

วิชา การเขียนภาษาโปรแกรม 1
รหัสวิชา ว 30289
เรื่อง ฟังก์ชันในภาษาซี

โดย
ครูเจตวัตร สวัสดิ์พาณิชย์

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยครูเจตวัตร สวัสดิ์พาณิชย์
 ▶ บรรณาธิปไตย ครูเจตวัตร สวัสดิ์พาณิชย์

ฟังก์ชันในภาษาซี

ฟังก์ชัน (Function) คือ การเขียนคำสั่งรวมกันไว้เป็นกลุ่มของคำสั่งเพื่อทำงานให้สำเร็จ โดยกลุ่มของคำสั่งที่เราเขียนจะอยู่ภายในเครื่องหมาย { } เพื่อบอกขอบเขต และมีการตั้งชื่อให้กับกลุ่มคำสั่งนั้น เพื่อความสะดวกในการเรียกใช้งาน

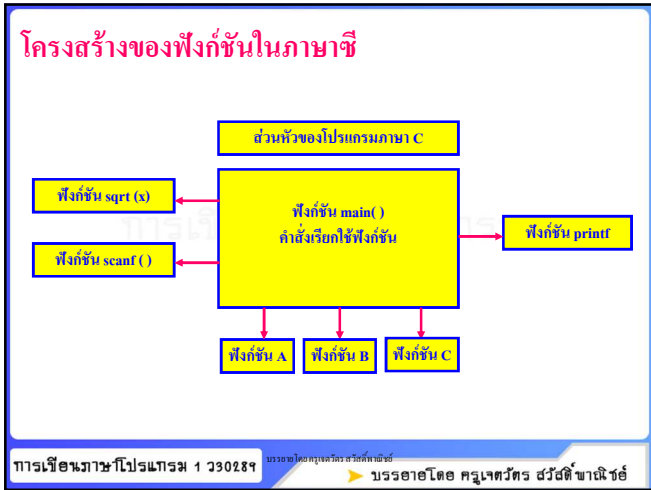
ซึ่งฟังก์ชันจะมีรูปแบบการส่งค่าข้อมูล หรือไม่มีการส่งค่าข้อมูลออกจากฟังก์ชัน ทั้งนี้ขึ้นอยู่กับหน้าที่และเป้าหมายการทำงานของฟังก์ชันนั้นๆ

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยครูเจตวัตร สวัสดิ์พาณิชย์
 ▶ บรรณาธิปไตย ครูเจตวัตร สวัสดิ์พาณิชย์

ฟังก์ชันในภาษาซี

ในการเขียนโปรแกรมภาษา C มีโครงสร้างประกอบด้วยฟังก์ชันการทำงาน โดยเริ่มต้นการทำงานที่ฟังก์ชัน main() เสมอ และ ฟังก์ชัน main() นั้นสามารถเรียกใช้ฟังก์ชันย่อยอื่นๆได้ ไม่ว่าจะเป็นฟังก์ชันที่ผู้อ่านสร้างขึ้นมาเอง (User-Defined Function) หรือฟังก์ชันมาตรฐานที่ภาษา C ได้สร้างมาให้แล้ว (Standard Library Function)

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยครูเจตวัตร สวัสดิ์พาณิชย์
 ▶ บรรณาธิปไตย ครูเจตวัตร สวัสดิ์พาณิชย์



รูปแบบของฟังก์ชันในภาษาซี

สำหรับฟังก์ชันในภาษานั้นถือได้ว่าเป็นโปรแกรมย่อย (Sub Program) โดยจะมีการทำงานเฉพาะอย่างของแต่ละฟังก์ชัน ซึ่งสามารถแบ่งฟังก์ชันออกเป็น 2 รูปแบบ ดังนี้

1. ฟังก์ชันมาตรฐาน (Standard Function) โดยเรียกใช้จาก Library
2. ฟังก์ชันที่ผู้ใช้งานกำหนดขึ้นเอง (User Defined Function)

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิกรรโดยครูเจตวัตร สวัสดิ์พิชาณิชย์
 > บรรณาธิกรรโดย ครูเจตวัตร สวัสดิ์พิชาณิชย์

