## การเขียนฟังก์ชัน

ปกติการเขียนโปรแกรมที่ผ่านมามีการเรียกใช้ฟังก์ชันบ้างแล้ว เช่น printf, scanf หรือ gets และยังมีอีกหลายฟังก์ชันสามารถเลือกใช้งานตามความเหมาะสมได้ แต่ในบางครั้งฟังก์ชันดังกล่าวอาจไม่ตอบโจทย์ความต้องการของเรา และเราสามารถเขียนฟังก์ชันขึ้นมาใหม่เพื่อให้ตรงกับความต้องการของเราได้ รูปแบบของฟังก์ชันแสดงดังนี้

```
ชนิดข้อมูลที่ส่งกลับ ชื่อฟังก์ชัน([argument 1], [argument 2],… , [argument n])
{
        [คำสั่ง;]
        [คำสั่ง;]
        …
        [คำสั่ง;]
        return ค่าที่ส่งกลับ
}
```

**ตัวอย่างที่ 1 ฟังก์ชันหาผลรวมของเลขสองจำนวน**

1. ให้ path อยู่ในตำแหน่ง /home/pi/c_codes
2. สร้างไฟล์ sum_exam.c ด้วยโปรแกรม nano
3. เขียนคำสั่งต่อไปนี้

```
#include <stdio.h>

int line();
int sum(int first, int second);

int main()
{
        int a, b;
        line();
        printf("enter first number:");
        scanf("%d",&a);
        printf("enter second number:");
        scanf("%d",&b);
        line();
        printf("summary is %d\n", sum(a,b));
        line();
}

int line()
{
        printf("==================\n");
}

int sum(int first, int second)
{
        int total;
        total = first + second;
        return total;
}
```

ประกาศฟังก์ชัน

จุดเรียกใช้ฟังก์ชัน

การทำงานของฟังก์ชัน line()

การทำงานของฟังก์ชัน sum()

จะเขียนฟังก์ชันต้องมีการประกาศก่อนที่จะขึ้นฟังก์ชัน main () ประกาศเพียงว่าฟังก์ชันชื่ออะไร มี argument เป็นข้อมูลประเภทอะไรบ้าง กี่ตัว เมื่อทำงานเสร็จแล้วจะส่งค่าเป็นข้อมูลชนิดใดกลับมา ส่วนคำสั่งต่างๆ ในแต่ละฟังก์ชันจึงมาเขียนลงในฟังก์ชันนั้นๆ ตามท้าย

การเรียกใช้ก็สอดคล้องรูปแบบของฟังก์ชันที่ประกาศไปในตอนต้นโปรแกรม ฟังก์ชัน line() ไม่มี argument เวลาเรียกใช้ ก็สามารถเรียกเพียงแค่ line() ได้เลย แต่ฟังก์ชัน sum() มี 2 arguments ทำให้เวลา

ที่เรียกใช้ฟังก์ชันดังกล่าว จะต้องระบุค่าหรือตัวแปรเพื่อส่งต่อเป็น argument ใช้ในการประมวลผลฟังก์ชันต่อไป ข้อดีอีกข้อของฟังก์ชัน คือการเรียกใช้ซ้ำๆ ได้ช่วยลดบรรทัด และเข้าใจโปรแกรมได้ง่ายขึ้น

4. บันทึก คอมไพล์ และรัน

```
pi@raspberrypi:~/c_codes $ gcc -o sum sum_exam.c
pi@raspberrypi:~/c_codes $ ./sum
==================
enter first number:5
enter second number:7
==================
summary is 12
==================
pi@raspberrypi:~/c_codes $
```



*** ทุกฟังก์ชันเมื่อทำงานเสร็จแล้วต้องมีการส่งค่ากลับ (return) และค่าที่ส่งกลับนั้นต้องมีชนิดข้อมูลที่ตรงกับที่ประกาศไว้หน้าชื่อฟังก์ชันด้วย แต่มีข้อสังเกตอย่างหนึ่งคือ ฟังก์ชันที่มี int นำหน้าชื่อฟังก์ชันถูกอนุโลมไม่ต้องใส่ return ก็ได้*