

บทที่ 6-1

ฟังก์ชันในภาษาซี

Function in C Language



ฟังก์ชันคืออะไร

ในการเขียนโปรแกรมคอมพิวเตอร์บางครั้งมีชุดคำสั่งบางชุดที่ถูกเรียกใช้งานบ่อย หากต้องเขียนชุดคำสั่งนั้นบ่อยครั้ง จะทำให้โปรแกรมมีขนาดใหญ่ จึงควรนำเอาชุดคำสั่งเหล่านั้นมารวมเป็นฟังก์ชัน

- ฟังก์ชัน คือ กลุ่มของคำสั่งที่สามารถถูกเรียกใช้ได้เพื่อทำงานเฉพาะอย่าง
- ประโยชน์ที่สำคัญของฟังก์ชันคือช่วยลดความซ้ำซ้อนของโปรแกรม อ่านเข้าใจ และแก้ไขได้ง่าย

ฟังก์ชันในภาษาซี

ฟังก์ชันในภาษาซี แบ่งออกได้เป็น 2 ประเภท คือ

- **ฟังก์ชันมาตรฐาน** (Standard Function) เป็นฟังก์ชันที่ Compiler มีมาให้ อยู่แล้ว ถูกจัดเป็นหมวดหมู่ในไลบรารีต่าง ๆ ในการใช้งานต้องเรียกใช้ Include Directives เพื่อเพิ่มไลบรารี ที่บรรจุฟังก์ชันนั้นอยู่ จึงสามารถใช้งานได้
- **ฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้น** (User-defined Function) เป็นฟังก์ชัน หรือโปรแกรมย่อยที่ผู้เขียนโปรแกรมสร้างขึ้นมาใช้งานเองตามวัตถุประสงค์ที่ต้องการ

main() ถือเป็น User-defined Function ชนิดหนึ่งที่เป็นต้องมี และเป็นจุดเริ่มต้นการทำงานของโปรแกรม



ฟังก์ชันมาตรฐาน (Standard Function)

Standard Function เป็นฟังก์ชันสำเร็จรูปที่มีมาให้พร้อมกับ Compiler โดยเก็บอยู่ในไฟล์ *.h ซึ่งมีรูปแบบดังนี้ `#include <file.h>`

Library Function ที่สำคัญ

- **math.h** ฟังก์ชันการคำนวณทางคณิตศาสตร์
- **string.h** ฟังก์ชันสำหรับข้อความ
- **ctype.h** ฟังก์ชันสำหรับอักขระ

ฟังก์ชันการคำนวณทางคณิตศาสตร์ – math.h

Function	Description
sin(x)	sine ของ x, x มีหน่วยเป็นเรเดียน
cos(x)	cosine ของ x, x มีหน่วยเป็นเรเดียน
tan(x)	tangent ของ x, x มีหน่วยเป็นเรเดียน
sqrt(x)	รากที่สองของ x, x >= 0
pow(x,y)	x ยกกำลัง y
log(x)	logarithm ฐาน e ของ x, x >= 0
log10(x)	logarithm ฐาน 10 ของ x, x >= 0
exp(x)	e ยกกำลัง x
abs(x)	ค่าสัมบูรณ์ของ x, x เป็นเลขจำนวนเต็ม
fabs(x)	ค่าสัมบูรณ์ของ x, x เป็นเลขจำนวนจริง

$$\text{Radians} = \left(\frac{\pi}{180^\circ} \right) \times \text{degrees}$$

$$\text{Degrees} = \left(\frac{180^\circ}{\pi} \right) \times \text{radians}$$

ตัวอย่างการใช้ฟังก์ชัน math.h

```
1 #include <stdio.h>
2 #include <math.h>
3 double x1 = 3.14159, x2 = 2.71828, y;
4 int main(void)
5 {
6     y = sin(x1);
7     printf("sin(pi)      = %7.4f\n", y);
8     printf("cos(pi/4)   = %7.4f\n", cos(x1/4));
9     printf("tan(pi/6)   = %7.4f\n", tan(x1/6));
10    printf("sqrt(2)     = %7.4f\n", sqrt(2));
11    printf("2.5^3       = %7.4f\n", pow(2.5,3));
12    printf("loge(e)     = %7.4f\n", log(x2));
13    printf("log(100)    = %7.4f\n", log10(100));
14    printf("e^3        = %7.4f\n", exp(3));
15    printf("abs(-7)     = %7.4f\n", abs(-7));
16    printf("fabs(-7.0)   = %7.4f\n", fabs(-7.0));
17    return(0);
18 }
```

```
sin(pi)      = 0.0000
cos(pi/4)    = 0.7071
tan(pi/6)    = 0.5774
sqrt(2)      = 1.4142
2.5^3        = 15.6250
loge(e)      = 1.0000
log(100)     = 2.0000
e^3          = 20.0855
abs(-7)      = 7.0000
fabs(-7.0)   = 7.0000
```

ฟังก์ชันสำหรับข้อความ – string.h

Function	Description
<code>strcpy(str1, str2)</code>	คัดลอกข้อความจาก str2 ไปเก็บที่ str1
<code>strcat(str1, str2)</code>	ต่อข้อความใน str1 ด้วย str2
<code>strncat(str1, str2, n)</code>	ต่อข้อความใน str1 ด้วย str2 จำนวน n อักขร
<code>strcmp(str1, str2)</code>	เปรียบเทียบตัวอักษรในข้อความ (case sensitive) ถ้า str1 = str2 จะได้ 0
<code>strlen(str)</code>	หาความยาวข้อความ