1. ฟังก์ชันมาตรฐานหรือไลบรารีฟังก์ชัน

ฟังก์ชันมาตรฐาน เป็น ฟังก์ชันที่มีมาให้พร้อมด้วยตัวแปลภาษา C เพื่อใช้งานได้ทันที และใช้ในงานด้านต่างๆ โดยเน้นงานพื้นฐาน เช่น ฟังก์ชันคำนวณทางคณิตศาสตร์ ฟังก์ชันสำหรับจัดการข้อความ ฟังก์ชันเวลา เป็นต้น เพื่อให้ผู้เขียนภาษา C มีความสะดวกมากขึ้น

โดยไลบรารีฟังก์ชันภาษา C จะเก็บอยู่ในไฟล์นามสกุล .h หรือที่เรียกว่า header file ยกตัวอย่างเช่น ฟังก์ชันเกี่ยวกับการคำนวณจะเก็บอยู่ในไฟล์ชื่อ math.h หรือฟังก์ชันเกี่ยวกับการจัดการข้อความอยู่ในไฟล์ชื่อ string.h เป็นต้น

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิกรรโดยครูเจตวัตร สวัสดิ์พิชาณิชย์
 > บรรณาธิกรรโดย ครูเจตวัตร สวัสดิ์พิชาณิชย์

การเรียกใช้งานฟังก์ชันมาตรฐาน

การเรียกใช้งานฟังก์ชันมาตรฐานจะต้องเรียกใช้ **Library** ที่ส่วนหัวของโปรแกรม เพื่อให้ตัวแปลภาษา C เข้าใจว่าภายใน โปรแกรมของเรามีการเรียกฟังก์ชันที่อยู่ใน Library นั้นๆ

รูปแบบ

```
#include <Library>
```

ตัวอย่างเช่น

#include <math.h> <= เรียกใช้ไลบรารี math.h

#include <string.h> <= เรียกใช้ไลบรารี string.h


ฟังก์ชันมาตรฐาน (Standard Function)

ตัวอย่าง ฟังก์ชันต่างๆ ที่มีอยู่ใน Library ของภาษาซี

| Library | Function | การทำงานของฟังก์ชัน |
|----------|------------|--|
| stdio.h | printf () | ใช้ในการแสดงผลข้อมูล |
| | scanf () | ใช้ในการรับข้อมูล |
| conio.h | getchar () | ใช้ในการรับข้อมูล 1 อักขระ โดยการกด Enter |
| | getch () | ใช้ในการรับข้อมูล 1 อักขระ ไม่ปรากฏให้เห็นในการรับข้อมูล |
| string.h | strlen () | ใช้ในการนับความยาวของตัวอักขระที่รับเข้ามา |
| | strcpy () | ใช้ในการสำเนาข้อความจากข้อความหนึ่งไปยังอีกข้อความหนึ่ง |
| math.h | pow () | ใช้ในการหาค่าเลขยกกำลัง |
| | sqrt () | ใช้ในการหาค่ารากที่สอง (Root) ของเลขจำนวนเต็ม |

โปรแกรมการเรียกใช้ฟังก์ชันมาตรฐานในการหาค่าเลขยกกำลัง

```
#include <stdio.h>
#include <math.h>
int main()
{
    int Number,Exponent,Power;
    printf("Input Number = ");
    scanf("%d",&Number);
    printf("Input Exponent = ");
    scanf("%d",&Exponent);
    Power=pow(Number,Exponent); <=เรียกใช้ฟังก์ชัน pow() ที่อยู่ในไลบรารี
    printf("%d Power %d = %d",Number,Exponent,Power);
    getch();
    return 0;
}
```



บรรณาธิปไตยคุณจักร ศวัสดีพัณี

2. ฟังก์ชันที่ผู้ใช้งานกำหนดขึ้นเอง (User Defined Function)

ไลบรารีฟังก์ชันในภาษา C เป็นฟังก์ชันที่สร้างขึ้นเพื่อทำงานพื้นฐานทั่วไป ซึ่งบางครั้งอาจไม่มีไลบรารีฟังก์ชันที่ทำงานได้อย่างที่เราต้องการ ดังนั้น ภาษา C จึงให้เราสร้างฟังก์ชันขึ้นมาใช้งานได้เอง

