

Section A : สัญลักษณ์ที่ใช้สร้างประโยค (Syntax notation)

ในตำราเล่มนี้รูปแบบของการเขียนคำสั่ง หรือ ส่วนต่าง ๆ ของคำสั่งอยู่ในสัญลักษณ์ที่เป็นระบบอย่างหนึ่งซึ่งเรากำหนดขึ้นมาใช้ สัญลักษณ์ที่ใช้นี้ไม่ใช่ส่วนหนึ่งของภาษา PL/I แต่เป็นเพียงรูปแบบมาตรฐานที่ใช้ในการบรรยาย รูปแบบหรือสร้างภาษาใดภาษาหนึ่งที่ใช้เขียนโปรแกรมเท่านั้น ซึ่งมีลักษณะเป็นข้อความสั้น ๆ กระชับรัดและได้ใจความชัดเจน แต่ไม่ได้บอกความหมายและโครงสร้างของ อีลิเมนต์ ในภาษานั้น นั่นคือเพียงแต่ระบุลำดับที่ของ อีลิเมนต์ ว่าอาจจะมีหรือต้องมีเครื่องหมายกำกับวรรคตอนที่ใช้ และส่วนไหนบ้าง ซึ่งอาจจะไม่มีก็ได้

กฎเกณฑ์ต่อไปนี้ อธิบายการใช้สัญลักษณ์นี้ สำหรับภาษาที่ใช้เขียนโปรแกรมภาษาหนึ่ง ส่วนตัวอย่างเป็นตัวอย่างในภาษา PL/I

1. A notation variable เป็นชื่อทั่ว ๆ ไปของชนิดของ อีลิเมนต์ ในภาษานั้น ตัวแปรชนิดนี้จะประกอบด้วย

a) ตัวพิมพ์เล็ก (lower-case letters), เลขฐานสิบ (decimal digits) และ hyphen แต่ต้องเริ่มต้นด้วยตัวอักษร

b) อาจจะเป็นอักษรตัวพิมพ์เล็กทั้งหมด หรือ combination ของตัวพิมพ์เล็ก กับตัวพิมพ์ใหญ่ (upper-case letters) ในกรณีหลังนี้ จะต้องมีส่วนหนึ่งที่อยู่ในตัวพิมพ์เล็กทั้งหมด และอีกส่วนหนึ่งอยู่ในตัวพิมพ์ใหญ่ทั้งหมด และทั้งสองส่วนนี้เขียนแยกกันด้วยเครื่องหมาย hyphen หนึ่งตัว

ตัวแปรทั้งหมดที่ใช้ defined ในหนังสือเล่มนี้ใช้สัญลักษณ์ดังกล่าวข้างต้น

ตัวอย่าง

digit ใช้สำหรับแทนตัวเลขตัวใดตัวหนึ่งในเลข 0 ถึงเลข 9

file-expression ใช้แทน a reference to a file

Do-statement ใช้แทนคำสั่ง DO ส่วนอักษรตัวพิมพ์ใหญ่ เป็นตัวบอกให้รู้ว่าคำนั้นเป็น keyword

2. A notation constant ใช้แทน literal occurrence สัญลักษณ์ตัวคงที่ในที่นี้อาจจะเป็นอักษรตัวพิมพ์ใหญ่ทั้งหมด หรือสัญลักษณ์พิเศษ ตัวใดตัวหนึ่ง (a special character) ก็ได้

ตัวอย่าง

```
DECLARE identifier FIXED;
```

ในที่นี้

DECLARE	เป็น	literal occurrence
identifier	เป็น	notation variable ซึ่งจะกล่าวถึงในที่อื่น ๆ อีก
FIXED	เป็น	literal occurrence
;	เป็น	literal occurrence

3. คำว่า "syntactic unit" ซึ่งจะใช้ในกฎเกณฑ์ข้อต่อไปนั้น หมายถึงข้อใดข้อหนึ่งข้างล่างนี้

- a) a single notation variable หรือ a single notation constant
- b) เป็นกลุ่มใดกลุ่มหนึ่งของ notation variables, notation constants, syntax-language symbols และ keywords ซึ่งอยู่ภายในเครื่องหมายวงเล็บ

4. เครื่องหมายวงเล็บปีกกา { } ใช้สำหรับบอกให้รู้ว่าเป็นกลุ่มของ อีลิเมนต์ ที่มากกว่าหนึ่งตัวขึ้นไป เพื่อทำให้เป็น a syntactic unit

ตัวอย่าง

```
identifier { FIXED | FLOAT }
```

เครื่องหมายวงเล็บปีกกาสองอันนี้หมายถึงให้เลือกเอาหนึ่งอย่าง ตัวอย่างข้างต้นจึงมีความหมายว่า ตัวแปร "identifier" ต้องตามด้วย literal occurrence ตัวใดตัวหนึ่งอาจจะเป็น FIXED หรือ FLOAT

5. เครื่องหมาย vertical stroke | หมายถึงให้เลือกเอาหนึ่งอย่าง

ตัวอย่าง

```
identifier { FIXED | FLOAT }
```

ตัวอย่างนี้มีความหมายอย่างเดียวกับตัวอย่างข้างต้น ทั้งข้อ 4 และข้อ 5 ในหนังสือเล่มนี้จะเลือกใช้วิธีใดก็ได้

6. เครื่องหมายวงเล็บใหญ่ [] หมายถึง options กล่าวคือ อะไรก็ตามที่อยู่ภายในเครื่องหมายวงเล็บนี้ อาจจะมีหรือไม่มีก็ได้ อยู่ในดุลยพินิจของโปรแกรมเมอร์

เครื่องหมายวงเล็บนี้ ยังเอาไปใช้ในวัตถุประสงค์ เพื่อให้เป็นเครื่องหมายใช้จำกัดขอบเขต a syntactic unit ได้

ตัวอย่าง

{ [lower-bound :] upper-bound } | *

ในที่นี้หมายความว่า ให้เลือกเอาระหว่าง literal asterisk (*) กับ ตัวแปร "upper-bound" แต่จะมีทั้งสองอย่างไม่ได้ ถ้ามี upper-bound อาจจะอยู่ข้างหลัง syntactic unit ที่ประกอบด้วยตัวแปร "lower-bound" และ literal colon (:) ก็ได้

7. จุดไข่ปลา 3 จุด ... หมายถึงจะตามด้วย syntactic unit ซ้ำกันได้ตั้งแต่หนึ่งชุดขึ้นไปตามลำดับ

ตัวอย่าง

[digit] ...

ในที่นี้ตัวแปร "digit" อาจจะมีหรือไม่มีก็ได้เพราะว่าอยู่ในเครื่องหมายวงเล็บใหญ่ แต่ถ้ามีอาจจะมีซ้ำเหมือนกันได้ตั้งแต่หนึ่งชุดขึ้นไป

8. ถ้าขีดเส้นใต้ อลิเมนต์ หมายถึง อลิเมนต์ตัวนั้นเป็น อลิเมนต์ตัวใดตัวหนึ่งของภาษานั้น ใช้ในกรณีที่มีการขัดแย้งระหว่าง อลิเมนต์ตัวนี้กับ อลิเมนต์ตัวใดตัวหนึ่งในภาษา syntax

ตัวอย่าง

operand { & | } operand

ในที่นี้ตัวแปร operands ทั้งสองตัว แยกกันด้วยเครื่องหมาย and (&) หรือเครื่องหมาย or (|) สำหรับ operator | ที่ขีดเส้นใต้ หมายถึงสัญลักษณ์ or ที่ใช้ในภาษา PL/I ไม่ใช่เครื่องหมาย or ในภาษา syntax