

บทที่2 ตรรกศาสตร์ (Logic)

บทที่หนึ่งเราเรียนรู้พื้นฐานของภาษา Python เช่น การเขียนข้อความอธิบายในโปรแกรม, การแสดงผลออกทางหน้าจอ, การสร้างและกำหนดค่าให้ตัวแปร และตัวดำเนินการทางคณิตศาสตร์ บทนี้เราจะเรียนเรื่อง ตัวดำเนินการเปรียบเทียบ, ประพจน์ และตัวดำเนินการทางตรรกศาสตร์

1. ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

การเปรียบเทียบ (Boolean) เป็นการเปรียบเทียบค่าด้านซ้ายและด้านขวาของตัวดำเนินการต่าง ๆ ซึ่งเครื่องหมายที่ใช้ในการเปรียบเทียบ (Operator) เป็นคนละกลุ่มกับตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator) ที่ได้ผลลัพธ์เป็นตัวเลขดังที่ได้เรียนไปในบทที่แล้ว สำหรับบทนี้การเปรียบเทียบจะแสดงค่าผลลัพธ์ออกเป็นค่าความจริง ได้แก่ จริง (True) และเท็จ (False) ในภาษา Python มีตัวดำเนินการดังนี้

1.1 เครื่องหมายน้อยกว่า (<)

เป็นการเปรียบเทียบค่าทางด้านซ้ายน้อยกว่าค่าทางด้านขวาหรือไม่

ตัวอย่าง `1 4 < 9`

```
1: print(4 < 9)
```

ผลลัพธ์

True

อธิบาย

Operator (<) เปรียบเทียบค่าด้านซ้าย (4) กับ ค่าด้านขวา (9) พบว่า ค่าด้านซ้ายน้อยกว่าค่าด้านขวจริง Operator (<) จึงให้ผลลัพธ์เป็น True

1.2 เครื่องหมายน้อยกว่าหรือเท่ากับ (<=)

เป็นการเปรียบเทียบค่าทางด้านซ้ายน้อยกว่าหรือเท่ากับค่าทางด้านขวาหรือไม่

ตัวอย่าง `2 2 <= 2`

```
1: print(2 <= 2)
```

ผลลัพธ์

True

อธิบาย

Operator (<=) เปรียบเทียบค่าด้านซ้าย (2) กับ ค่าด้านขวา (2) พบว่า ค่าด้านซ้ายน้อยกว่าหรือเท่ากับค่าด้านขวจริง Operator (<=) จึงให้ผลลัพธ์เป็น True

1.3 เครื่องหมายมากกว่า (>)

เป็นการเปรียบเทียบค่าทางด้านซ้ายมากกว่าค่าทางด้านขวาหรือไม่

ตัวอย่าง `3 5 > 3`

```
1: print(5 > 3)
```

ผลลัพธ์

True

อธิบาย

Operator (>) เปรียบเทียบค่าด้านซ้าย (5) กับ ค่าด้านขวา (3) พบว่า ค่าด้านซ้ายมากกว่าค่าด้านขวา
จริง Operator (>) จึงให้ผลลัพธ์เป็น True

1.4 เครื่องหมายมากกว่าหรือเท่ากับ (>=)

เป็นการเปรียบเทียบค่าทางด้านซ้ายมากกว่าหรือเท่ากับค่าทางด้านขวาหรือไม่

ตัวอย่าง 4 $5 \geq 5$

```
1: print(5 >=5)
```

ผลลัพธ์

True

อธิบาย

Operator (>=) เปรียบเทียบค่าด้านซ้าย (5) กับ ค่าด้านขวา (5) พบว่า ค่าด้านซ้ายมากกว่าหรือ
เท่ากับค่าด้านขวาจริง Operator (>=) จึงให้ผลลัพธ์เป็น True

1.5 เครื่องหมายเท่ากับ (==)

