

บทที่ 3

การโปรแกรมแบบเงื่อนไขในภาษา Python

Chapter 3 Python and Conditional Statement



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตรบัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCNI-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตรข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- **Chapter 3 – Conditional Statement**
- Chapter 4 – Iteration Statement
- Chapter 5 – Strings
- Chapter 6 – Lists, Tuples, Sets, Dictionaries
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Chapter 3 – Conditional Statement

1. Comparison Operator
2. Conditional Statement
3. if
4. if else
5. if elif
6. if elif else
7. Random number
8. unicode

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

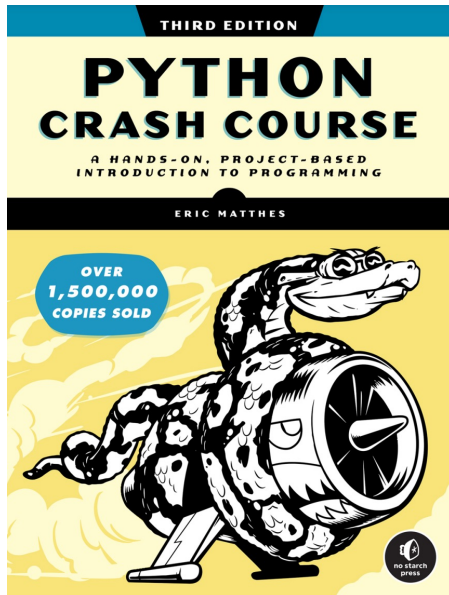
Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
1 พท 7 ธค 66	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 การโปรแกรมคอมพิวเตอร์	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
2 พท 14 ธค 66	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
3 พท 21 ธค 66	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
4 พท 28 ธค 66 ONLINE	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement) (ต่อ)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
5 พท 4 มค 67	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
6 พท 11 มค 67	บทที่ 5 การโปรแกรมสายอักขระ (String)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
7 พท 18 มค 67	บทที่ 6 ลิสต์ ทัวเพิล เซต และดิกชันนารี	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
8 พท 25 มค 67	สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			

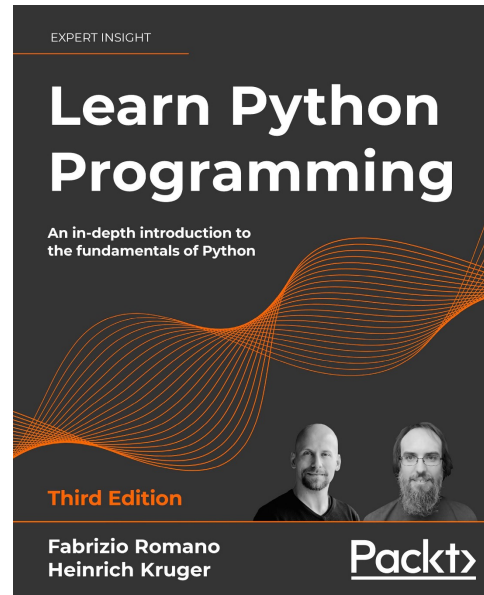
Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
9 พฤษ 1 กพ 67	บทที่ 7 ฟังก์ชัน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
10 พฤษ 8 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
12 พฤษ 15 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
13 พฤษ 22 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
14 พฤษ 29 กพ 67	บทที่ 10 วงจรการพัฒนาระบบ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
15 พฤษ 7 มีค 67	สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
16 พฤษ 14 มีค 67	ส่งโครงงาน และนำเสนอ 20%			
17 พฤษ 21 มีค 67				

Learning Materials – References



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- <https://www.w3schools.com/python/>
- <https://realpython.com/>
- <https://docs.python.org/3/tutorial/index.html>
- <https://pythontutor.com/>

โครงสร้างการควบคุมโปรแกรม (Program Control Structures)

โครงสร้างการควบคุมโปรแกรม (Program Control Structures) ในการเขียนโปรแกรม มี 3 ลักษณะหลัก ได้แก่

- การโปรแกรมแบบลำดับ (Sequential Statement)
- การโปรแกรมแบบเงื่อนไข (Conditional Statement)
- การโปรแกรมแบบทำซ้ำ (Iteration Statement)

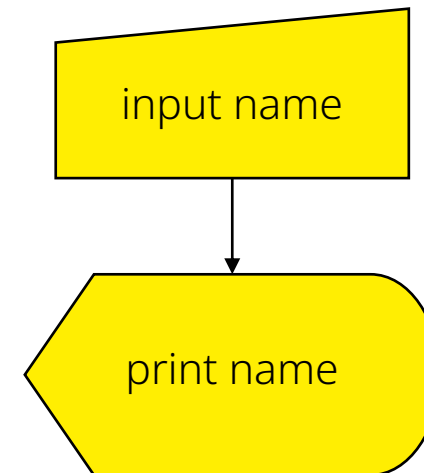
แต่ละลักษณะมีความสำคัญและทำหน้าที่ที่แตกต่างกันในการควบคุมการทำงานของโปรแกรม

โครงสร้างการควบคุมโปรแกรม (Program Control Structures)

1. Sequential Statement (คำสั่งแบบลำดับ)

- คำสั่งแบบลำดับเป็นรูปแบบพื้นฐานที่สุดของโครงสร้างการควบคุมโปรแกรม
- โปรแกรมจะทำงานเรียงตามลำดับของคำสั่งที่เขียนไว้ในโปรแกรม เรียกว่า Top-Down
- ไม่มีการตรวจสอบเงื่อนไขหรือการซ้ำ
- ทุกคำสั่งจะถูกทำงานหนึ่งครั้งตามลำดับที่ปรากฏ

```
# Sequential Statement  
# Code 1  
name = input("Please enter your name:")  
print("Hello,", name)
```



โครงสร้างการควบคุมโปรแกรม (Program Control Structures)

2. Conditional Statement (คำสั่งเงื่อนไข)

คำสั่งเงื่อนไขช่วยให้โปรแกรมสามารถเลือกทำงานทางเลือกต่างๆ ตามเงื่อนไขที่กำหนด

โครงสร้างนี้ประกอบด้วยคำสั่ง **if**, **elif**, และ **else** ในภาษา Python

ตัวอย่าง เช่น ถ้าเงื่อนไขเป็นจริง (True) บล็อกของคำสั่งใน if จะถูกทำงาน;
 ถ้าเงื่อนไขเป็นเท็จ (False), โปรแกรมอาจเลือกทำงานบล็อกใน elif หรือ else

Conditional Statement (คำสั่งเงื่อนไข) – if

```
1.# Code 2.1  
2.sun = True  
3.if sun:  
4.print("Morning")
```

Morning

```
1.# Code 2.2  
2.sun = False  
3.if sun:  
4.print("Morning")
```

Conditional Statement (คำสั่งเงื่อนไข) – if

```
height = 175
weight = 62
if (height > 170 and height <= 180) :
    print("Height: Perfect")
```

A

Height: Perfect

B

Nil

Conditional Statement (คำสั่งเงื่อนไข) – if

Set a condition to print 'You are tall.' if the height in the variable my_height is greater than 170

```
my_height = 175
```

```
if _____ :  
    print('You are tall.')
```

Conditional Statement (คำสั่งเงื่อนไข) – if

Set a condition to print 'You are wealthy and tall.' if the salary in my_salary is greater than 35000 and my_height is greater than 170.

```
my_salary = _____
```

```
my_height = _____
```

```
if _____ :  
    print('You are wealthy and tall.')
```


Conditional Statement (คำสั่งเงื่อนไข) – if

```
object1="strawberry"
```

```
object2="pencil"
```

```
if (object1=="strawberry" or object2=="watermelon") :  
    print("There is fruit in the basket")
```

