



#4 Function

What is Function?

Function คือ บล็อกของคำสั่งที่มีการทำงานตามสิ่งที่กำหนด

Build-in function

เป็นฟังก์ชันที่มีอยู่ในตัว Library ของภาษาเช่น

- `printf()`
- `getchar()`

User-defined function

เป็นฟังก์ชันที่เรากำหนดขึ้นมาเอง

Structure of Function

```
returnType functionName(parameterList) { // Function Header // Function Body }
```

Function Prototype

คือ Function Header ที่ใช้เพื่อบอก Compiler ไว้ก่อนว่า Function ดังกล่าว ว่าจะต้องการรับ Parameter ใด return เป็น type ใด

ต้องใช้เพราะภาษา C ใช้การ compile แบบ One-pass

```
returnType functionName(parameterList)
```

Recursive Function

คือ Function ที่เรียกใช้ Function ของตัวเอง

```
long factorial(int n) { if (n == 0) { // Base case: factorial of 0 is 1
return 1; } else { return n * factorial(n - 1); } }
```

Call `factorial(5)`

- `factorial(5)` returns `5 * factorial(4)`

Call `factorial(4)`

- `factorial(4)` returns `4 * factorial(3)`

Call `factorial(3)`

- `factorial(3)` returns `3 * factorial(2)`

Call `factorial(2)`

- `factorial(2)` returns `2 * factorial(1)`

Call `factorial(1)`

- `factorial(1)` returns `1 * factorial(0)`

Call `factorial(0)` (Base case)

- `factorial(0)` returns `1`

เมื่อถึง Base case จะเริ่ม return ค่าย้อนกลับขึ้นไปเรื่อยๆ :

- `factorial(1)` becomes `1 * 1 = 1`
- `factorial(2)` becomes `2 * 1 = 2`
- `factorial(3)` becomes `3 * 2 = 6`
- `factorial(4)` becomes `4 * 6 = 24`
- `factorial(5)` becomes `5 * 24 = 120`

** การทำ recursive function อาจทำให้เกิด overhead เนื่องจากการ call function ต่อกัน เป็นทอด ๆ แบบ recursive จะเกิดการสร้าง stack frame ของการเรียก function แต่ละครั้ง เพื่อใช้ในการเก็บ local variables