อนค่า Factorial:

8

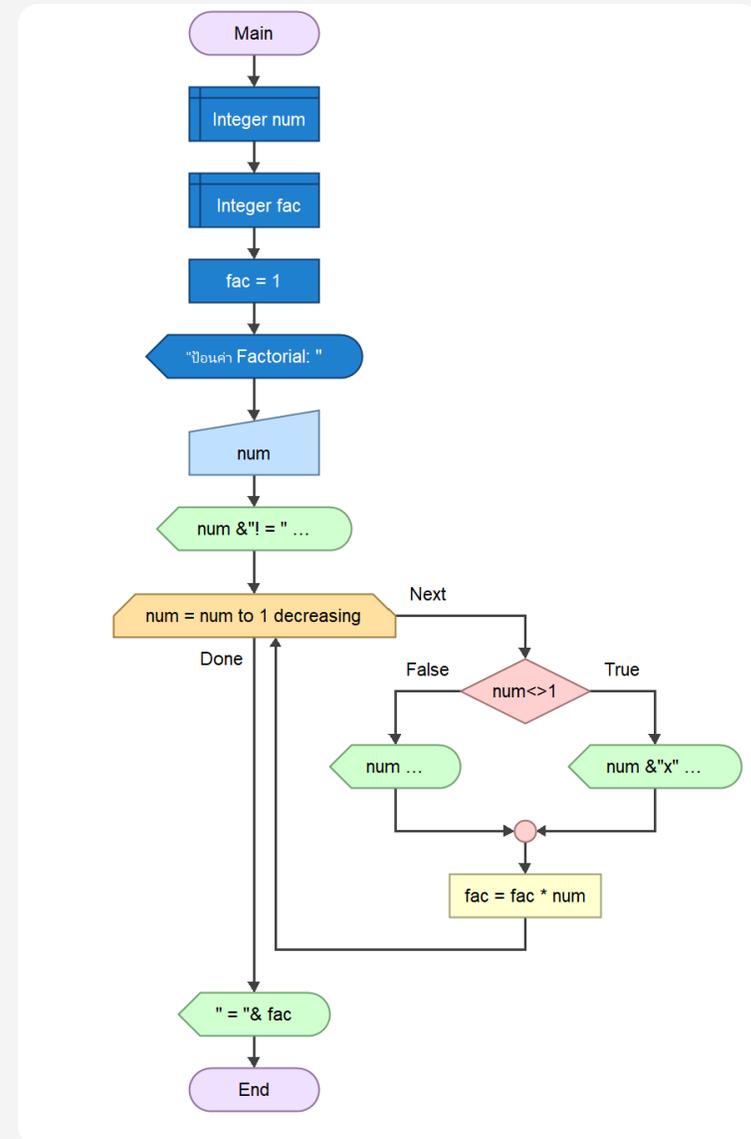
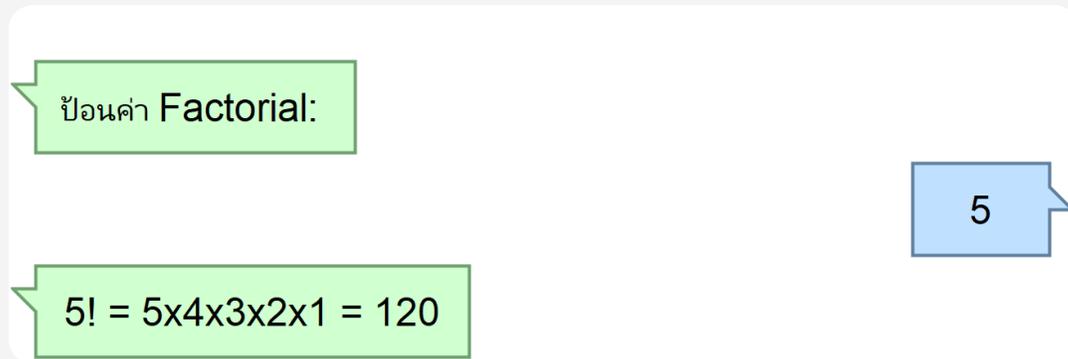
8x7x6x5x4x3x2x1