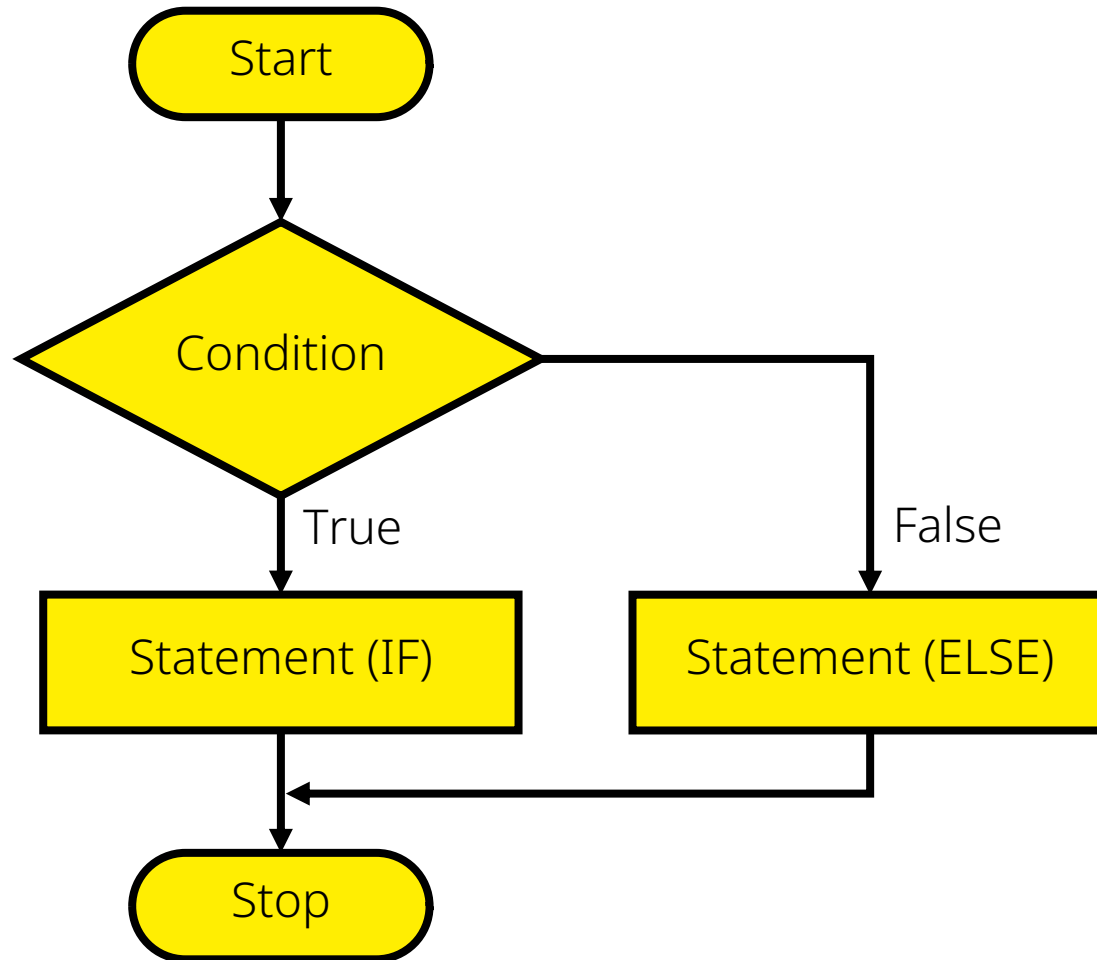
A

There is fruit in the basket

B

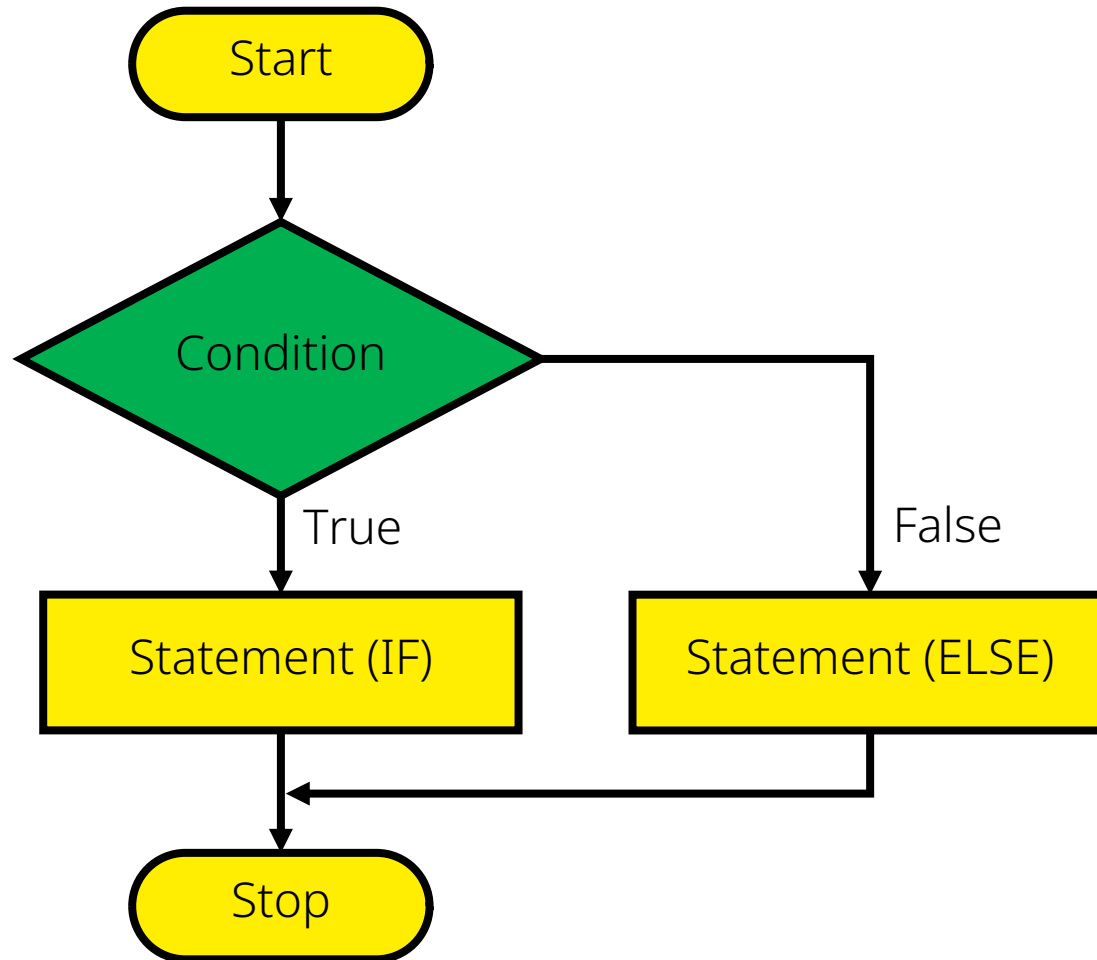
Nil

Conditional Statement (คำสั่งเงื่อนไข) – if else



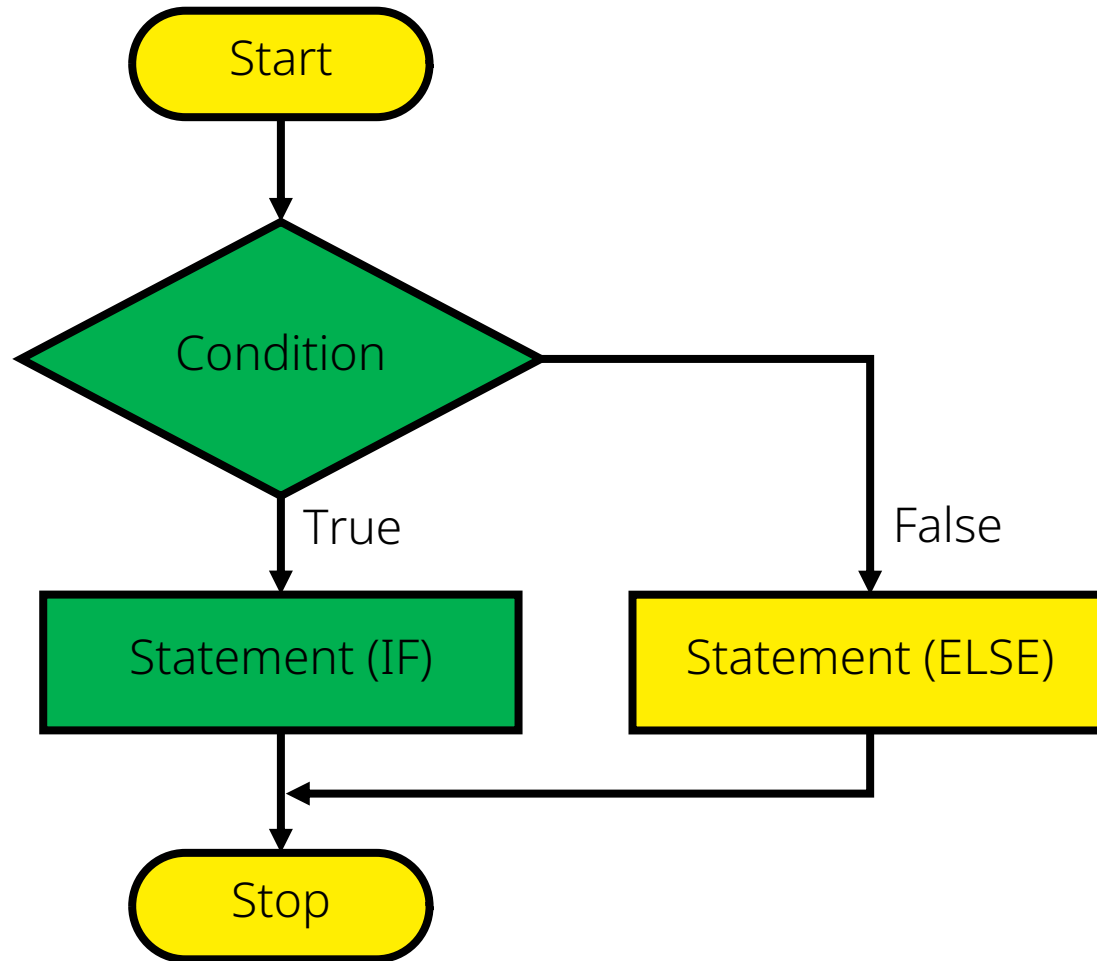
```
if( Condition ):  
    Statement (IF)  
else:  
    Statement (ELSE)
```

