

ใบความรู้เรื่อง ฟังก์ชัน (Function)

โรงเรียนศรีอยุธยา ในพระอุปถัมภ์ฯ วิชา เทคโนโลยีสารสนเทศ 2 (ง33102)

ในการเขียนโปรแกรมคอมพิวเตอร์บางครั้งจำเป็นต้องมีชุดคำสั่งบางชุดที่จะต้องใช้งานบ่อยๆ เมื่อโปรแกรมต้องการทำชุดคำสั่งเหล่านั้นและต้องเขียนชุดคำสั่งนั้นใหม่จะทำให้เสียเวลาและโปรแกรมมีขนาดใหญ่ เราสามารถนำชุดคำสั่งที่ต้องทำบ่อยๆ มารวมเป็นฟังก์ชันได้ ถ้าหากต้องการทำชุดคำสั่งนั้นเมื่อใดก็ได้ เรียกชื่อฟังก์ชันนั้นได้เลย ฟังก์ชันถือว่าเป็นส่วนหนึ่งของโปรแกรม ฟังก์ชันในภาษาซีอาจเรียกว่าเป็น โปรแกรมย่อยที่สามารถทำงานได้อย่างหนึ่งได้ มีชื่อเฉพาะตัว สามารถเรียกใช้ได้ เมื่อคอมพิวเตอร์ทำงาน ฟังก์ชันใดฟังก์ชันหนึ่งเสร็จแล้วจะต้องกลับไปทำงานยังจุดที่เรียกใช้ฟังก์ชัน ในการเขียนโปรแกรมด้วย ภาษาซีจะต้องมีฟังก์ชัน `main()` เสมอ และถ้าต้องการให้โปรแกรมทำงานใด ๆ ฟังก์ชัน `main()` จะไปเรียก ฟังก์ชันอื่น ๆ ให้โปรแกรมทำงาน ในภาษาซีจะมีไลบรารีสำหรับเก็บฟังก์ชันต่าง ๆ ให้เราเลือกใช้งาน เช่น ฟังก์ชัน `printf()` จะเก็บไว้ใน `stdio.h`

ฟังก์ชันในภาษาซีแบ่งตามที่มาของฟังก์ชันสามารถแบ่งออกได้เป็นสองประเภท คือ

1. ฟังก์ชันมาตรฐาน (Standard Function) เป็นฟังก์ชันที่ถูกสร้างขึ้น และเก็บไว้ในไลบรารีในการใช้งาน เราต้องเรียกใช้ Include Directive เพื่อเรียก Header file ขึ้นมาก่อน จึงสามารถใช้งานฟังก์ชันได้
2. ฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้น (Uses-defined Function) เป็นฟังก์ชันหรือโปรแกรมย่อยที่ผู้ใช้สร้าง ขึ้นมาในการเขียนโปรแกรมเพื่อทำงานใดงานหนึ่ง

ฟังก์ชันมาตรฐาน (Standard Function)

1.) `printf()`;

เป็นฟังก์ชันที่ใช้พิมพ์หรือแสดงข้อความออกทางจอภาพถูกนิยามไว้ใน `stdio.h`

`printf("ข้อความ");`

`printf("รหัสควบคุมการแสดงผล",ตัวแปร);`

`printf("ข้อความ รหัสควบคุมการแสดงผล",ตัวแปร);`

เช่น

`printf("Hello this is number = %d",x);`

Hello this is number = คือ ข้อความ โปรแกรมจะแสดงข้อความดังกล่าวออกทางจอภาพ

%d คือ รหัสควบคุมการแสดงผลซึ่งจะดึงค่าจากตัวแปร x (ดูที่ใบงานตัวควบคุมการแสดงผลและการรับข้อมูล)

x คือ ตัวแปรที่ตัวกำหนดตัวแปรจะดึงค่าไปแสดงผล

2.) `scanf()`;

เป็นฟังก์ชันที่ใช้รอรับค่าข้อมูลจากแป้นพิมพ์มาเก็บไว้ในตัวแปรเพื่อใช้ในการคำนวณและการประมวลผล ถูกนิยามไว้ใน `stdio.h`

`scanf("รหัสควบคุมการแสดงผล",&ตัวแปร);`

เช่น

`scanf("%d %c",&x,&y);`

%d คือ รหัสควบคุมในการกำหนดค่าซึ่งจะใส่ค่าเข้าไปที่ตัวแปร x โดยจะต้องตรงกับการประกาศตัวแปร

%c คือ รหัสควบคุมในการกำหนดค่าซึ่งจะใส่ค่าเข้าไปที่ตัวแปร y โดยจะต้องตรงกับการประกาศตัวแปร

