

ข้อมูลชนิดสตริง (String) ในภาษาซี

- ลักษณะของข้อมูลที่มาจากกลุ่มของตัวอักษร
- Array of characters
- ใช้เก็บข้อมูลเป็นข้อความ เช่น ชื่อ นามสกุล ที่อยู่ เบอร์โทร เพศ ฯลฯ

ตัวอย่าง

Watcharee Shapphum

0-2956-3340-5

Teach Computer

32 Moo 1, Bankao, Muang Kanchanaburi 71000 Thailand

การประกาศตัวแปรแบบสตริง

การประกาศตัวแปรแบบสตริง มีรูปแบบดังนี้

```
char    var_name[size] ;
```

ตัวอย่าง

```
Char Number[100] ;  
char Name[50] ;  
char SurName[50] , Address[50] ;
```

หมายเหตุ

- size (ขนาดสตริง) = จำนวนตัวอักษร + 1 ตัวอักษรปิดสตริง
- ตัวอักษรปิดสตริง คือ '\0' (Null character)

การกำหนดค่าให้กับตัวแปรแบบสตริง

Array of characters ใช้ ' (Single quotes) และใช้ตัวอักษร '\0' ปิดสตริง)

```
char FirstName[15] = {'M', 'i', 's', 's', ' ', 'W', 'a', 'n', 'g', '\0'} ;  
char PhoneNo[ ] = {'0', '8', '9', '4', '6', '3', '9', '3', '9', '6', '\0' } ;
```

ใช้ " (Double quotes)

```
char Address[ ] = "Muang Kanchanaburi Thailand" ;
```

หมายเหตุ

- Array of characters หนึ่งมิติใช้เก็บข้อความสตริงได้
- ใช้ตัวอักษร '\0' (Null character) ปิดสตริง

การใช้คำสั่ง printf() แสดงผลตัวแปรแบบสตริง

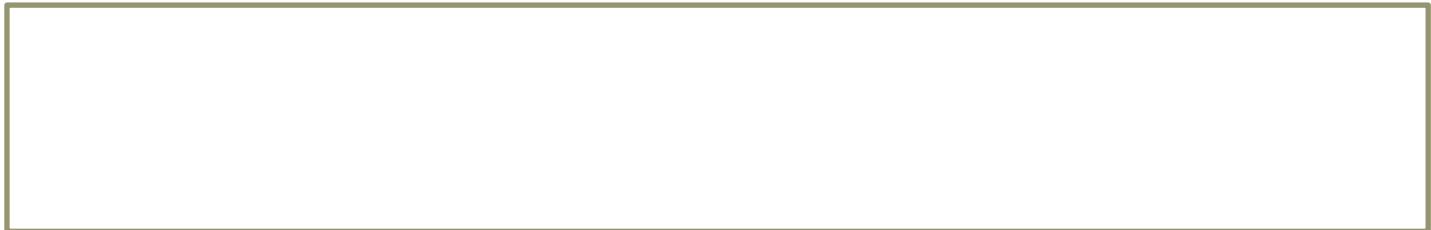
การใช้คำสั่ง printf() แสดงผลตัวแปรสตริง

```
#include <stdio.h>
int main(void)
{
    char Province[100] = "Kanchanaburi";
    printf("Province name is %s", Province);
}
```

Format string is %s



Output



การใช้คำสั่ง printf() แสดงผลตัวแปรแบบสตริง

การใช้คำสั่ง printf() แสดงผลตัวแปรสตริง

```
#include <stdio.h>
int main(void)
{
    char Province[100] = "Kanchanaburi";
    printf("Province name is %s", Province);
}
```

Output

Format string is %s



```
Province name is Kanchanaburi
```

ตัวอย่าง โปรแกรมเก็บข้อมูลไว้ในตัวแปรสตริง และแสดงผลตัวแปรแบบสตริง

```
#include <stdio.h>
int main(void)
{
    char str[80];
    printf("Enter a string : ");
    gets(str); // เก็บข้อมูลสตริงไว้ในตัวแปร str
    printf("You enter is : %s", str);
    return(0);
}
```

ตัวอย่าง โปรแกรมเก็บข้อมูลไว้ในตัวแปรสตริง และแสดงผลตัวแปรแบบสตริง

```
#include <stdio.h>
int main(void)
{
    char str[80];
    printf("Enter a string : ");
    gets(str); // เก็บข้อมูลสตริงไว้ในตัวแปร str
    printf("You enter is : %s", str);
    return(0);
}
```

Output

```
Enter a string : WATCHAREE
You enter is : WATCHAREE
```

ตัวอย่างการใช้งานตัวแปรสตริง

```
#include <stdio.h>
char message[21];
int x=0;
int main(void){
    //measure the length of message
    printf("Enter your message : ");
    gets(message);
    while(message[x] != '\0')
    {
        x++;
    }
    printf("Message Length = %d\n", x);
    return(0);
}
```