Conditional Statement (คำสั่งเงื่อนไข) – if else



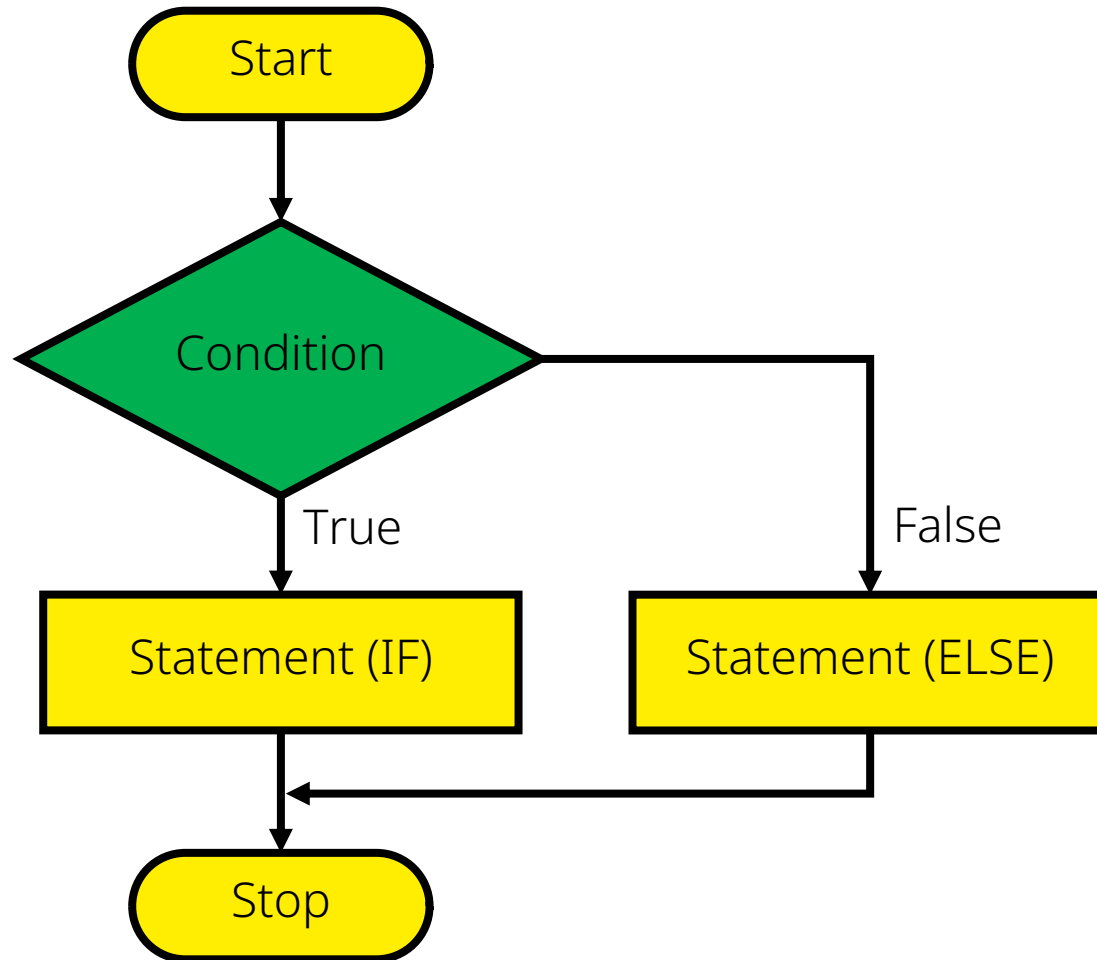
if(Condition):
 Statement (IF)
else:
 Statement (ELSE)

Conditional Statement (คำสั่งเงื่อนไข) – if else



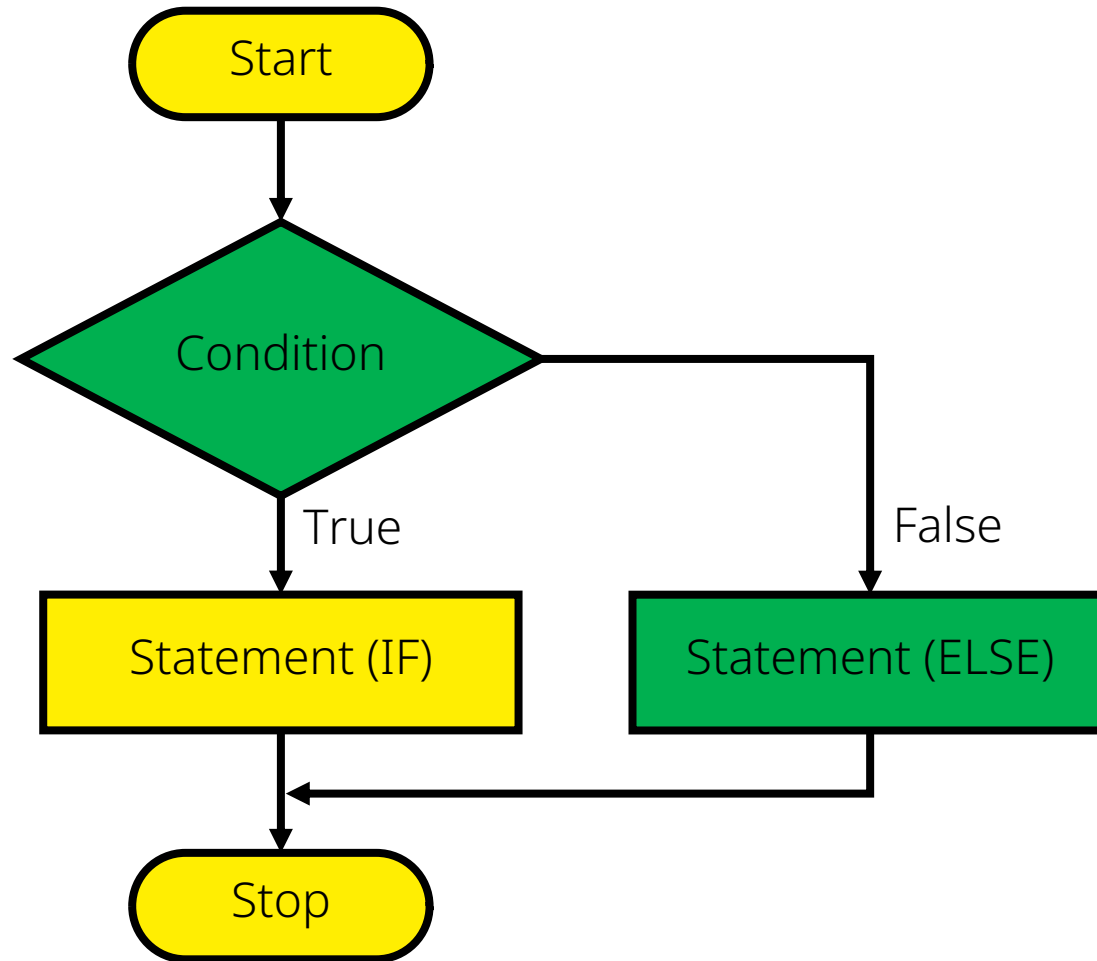
if(Condition):
 Statement (IF)
else:
 Statement (ELSE)

Conditional Statement (คำสั่งเงื่อนไข) – if else



```
if( Condition ):  
    Statement (IF)  
else:  
    Statement (ELSE)
```

Conditional Statement (คำสั่งเงื่อนไข) – if else



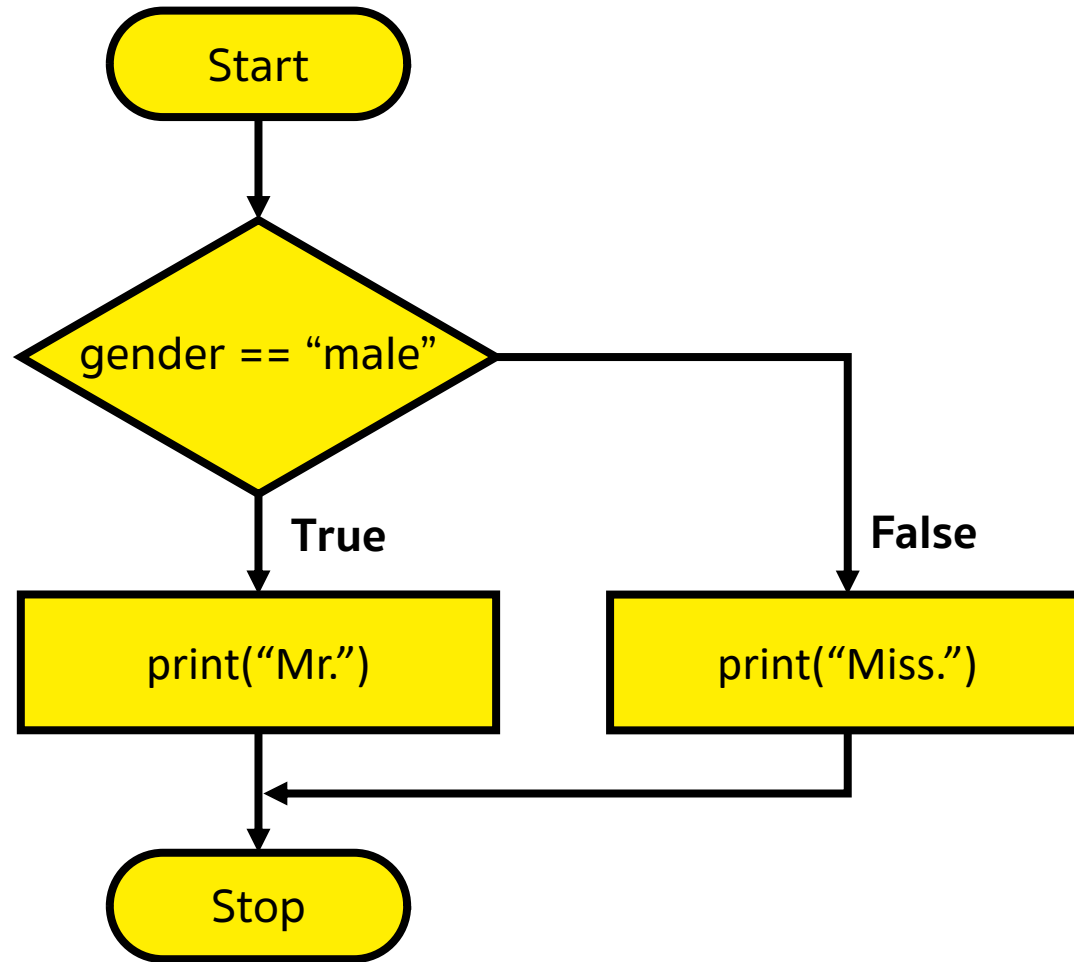
```
if( Condition ):  
    Statement (IF)  
else:  
    Statement (ELSE)
```

ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

ตัวดำเนินการเปรียบเทียบ (Comparison Operators) เป็นตัวดำเนินการที่ใช้ในการเปรียบเทียบค่าของตัวแปรหรือค่าที่มีอยู่ในนิพจน์ เพื่อตรวจสอบว่าสอดคล้องกับเงื่อนไขที่กำหนดหรือไม่ ผลลัพธ์ของการเปรียบเทียบจะเป็นค่าบูลีน (Boolean) ซึ่งอาจเป็น True (จริง) หรือ False (เท็จ) ตามผลการเปรียบเทียบ

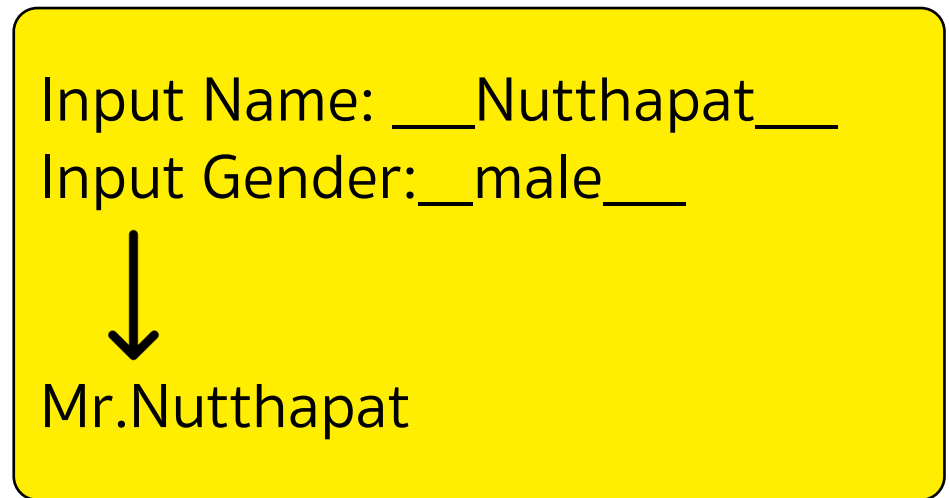
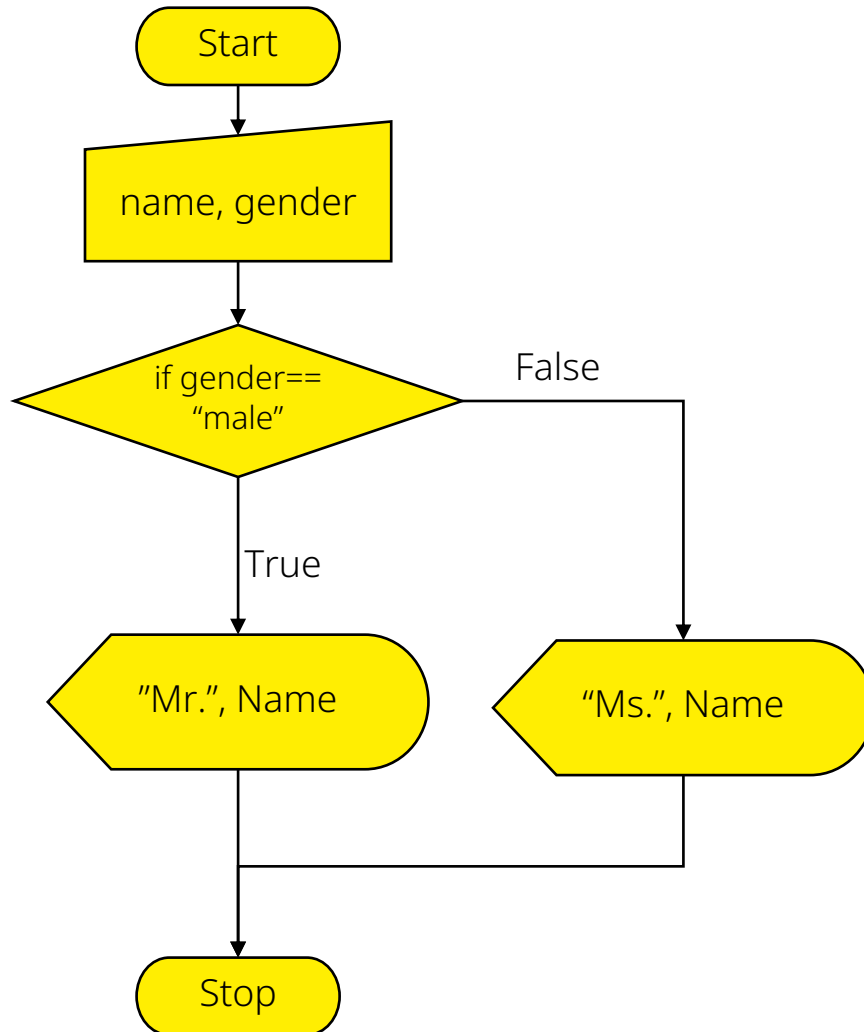
1. เท่ากับ (==): ตรวจสอบว่าค่าของสองค่าเท่ากันหรือไม่
ตัวอย่าง: $x == y$ จะเป็น True ถ้า x มีค่าเท่ากับ y
2. ไม่เท่ากับ (!=): ตรวจสอบว่าค่าของสองค่าไม่เท่ากัน
ตัวอย่าง: $x != y$ จะเป็น True ถ้า x ไม่เท่ากับ y
3. มากกว่า (>): ตรวจสอบว่าค่าด้านซ้ายมากกว่าค่าด้านขวา
ตัวอย่าง: $x > y$ จะเป็น True ถ้า x มากกว่า y
4. น้อยกว่า (<): ตรวจสอบว่าค่าด้านซ้ายน้อยกว่าค่าด้านขวา
ตัวอย่าง: $x < y$ จะเป็น True ถ้า x น้อยกว่า y
5. มากกว่าหรือเท่ากับ (>=): ตรวจสอบว่าค่าด้านซ้ายมากกว่าหรือเท่ากับค่าด้านขวา
ตัวอย่าง: $x >= y$ จะเป็น True ถ้า x มากกว่าหรือเท่ากับ y
6. น้อยกว่าหรือเท่ากับ (<=): ตรวจสอบว่าค่าด้านซ้ายน้อยกว่าหรือเท่ากับค่าด้านขวา
ตัวอย่าง: $x <= y$ จะเป็น True ถ้า x น้อยกว่าหรือเท่ากับ y

Conditional Statement (คำสั่งเงื่อนไข) – if else

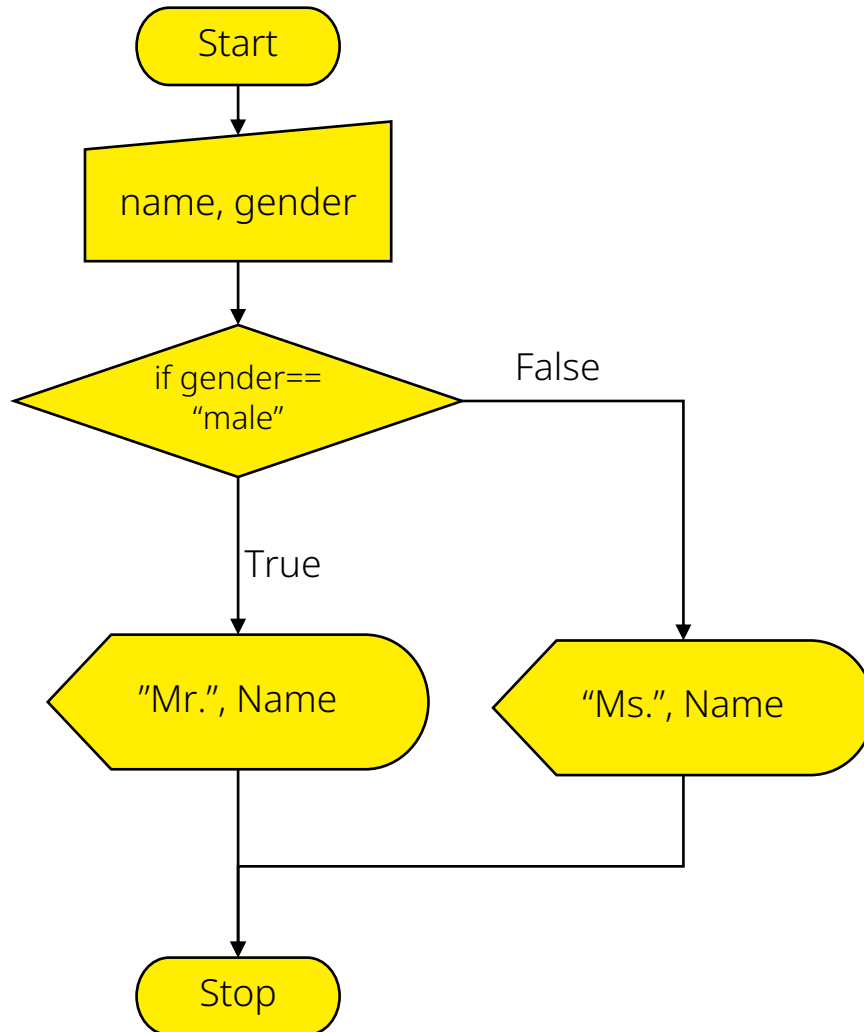


```
if( gender == "male" ):  
    print("Mr.")  
else:  
    print("Miss.")
```