การสร้างฟังก์ชันขึ้นมาใช้งานโปรแกรม เป็นการแบ่งการทำงานเป็นส่วนเล็กๆ ทำให้ง่ายและสะดวกต่อการตรวจสอบ และแก้ไขโปรแกรม โดยเฉพาะโปรแกรมที่มีขนาดใหญ่

คำสั่งการสร้างฟังก์ชัน

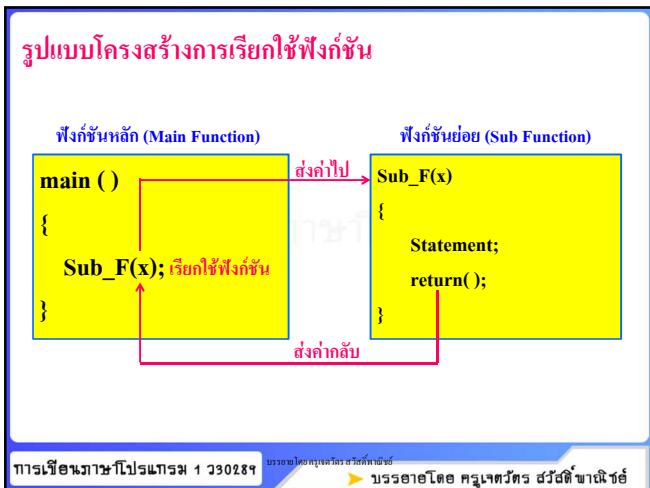
การสร้างฟังก์ชันมาตรฐานและฟังก์ชันที่ผู้ใช้สร้างขึ้นมีรูปแบบดังนี้

รูปแบบ

```
ประเภทของข้อมูล ชื่อฟังก์ชัน (พารามิเตอร์)
{
    คำสั่ง;
    [ return ] ;
}
```

อธิบายคำสั่งการสร้างฟังก์ชัน

| | |
|-----------------|--|
| ประเภทของข้อมูล | ประเภทของข้อมูลที่ส่งกลับ ส่วนนี้จะไม่มีหรือไม่มีก็ได้ ขึ้นอยู่กับการทำงาน ถ้าหากฟังก์ชันไม่มีการส่งกลับ ส่วนนี้จะประกาศเป็น void |
| ชื่อฟังก์ชัน | เป็นชื่อฟังก์ชันที่สร้างขึ้น สามารถเรียกใช้งานผ่านชื่อนี้ได้ โดยการตั้งชื่อจะต้องเป็นไปตามกฎการตั้งชื่อของภาษาซี |
| พารามิเตอร์ | เป็นตัวแปรหรือพารามิเตอร์ที่รับค่าผ่านเข้ามาในฟังก์ชันส่วนนี้จะไม่มีหรือไม่มีก็ได้ ขึ้นอยู่กับการทำงานของฟังก์ชัน |
| return | ใช้ในการส่งค่ากลับ ส่วนนี้จะไม่มีหรือไม่มีก็ได้ ขึ้นอยู่กับการทำงาน สำหรับในกรณีที่มีฟังก์ชันมีการส่งค่ากลับจะต้องเขียนค่าที่ส่งกลับตามหลัง return |



ตัวอย่างการเรียกใช้ฟังก์ชัน

```

#include<stdio.h>
int main( )
{
    int i;
    for(i = 1; i <= 10; i++)
    {
        printf("%d ", i * i);
    }
    printf("\n");
    return 0;
}
    
```

→

```

#include<stdio.h>
int square(int);
int main( )
{
    int i;
    for(i = 1; i <= 10; i++)
    {
        printf("%d ", square(i));
    }
    printf("\n");
    getch();
    return 0;
}
int square(int y)
{
    return(y*y);
}
    
```

การเขียนภาษาโปรแกรม 1 230289 บรรณานุกรมโดยครูเจตวัตร สวัสดิ์พิชาณิษฐ์
▶ บรรณานุกรมโดย ครูเจตวัตร สวัสดิ์พิชาณิษฐ์

โปรโตไทป์ (Prototypes)