ตัวอย่างการใช้งานตัวแปรสตริง

```
#include <stdio.h>
char message[21];
int x=0;
int main(void){
    //measure the length of message
    printf("Enter your message : ");
    gets(message);
    while(message[x] != '\0')
    {
        x++;
    }
    printf("Message Length = %d\n", x);
    return(0);
}
```

Output

```
Enter your message : WATCHAREE
Message Length = 9
```

ตัวอย่าง การรับข้อมูลที่เป็นสตริง และพิมพ์ข้อมูลในสตริงออกมาทีละเซลล์

```
#include <stdio.h>
int main(void)
{
    char str[80];
    int i;
    printf("Enter a string : ");
    gets(str);
    for (i=0;str[i];i++)
        printf("str[%d] = %c\n" , i, str[i]);
    return(0);
}
```

ตัวอย่าง การรับข้อมูลที่เป็นสตริง และพิมพ์ข้อมูลในสตริงออกมาทีละเซลล์

```
#include <stdio.h>
int main(void)
{
    char str[80];
    int i;
    printf("Enter a string : ");
    gets(str);
    for (i=0;str[i];i++)
        printf("str[%d] = %c\n" , i, str[i]);
    return(0);
}
```

Output

หมายเหตุ

'\0' = null = 0 = False

```
Enter a string : I am Kea
str[0] = I
str[1] = 
str[2] = a
str[3] = m
str[4] = 
str[5] = K
str[6] = e
str[7] = a
```

ตัวอย่างการกำหนดค่าให้กับตัวแปรแบบสตริง

```
#include <stdio.h>
char sentence[21] = "Computer Programming";
char word[9] = {'T', 'h', 'a', 'i', 'l', 'a', 'n', 'd', '\0'};
char nword[4] = {'B', 'i', 'l', 'l'};
int main(void)
{
    printf("%s\n", sentence);
    printf("%s\n", word);
    printf("%s\n", nword);
    return(0);
}
```

Output

หมายเหตุ

- ตัวแปรสตริง nword ไม่มีตัวอักษร '\0' (Null character) ปิดสตริง

ตัวอย่างการกำหนดค่าให้กับตัวแปรแบบสตริง

```
#include <stdio.h>
char sentence[21] = "Computer Programming";
char word[9] = {'T', 'h', 'a', 'i', 'l', 'a', 'n', 'd', '\0'};
char nword[4] = {'B', 'i', 'l', 'l'};
int main(void)
{
    printf("%s\n", sentence);
    printf("%s\n", word);
    printf("%s\n", nword);
    return(0);
}
```

Output

```
Computer Programming
Thailand
Bill$#!2^&*
```

หมายเหตุ

- ตัวแปรสตริง nword ไม่มีตัวอักษร '\0' (Null character) ปิดสตริง

การใช้คำสั่ง gets() และ scanf() รับข้อมูลสตริง

```
gets(str); //แบบที่ 1
#include <stdio.h>
int main(void)
{
    char s[30];
    printf("Input: ");
    gets(s);
    printf("Output: %s\n", s);
    return 0;
}
```

การใช้คำสั่ง gets() และ scanf() รับข้อมูลสตริง

```
gets(str); //แบบที่ 1
#include <stdio.h>
int main(void)
{
    char s[30];
    printf("Input: ");
    gets(s);
    printf("Output: %s\n", s);
    return 0;
}
```

Output

```
Input: Watcharee
Output: Watcharee
```

การใช้คำสั่ง gets() และ scanf() รับข้อมูลสตริง

scanf("%s", str); //แบบที่ 2

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

```
int main(void)
```

```
{
```

```
    char s[30];
```

```
    printf(" Input: ");
```

```
    scanf("%s", &s);
```

```
    printf(" Output: %s\n", s);
```

```
    return 0;
```

```
}
```


การใช้คำสั่ง gets() และ scanf() รับข้อมูลสตริง

```
scanf("%s", str); //แบบที่ 2
```

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

```
int main(void)
```

```
{
```

```
    char s[30];
```

```
    printf(" Input: ");
```

```
    scanf("%s", &s);
```

```
    printf(" Output: %s\n", s);
```

```
    return 0;
```

```
}
```

Output

Input: Watcharee Shapphum

Output: Watcharee

การใช้คำสั่ง gets() และ scanf()

ต่างกันอย่างไร

gets():