& เป็นสัญลักษณ์ในการใส่ค่าไปที่ตัวแปรจะใช้กับฟังก์ชัน scanf(); อ่านว่า address

*การเรียกแสดงผลค่าหรือกำหนดค่าของตัวแปรจะเรียกใช้ทีละตัวดังตัวอย่าง scanf();

3.) clrscr();

เป็นฟังก์ชันลบข้อความบนจอภาพ ถูกนิยามไว้ใน conio.h

clrscr();

4.) getch();

เป็นฟังก์ชันที่รับข้อมูลที่เป็นตัวอักขระ 1 ตัวอักษร เข้ามาทางแป้นพิมพ์โดยเมื่อป้อนข้อมูลเสร็จ ไม่ต้อง กดปุ่ม Enter และอักขระที่ป้อนเข้ามาจะไม่ปรากฏบนจอภาพ

getch();

5.) puts();

เป็นฟังก์ชันที่ใช้แสดงผลข้อมูลที่เป็นข้อความที่เก็บไว้ในตัวแปรชุดออกมาบนจอภาพ จะบรรทัดใหม่ให้เอง

puts(n);

puts(“ข้อความ”);

n คือ ชื่อตัวแปรชุดที่ต้องการนำมาแสดงผล ต้องเป็นชนิด char

*เนื่องจากการกำหนดตัวแปรของชนิดข้อมูล char นั้นหากไม่กำหนดขนาดตัวอักษรจะบันทึกได้เพียง 1 ตัว

เช่น char x = ‘A’ , char y[10] = “Thailand” เป็นต้น

6.) putchar();

เป็นฟังก์ชันที่ใช้แสดงอักขระเพียง 1 ตัว ออกทางจอคอมพิวเตอร์

putchar(ตัวแปรประเภท char);

putchar(‘อักขระ 1 ตัว’);

7.) textcolor();

เป็นฟังก์ชันที่ใช้สำหรับกำหนดสีของตัวอักษรจะต้องใช้ร่วมกับฟังก์ชัน cprint(); ถูกนิยามไว้ใน conio.h

textcolor(ค่าสี);

8.) cprintf();

เป็นฟังก์ชันที่ใช้แสดงข้อความหรือตัวอักษรเป็นสีออกทางจอภาพตามค่าของสีที่กำหนดไว้ใน ฟังก์ชัน textcolor(); และมีรูปแบบการใช้งานเหมือนฟังก์ชัน printf(); ต่างกันตรงที่จะแสดงข้อความออกมา เป็นสี การใช้งานฟังก์ชัน cprintf(); ต้องกำหนดสีของตัวอักษรก่อนเสมอ ถูกนิยามไว้ใน conio.h

```

cprintf("ข้อความ");
cprintf("ตัวกำหนดชนิดตัวแปร",ตัวแปร);
cprintf("ข้อความ ตัวกำหนดชนิดตัวแปร",ตัวแปร);

```

เช่น

```

textcolor(4);
cprintf(" computer ");

```

ผลการทำงานของโปรแกรมจะแสดงข้อความ computer เป็นสีแดง

ตารางแสดงค่าสี ค่าสี	สีที่แสดง
0(black)	ดำ
1(blue)	น้ำเงิน
2(green)	เขียว
3(cyan)	น้ำเงินเขียว
4(red)	แดง
5(magenta)	ม่วง
6(brown)	น้ำตาล
7(lightgray)	เทาสว่าง
8(darkgray)	เทาดำ
9(lightblue)	น้ำเงินสว่าง
10(lightgreen)	เขียวสว่าง
11(lightcyan)	น้ำเงินเขียวสว่าง
12(lightred)	แดงสว่าง
13(lightmagenta)	ม่วงสว่าง
14(yellow)	เหลือง
15(white)	ขาว
128(blink)	กระพริบ

9.) gotoxy();

เป็นฟังก์ชันที่ใช้ในการกำหนดตำแหน่งในหน้าจอ ด้วยแกน x แกน y

```
gotoxy(แกน x , แกน y);
```

เช่น

```
gotoxy(10,5);
```

10 คือ ค่าแกน x นับ 1 หน่วย ต่อ 1 ตัวอักษร ในที่นี้จึงเท่ากับเว้น 10 ตัวอักษร ในแนวนอน

5 คือ ค่าแกน y นับ 1 หน่วย ต่อ 1 ตัวอักษร ในที่นี้จึงเท่ากับเว้น 5 ตัวอักษร ในแนวตั้ง