



การสร้างฟังก์ชันในภาษาซี



วัตถุประสงค์



- อธิบายความหมายและประเภทของฟังก์ชันได้
- สามารถสร้างฟังก์ชันเพื่อใช้งานเองได้
- บอกความแตกต่างของฟังก์ชันลักษณะต่าง ๆ ได้
- สามารถใช้ฟังก์ชันมาตรฐานต่าง ๆ ในภาษาซีได้
- รู้จักข้อแตกต่างระหว่างตัวแปรแบบโลคอลและตัวแปรแบบโกลบอล



ฟังก์ชัน คือ อะไร



- เป็นกลุ่มคำสั่งที่สร้างขึ้นมาเพื่อให้ทำงานอย่างใดอย่างหนึ่ง
- สามารถเรียกใช้งานได้หลายครั้ง ทำให้ง่ายต่อการทำงาน
- สามารถปรับปรุงแก้ไขโปรแกรมได้ง่าย



ประเภทของฟังก์ชัน



- ฟังก์ชันมาตรฐาน เป็นฟังก์ชันที่มีอยู่แล้วและเก็บอยู่ในไลบรารี เช่น ฟังก์ชันรับค่าและฟังก์ชันแสดงผล
- ฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้น เพื่อทำงานบางอย่างใดอย่างหนึ่ง ทำได้ง่ายและลดความซับซ้อนลง



การสร้างฟังก์ชัน



ประเภทของข้อมูล ชื่อฟังก์ชัน (พารามิเตอร์)

```
{  
    statement;  
    [return];  
}
```

- ประเภทข้อมูล คือ ประเภทข้อมูลที่จะส่งกลับ
- ชื่อฟังก์ชัน คือ ชื่อของฟังก์ชันตามกฎของภาษาซี
- พารามิเตอร์ คือ ตัวแปรที่รับค่าผ่านเข้ามาในฟังก์ชัน
- Return ใช้ในการส่งค่ากลับ มีหรือไม่มีก็ได้



ตัวอย่างฟังก์ชัน



```
void Show_A
{
    printf("AAA\n");
    printf("BBB\n");
}
```

```
void Show_A
{
    int i;
    for(i=1; i<20; i++)
        printf("A");
}
```



ตัวอย่างฟังก์ชันบวกเลข



```
int add(int a, int b)
{
    int c;
    c = a + b;
    return c;
}
main()
{
    printf("3 + 5 = %d\n",add(3.5));
    getch();
    return 0;
}
```



โปรโตไทป์



- จะต้องมีการเขียนชื่อฟังก์ชันรวมไว้ที่ส่วนหัวของโปรแกรม
- เพื่อบอกให้ตัวแปลภาษารู้จักฟังก์ชันก่อน
- และสามารถเรียกใช้งานฟังก์ชันที่อยู่ส่วนท้ายของโปรแกรมได้

ประเภทของข้อมูล ชื่อฟังก์ชัน(ประเภท พารามิเตอร์ 1, ..., พารามิเตอร์ N);



ตัวอย่างการกำหนดโปรโตไทป์



```
void p_aaa();  
void p_bbb();  
main()  
{  
    p_aaa();  
    p_bbb();  
    getch();  
    return 0;  
}
```

```
void p_aaa()  
{  
    int j;  
    for(j=1;j<20;j++)  
        printf("A\n");  
}  
void p_bbb()  
{  
    int j;  
    for(j=1;j<20;j++)  
        printf("B\n");  
}
```



ตัวอย่างโปรแกรม Show text



```
void Show_text(int num);  
main()  
{  
    show_text(5);  
    show_text(15);  
    getch();  
    return 0;  
}
```

```
void Show_text(int num);  
{  
    int i;  
    for(i=1; i<num; i++)  
        printf("%d\n", i);  
}
```



ตัวแปรทั่วไปและตัวแปรเฉพาะที่



• ตัวแปรทั่วไป

- ประกาศไว้นอกฟังก์ชัน
- ทุกส่วนของโปรแกรมสามารถเรียกใช้งานได้
- ค่าของมันจะเปลี่ยนแปลงไปตามที่ฟังก์ชันนั้นเรียกใช้งาน

• ตัวแปรเฉพาะที่

- ตัวแปรที่ใช้งานในฟังก์ชันเท่านั้น
- ฟังก์ชันอื่น ๆ ไม่สามารถเรียกใช้งานได้
- เมื่อออกจากฟังก์ชันนี้ข้อมูลจะหายไป



ตัวอย่าง ตัวแปรแบบทั่วไป



```
int a;  
void ex()  
{  
    a=5;  
    printf(“%d\n”,a);  
}
```

3
5
5

```
main()  
{  
    a=3;  
    printf(“%d\n”,a);  
    ex();  
    printf(“%d\n”,a);  
    return 0;  
}
```



ตัวอย่าง ตัวแปรแบบทเฉพาะที่



```
int a;  
void ex()  
{  
    int a;  
    a=5;  
    printf(“%d\n”,a);  
}
```

```
main()  
{  
    a=3;  
    printf(“%d\n”,a);  
    ex();  
    printf(“%d\n”,a);  
    return 0;  
}
```

3
5
3



ฟังก์ชันประเภทต่าง ๆ



- ฟังก์ชันแบบไม่มีการส่งค่า
 - เช่น printf, scanf, if อื่น ๆ
- ฟังก์ชันแบบส่งค่าผ่านทางเดียว

```
int add(int a, int b)  
{ return a+b; }
```

- ฟังก์ชันแบบส่งค่าสองทาง



ตัวอย่างฟังก์ชันแบบส่งค่าสองทาง



```
int sqr(int x);  
main()  
{  
int a;  
printf("Input Number:");  
scanf("%d",&a);  
printf("%d ^ 2 =%d\n",a,sqr(a));  
return 0;  
}
```

```
int sqr(int x)  
{  
    x=x*x;  
    return x;  
}
```





Thank You!