# ปัญหา: ต้องการรับค่าจากผู้ใช้เพื่อแสดงจำนวน factorial ดังตัวอย่างต่อไปนี้

รับค่า: 5

แสดงผล:  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$



# การสอน Coding สำหรับผู้เริ่มต้น

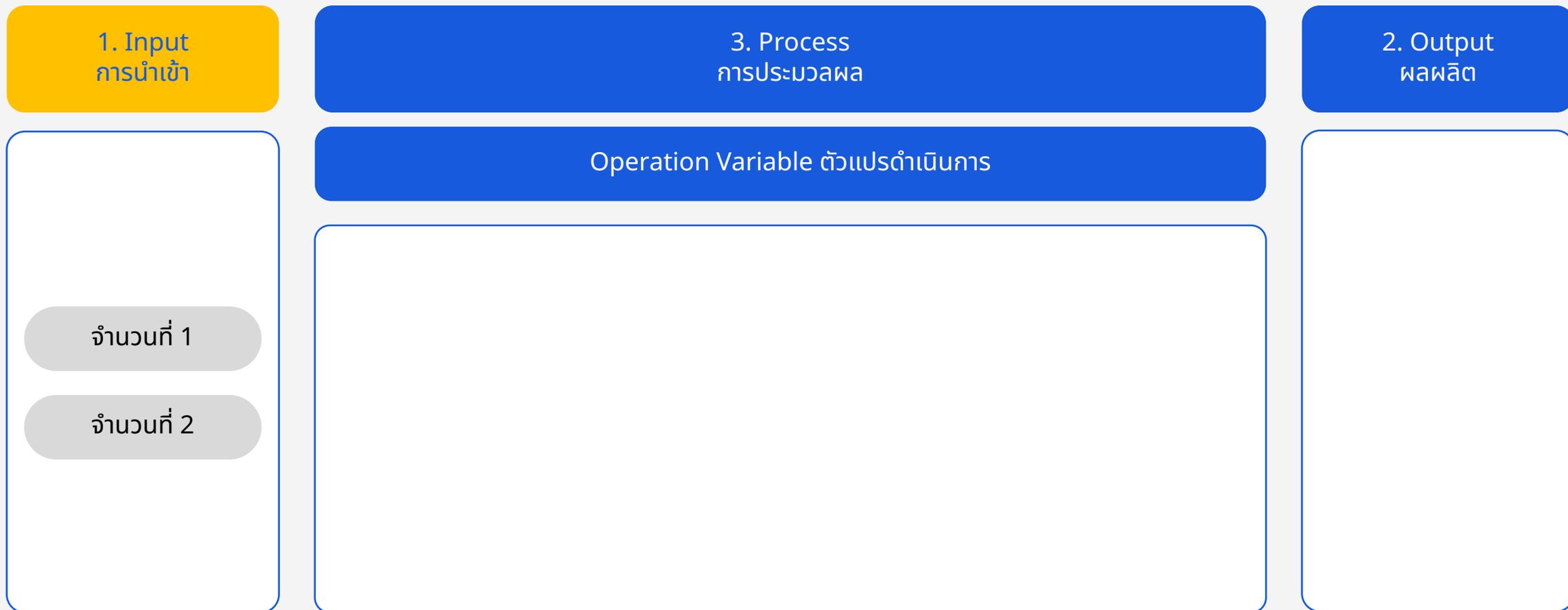
1. Input  
การนำเข้า

3. Process  
การประมวลผล

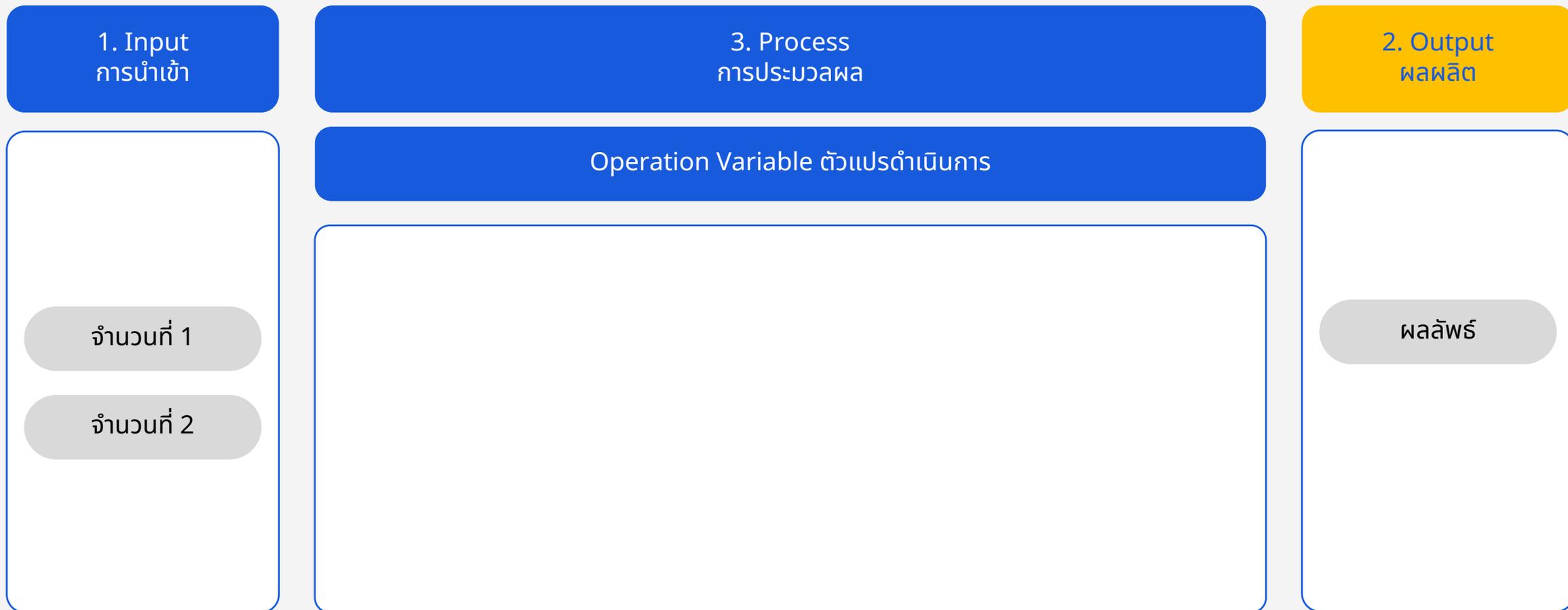
2. Output  
ผลผลิต

Operation Variable ตัวแปรดำเนินการ

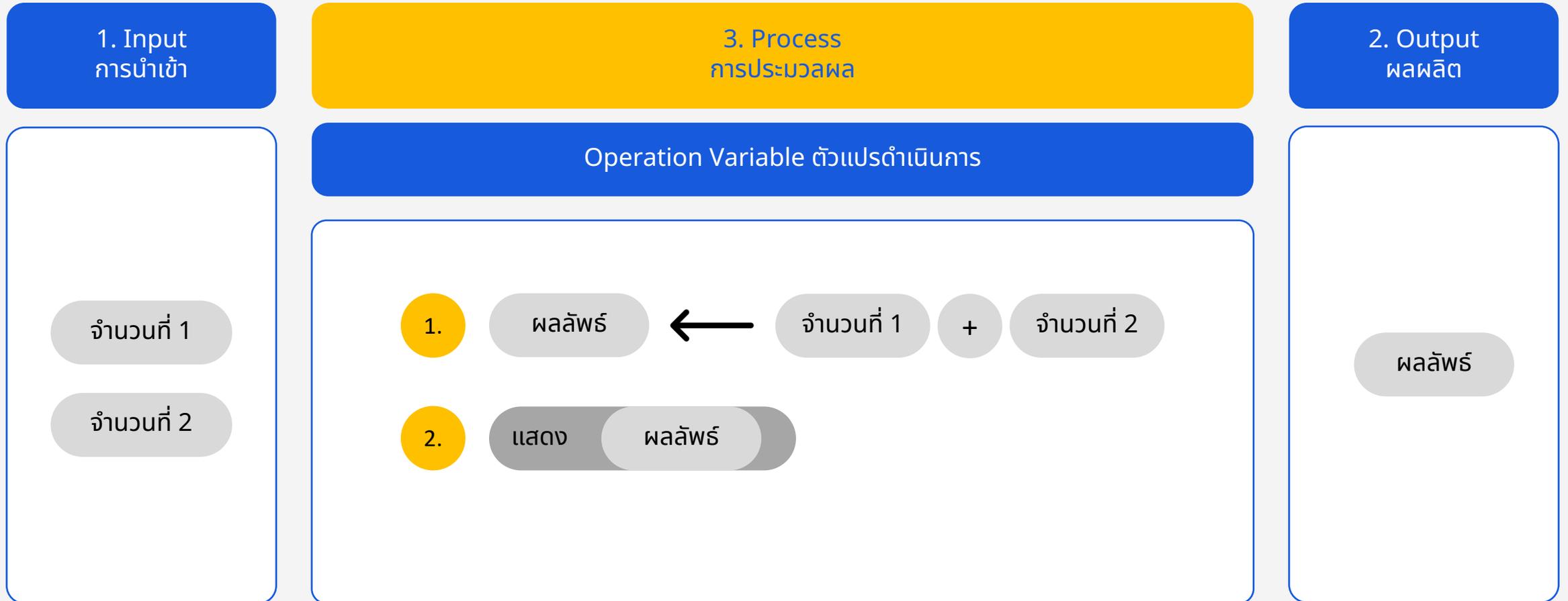
# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการคำนวณส่วนลดราคาสินค้า

## ราคาสินค้า 200 บาท ส่วนลด 10% ต้องจ่ายกี่บาท

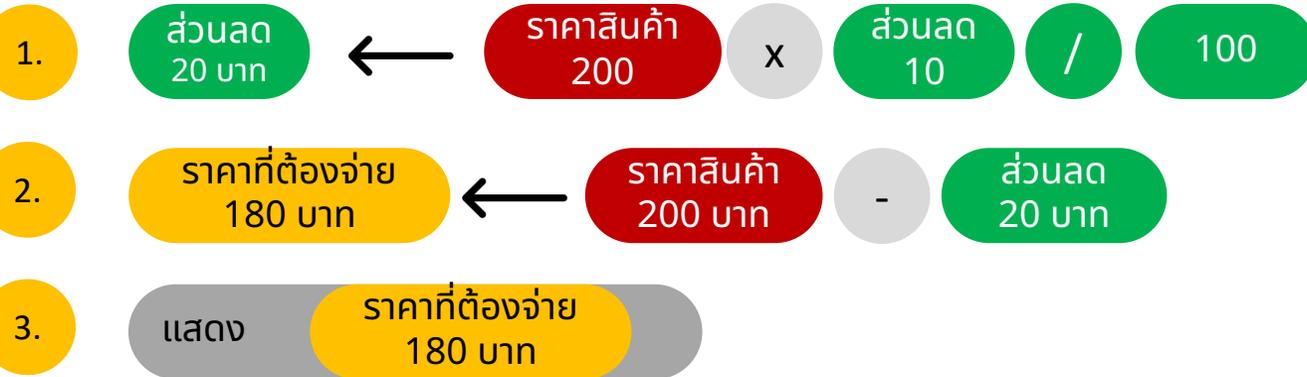
1. Input  
การนำเข้า

ราคาสินค้า

ส่วนลด

3. Process  
การประมวลผล

Operation Variable ตัวแปรดำเนินการ



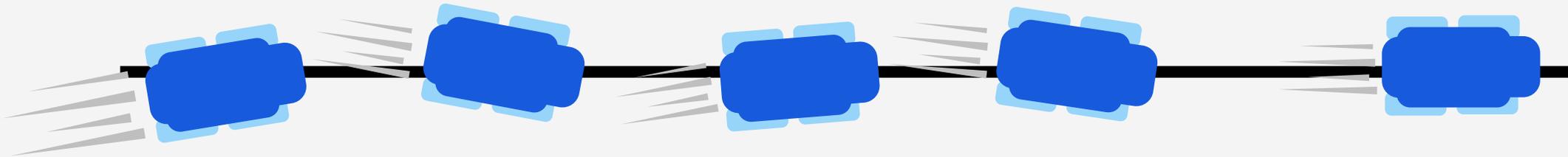
ตัวแปร (Variable) เช่น ราคาสินค้า ส่วนลด ส่วน 50, 100 เป็นค่าของตัวแปร (Value)