เป็นการเปรียบเทียบค่าทางด้านซ้ายเท่ากับค่าทางด้านขวาหรือไม่

ตัวอย่าง 5 $7 == 8$

```
1: print(7 ==8)
```

ผลลัพธ์

False

อธิบาย

Operator (==) เปรียบเทียบค่าด้านซ้าย (7) กับ ค่าด้านขวา (8) พบว่า ค่าด้านซ้ายไม่เท่ากับค่า
ด้านขวาจริง Operator (==) จึงให้ผลลัพธ์มีค่าเป็น False

นอกจากนั้น Operator (==) ยังสามารถใช้เปรียบเทียบข้อความว่า ข้อความด้านซ้าย เหมือนกับ
ข้อความด้านขวาหรือไม่

ตัวอย่าง 6 `informatics = informatics`

```
1: print('informatics' == 'informatics')
```

ผลลัพธ์

True

อธิบาย

Operator (==) เปรียบเทียบค่าทางซ้าย ('informatics') กับ ค่าขวา ('informatics') พบว่า ค่าด้านซ้าย
เหมือนกับค่าด้านขวา Operator (==) จึงทำให้ผลลัพธ์เป็น True

1.6 เครื่องหมายไม่เท่ากับ (!=)

เป็นการเปรียบเทียบค่าทางด้านซ้ายไม่เท่ากับค่าทางด้านขวาหรือไม่

ตัวอย่าง 7 $4 \neq 10$

```
1: print(4 !=10)
```

ผลลัพธ์

True

อธิบาย

Operator (!=) เปรียบเทียบค่าซ้าย (4) กับ ค่าขวา (10) พบว่า ค่าด้านซ้ายไม่เท่ากับค่าด้านขวา
Operator (!=) จึงให้ผลลัพธ์เป็น True

1.7 in

เป็นการตรวจสอบค่าทางซ้ายของคำสั่งว่าอยู่ในกลุ่มข้อมูลด้านขวาหรือไม่

ตัวอย่าง 8 แสดงวิธีการใช้งาน in

```
1: x =[1, 2, 3, 4]
2: print(2 in x)
```

ผลลัพธ์

True

อธิบาย

Operator (in) เปรียบเทียบค่าทางซ้าย (2) กับ ค่าขวา (x) ที่เป็นตัวแปร มีค่า (1, 2, 3, 4) พบว่า ค่าซ้ายมีอยู่ในกลุ่มข้อมูลในตัวแปรด้านขวา (x) Operator (in) จึงให้ผลลัพธ์เป็น True

2. ประพจน์ (Propositions)

ประพจน์ (Propositions) คือ ประโยคบอกเล่าหรือข้อความที่เป็นจริง (True) หรือเท็จ (False) อย่างใดอย่างหนึ่งเท่านั้น และเรียก “จริง (True)” และ “เท็จ (False)” ว่าเป็นค่าความจริงของประพจน์

ตัวอย่าง 9 แสดงตัวอย่างของประโยคที่เป็นประพจน์

$1 + 1 = 2$	เป็นประพจน์
0.2 เป็นจำนวนเต็ม	เป็นประพจน์
วาฬเป็นสัตว์เลี้ยงลูกด้วยนม	เป็นประพจน์
$X > 15$	ไม่เป็นประพจน์

จงพิจารณาประโยคต่อไปนี้ว่าเป็นประพจน์หรือไม่ ถ้าเป็นจริงให้บอกค่าความจริงของประพจน์
ตาราง 2 - 1 ตรวจสอบประโยคเป็นประพจน์

ประโยค	เป็นประพจน์หรือไม่	ค่าความจริง
เชียงใหม่เป็นเมืองหลวงของประเทศไทย	เป็น	เท็จ
แมงมุมเป็นแมลง	เป็น	เท็จ
นกเพนกวินบินได้		
0 เป็นจำนวนเต็มบวก		
วันที่ 1 มกราคม ตรงกับวันจันทร์		

การแทนประพจน์ด้วยภาษา python

```
1: p = True      #ให้ p มีความจริงเป็น True
2: print(p)      #แสดงค่าความจริงของ p
```