การเขียนโปรแกรมที่มีการสร้างฟังก์ชันจะมีการเขียนชื่อฟังก์ชันรวมไว้ที่ส่วนหัวของโปรแกรมเพื่อบอกให้ตัวแปลภาษารู้จักฟังก์ชันก่อน และสามารถเรียกใช้ฟังก์ชันได้ แม้ตัวฟังก์ชันจะอยู่ส่วนท้ายของโปรแกรม เราเรียกโปรโตไทป์ (Prototypes) โดยจะเขียนว่าฟังก์ชันนี้ชื่ออะไร มีพารามิเตอร์บ้าง และฟังก์ชันมีการคืนค่ากลับมาอย่างไร ซึ่งจะต้องประกาศก่อนฟังก์ชัน main () โดยรูปแบบการประกาศโปรโตไทป์ (Prototypes) มีดังนี้

รูปแบบ
ประเภทของข้อมูล ชื่อฟังก์ชัน (ประเภท พารามิเตอร์มีเตอร์1, ... , พารามิเตอร์ N)

ตัวอย่างเช่น

```

int Sum (int x, int y);    <= ประกาศโปรโตไทป์
int (int x, int y);      <= ประกาศโปรโตไทป์
    
```

การเขียนภาษาโปรแกรม 1 230289 บรรณานุกรมโดยครูเจตวัตร สวัสดิ์พิชาณิษฐ์
▶ บรรณานุกรมโดย ครูเจตวัตร สวัสดิ์พิชาณิษฐ์

```

โปรแกรมการเรียกประกาศโปรโตไทป์ (Prototypes)
#include <stdio.h>

int maximum(int x, int y, int z);    //การประกาศโปรโตไทป์ ซึ่งอยู่ก่อนฟังก์ชัน main ()

int main()
{
    int a, b, c;
    printf("Enter three integers: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("Maximum is: %d\n", maximum(a, b, c));
    getch ();
    return 0;
}

/* Function maximum definition */
int maximum(int x, int y, int z)
{
    int max = x;
    if(y > max)
        max = y;
    if(z > max)
        max = z;
    return max;
}
    
```

ข้อสังเกตของการประกาศโปรโตไทป์ (Prototypes)

```

#include <stdio.h>

int maximum(int x, int y, int z)
{
    int max = x;
    if(y > max)
        max = y;
    if(z > max)
        max = z;
    return max;
}

int main()
{
    int a, b, c;
    printf("Enter three integers: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("Maximum is: %d\n", maximum(a, b, c));
    return 0;
}
    
```

โปรแกรมได้สร้างฟังก์ชันไว้ก่อนการเรียกใช้ จึงไม่จำเป็นต้องมีต้นแบบฟังก์ชันแต่ข้อเสียคือฟังก์ชัน main () จะถูกเลื่อนไปท้ายโปรแกรม

เนื่องจากฟังก์ชัน main () เป็นฟังก์ชันหลัก ในการอ่านโปรแกรมจึงควรพบฟังก์ชันนี้ก่อน นอกจากนี้การเขียนในลักษณะนี้ยังไม่เหมาะสมในกรณีที่มีโปรแกรมหมีขนาดใหญ่มากและซับซ้อนด้วย

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยศุภจวิตร สวัสดิ์พิฒธิชัย

▶ บรรณาธิปไตยศุภจวิตร สวัสดิ์พิฒธิชัย

ตัวแปรภายใน (Local Variable) และตัวแปรภายนอก (Global Variable)

ตัวแปรภายใน (Local Variable) คือ ตัวแปรที่ถูกสร้างขึ้นภายในฟังก์ชัน สามารถเรียกใช้งานได้เฉพาะภายในฟังก์ชันที่สร้างขึ้น และจะถูกทำลายลงเมื่อเสร็จสิ้นการทำงานของฟังก์ชันนั้นๆ

ตัวแปรภายนอก (Global Variable) คือ ตัวแปรที่ถูกสร้างขึ้นภายนอกฟังก์ชัน สามารถใช้งานได้ในทุกฟังก์ชัน หรือทั้ง โปรแกรม (ยกเว้นฟังก์ชันที่มีตัวแปรภายในชื่อเดียวกับตัวแปรภายนอก ซึ่งมีความเป็นไปได้) และจะคงอยู่ตลอดการทำงาน of โปรแกรม