2. Output  
ผลผลิต

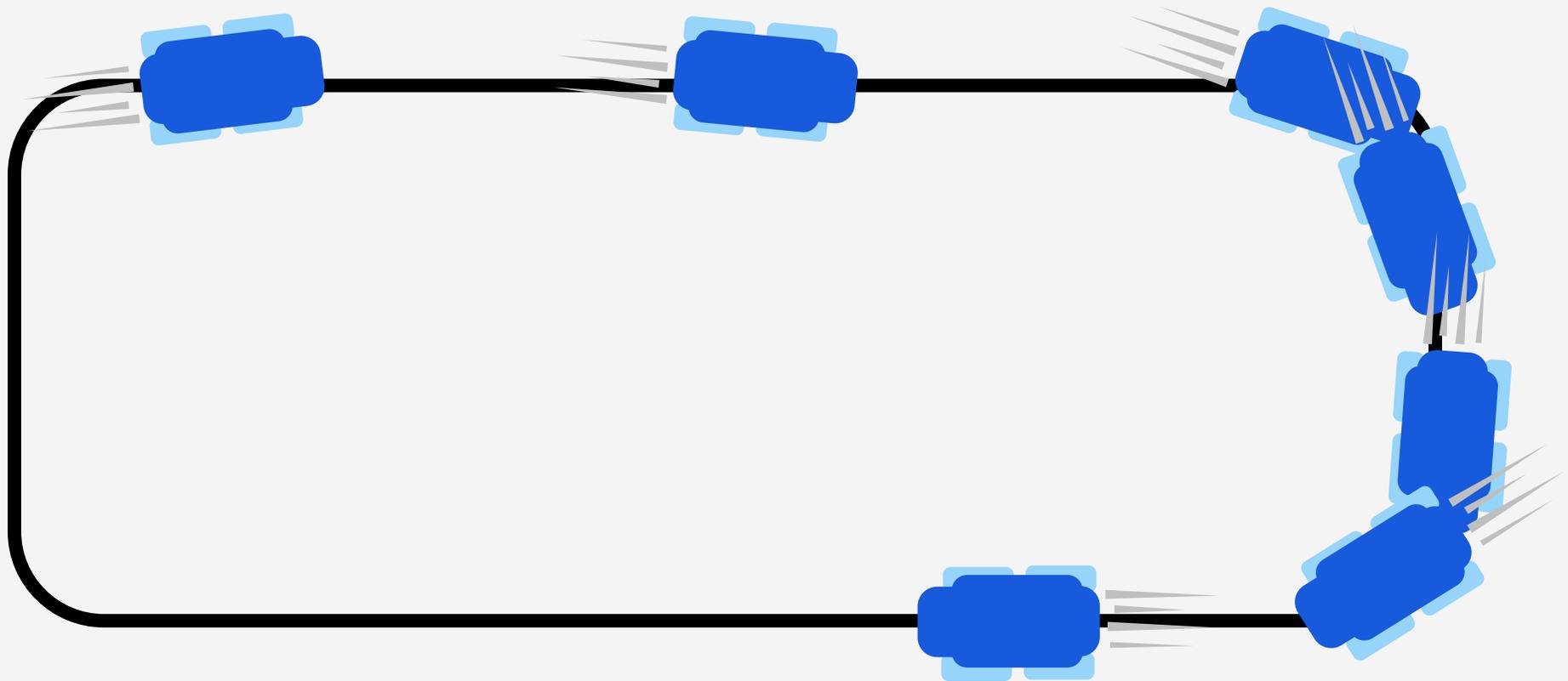
ราคาที่ต้องจ่าย



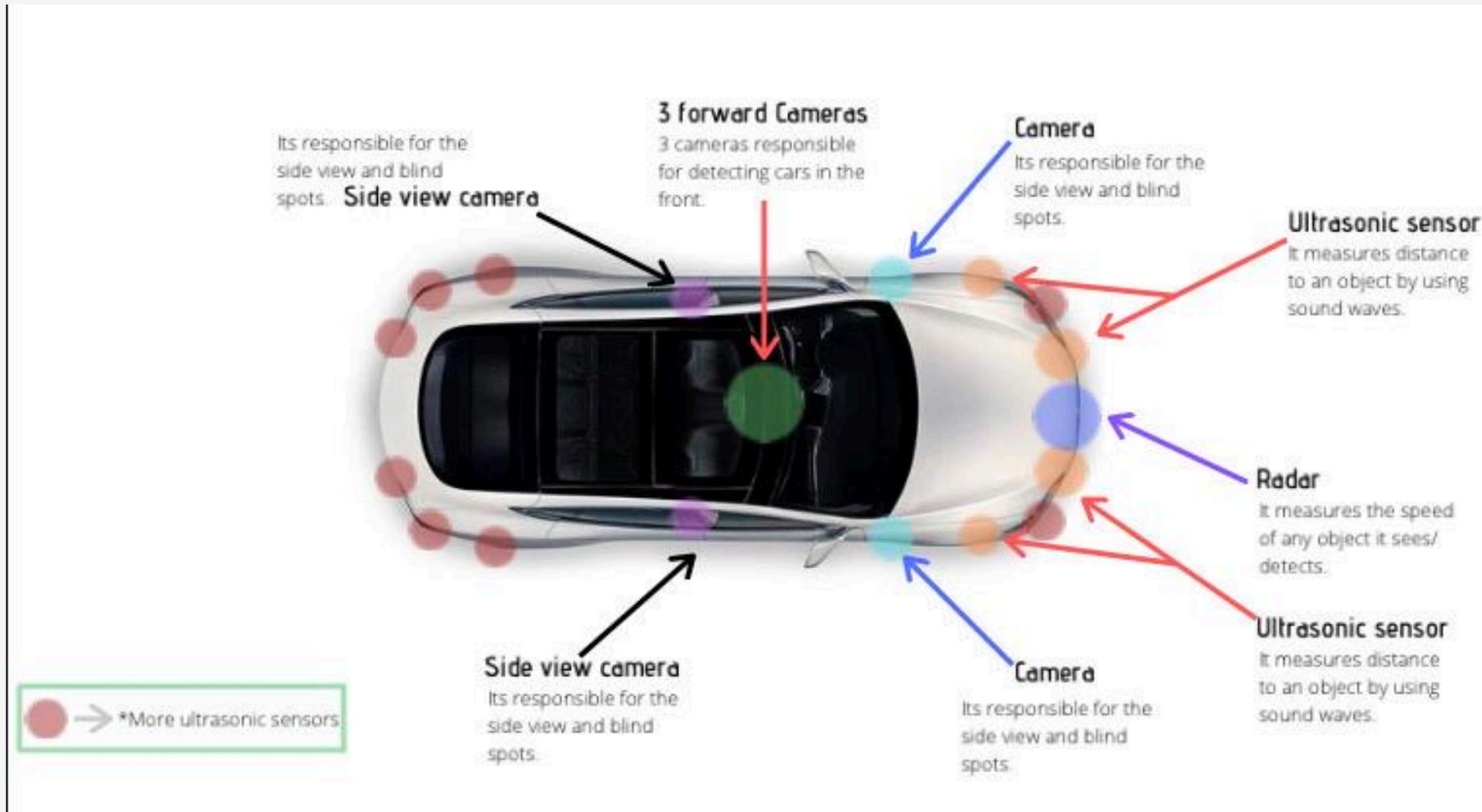
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



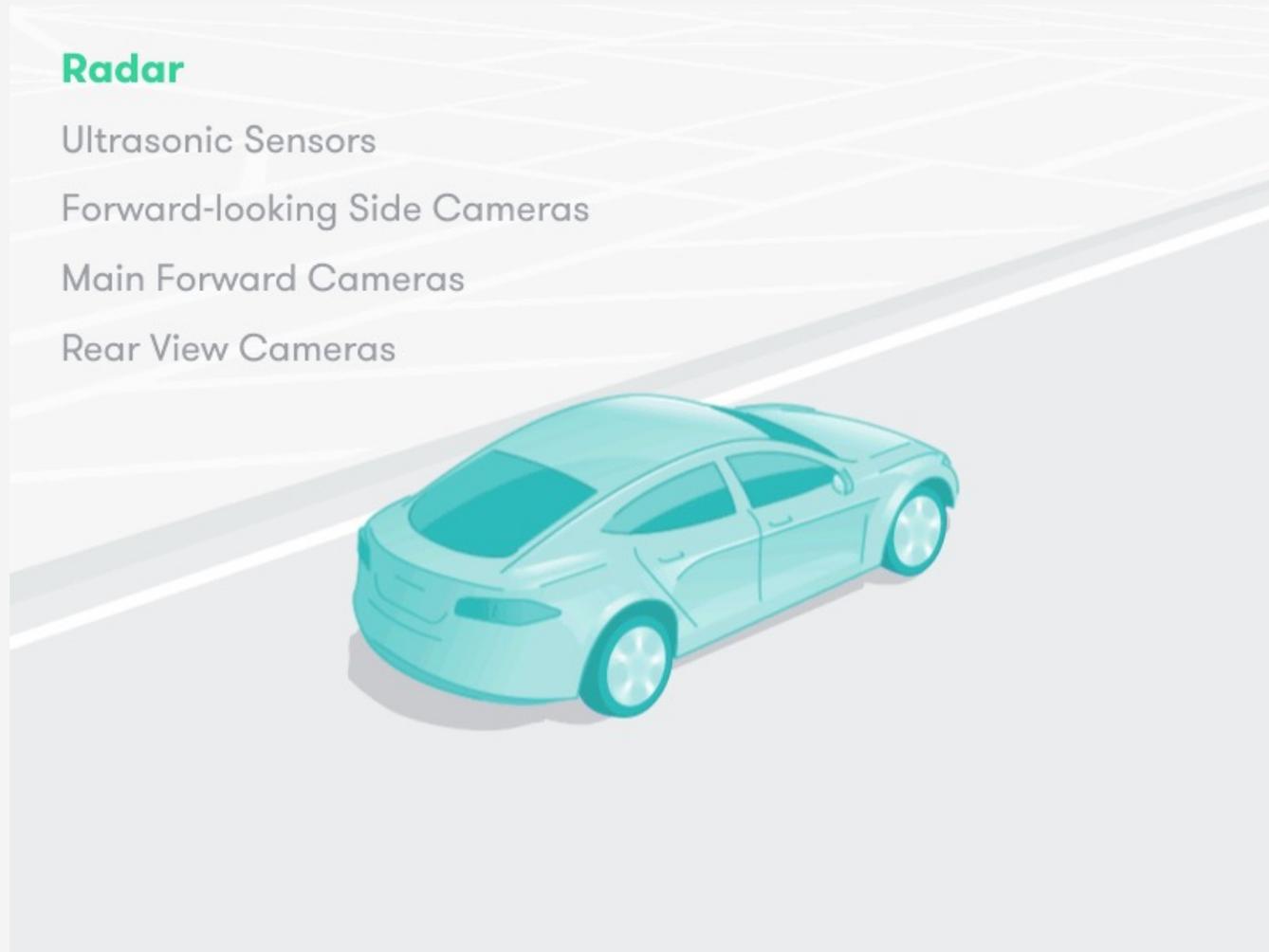
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