ผลลัพธ์
True

3. ตัวดำเนินการทางตรรกศาสตร์

3.1 นิเสธ (Not)

$$\neg p$$

นิเสธ คือ การเปลี่ยนค่าความจริงของประพจน์ ให้เป็นตรงกันข้ามจาก True เป็น False และ จาก False เป็น True ใช้สัญลักษณ์แทนด้วย “ \neg ”

ตาราง 2 - 2 ตารางค่าความจริงของ not

p	$\neg p$
F	T
T	F

การเขียนโปรแกรมใน python จะใช้ “not” ในการบ่งบอกว่าเป็นนิเสธ

การแทนประพจน์ด้วยภาษา python ในโปรแกรม PyCharm

```
1: p = True      #ให้ p มีค่าความจริงเป็น True
2: print(not p)  #แสดงค่าความจริงของ  $\neg p$ 
```

ผลลัพธ์
False

3.2 Conjunction (AND)

$$p \wedge q$$

ประพจน์ที่ได้จากการเชื่อมประพจน์ p กับ q ด้วยกันด้วยตัวเชื่อม “และ” จะเขียนแทนด้วยรูปแบบ $p \wedge q$ มีค่าความจริง ดังตาราง 2 - 3

ตาราง 2 - 3 ตารางค่าความจริงของ AND

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

จากตารางค่าความจริงจะเห็นได้ว่า AND มีค่าความจริงเป็น True กรณีเดียวนั้นคือประพจน์ทั้งหมดมีค่าความจริงเป็น True

ในภาษา Python เชื่อมประพจน์ p กับ q โดยใช้ตัวเชื่อม “และ” จะเขียนรูปแบบ p and q ดังนี้

```
1: p = True      #ให้ p มีค่าความจริงเป็น True
2: q = True      #ให้ q มีค่าความจริงเป็น True
3: print(p and q)  #แสดงค่าความจริงของ p และ q
```

ผลลัพธ์
True

```

1:    p = True      #ให้ p มีค่าความจริงเป็น True
2:    q = False     #ให้ q มีค่าความจริงเป็น False
3:    print(p and q)    #แสดงค่าความจริงของ p และ q

```

ผลลัพธ์
False

3.3 Disjunction (OR)

$$p \vee q$$

ประพจน์ที่ได้จากการเชื่อมประพจน์ p กับ q ด้วยกันด้วยตัวเชื่อม “หรือ” จะเขียนแทนด้วย

รูปแบบ $p \vee q$ มีค่าความจริง ดังตาราง 2 - 4

ตาราง 2 - 4 ตารางค่าความจริงของ OR

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

จากตารางค่าความจริงจะเห็นได้ว่า OR มีค่าความจริงเป็น False กรณีเดียวนั้นคือประพจน์ทั้งหมดมีค่าความจริงเป็น False ในภาษา python เชื่อมประพจน์ p กับ q โดยใช้ตัวเชื่อม “หรือ” จะเขียนรูปแบบ $p \text{ or } q$ ดังนี้

```

1:    p = True      #ให้ p มีค่าความจริงเป็น True
2:    q = False     #ให้ q มีค่าความจริงเป็น False
3:    print(p or q)    #แสดงค่าความจริงของ p หรือ q

```

ผลลัพธ์
True

```

1:    p = False     #ให้ p มีค่าความจริงเป็น False
2:    q = False     #ให้ q มีค่าความจริงเป็น False
3:    print(p or q)    #แสดงค่าความจริงของ p หรือ q

```

ผลลัพธ์
False

3.4 Exclusive-OR (XOR)

$$p \oplus q$$

ประพจน์ที่ได้จากการเชื่อมประพจน์ p กับ q ด้วยกันด้วยตัวเชื่อม “XOR” จะเขียนแทนด้วยรูปแบบ $p \oplus q$

มีค่าความจริง ดังตาราง 2 - 5

ตาราง 2 - 5 ตารางค่าความจริงของ XOR

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	F

จากตารางค่าความจริงจะเห็นได้ว่า XOR จะมีค่าความจริงเป็น True ก็ต่อเมื่อประพจน์มีค่าความจริงต่างกัน
ในภาษา python การดำเนินการตัวเชื่อม “XOR” เขียนได้ว่า $p \wedge q$ เมื่อนำมาเขียนโปรแกรมจะได้ดังนี้