ฟังก์ชันสำหรับข้อความ – ctype.h

Function	Description
<code>tolower(ch)</code>	เปลี่ยนอักขระเป็นตัวพิมพ์เล็ก
<code>toupper(ch)</code>	เปลี่ยนอักขระเป็นตัวพิมพ์ใหญ่

ตัวอย่างการใช้ฟังก์ชัน strcpy และ strcat

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(void)
4 {
5     char str1[30] = "My Birthday ";
6     char str2[30] = "18 August 1991";
7     char str3[30];
8     strcpy(str3, str1);
9     strcat(str1, str2);
10    printf("str1 = %s\n", str1);
11    printf("str3 = %s\n", str3);
12    return(0);
13 }
```

ผลการทำงาน ?

```
str1 = My Birthday 18 August 1991
str3 = My Birthday
```

ตัวอย่างการใช้ฟังก์ชัน strcmp และ strlen

```
1
2 #include <string.h>
3 int main(void)
4 {
5     char buf1[] = "Nanometer", buf2[] = "Nanometer";
6     int ptr;
7     printf("buffer 1 : \"%s\"\n",buf1);
8     printf("buffer 2 : \"%s\"\n",buf2);
9     ptr = strcmp(buf2, buf1);
10    if(ptr == 0)
11        printf("buffer 2 is equal to buffer 1\n");
12    printf("Length of buffer 1 : %d\n", strlen(buf1));
13    return(0);
14 }
```

ผลการทำงาน ?

```
buffer 1 : "Nanometer"
buffer 2 : "Nanometer"
buffer 2 is equal to buffer 1
Length of buffer 1 : 9
```

ตัวอย่างการใช้ฟังก์ชัน tolower และ toupper

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4 int main()
5 {
6     int length, i;
7     char string[] = "This Is a String";
8     length = strlen(string);
9     for(i=0; i<length; i++)
10         string[i] = toupper(string[i]);
11     printf("%s\n",string);
12     for(i=0; i<length; i++)
13         string[i] = tolower(string[i]);
14     printf("%s\n",string);
15     return(0);
16 }
```

ผลการทำงาน ?

```
THIS IS A STRING
this is a string
```

ฟังก์ชันที่ใช้งานกับข้อมูลแบบสตริง

- ฟังก์ชันที่ทำงานกับข้อมูลแบบสตริงมีอยู่มากมายเช่น การตัดลอก การค้นหา การเปรียบเทียบ การทำให้เป็นตัวอักษรเล็กหรือใหญ่ ต่าง ๆ ซึ่งอยู่ใน `string.h` ก่อนนำฟังก์ชันมาใช้งานจะต้องมีการกำหนด include directive ก่อนเสมอ คือ

```
#include <string.h>
```

```
void main() {  
    Statement;  
  
    ...  
  
    Statement;  
}
```

ตัวอย่าง ฟังก์ชันที่ใช้งานกับข้อมูลแบบสตริง

- สำเนาสตริงจาก str2 ไปยัง str1

```
char *strcpy(char *str1, const char *str2)
```

- สำเนาสตริงจาก str2 ไปต่อท้ายสตริง str1

```
char *strcat(char *str1, const char *str2)
```

- นำค่าของ str1 และ str2 มาเปรียบเทียบกัน

```
int strcmp(const char *str1, const char *str2)
```

- ถ้า $str1 < str2$ จะคืนค่า < 0 ตัวอย่าง $str1[] = "AA"$, $str2[] = "AC"$
- ถ้า $str1 = str2$ จะคืนค่า $= 0$ ตัวอย่าง $str1[] = "AB"$, $str2[] = "AB"$
- ถ้า $str1 > str2$ จะคืนค่า > 0 ตัวอย่าง $str1[] = "AC"$, $str2[] = "AB"$

ตัวอย่าง ฟังก์ชันที่ใช้งานกับข้อมูลแบบสตริง

- ค้นหาตำแหน่งตัวอักษรที่อยู่ในสตริง str และได้ข้อมูลตั้งแต่ตำแหน่งที่พบไปทั้งหมด

```
char *strchr(const char * str,int ch)
```

- ค้นหาตำแหน่งของสตริง str2 ใน str1 และได้ข้อมูลตั้งแต่ตำแหน่งที่พบไปทั้งหมด

```
char *strstr(const char *str1, const char *str2)
```

ตัวอย่าง การใช้งานฟังก์ชัน

```
1 #include <stdio.h>
2 #include <string.h>
3 void main (void)
4 {   char str1[30]="Superman!! Help me!!";
5     char str2[30]="Help";
6     char str3[30];
7     printf("STRSTR : %s\n" , strstr(str1,str2));
8     printf("STRCHR : %s\n" , strchr(str1,'m'));
9     printf("STRCPY : %s\n" , strcpy(str3,str2));
10    printf("STRCMP : %d\n" , strcmp(str3,str2));
11    printf("STRCAT : %s\n" , strcat(str1,str2));
12    return(0);
13 }
```

ผลการทำงาน ?

STRSTR : Help me!!

STRCHR : man!! Help me!!

STRCPY : Help

STRCMP : 0

STRCAT : Superman!! Help me!!Help

จบบทที่ 6-1

ฟังก์ชันในภาษาซี

Function in C Language