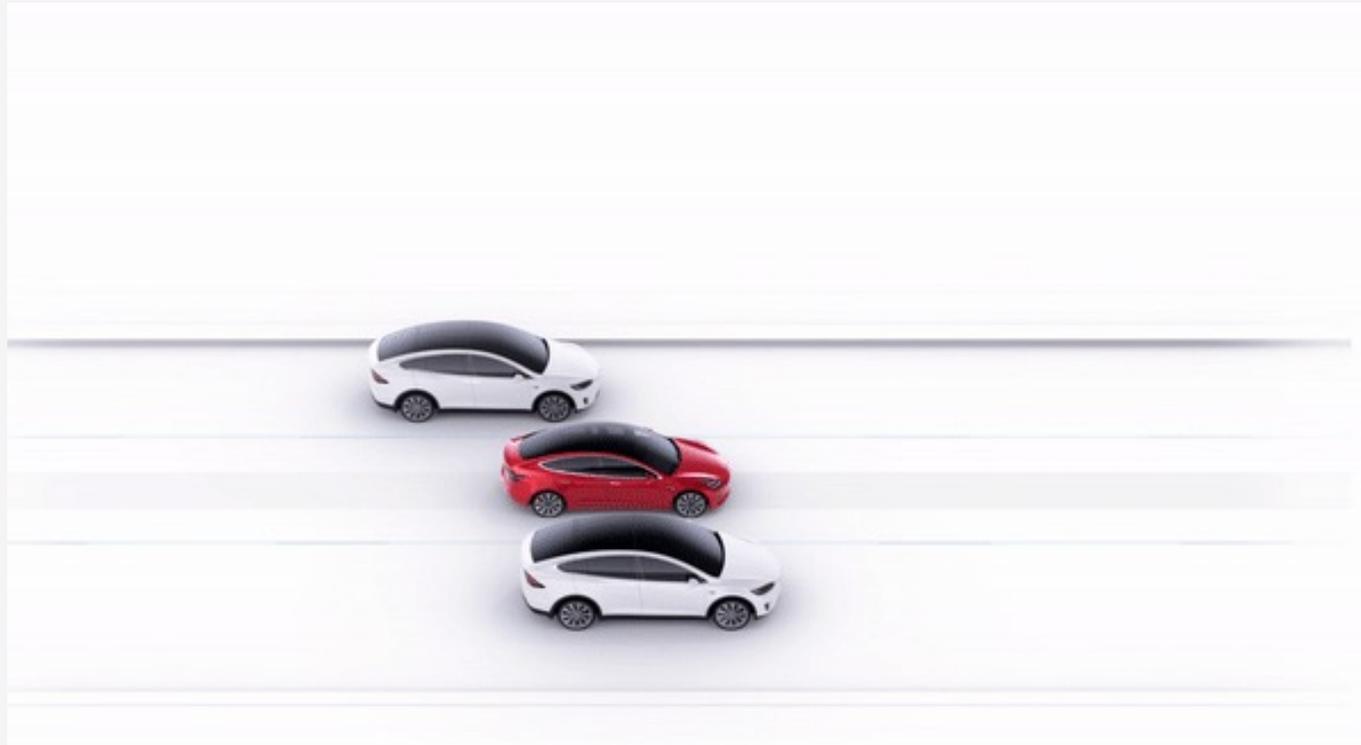
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



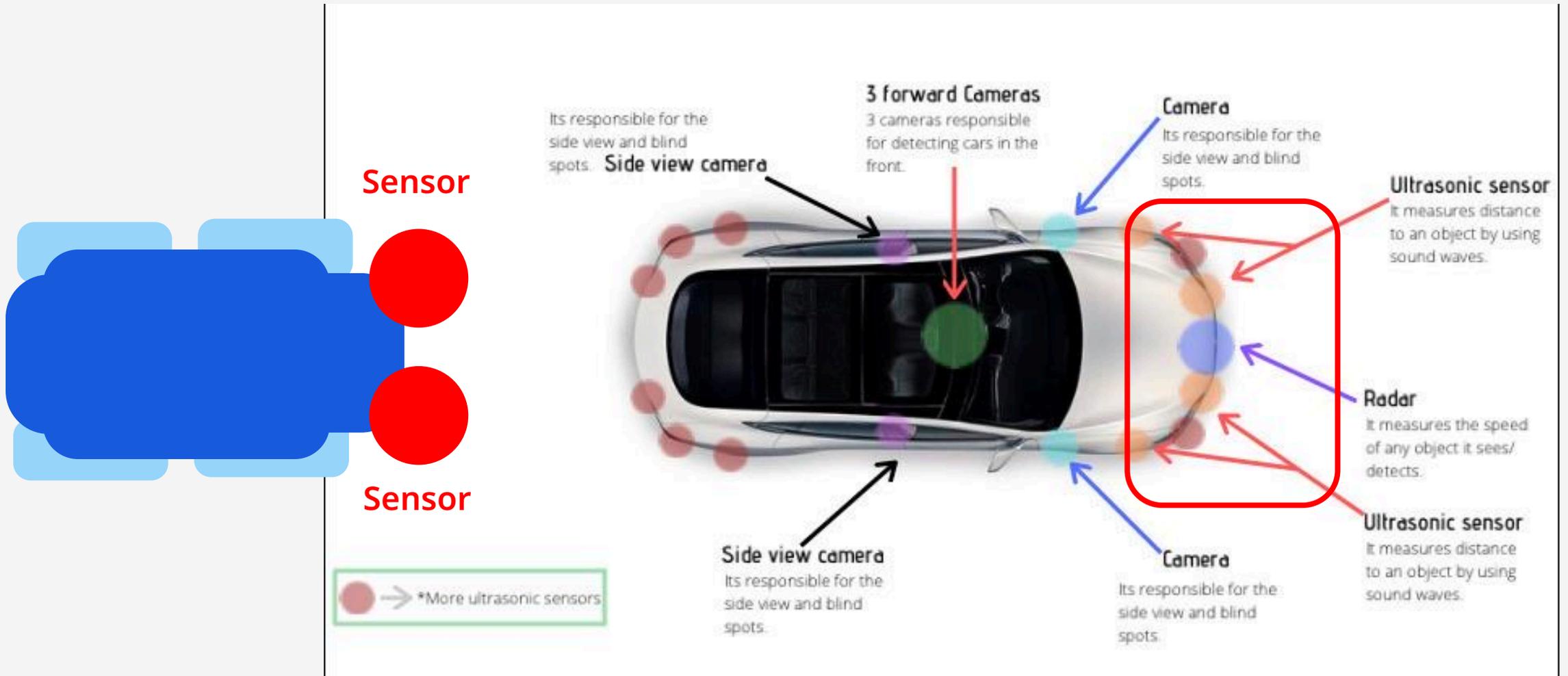
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



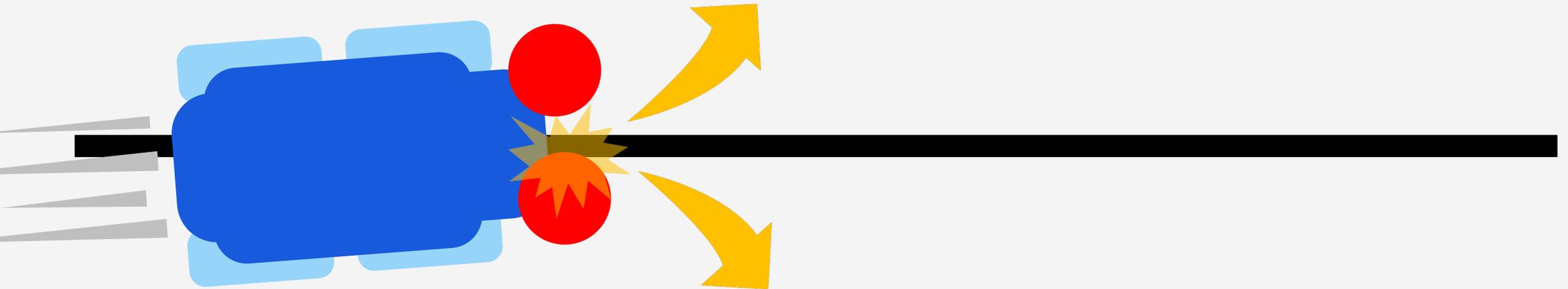
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?