<pre>1: p = True 2: q = False 3: print(p ^ q)</pre>	ผลลัพธ์ True
<pre>1: p = False 2: q = False 3: print(p ^ q)</pre>	ผลลัพธ์ False

3.5 Implication (IMPLIES)

$$p \rightarrow q$$

ประพจน์ที่ได้จากการเชื่อมประพจน์ p กับ q ด้วยกันด้วยตัวเชื่อม “ถ้า...แล้ว...” จะเขียนแทนด้วยรูปแบบ

$p \rightarrow q$ มีค่าความจริง ดังตาราง 2 - 6

ตาราง 2 - 6 ตารางค่าความจริงของ IMPLIES

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

จากตารางค่าความจริงจะเห็นได้ว่า IMPLIES จะมีค่าความจริงเป็น False ก็ต่อเมื่อประพจน์แรกมีค่าความจริงเป็น True และประพจน์หลังมีค่าความจริงเป็น False

ภาษา python ไม่มีการดำเนินการตัวเชื่อม “IMPLIES” โดยเฉพาะ แต่เนื่องจาก $p \rightarrow q$ มีความหมายเหมือนกับ $\neg p \vee q$ ซึ่งสามารถใช้ในการเขียนโปรแกรมเพื่อหาค่าความจริงแทนได้ ดังนี้

<pre>1: p = True 2: q = False 3: print((not p) or q)</pre>	ผลลัพธ์ False
<pre>1: p = False 2: q = False 3: print((not p) or q)</pre>	ผลลัพธ์ True

3.6 Biconditional (IFF)

$$p \leftrightarrow q$$

ประพจน์ที่ได้จากการเชื่อมประพจน์ p กับ q ด้วยกันด้วยตัวเชื่อม “ก็ต่อเมื่อ” จะเขียนแทนด้วยรูปแบบ $p \leftrightarrow q$ มีค่าความจริง ดังตาราง 2 - 7

ตาราง 2 - 7 ตารางค่าความจริงของ IFF

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

จากตารางค่าความจริงจะเห็นได้ว่า IFF จะมีค่าความจริงเป็น True ก็ต่อเมื่อประพจน์ทั้งสองมีค่าความจริงเหมือนกัน หรืออาจกล่าวได้ว่า XOR เป็นนิเสธของ IFF

ในภาษา python ไม่มีการดำเนินการตัวเชื่อม “IFF” โดยเฉพาะ แต่เนื่องจาก IFF เป็นนิเสธของ XOR จึงสามารถเขียนได้ดังนี้ $\neg(p \oplus q)$ สามารถใช้ในการเขียนโปรแกรมเพื่อหาค่าความจริงแทน IFF เป็นนิเสธของ XOR เขียนโปรแกรมด้วยภาษา Python ได้ดังนี้

1: <code>p = True</code> 2: <code>q = False</code> 3: <code>print(not(p ^ q))</code>	ผลลัพธ์ False
1: <code>p = False</code> 2: <code>q = False</code> 3: <code>print(not(p ^ q))</code>	ผลลัพธ์ True

นิสิตสามารถใช้เว็บไซต์ <https://sagecell.sagemath.org> ในการแสดงตารางค่าความจริง เมื่อเข้าเว็บไซต์แล้ว ทำการเปลี่ยนภาษาเป็นภาษา Python ดังแสดงในวงกลมสีแดงในรูปภาพ 2 - 1 เขียนโปรแกรมลงในช่องว่างและกดปุ่ม Evaluate เพื่อรันโปรแกรม



รูปภาพ 2 - 1 เว็บไซต์ sagecell.sagemath.org

หมายเหตุ ในเว็บไซต์ <https://sagecell.sagemath.org> จะใช้เครื่องหมาย ! แทน not, & แทน and, | แทน or, -> แทน implies, <-> แทน IFF