Practice 1: ให้รับข้อมูลเพศ และเติมคำนำหน้าชื่อตาม Flowchart

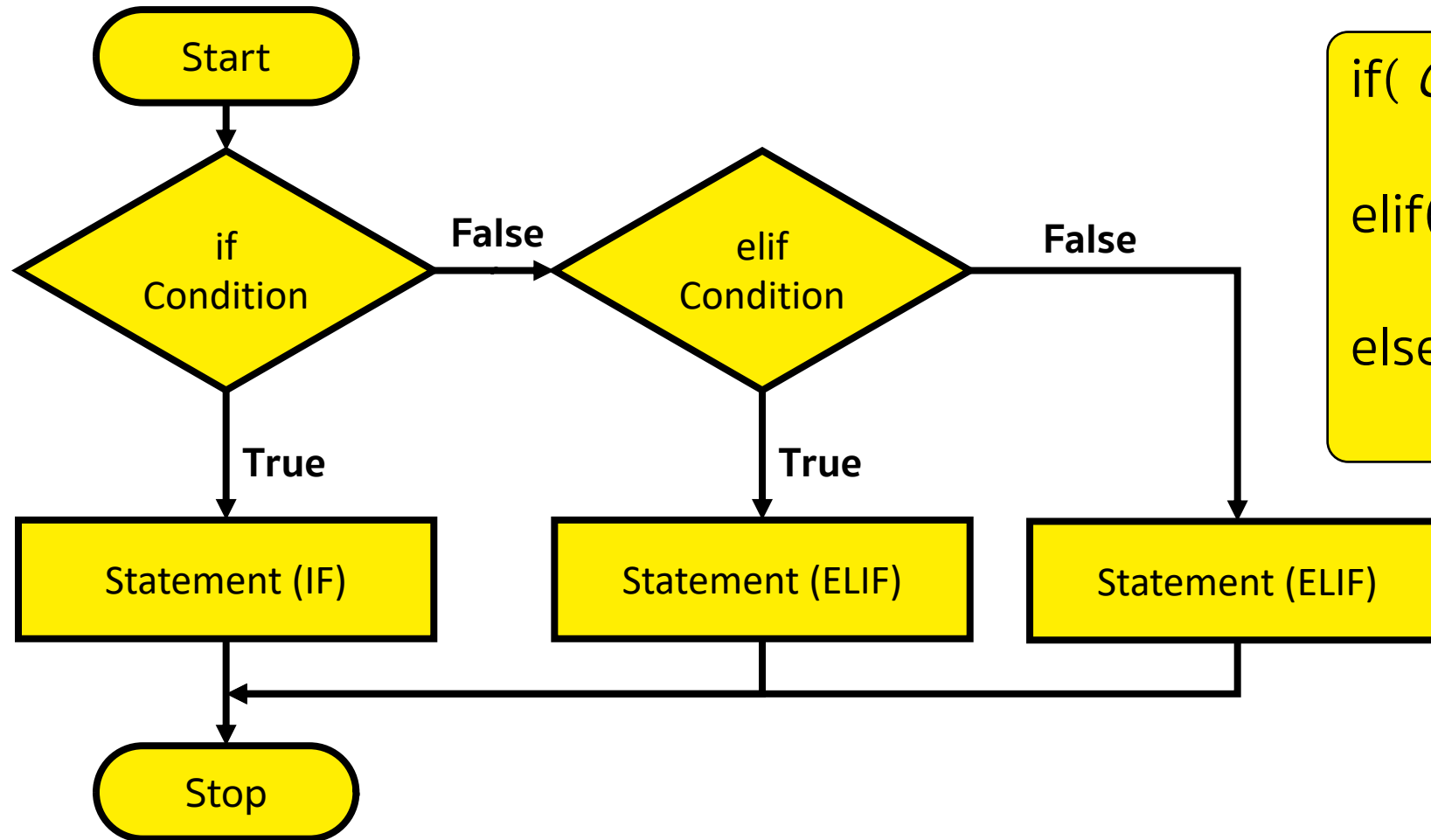


Practice 1: ให้รับข้อมูลเพศ และเติมคำนำหน้าชื่อตาม Flowchart



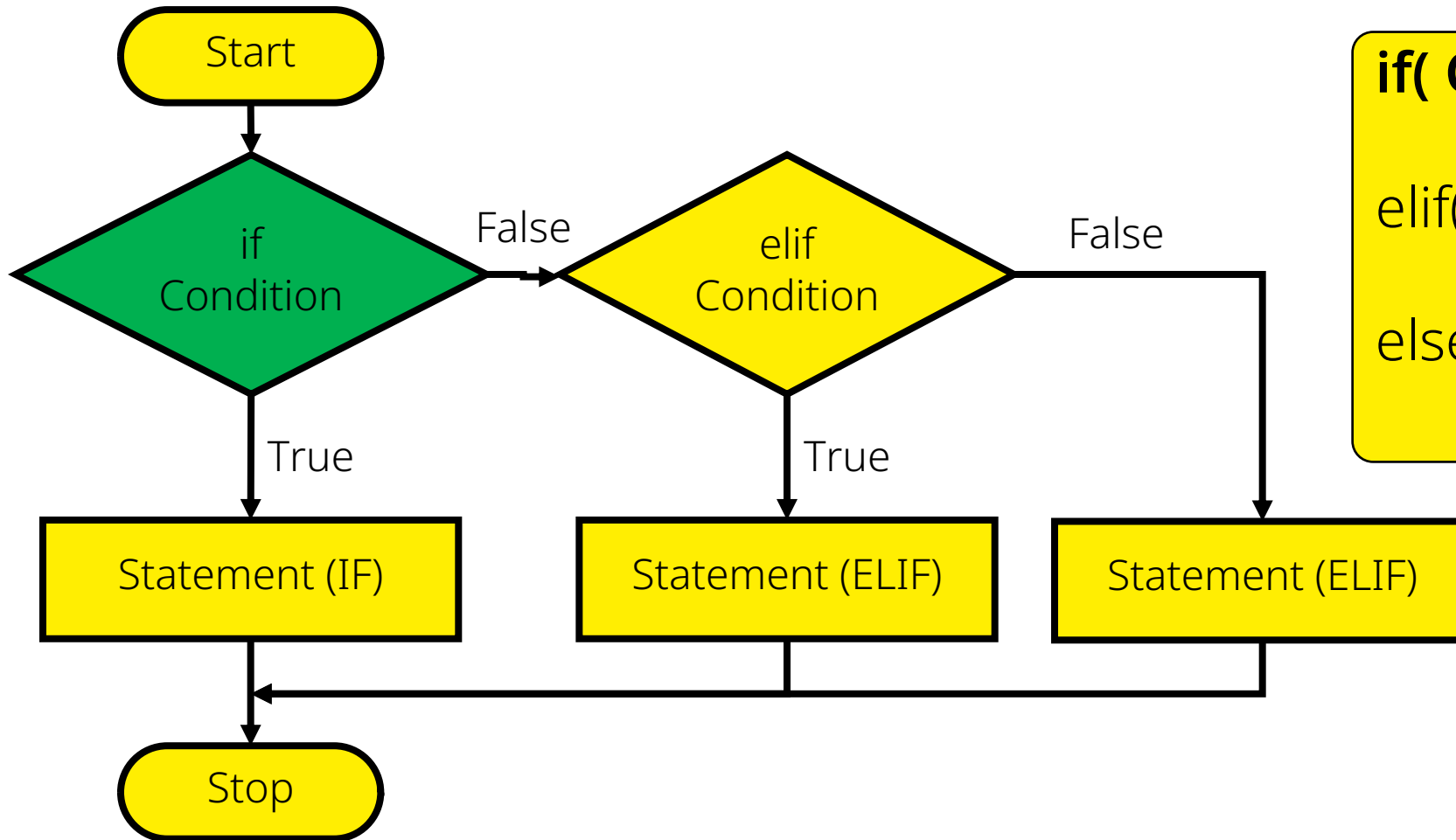
```
name = input("Name: ")
gender = input("Gender: ")
if(gender == "male"):
    print("Mr. " + name)
else:
    print("Ms. " + name)
```