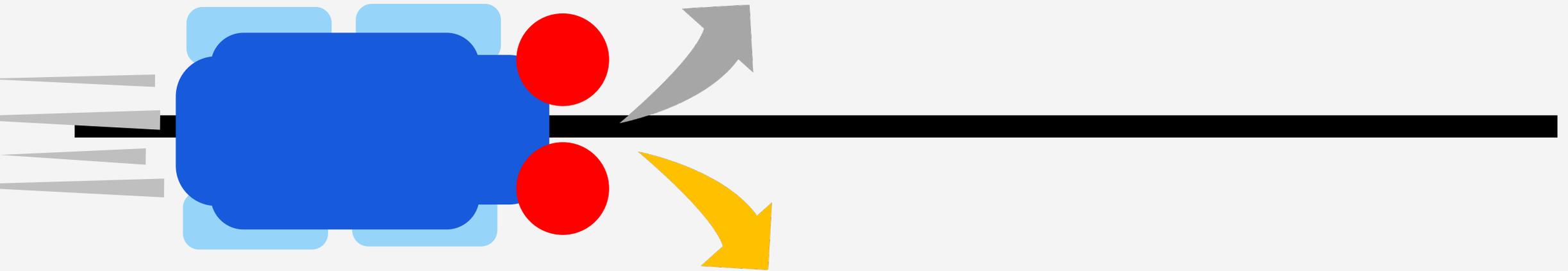


# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



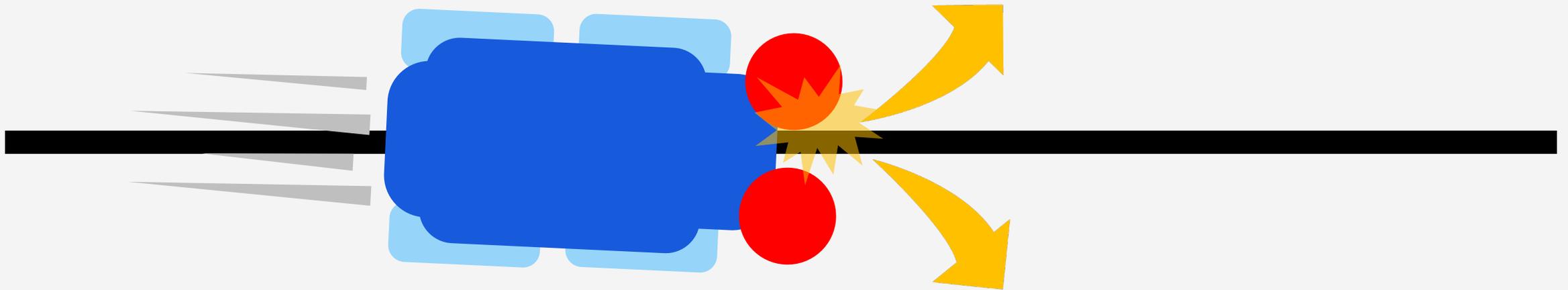
sensor (red) touch road (black) -> turn right 15 angle

# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?

sensor (red) touch road (black) -> turn left 15 angle

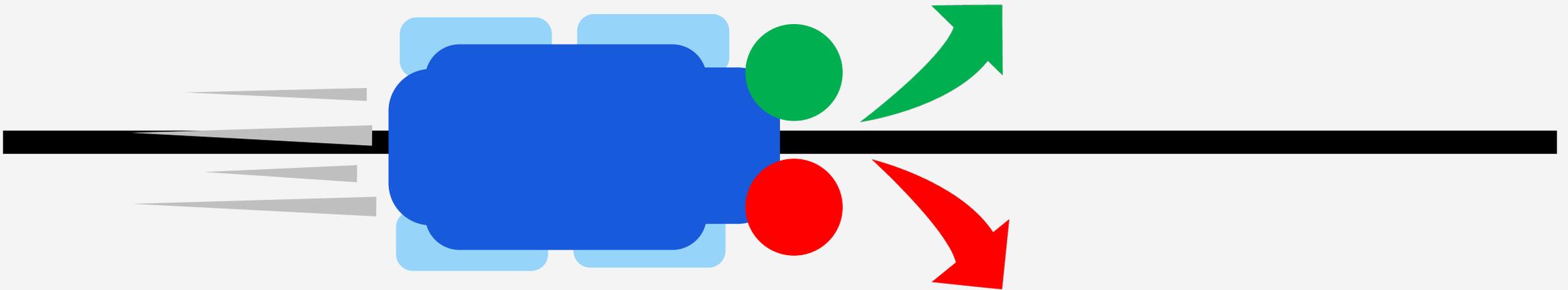


sensor (red) touch road (black) -> turn right 15 angle



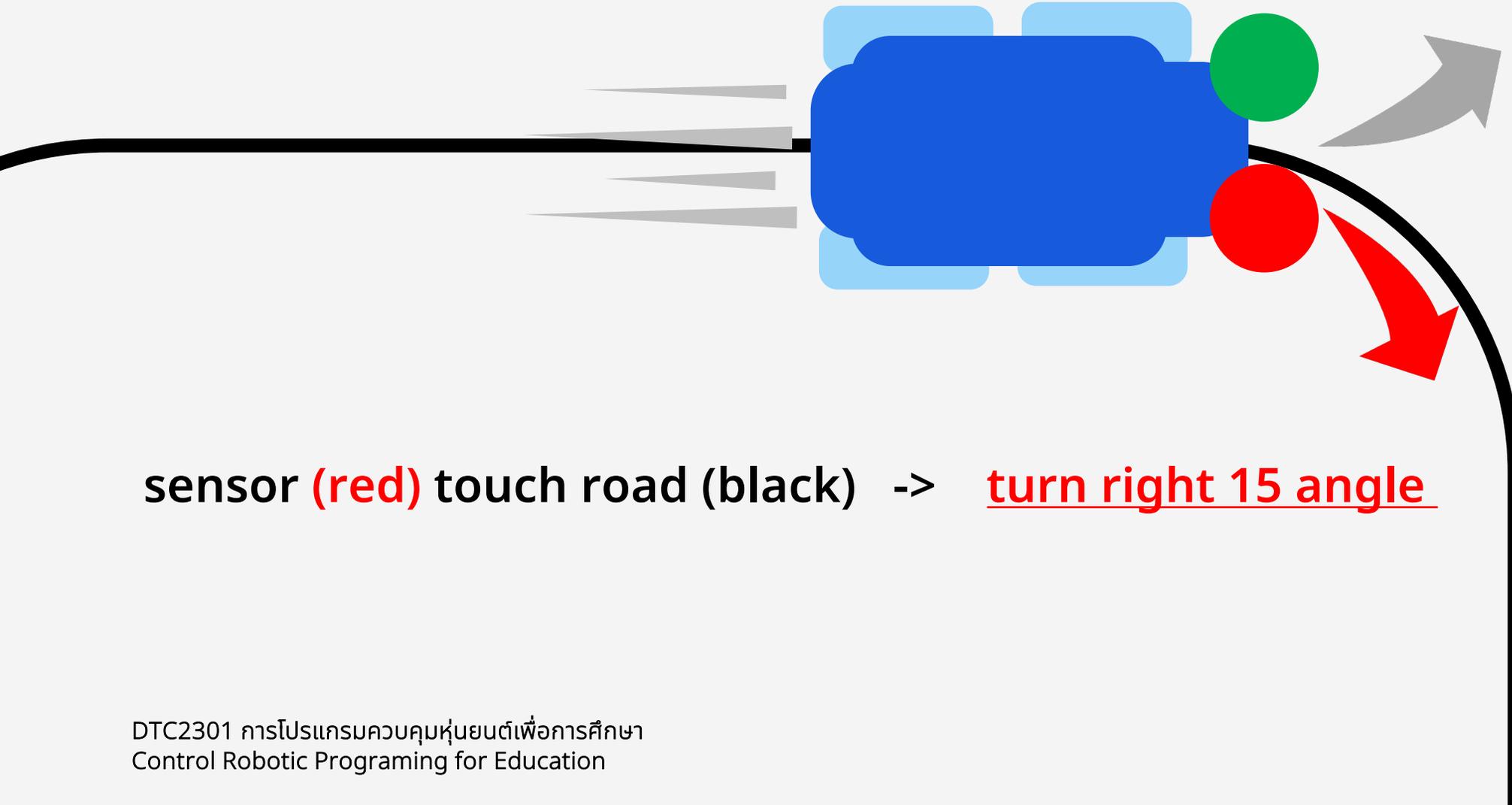
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?