ตัวอย่าง 10 $\neg p \wedge (q \vee r)$

เขียนด้วยภาษา Python ได้ดังนี้ ในเว็บไซต์ <https://sagecell.sagemath.org> ได้ดังนี้

```
1: logic = SymbolicLogic()
2: s = logic.statement("p&(q|r)")
3: t = logic.truthtable(s)
4: logic.print_table(t)
```

อธิบาย

ให้นิสิตทดลองเปลี่ยนประพจน์ในเครื่องหมาย " " ในบรรทัดที่ 2 และสังเกตผลลัพธ์ที่ได้

4. การสมมูล

การสมมูล คือ ประพจน์ประกอบที่เขียนต่างกัน แต่มีความหมายเหมือนกัน โดยประพจน์ประกอบทั้งสองจะต้องมีค่าความจริงเหมือนกันทุกกรณี โดยจะใช้สัญลักษณ์ " \Leftrightarrow " เพื่อแสดงว่าประพจน์ทั้งสองสมมูลกัน

ตัวอย่าง 11 ตรวจสอบ $(p \vee q) \wedge \neg(p \wedge q)$ และ $p \oplus q$ สมมูลกัน

ตาราง 2 - 8 ตรวจสอบ $(p \vee q) \wedge \neg(p \wedge q)$ และ $p \oplus q$ สมมูลกัน

p	q	$p \vee q$	$p \wedge q$	$\neg(p \wedge q)$	$(p \vee q) \wedge \neg(p \wedge q)$	$p \oplus q$
F	F	F	F	T	F	F
F	T	T	F	T	T	T
T	F	T	F	T	T	T
T	T	T	T	F	F	F

จากตารางค่าความจริงจะเห็นได้ว่า $p \oplus q$ และ $(p \vee q) \wedge \neg(p \wedge q)$ มีค่าความจริงเหมือนกันทุกแถว เพราะฉะนั้น $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$

เขียนด้วยภาษา Python ได้ดังนี้ ในเว็บไซต์ <https://sagecell.sagemath.org> ได้ดังนี้

```
1: logic = SymbolicLogic()
2: s = logic.statement("(p|q) &! (p&q) <->! (p<->q)")
3: t = logic.truthtable(s)
4: logic.print_table(t)
```

สังเกตผลลัพธ์ที่ได้ ซึ่งจะเห็นว่าให้ค่าความจริงเป็น True ทุกแถว แสดงว่าประพจน์ $(p|q)\&(p\&q) \Leftrightarrow !(p<->q)$ มีค่าความจริงเหมือนกันทุกแถว จึงสรุปได้ว่า $(p \vee q) \wedge \neg(p \wedge q) \Leftrightarrow p \oplus q$

ตัวอย่าง 12 ตรวจสอบ $\neg p \vee q$ และ $p \rightarrow q$ สมมูลกัน

ตาราง 2 - 9 ตรวจสอบ $\neg p \vee q$ และ $p \rightarrow q$ สมมูลกัน

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
F	F	T	T	T
F	T	T	T	T
T	F	F	F	F
T	T	F	T	T

จากตารางค่าความจริงจะเห็นได้ว่า $p \rightarrow q$ และ $\neg p \vee q$ มีค่าความจริงเหมือนกันทุกแถว เพราะฉะนั้น

$$p \rightarrow q \Leftrightarrow \neg p \vee q$$

ตัวอย่าง 13 ตรวจสอบ $p \leftrightarrow q$ และ $\neg(p \oplus q)$ สมมูลกัน

ตาราง 2 - 10 ตรวจสอบ $p \leftrightarrow q$ และ $\neg(p \oplus q)$ สมมูลกัน

p	q	$p \oplus q$	$\neg(p \oplus q)$	$p \leftrightarrow q$
F	F	F	T	T
F	T	T	F	F
T	F	T	F	F
T	T	F	T	T

จากตารางค่าความจริงจะเห็นได้ว่า $p \leftrightarrow q$ และ $\neg(p \oplus q)$ มีค่าความจริงเหมือนกันทุกแถว เพราะฉะนั้น

$$p \leftrightarrow q \Leftrightarrow \neg(p \oplus q)$$