การเขียนภาษาโปรแกรม 1 230289 18

▶ บรรณาธิปไตยศุภจวิตร สวัสดิ์พิฒธิชัย

ตัวอย่างการใช้แปรภายใน และตัวแปรภายนอก

```
#include<stdio.h>

int ans = 0;

int inc_one(int); /* function prototype */

void main()
{
    ans = inc_one(3);
    printf("Answer is %d \n", ans);
}

/* function definition: return x+1 */
int inc_one(int x)
{
    int ans, b;

    b = 2;
    ans = x + b;
    return ans;
}
```

ans
เป็นตัวแปรภายนอกเพราะประกาศไว้ข้างนอก ซึ่งจะเห็นว่าฟังก์ชัน main () สามารถเรียกใช้ได้โดยไม่ต้องประกาศตัวแปร ans อื่น

x, ans, b
เป็นตัวแปรภายในฟังก์ชัน inc_one ที่รู้จักแต่ในฟังก์ชันนี้เท่านั้น และตัวแปร ans ที่ประกาศในฟังก์ชันนี้จะเป็นคนละตัวกับตัวแปร ans ที่ประกาศข้างนอกซึ่งจะไม่เกี่ยวข้องกัน

บรรณาธิ โดยครูเจตวัตร สวัสดิ์พิทักษ์

ตัวอย่างการใช้แปรภายใน และตัวแปรภายนอก

```
#include<stdio.h>

void my_func(); //prototype of my_func

void main()
{
    int x=3;
    printf("Main: Before call function x=%d \n", x);
    my_func(); //call my_func
    printf("Main: After call function x=%d \n", x);
}

void my_func()
{
    int x;
    x=2;
    printf("My_func: x=%d \n", x);
}
```

ตัวอย่างนี้แสดงถึงตัวแปรภายในฟังก์ชัน main ชื่อ x และตัวแปรภายในฟังก์ชัน my_func ชื่อ x ซึ่งแม้จะมีชื่อเดียวกันแต่ค่าที่เก็บก็ไม่เกี่ยวข้องกัน

ดังนั้นเมื่อฟังก์ชัน my_func ถูกเรียกใช้ซึ่งค่าในตัวแปร x ในฟังก์ชัน my_func ถูกกำหนดเป็น 2 ค่า x ในฟังก์ชัน main ก็ยังมีค่าเป็น 3 เหมือนเดิม ดังนั้นการทำงานของโปรแกรม

```
Main: Before call function x=3
My_func: x=2
Main: After call function x=3
```

บรรณาธิ โดยครูเจตวัตร สวัสดิ์พิทักษ์

ตัวอย่างการใช้แปรภายใน และตัวแปรภายนอก

```
#include<stdio.h>

void my_func(); //prototype of my_func
int x;

void main()
{
    printf("Main: Before call function x=%d \n", x);
    my_func(); //call my_func
    printf("Main: After call function x=%d \n", x);
}

void my_func()
{
    x=2;
    printf("My_func: x=%d \n", x);
}
```

ตัวอย่างนี้แสดงถึงตัวแปรภายนอกชื่อ x ซึ่งเป็นที่รู้จักในทุก ๆ ฟังก์ชันของโปรแกรม (ฟังก์ชันไม่ต้องประกาศก็สามารถเรียกตัวแปร x ใช้ได้)

ดังนั้นเมื่อฟังก์ชัน my_func ถูกเรียกใช้ซึ่งค่าในตัวแปร x ในฟังก์ชัน my_func ถูกเปลี่ยนค่าให้เป็น 2 ดังนั้นค่า x ที่แสดงในฟังก์ชัน main จึงเปลี่ยนเป็น 2 เหมือนกัน ดังนั้นการทำงานของโปรแกรม

```
Main: Before call function x=3
My_func: x=2
Main: After call function x=2
```

บรรณาธิ โดยครูเจตวัตร สวัสดิ์พิทักษ์

ประเภทของฟังก์ชันในการรับ/ส่งค่า

สำหรับภาษา C ใช้การรับ/ส่งค่าของฟังก์ชันเป็นเกณฑ์ สามารถแบ่งได้ 3 ประเภทด้วยกัน ดังนี้

1. ฟังก์ชันที่ไม่มีการรับ/ส่งค่า
2. ฟังก์ชันที่มีการรับค่าเข้าไปในฟังก์ชัน
3. ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

