

ฟังก์ชันในภาษา C

คำสั่งที่เขียนรวมกันไว้ใช้งาน เรียกว่า (Function)

กลุ่มของคำสั่งที่เขียนขึ้นอยู่ภายในเครื่องหมาย { } เพื่อบอกขอบเขต และมีการตั้งชื่อให้กลุ่มคำสั่งเพื่อ เรียกออกมาใช้งาน

ข้อดี ลดการทำงานซ้ำซ้อน

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com

ตย 1 โปรแกรมสำหรับหาพื้นที่สี่เหลี่ยม

```
#include<stdio.h>
```

```
float width = 2.5, length = 3.45, area;
```

```
main()
```

```
{
```

```
    area = width*length;
```

```
    printf("Area of rectangle = %f\n",area);
```

```
}
```

ฟังก์ชันที่สร้างขึ้นเอง

โครงสร้างของฟังก์ชัน

```
type function-name (arg-1,arg-2,...,arg-n)
```

```
type arg-1
```

```
.....
```

```
type arg-n;
```

```
{
```

```
    variable-declaration;
```

```
    statement-1;
```

```
    statement-2;
```

```
    -----
```

```
    statement-n;
```

```
    return ();
```

```
}
```

type เป็นชนิดฟังก์ชัน เช่น **float** เป็นต้นถ้าไม่ระบุภาษา **C** กำหนดให้เป็น **int** อัตโนมัติ

Function-name ชื่อของฟังก์ชันที่ตั้งขึ้น

arg-1, arg-n; ระบุว่าในการเรียกใช้ฟังก์ชันต้องส่งตัวแปรใดเข้ามาให้กับฟังก์ชันบ้าง เนื่องจากฟังก์ชันต้องรับข้อมูลเข้ามาในฟังก์ชันด้วย

{ เครื่องหมายสำหรับเริ่มต้นฟังก์ชัน

variable-declaration; เป็นคำสั่งประกาศสร้างตัวแปรสำหรับใช้งานภายในฟังก์ชัน

statement-1; statement-2; statement-n; คำสั่งการทำงานของฟังก์ชัน

return (); คำสั่งสำหรับส่งผลลัพธ์การทำงาน ของฟังก์ชันกลับไปยังคำสั่งที่เรียกใช้ฟังก์ชัน โดย **value** ผลลัพธ์อยู่ในรูปตัวแปรหรือค่าคงที่ แต่ต้องเป็นชนิดแรก ถ้าไม่มีส่งค่ากลับก็ไม่ต้องระบุส่วน **return**

} เป็นเครื่องหมายสิ้นสุดของฟังก์ชัน

ทดลองเขียนฟังก์ชัน

```
#include <stdio.h>
main()
{
printf("First message\n");
    message();
printf("Second
message\n");
    message();
printf("Third messag\n");
    message();
}
message()
{
```

```
int i=0;
while (i<30)
{
printf("*");
    i++;
}
printf("\n\t");
printf("C Language");
printf("\n");
    i=0;
while (i<30)
{
printf("*");
    i++;
}
printf("\n");
}
```

by Aj.Sanchai Ramphueiphad ,Program in Air C

Department of Mecanical Engi

www.acr.rmuti.ac.th,Email:acr501@gmail.com

1. สร้างฟังก์ชันชื่อ **message()** ไม่ระบุชนิดของฟังก์ชัน ซึ่งไม่มีการส่งค่ากลับจากฟังก์ชัน และไม่ต้องส่งส่งอาร์กิวเมนต์ใด ๆ เข้ามาให้กับฟังก์ชัน สังเกตได้จากวงเล็บท้ายชื่อฟังก์ชันไม่กำหนดค่าตัวแปรใด ๆ เลย
2. สร้างตัวแปรชนิด **int** ชื่อ **i** พร้อมกำหนดค่าเริ่มต้นให้เป็นศูนย์ ซึ่งเป็นตัวแปรที่สร้างขึ้นเพื่อใช้งานในฟังก์ชัน **message()**
3. การทำงานของฟังก์ชันเริ่มจากใช้คำสั่งวนลูป **while** แสดงเครื่องหมาย * ออกทางหน้าจอจำนวน **30** ครั้ง
4. แสดงข้อความ **C Language** ออกทางหน้าจอ
5. ปิดท้ายข้อความด้วยการใช้คำสั่งวนรอบ **while** แสดงเครื่องหมาย * ออกทางหน้าจออีกครั้ง
6. ในส่วนของฟังก์ชัน **main()** ,มีการเรียกใช้ ฟังก์ชัน **message()** **3** ครั้ง

ฟังก์ชันชนิดต่าง ๆ

ฟังก์ชันที่ไม่มีการรับส่งค่า

หมายถึงไม่ต้องส่งอาร์กิวเมนต์ใด ๆ เข้ามาให้กับฟังก์ชัน และฟังก์ชันจะทำงานโดยไม่ส่งผลลัพธ์ใดใดออกมา

```
void funct_name(void);
```

หรือ

```
void funct_name();
```

ฟังก์ชันที่ไม่มีการรับส่งค่า

```
#include <stdio.h>
main()
{
    char ans;
    printf("do you want to
check movie program?
(y/n):");
    ans=getchar();
    if (ans=='y')
movie_program();
    else
    printf("Thank you\n");
}
```

```
movie_program()
{
    int i;
    printf("\n");
    for (i=0;i<4;i++)
    {
        printf("*");
    }
    printf("\n");
    printf("1.Battle\n");
    printf("2.shallow\n");
    printf("3.Captain\n");
    printf("4.Harry\n");
}
```