Conditional Statement (คำสั่งเงื่อนไข) – if elif else



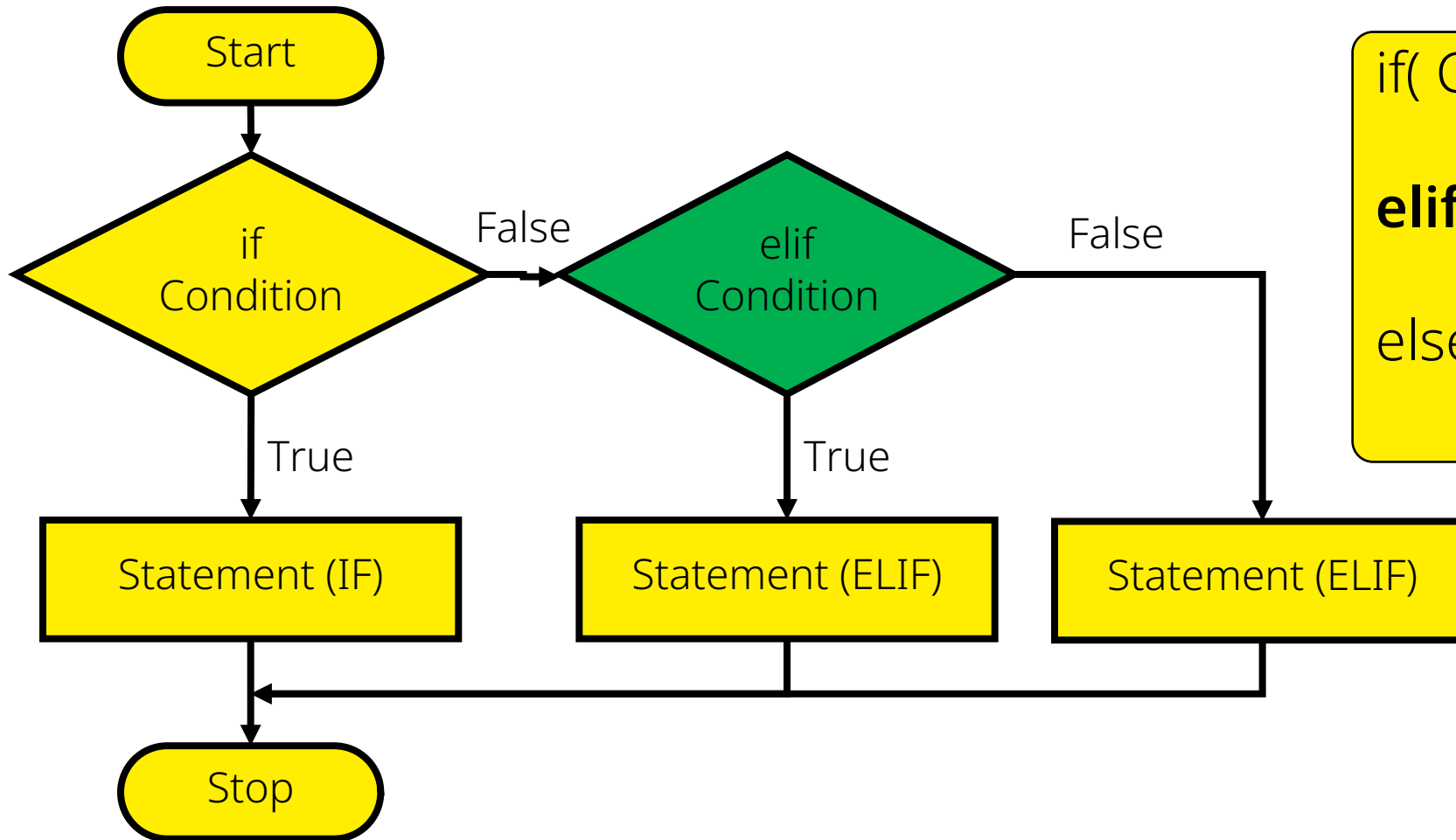
```
if( Condition ):
    Statement (IF)
elif( Condition ):
    Statement (ELIF)
else:
    Statement (ELSE)
```

Conditional Statement (คำสั่งเงื่อนไข) – if elif else



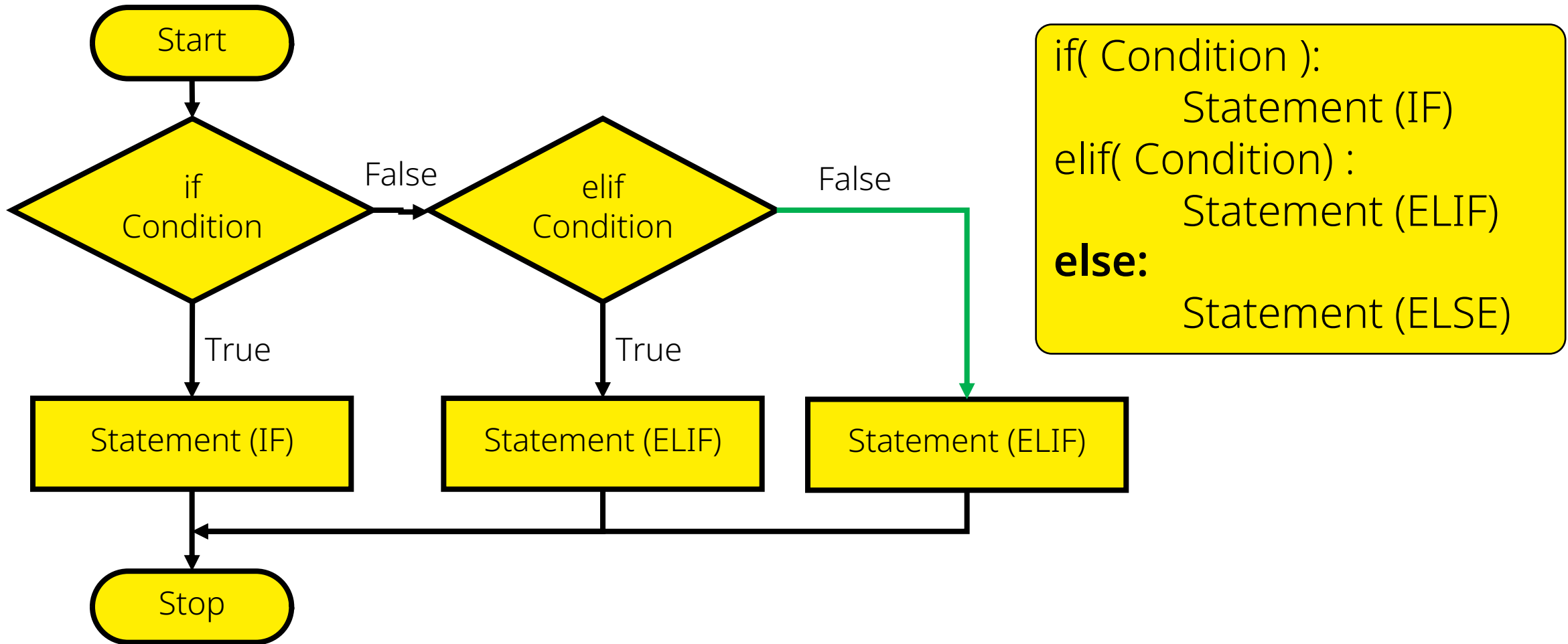
```
if( Condition ):
    Statement (IF)
elif( Condition) :
    Statement (ELIF)
else:
    Statement (ELSE)
```

Conditional Statement (คำสั่งเงื่อนไข) – if elif else



if(Condition):
 Statement (IF)
elif(Condition) :
 Statement (ELIF)
else:
 Statement (ELSE)