1. ฟังก์ชันที่ไม่มีการรับ/ส่งค่า

ฟังก์ชันประเภทนี้จะไม่มีการส่งค่าผ่านพารามิเตอร์ (parameter) ไปกลับจะมีคำว่า void นำหน้าชื่อฟังก์ชัน การเรียกใช้ฟังก์ชันจะต้องมีการประกาศฟังก์ชันก่อน

| ฟังก์ชันหลัก (Main Function) | ฟังก์ชันย่อย (Sub Function) |
|--|--|
| <pre>main () { Sub_F(); เรียกใช้ฟังก์ชัน }</pre> | <pre>void Sub_F() { Statement; }</pre> |

รูปแบบการประกาศฟังก์ชันที่ไม่มีการรับ/ส่งค่า

รูปแบบการประกาศฟังก์ชัน

void ชื่อฟังก์ชัน (); หรือ void ชื่อฟังก์ชัน (void);

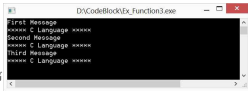
รูปแบบฟังก์ชัน

```
void ชื่อฟังก์ชัน ()
{
    statement;
}
```

โปรแกรมการเรียกใช้ฟังก์ชันที่ไม่มีการรับ/ส่งค่า

```
#include<stdio.h>
void message();
int main()
{
    printf("First Message\n");
    message();
    printf("Second Message\n");
    message();
    printf("Third Message\n");
    message();
    getch();
    return 0;
}

void message()
{
    printf("***** C Language *****\n");
}
```



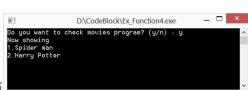
บรรณาช โดยคุณเจตวัตร สวัสดิ์พิพัฒน์ชัย

ตัวอย่าง เป็นฟังก์ชันแสดงรายชื่อภาพยนตร์ที่กำลังเข้าฉาย โดยให้ผู้ใช้เลือกว่าใช่หรือไม่

```
#include<stdio.h>
void movie_program();

void main()
{
    char ans;
    printf("Do you want to check movies program? (y/n) : ");
    scanf("%c",&ans);
    if(ans=='y')
    {
        movie_program();
    }
    else
    {
        printf("Thank you");
    }
    getch();
}

void movie_program()
{
    printf("Now showing \n");
    printf("1.Spider man \n");
    printf("2.Harry Potter \n");
}
```



บรรณาช โดยคุณเจตวัตร สวัสดิ์พิพัฒน์ชัย

อธิบายเพิ่มเติม

void หน้าชื่อฟังก์ชัน เป็นการกำหนดว่า ฟังก์ชันนั้น ไม่มีการคืนค่าใดๆ กลับไปยังฟังก์ชันที่เรียกใช้งาน

void หลังชื่อฟังก์ชัน เป็นการกำหนดว่า ฟังก์ชันนั้น ไม่มีการรับพารามิเตอร์ใดๆ จากฟังก์ชันที่เรียกใช้งาน

ตัวอย่างการใช้งาน

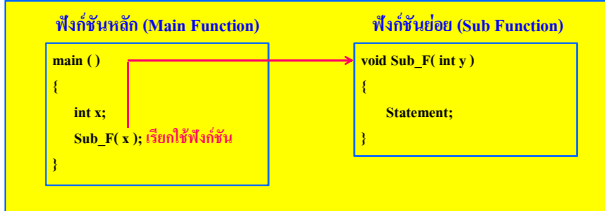
```
void MyName (void )
{
    printf ("Programming Language");
}
```

การเขียนภาษาโปรแกรม 1 230289 บรรณาช โดยคุณเจตวัตร สวัสดิ์พิพัฒน์ชัย

▶ บรรณาชโดย คุณเจตวัตร สวัสดิ์พิพัฒน์ชัย

2. ฟังก์ชันที่มีการรับค่าเข้าไปในฟังก์ชัน

ฟังก์ชันประเภทนี้จะเป็นการส่งค่าผ่านพารามิเตอร์ (parameter) ให้แก่ฟังก์ชัน โดยไม่มีการส่งค่ากลับ จะมีคำว่า **void** นำหน้าชื่อฟังก์ชัน โดยในวงเล็บจะระบุชนิดตัวแปรและชื่อตัวแปร ถ้ามีหลายตัวจะใช้ , คั่น