sensor (green) touch road (black) -> turn left 15 angle



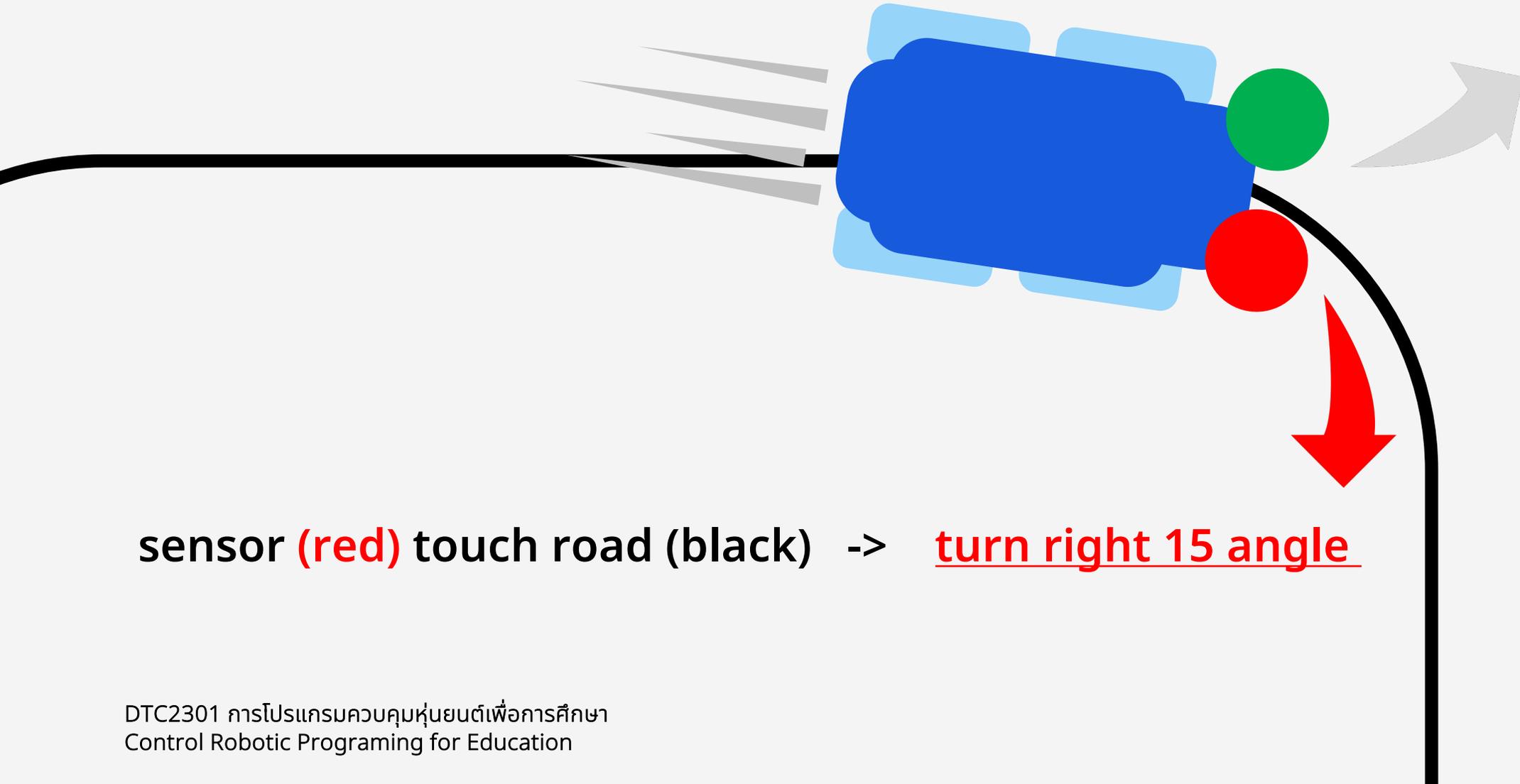
sensor (red) touch road (black) -> turn right 15 angle

# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



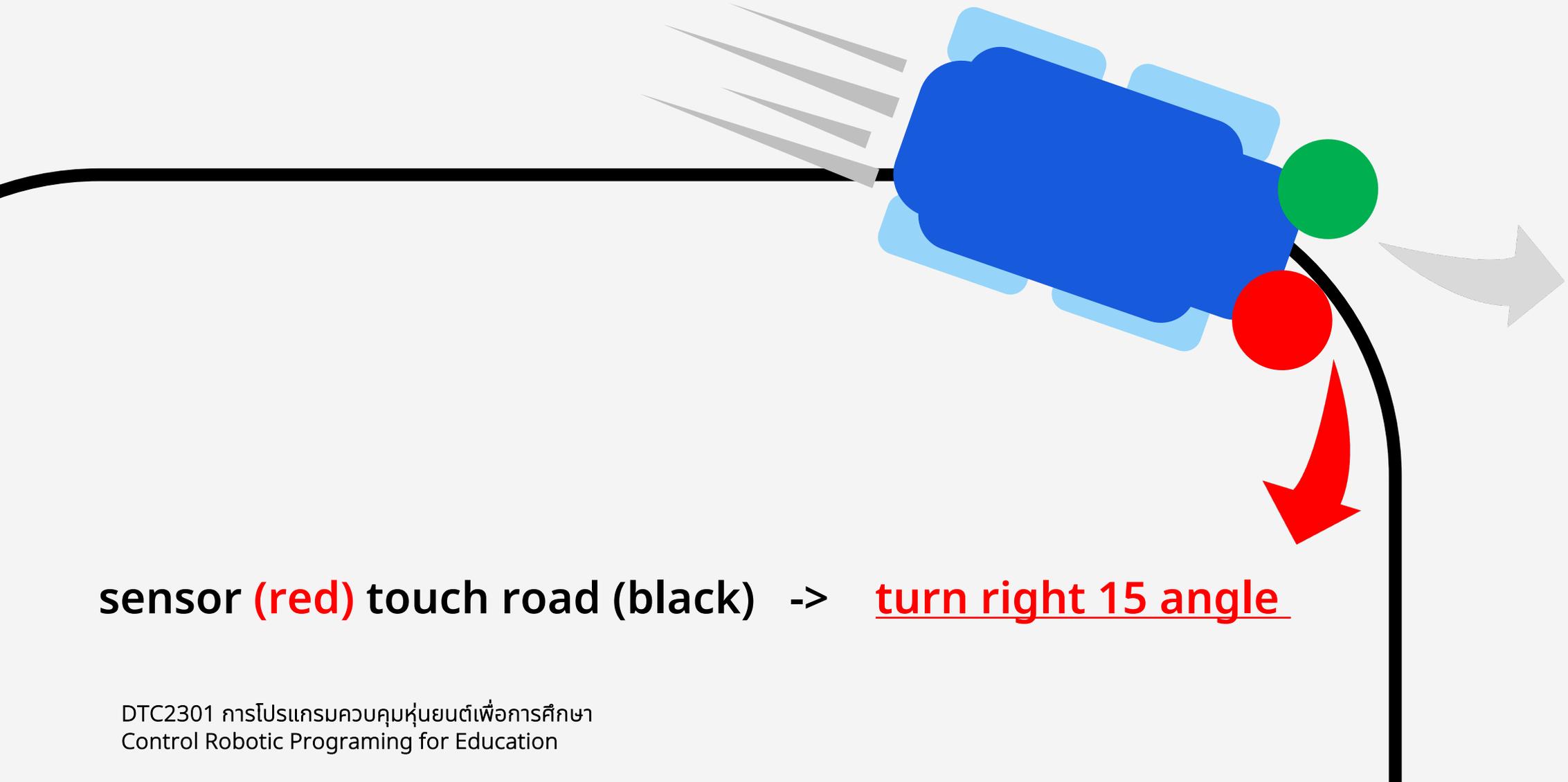
sensor (red) touch road (black) -> turn right 15 angle

# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



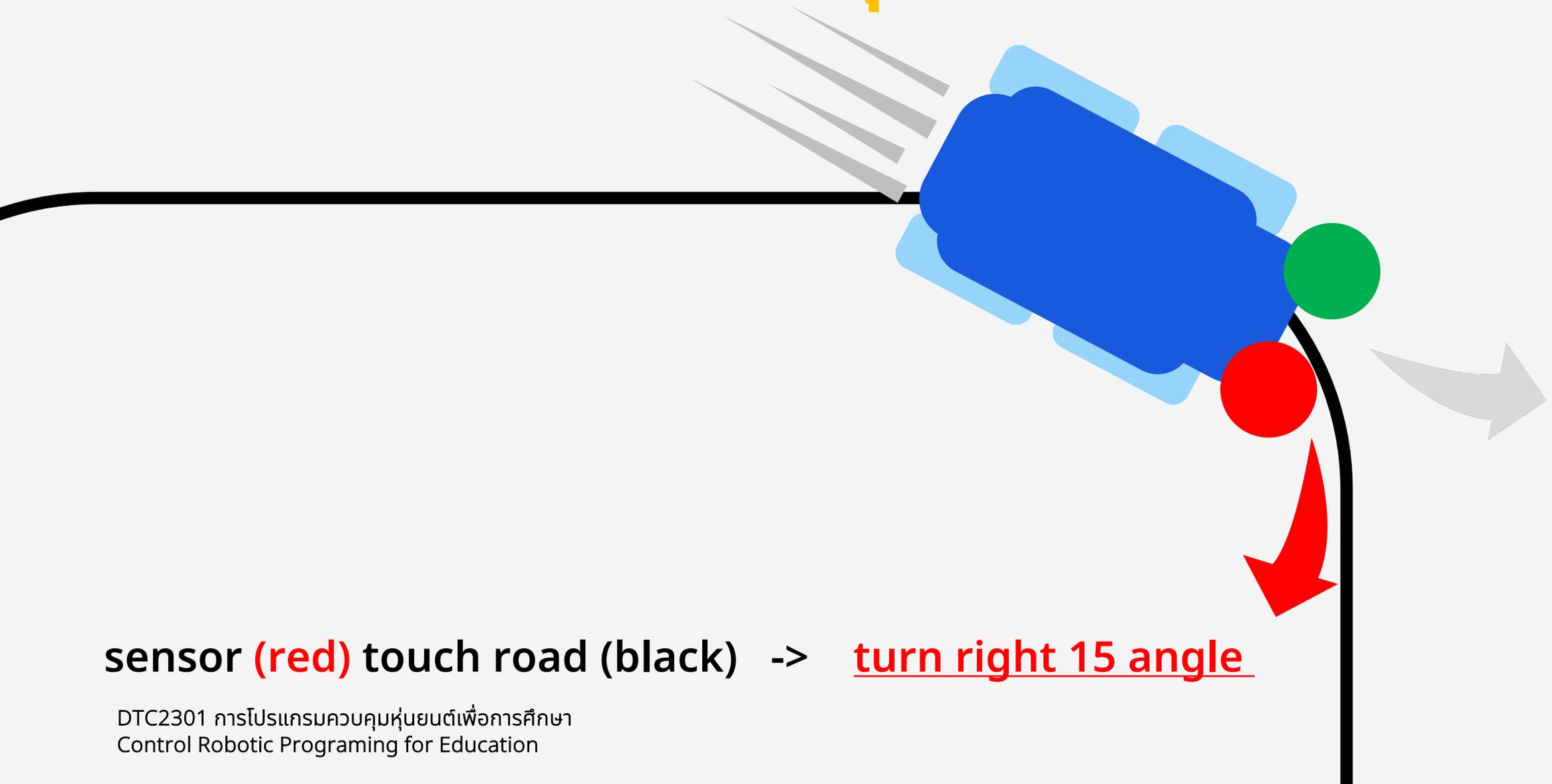
sensor (red) touch road (black) -> turn right 15 angle

# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?



sensor (red) touch road (black) -> turn right 15 angle

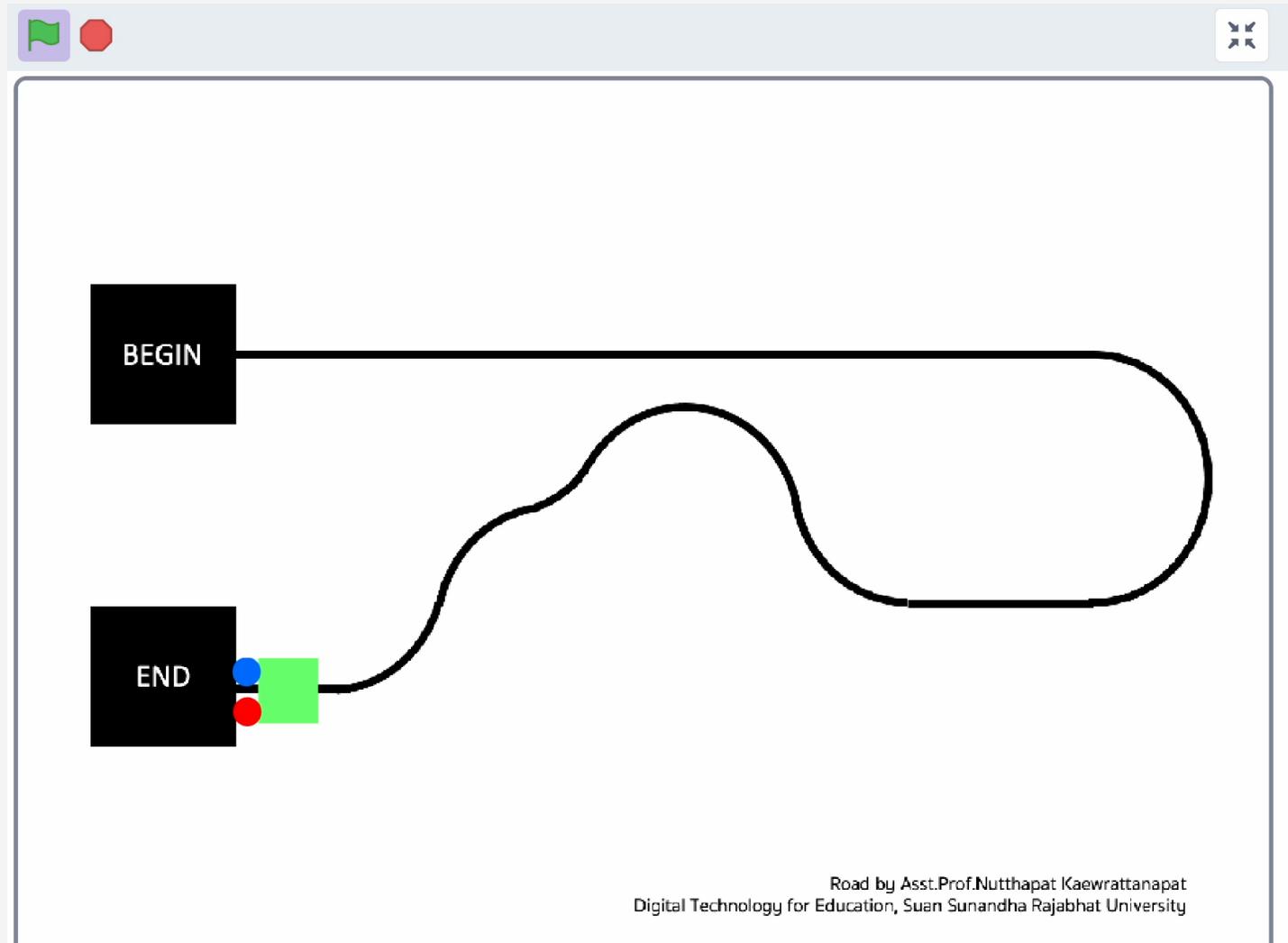
# รถวิ่งตามเส้นทาง ไม่หลุดทาง ได้อย่างไร?