Conditional Statement (คำสั่งเงื่อนไข) – if elif else



Conditional Statement (คำสั่งเงื่อนไข) – if elif else

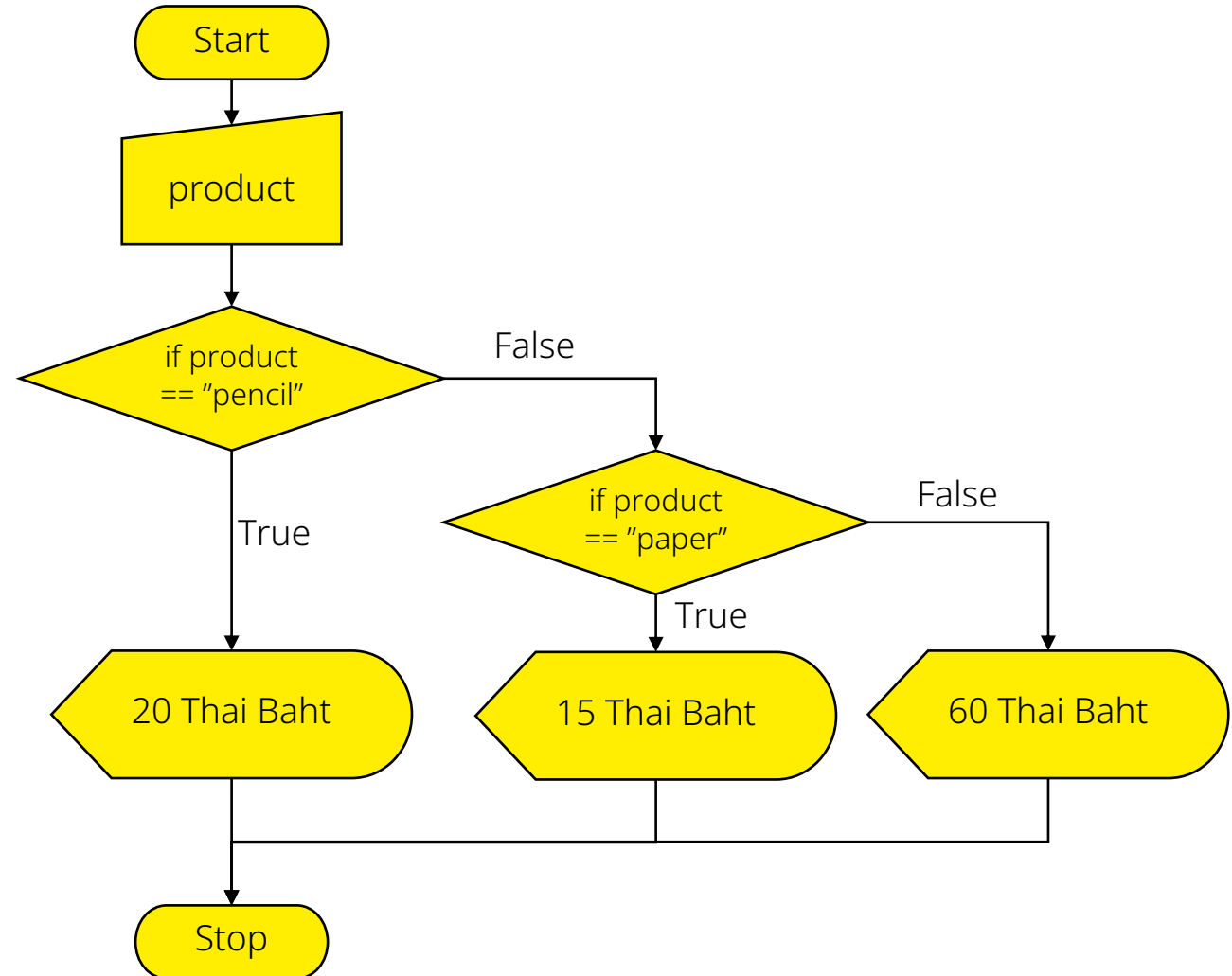
```
# Condition Statement
# Code 2
product = input("Please enter product:")
if product == "pencil":
    print(product, "20 Thai Baht")
elif product == "paper":
    print(product, "15 Thai Baht")
else:
    print(product, "60 Thai Baht")
```

คำอธิบาย Flowchart

1. เริ่ม
2. รับข้อมูล (input) ชื่อสินค้า โดยใช้คำสั่ง
product = input("Please enter product:")
3. ตรวจสอบเงื่อนไขแรก:
ถ้า product == "pencil" เป็นจริง ให้ พิมพ์
ข้อความ product, "20 Thai Baht"
4. ตรวจสอบเงื่อนไขที่สอง (ถ้าเงื่อนไขแรกไม่เป็นจริง):
ถ้า product == "paper" เป็นจริง ให้ พิมพ์
ข้อความ product, "15 Thai Baht"
5. ถ้าเงื่อนไขแรกและเงื่อนไขที่สองไม่เป็นจริง ให้ไปที่ else:
พิมพ์ข้อความ product, "60 Thai Baht"
6. จบการทำงาน

Conditional Statement (คำสั่งเงื่อนไข) – if elif else

```
# Condition Statement
# Code 2
product = input("Please enter product:")
if product == "pencil":
    print(product, "20 Thai Baht")
elif product == "paper":
    print(product, "15 Thai Baht")
else:
    print(product, "60 Thai Baht")
```



modulus (หรือเครื่องหมาย %)

การใช้ modulus (หรือเครื่องหมาย %) เป็นการหารเลขและคืนเศษของการหาร โดยการใช้ modulus คุณจะได้ผลลัพธ์เป็นเศษที่เหลือจากการหารของเลข 2 จำนวน โดยทั่วไปการใช้ modulus จะมีรูปแบบดังนี้

$a \% b$

a คือตัวเลขที่จะนำไปหาร

b คือตัวเลขที่ใช้ในการหาร a

ผลลัพธ์ของ $a \% b$ คือเศษที่เหลือจากการหาร a ด้วย b .

modulus (หรือเครื่องหมาย %)

10 % 3 จะได้ผลลัพธ์เป็น 1 เพราะ 10หาร 3 แล้วเหลือเศษ 1

20 % 4 จะได้ผลลัพธ์เป็น 0 เพราะ 20หาร 4 ลงตัว ไม่มีเศษ

15 % 7 จะได้ผลลัพธ์เป็น 1 เพราะ 15หาร 7 แล้วเหลือเศษ 1

4 % 2 ได้ ?

6 % 2 ได้?

8 % 2 ได้?

100 % 2 ได้?

modulus (หรือเครื่องหมาย %)

```
number = 8
if number % 2 == 0:
    print(number, "เป็นเลขคู่")
else:
    print(number, "เป็นเลขคี่")
```

8 เป็นเลขคู่

modulus (หรือเครื่องหมาย %)

```
#ตรวจสอบเลขเหล่านี้เป็นเลขคู่ หรือ คี่  
num1 = 204  
num2 = -8  
num3 = 22.22
```

modulus (หรือเครื่องหมาย %)

```
#ตรวจสอบเลขเหล่านี้หาร 3 ลงตัวไหม  
num1 = 27  
num2 = 21  
num3 = 972
```

modulus (หรือเครื่องหมาย %)

```
#ตรวจสอบเลขเหล่านี้หาร 5 ลงตัวไหม  
num1 = 225  
num2 = 1225  
num3 = 100.105
```

modulus (หรือเครื่องหมาย %)

การใช้ modulus มีประโยชน์ในหลายสถานการณ์ เช่น

- ตรวจสอบว่าเลขคู่หรือเลขคี่: ในการตรวจสอบว่าเลขหนึ่งเป็นเลขคู่หรือเลขคี่ โดยสามารถใช้ **$n \% 2$** โดยถ้าผลลัพธ์เป็น 0 แสดงว่าเป็นเลขคู่ และถ้าผลลัพธ์ไม่เท่ากับ 0 แสดงว่าเป็นเลขคี่
- การสร้างรอบ (loop): การใช้ modulus เป็นวิธีที่ดีในการสร้างรอบหรือลูปที่ทำงานซ้ำเมื่อมีจำนวนตัวเลขที่ต้องการประมวลผลเท่าๆ กัน เช่นในการแสดงตัวเลขคู่ทั้งหมดหรือเลขคี่ทั้งหมดจาก 1 ถึง n
- การใช้ modulus เป็นเครื่องมือที่มีประโยชน์ในการทำงานกับตัวเลขและจำนวนต่างๆ ในการเขียนโปรแกรมและการแก้ปัญหาทางคณิตศาสตร์

การสุ่มจำนวน (Random numbers)

- **การสุ่มจำนวน (random number)** คือ กระบวนการที่ใช้เลือกหรือสร้างตัวเลขที่มีค่าสุ่มจากช่วงหรือชุดข้อมูลที่กำหนด โดยความสุ่มนี้ทำให้ตัวเลขที่ได้มีค่าไม่แน่นอนและไม่สามารถทำนายล่วงหน้าได้ สิ่งที่สำคัญในการสุ่มคือความคล้ายกันในการเกิดตัวเลขที่มีค่าสุ่มอย่างเท่าเทียมกันซึ่งไม่มีลำดับหรือรูปแบบที่ซ้ำกัน
- การสุ่มจำนวนมักถูกนำมาใช้ในหลายสถานการณ์ เช่น การเล่นเกม, การทำการทดลองทางวิทยาศาสตร์, การสร้างข้อมูลทดสอบ (testing data), และอื่น ๆ อีกมากมาย
- ในโปรแกรมคอมพิวเตอร์ การสุ่มจำนวนจะใช้ฟังก์ชันสุ่ม (random function) เพื่อสร้างตัวเลขที่มีค่าสุ่ม ในภาษา Python

```
1. # Code 3
2. import random

3. # สุ่มจำนวนเต็มในช่วง 1 ถึง 10
4. random_int = random.randint(1, 10)
5. print(random_int)

6. # สุ่มจำนวนทศนิยมในช่วง 0.0 ถึง 1.0
7. random_float = random.random()
8. print(random_float)
```


Practice 2: สุ่มและบอกว่าค่านั้นเป็นเลขคู่ หรือ เลขคี่

1. ให้ทำการเขียนโปรแกรมเพื่อสุ่มเลข ระหว่าง 1 ถึง 100
2. นำเลขสุ่มที่ได้ ไปพิจารณาหาว่า ค่านั้นเป็นเลขคู่ หรือ เลขคี่
3. แจ้งออกทางหน้าจอว่าสุ่มได้เลขอะไร และเลขนั้นเป็นเลขคู่ (Even) หรือ เลขคี่ (Odd)

Practice 2: สุ่มและบอกว่าค่านั้นเป็นเลขคู่ หรือ เลขคี่

1. ให้ทำการเขียนโปรแกรมเพื่อสุ่มเลข ระหว่าง 1 ถึง 100
2. นำเลขสุ่มที่ได้ ไปพิจารณาหาว่า ค่านั้นเป็นเลขคู่ หรือ เลขคี่
3. แจ้งออกทางหน้าจอว่าสุ่มได้เลขอะไร และเลขนั้นเป็นเลขคู่ (Even) หรือ เลขคี่ (Odd)