ฟังก์ชันที่มีการรับค่า

หมายถึงเวลาใช้งานต้องส่งอาร์กิวเมนต์ไปให้กับฟังก์ชันนั้น ซึ่งอาร์กิวเมนต์ต้องเป็นไปตามข้อกำหนดของฟังก์ชัน เช่น

ถ้าฟังก์ชันกำหนดให้เป็น อาร์กิวเมนต์ 3 ตัว เป็น `int, float, int` เวลาส่งข้อมูลก็ต้องอยู่ในรูปตัวแปร นิพจน์ หรือค่าคง แต่ต้องเรียง `int, float, int` เรียงตามลำดับ

การรับค่าโดยตรง

ฟังก์ชันที่มีการรับค่าเป็นตัวแปร ค่าคงที่ หรือนิพจน์ ให้ผลออกมาเป็นค่าคงที่เรียกว่า
เป็นการรับค่าโดยตรง

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com

ฟังก์ชันที่มีการรับค่าโดยตรง

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    int item;
```

```
    printf("Enter movie  
program (1-2) :");
```

```
    scanf("%d",&item);
```

```
    movie_time(item);
```

```
}
```

```
movie_time(agr1)
```

```
int agr1;
```

```
{
```

```
    int x;
```

```
    x=agr1;
```

```
    switch (x)
```

```
{
```

```
    case 1:
```

```
        printf("Battle Royale\n");
```

```
    printf("11.05 14.00 17.15 20.30\n");
```

```
        break;
```

```
    case 2:
```

```
        printf("shallow Ha\n");
```

```
    printf("10.45 13.30 16.15  
19.30\n");
```

```
        break;
```

```
}
```

```
    printf("Enjoy the movie :\n");
```

```
}
```

การรับค่าเป็นตำแหน่งของหน่วยความจำ

เราสามารถส่งค่าตำแหน่งในหน่วยความจำของตัวแปรเป็นอาร์กิวเมนต์ในการเรียกใช้งานฟังก์ชันได้ โดยใช้ตัวแปรพอยน์เตอร์ และเครื่องหมายตัวแปรพอยเตอร์เข้ามาช่วย(&,*) เข้ามาช่วย

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com

การรับค่าเป็นตำแหน่งของหน่วยความจำ

```
#include<stdio.h>
main()
{
    float base,height;
    float *pt;
    printf("Enter base of
triangle:");
    scanf("%f",&base);
    printf("Enter height of
triangle:");
    scanf("%f",&height);
    pt = &base;
    triangle(pt,&height);
```

```
}
triangle(float
*arg1,float *arg2)
{
    float x,y;
    x=*arg1;
    y=*arg2;
    printf("Area of
triangle = %f",x*y/2);
}
```

การรับค่าเป็นชุด

ในกรณีที่ฟังก์ชันต้องการอาร์กิวเมนต์จำนวนหลายค่าโดยเฉพาะถ้าเป็นอาร์กิวเมนต์ชนิดเดียวกัน เราสามารถกำหนดอาร์กิวเมนต์เป็นตัวแปรอาร์เรย์ได้

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com

การรับค่าเป็นชุด

```
#include<stdio.h>
main()
{
    float x[10];
    int i;
    for (i=0; i<10; i++)
    {
        printf("Enter
number :");
        scanf("%f",&x[i]);
    }
    average(x);
}
```

```
average(float num[10])
{
    float sum = 0;
    int j;
    for(j=0; j<10;j++)
    {
        sum+=num[j];
    }
    printf("Average
=%f\n",sum/10);
}
```

ฟังก์ชันที่มีการส่งค่าไปแต่ไม่มีการรับค่ากลับ

```
Void funct_name(type1, type2,...typeN);
```

Or

```
Void funct_name(type1 arg1, type2 arg2,...,typeN argN);
```

ตัวอย่างการประกาศตัวแปร

```
void multiply(int,int);
```

```
void add(float,int);
```

```
void a(int*,int);
```


ฟังก์ชันที่มีการส่งค่ากลับ

```
#include<stdio.h>
main()
{
```

```
average(float num[10])
{
    float sum = 0;
    int j;
    for(j=0; j<10;j++)
    {

        sum+=num[j];
    }
    printf("Average
    =%f\n",sum/10);
}
```

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com

ฟังก์ชันที่มีการส่งค่าไปและรับค่ากลับ

```
type funct_name(type1, type2,...typeN);
```

Or

```
type funct_name(type1 arg1, type2 arg2,...,typeN argN);
```

ตัวอย่างการประกาศตัวแปร

```
int multiply(int,int);
```

```
float add(float,int);
```

```
int a(int*,int);
```

ฟังก์ชันที่มีการส่งค่ากลับ

```
#include<stdio.h>
main()
{
```

```
average(float num[10])
{
    float sum = 0;
    int j;
    for(j=0; j<10;j++)
    {

        sum+=num[j];
    }
    printf("Average
    =%f\n",sum/10);
}
```

by Aj.Sanchai Ramphueiphad ,Program in Air Conditioning & Refrigeration
Department of Mecanical Engineering,
www.acr.rmuti.ac.th,Email:acr501@gmail.com