การเขียนภาษาโปรแกรม 1 230289 บรรณาธิโดยคุณเจตวิตร สวัสดิ์พิฒธิชัย > บรรณาธิโดย คุณเจตวิตร สวัสดิ์พิฒธิชัย

รูปแบบการประกาศฟังก์ชันที่มีการรับค่าเข้าไปในฟังก์ชัน

รูปแบบการประกาศฟังก์ชัน

void ชื่อฟังก์ชัน (ชนิดตัวแปร ชื่อตัวแปร, ...);

รูปแบบฟังก์ชัน

```

void ชื่อฟังก์ชัน ( ชนิดตัวแปร ชื่อตัวแปร, ... )
{
    statement;
}
    
```

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิโดยคุณเจตวิตร สวัสดิ์พิฒธิชัย > บรรณาธิโดย คุณเจตวิตร สวัสดิ์พิฒธิชัย

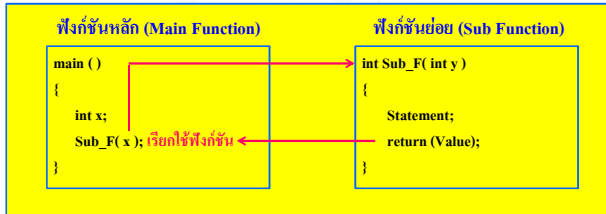
```

โปรแกรมการเรียกใช้ฟังก์ชันที่มีการรับค่าเข้าไปในฟังก์ชัน
#include<stdio.h>
void change_number(int num);

int main()
{
    int number=1000;
    printf("Number before call function = %d \n",number);
    change_number(number);
    getch();
    return 0;
}
void change_number(int num)
{
    printf("Number = %d \n",num);
    num=num*2;
    printf("Values of Number after change = %d \n",num);
}
    
```

3. ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

ฟังก์ชันประเภทนี้จะเป็นการส่งค่าผ่านพารามิเตอร์ (parameter) ไปกลับระหว่างฟังก์ชัน จะมีชนิดตัวแปรนำหน้าชื่อฟังก์ชัน แล้วตามด้วยชนิดตัวแปรและชื่อตัวแปร ถ้ามีหลายตัวจะใช้ , คั่น



การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยคุณจักร สวัสดิ์พาณิชย์ > บรรณาธิปไตย คุณเจตวัตร สวัสดิ์พาณิชย์

รูปแบบการประกาศฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

รูปแบบการประกาศฟังก์ชัน

ชนิดตัวแปร ชื่อฟังก์ชัน (ชนิดตัวแปร ชื่อตัวแปร, ...);

รูปแบบฟังก์ชัน

ชนิดตัวแปร ชื่อฟังก์ชัน (ชนิดตัวแปร ชื่อตัวแปร, ...)

```

{
    statement;
    return (Value);
}
    
```

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยคุณจักร สวัสดิ์พาณิชย์ > บรรณาธิปไตย คุณเจตวัตร สวัสดิ์พาณิชย์

ข้อสังเกต



ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน เป็นฟังก์ชันที่ต้องส่งผลลัพธ์การใช้งานออกไปนอกฟังก์ชันด้วย ส่วนใหญ่จะเป็นฟังก์ชันที่ต้องคำนวณ โดยจะต้องสร้างตัวแปรเพื่อรองรับค่าผลลัพธ์จากการเรียกใช้ฟังก์ชันด้วย

การสร้างฟังก์ชันแบบส่งค่ากลับออกจากฟังก์ชัน จะต้องฟังก์ชันไว้ก่อนหน้า main () หรือต้องมีการประกาศโปรโตไทป์ เพื่อให้ตัวแปลภาษา C เรียกฟังก์ชันเหล่านั้นทำงานได้อย่างถูกต้อง

การเขียนภาษาโปรแกรม 1 230289 บรรณาธิปไตยคุณจักร สวัสดิ์พาณิชย์ > บรรณาธิปไตย คุณเจตวัตร สวัสดิ์พาณิชย์