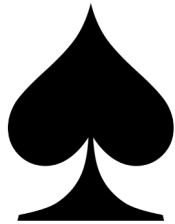
```
1. import random
2. # สุ่มเลขในช่วง 1 ถึง 100
3. random_number = random.randint(1, 100)

4. # ตรวจสอบว่าเลขนี้เป็นเลขคู่หรือเลขคี่
5. if random_number % 2 == 0:
6.     result = "เลขคู่ (Even) "
7. else:
8.     result = "เลขคี่ (Odd) "

9. # แสดงผลลัพธ์
10. print("เลขที่สุ่มได้:", random_number)
11. print("ผลลัพธ์:", result)
```

print() and Unicode

```
print("\u2660 ")
```

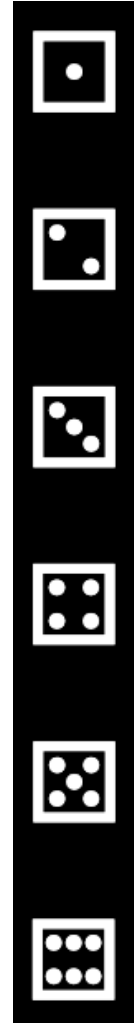


```
1 print("\u2660 ")
2 print("\u2665 ")
3 print("\u2666 ")
4 print("\u2663 ")
```



print() and Unicode

```
print("\u2680")  
print("\u2681")  
print("\u2682")  
print("\u2683")  
print("\u2684")  
print("\u2685")
```



Practice 3 : Randomize numbers and convert them to the sides of the dice

ให้เขียนโปรแกรมเพื่อสุ่มตัวเลขและแปลงให้เป็นด้านของลูกเต๋า

Randomize numbers



Randomize numbers



Practice 3 : Randomize numbers and convert them to the sides of the dice

ให้เขียนโปรแกรมเพื่อสุ่มตัวเลขและแปลงให้เป็นด้านของลูกเต๋า

```
import random
dice = random.randrange(1, 7)
if(dice==1):
    ucode = "\u2680"
elif(dice==2):
    ucode = "\u2681"
elif(dice==3):
    ucode = "\u2682"
elif(dice==4):
    ucode = "\u2683"
elif(dice==5):
    ucode = "\u2684"
elif(dice==6):
    ucode = "\u2685"
print(dice, ucode)
```

Randomize numbers



Assignment 3: Rock-Paper-Scissors

ให้เขียนโปรแกรมเพื่อเป่ายิงฉุบ กับ โปรแกรม โดยมีขั้นตอนดังนี้

1. ให้รับค่า input จากมนุษย์

1 คือ paper

2 คือ rock

3 คือ scissor

2. ให้โปรแกรมสุ่มตัวเลข 1 ถึง 3 มาต่อสู้

1 หมายถึง paper

2 หมายถึง rock

3 หมายถึง scissor

Rock-Paper-Scissors by Nutthapat

1 = Rock

2 = Scissors

3 = Paper

Your turn:

Rock-Paper-Scissors by Nutthapat

1 = Rock

2 = Scissors

3 = Paper

Your turn: 1

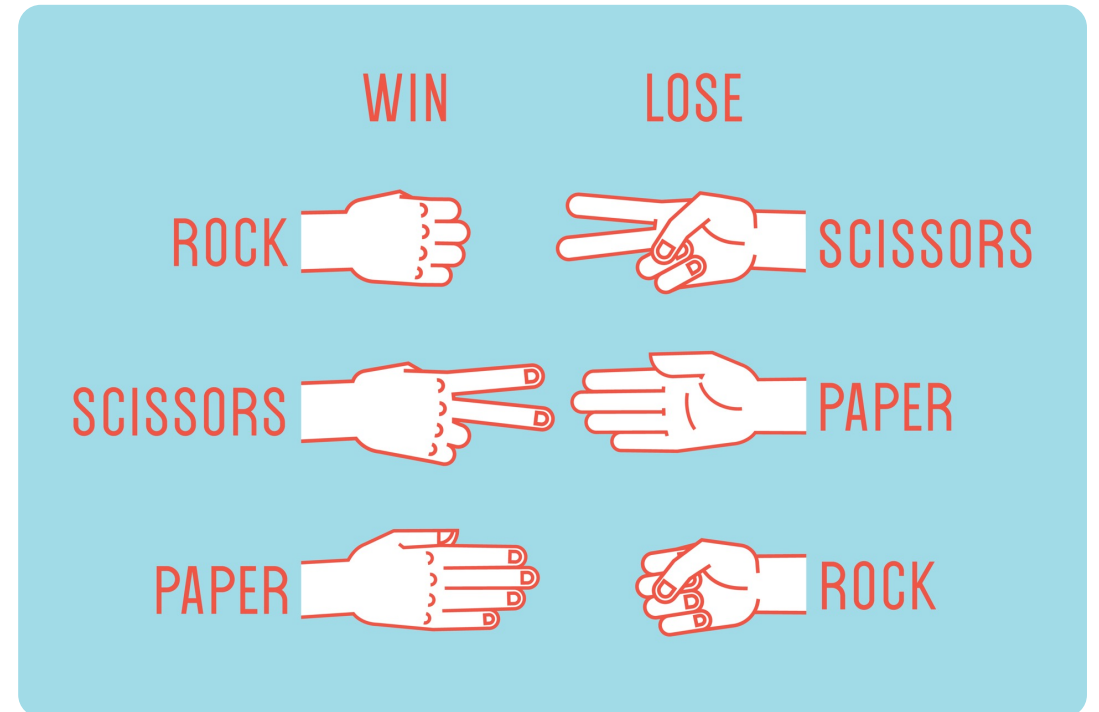
Your choice: Rock and Bot: Scissors

You Win!

Assignment 3: Rock-Paper-Scissors

```
1 import random
2 print("Rock-Paper-Scissors by Nutthapat")
3 print("1 = Rock\n2 = Scissors\n3 = Paper")
4 user = int(input("Your turn: "))
5 bot = random.randrange(1,3)
6 if(user == 1 and bot == 2):
7     print("Your choice: Rock and Bot: Scissors\n You Win!")
8 elif(user == 2 and bot == 3):
9     print("Your choice: Scissors and Bot: Paper\n You Win!")
10 elif(user == 3 and bot == 1):
11     print("Your choice: Paper and Bot: Rock\n You Win!")
12 elif(user == 2 and bot == 1):
13     print("Your choice: Scissors and Bot: Rock\n You Lose!")
14 elif(user == 3 and bot == 2):
15     print("Your choice: Paper and Bot: Scissors\n You Lose!")
16 elif(user == 1 and bot == 3):
17     print("Your choice: Rock and Bot: Paper\n You Lose!")
18 else:
19     print("Draw!")
```

```
Rock-Paper-Scissors by Nutthapat
1 = Rock
2 = Scissors
3 = Paper
Your turn: 1
Your choice: Rock and Bot: Scissors
You Win!
```



TEST

1. ข้อใดไม่ใช่ keywords ใน Conditional Statement ในภาษา Python

A

if

B

if elif

C

else

D

if else

E

if elif else

F

if elseif else

2. จากคำสั่ง แสดงผลข้อใด

```
x = 200
y = 300
x = 1
y = 2
if x == x:
    print(x==x)
else:
    print(x!=x)
```

A 200==1

B True

C False

D 200!=1

E 200==200

F 1==1

3. จากคำสั่ง แสดงผลข้อใด

```
x = 200
y = 300
x = 1
y = 2
if y != y:
    print(y!=y)
else:
    print(y==y)
```

A 300==2

B True

C False

D 300!=2

E 300==300

F 2==2

4. จากคำสั่ง แสดงผลข้อใด

```
n = 10
if (n%2)==0:
    print("2")
elif (n%5)==0:
    print("5")
elif (n%10)==0:
    print("10")
else:
    print("Noting")
```

A

2

B

5

C

10

D

Nothing

E

1

F

2 5 10

5. จากคำสั่ง แสดงผลข้อใด

```
print ("\u2660")
```

A 2660

B 

C 

D \u2660

E error

F u2260

6. จากคำสั่ง แสดงผลข้อใด

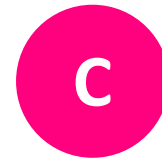
```
import random
random_int = random.randint(1, 10)
print(random_int)
```



1



2



3



4



10



ถูกทุกข้อ

7. จากคำสั่ง ไม่สามารถแสดงผลข้อใด

```
import random  
num = random.randrange(5, 10)  
print(num)
```

A	5	D	9
B	7	E	8
C	10	F	6

8. จากคำสั่ง ไม่สามารถแสดงผลข้อใด

```
import random
num = random.randrange(5,10)
if (num%5)==0:
    print("5")
else
    print("6,7,8,9")
```



5



0



6



8



7



9

9. จากคำสั่ง มีโอกาสแสดงผลข้อใด

```
import random
num = random.randrange(1,10)
num2 = random.randrange(10,20)
if num == num2:
    print("=")
elif num >= num2:
    print(">=")
elif num <= num2:
    print("<=")
else:
    print("<")
```