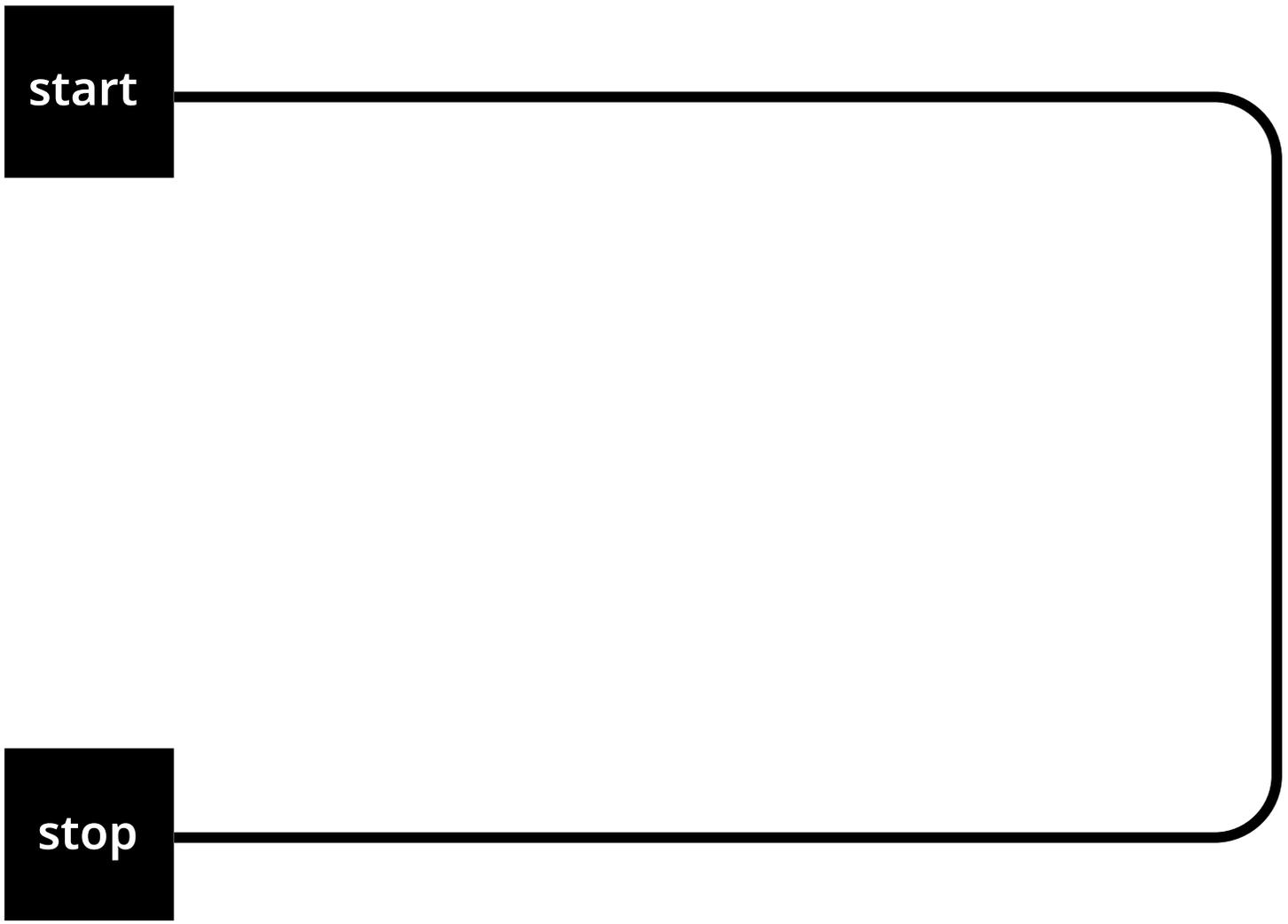


sensor (red) touch road (black) -> turn right 15 angle

# ตัวอย่าง

<https://scratch.mit.edu/projects/715600028>





map: Asst.Prof.Dr.Nutthapat Kaewrattanapat

AutoPilot Car

when green flag clicked

go to x: -131 y: 118

point in direction 90

forever loop:

- if color red is touching black? then turn 5 degrees
- if color blue is touching black? then turn 5 degrees
- if color red is touching black? and color blue is touching black? then move 0 steps; else move 2 steps

start

stop

Sprite: RoboticCar, x: 31, y: 118, Size: 45, Direction: 90

Stage

Backdrops: 4

# ห้ามชนรถคันหน้า พบแล้วให้ออдрอ



map: Asst.Prof.Dr.Nutthapat Kaewrattanapat

```
when clicked
  go to x: -142 y: 126
  point in direction 90
  forever
    if color red is touching black ? then
      turn 5 degrees
    if color blue is touching black ? then
      turn 5 degrees
    if color red is touching purple ? or color blue is touching purple ? then
      move -2 steps
    if color red is touching black ? and color blue is touching black ? then
      move 0 steps
    else
      move 2 steps
```

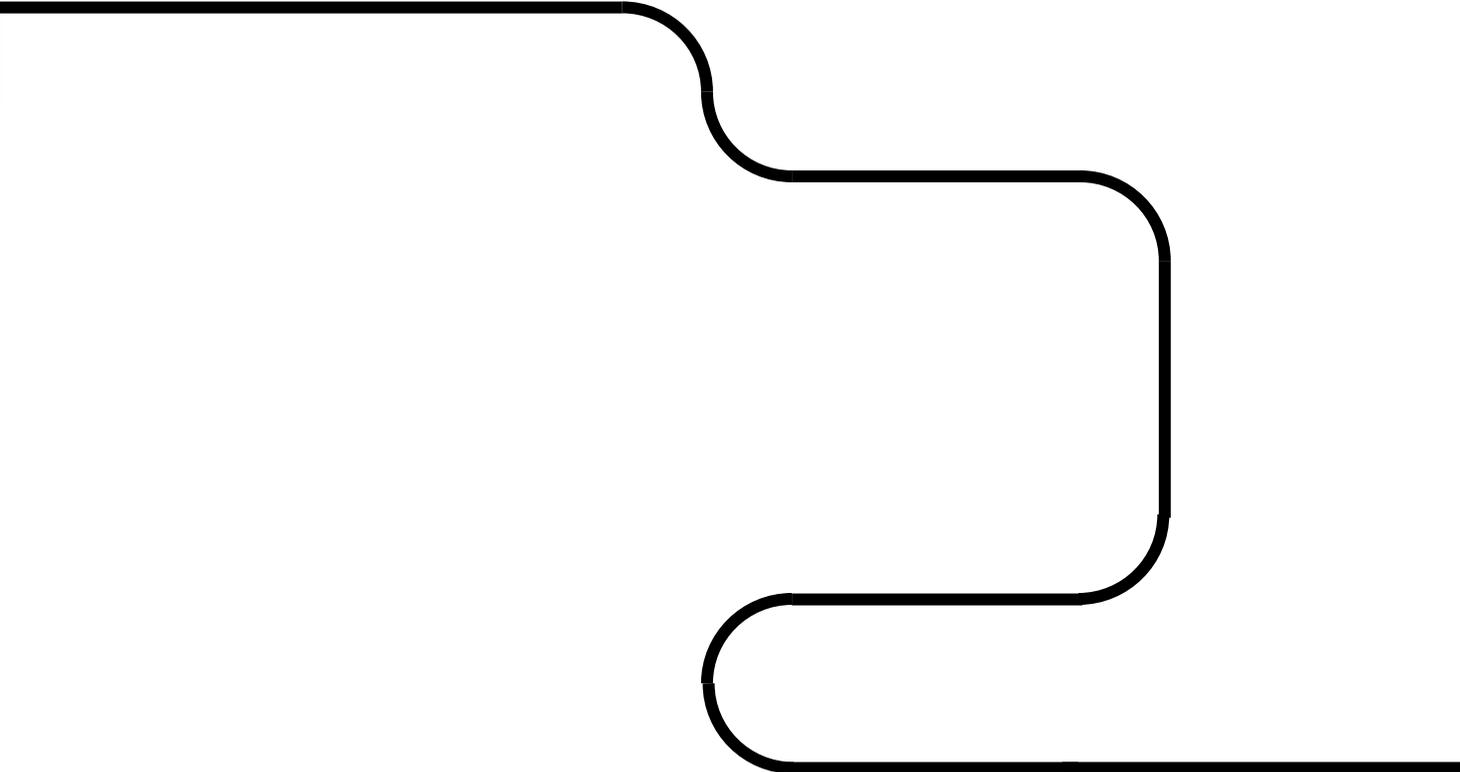


Sprite: RoboticCar  
x: 212 y: 32  
Show:    
Size: 45 Direction: 180



Stage  
Backdrops: 5

**start**



**stop**

map: Asst.Prof.Dr.Nutthapat Kaewrattanapat



## ห้ามวิ่งออกจากถนน



map: Asst.Prof.Dr.Nutthapat Kaewrattanapat

## ห้ามวิ่งออกจากถนน และต้องสลับเลนเมื่อพบรถในเลนเดียวกันเพื่อวิ่งไปจนสุดทาง



map: Asst.Prof.Dr.Nutthapat Kaewrattanapat

# ห้ามวิ่งออกจากถนน และต้องสลับเลนเมื่อพบรถในเลนเดียวกันเพื่อวิ่งไปจนสุดทาง



map: Asst.Prof.Dr.Nutthapat Kaewrattanapat