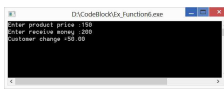
โปรแกรมการเรียกใช้ฟังก์ชันที่มีการส่งค่ากลับออกจากฟังก์ชัน

```
#include<stdio.h>
float cal_change(float price,float money);

int main()
{
    float price,money,change;
    printf("Enter product price :");
    scanf("%f",&price);
    printf("Enter receive money :");
    scanf("%f",&money);
    change = cal_change(price,money);
    printf("Customer change =%.2f \n",change);
    getch();
    return 0;
}

float cal_change(float price,float money)
{
    float z;
    z = money-price;
    return(z);
}
```

โปรแกรมคำนวณเงินทอนลูกค้า โดยเมื่อลูกค้าซื้อสินค้าโปรแกรมจะทำการรับค่าราคาสินค้า และจำนวนเงินลูกค้า และสร้างฟังก์ชันคิดเงินทอนเงิน



บรรณาช โดยคุณเจตวัตร สวัสดิ์พิชาณิชย์

อธิบายเพิ่มเติม

จากฟังก์ชันตัวอย่าง **cal_change ()** หน้าฟังก์ชันจาก **void** จะเปลี่ยนเป็นชนิดข้อมูลแทน คือฟังก์ชัน **cal_change ()** มีข้อมูลชนิด float และมีการ **return** ค่าตัวแปร โดยจะตัดการแสดงผล (**printf**) ออกไป หมายความว่าฟังก์ชัน **cal_change ()** จะทำการประมวลผลและส่งค่าผลลัพธ์กลับไปฟังก์ชัน **main ()**

จะเห็นว่าเวลาเรียกใช้ฟังก์ชันจะมีการส่งค่าไปตามปกติ และในขณะเดียวกัน เราก็จะให้มีการรับค่าที่คืนจากฟังก์ชันด้วย (**return**) โดยค่าที่คืนกลับมาคือค่าผลลัพธ์ที่ฟังก์ชันได้ประมวลผลนั่นเอง

การเขียนภาษาโปรแกรม 1 230289 บรรณาช โดยคุณเจตวัตร สวัสดิ์พิชาณิชย์
▶ บรรณาชโดย คุณเจตวัตร สวัสดิ์พิชาณิชย์

โจทย์ปัญหาแนวคิด ชุดที่ 8

- จงอธิบายความหมายของฟังก์ชันในภาษาซี มาพอสังเขป ?
- ฟังก์ชันในภาษาซีมีกี่รูปแบบ อะไรบ้าง ?
- จงเขียนโปรแกรมของการหารากที่สอง (Root) โดยใช้ฟังก์ชันมาตรฐาน ?
- จงเขียนโปรแกรมให้พิมพ์ชื่อของนักเรียน 10 ครั้งทางจอภาพ โดยรูปแบบฟังก์ชันแบบกำหนดเอง ?

การเขียนภาษาโปรแกรม 1 230289 บรรณาช โดยคุณเจตวัตร สวัสดิ์พิชาณิชย์
▶ บรรณาชโดย คุณเจตวัตร สวัสดิ์พิชาณิชย์

โจทย์ปัญหาแนวคิด ชุดที่ 8

5. เขียนโปรแกรมรับค่าสองค่าจากฟังก์ชัน `main()` และให้สร้างเมนูและฟังก์ชันในการคำนวณต่อไปนี้

ฟังก์ชัน `Add ()` ในการบวกค่า

ฟังก์ชัน `Difference ()` ในการลบค่า

ฟังก์ชัน `Multiply ()` ในการคูณ

ฟังก์ชัน `Division ()` ในการหาร

โดยให้ส่งค่าผลลัพธ์การคำนวณกลับมาแสดงผล

ที่ฟังก์ชัน `main ()` ตัวอย่างหน้าจอผลลัพธ์

```
Please Enter Number 1 : 25
Please Enter Number 2 : 20
Menu
1.Addition
2.Difference
3.Multiply
4.Division
Please select Menu :
```

การเขียนภาษาโปรแกรม 1 230289

บรรณาธิกรรโดยครูเจตวัตร สวัสดิ์พิพนิชย์

➤ บรรณาธิกรรโดย ครูเจตวัตร สวัสดิ์พิพนิชย์