- ใช้สำหรับรับข้อมูลประเภทสตริง (String) เท่านั้น โดยจะอ่านข้อมูลจากคีย์บอร์ดจนกว่าจะเจอเครื่องหมาย newline (\n) หรือกด Enter
- มีความไม่ปลอดภัย เพราะไม่มีการจำกัดความยาวของสตริงที่รับเข้ามา ซึ่งอาจทำให้เกิด Buffer Overflow ได้
- รับข้อมูลที่มีช่องว่าง (Whitespace) ได้ทั้งหมดจนกว่าจะเจอ Enter

scanf():

- ใช้สำหรับรับข้อมูลหลากหลายประเภท (เช่น จำนวนเต็ม, จำนวนทศนิยม, สตริง)
- ปลอดภัยกว่า gets() แต่ถ้าใช้ %s ใน scanf() อาจเกิด Buffer Overflow ได้เช่นกันหากไม่ได้จำกัดความยาวของข้อมูลที่รับเข้า จึงควรใช้ฟังก์ชันเช่น fgets() หรือใช้การระบุขนาดในการรับข้อมูล
- จะรับข้อมูลจนกว่าจะเจอช่องว่าง (Whitespace) ตัวแรก เช่น ถ้าป้อน "Hello World" จะรับมาเฉพาะ "Hello"

ตัวอย่าง fgets()

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

```
int main() {
```

```
    char str[50];
```

```
    // รับข้อมูลจากคีย์บอร์ด (stdin) เข้ามาในอาร์เรย์ str โดยอ่านสูงสุด 49 ตัวอักษร
```

```
    printf("Enter your name: ");
```

```
    fgets(str, sizeof(str), stdin);
```

```
    // แสดงผลข้อมูลที่รับเข้ามา
```

```
    printf("Your name is: %s", str);
```

```
    return 0;
```

```
}
```

```
Enter your name: WATCHAREE
Your name is: WATCHAREE
```

ฟังก์ชันมาตรฐาน (Standard Function)

Standard Function เป็นฟังก์ชันสำเร็จรูปที่มีมาให้พร้อมกับ Compiler

โดยเก็บอยู่ในไฟล์ *.h ซึ่งมีรูปแบบดังนี้ `#include <file.h>`

Library Function ที่สำคัญ

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

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, $x \geq 0$
pow(x,y)	x ยกกำลัง y
log(x)	logarithm ฐาน e ของ x, $x \geq 0$
log10(x)	logarithm ฐาน 10 ของ x, $x \geq 0$
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

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

```
#include <math.h>
```

```
double x1 = 3.14159, x2 = 2.71828, y;
```

```
int main(void)
```

```
{
```

```
    y = sin(x1);
```

```
    printf("sin(pi) = %10.4f\n", y);
```

```
    printf("cos(pi/4) = %10.4f\n", cos(x1/4));
```

```
    printf("tan(pi/6) = %10.4f\n", tan(x1/6));
```

```
    printf("sqrt(2) = %10.4f\n", sqrt(2));
```

```
    printf("2.5^3 = %10.4f\n", pow(2.5,3));
```

```
    printf("loge(e) = %10.4f\n", log(x2));
```

```
    printf("log(100) = %10.4f\n", log10(100));
```

```
    printf("e^3 = %10.4f\n", exp(3));
```

```
    printf("abs(-7) = %10.4f\n", abs(-7));
```

```
    printf("fabs(-7.0) = %10.4f\n", fabs(-7.0));
```

```
    return(0);
```

```
}
```

```
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 – ctype.h

Function	Description
tolower(ch)	เปลี่ยนอักขระเป็นตัวพิมพ์เล็ก
toupper(ch)	เปลี่ยนอักขระเป็นตัวพิมพ์ใหญ่

ตัวอย่างการใช้ฟังก์ชัน tolower และ toupper

```
#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main()
{
    int length, i;
    char string[] = "Watcharee Shapphum";
    length = strlen(string);
    for(i=0; i<length; i++)
        string[i] = toupper(string[i]);
    printf("%s\n",string);
    for(i=0; i<length; i++)
        string[i] = tolower(string[i]);
    printf("%s\n",string);
    return(0);
}
```

Output

WATCHAREE SHAPPHUM

watcharee shapphum

ฟังก์ชันสำหรับข้อความ – string.h

Function	Description
strlen	ความยาวสตริง
strncpy, strncpy	คัดลอกสตริง
strcmp, strncmp	เปรียบเทียบสตริง
strcat, strncat	ต่อสตริง

ฟังก์ชันที่ใช้งานกับข้อมูลแบบสตริง

ฟังก์ชันที่ทำงานกับข้อมูลแบบสตริงมีอยู่มากมายเช่น การคัดลอก การค้นหา การเปรียบเทียบ การทำให้เป็นตัวอักษรเล็กหรือใหญ่ ต่าง ๆ ซึ่งอยู่ใน string.h ก่อนนำฟังก์ชันมาใช้งานจะต้องมีการกำหนด include directive ก่อนเสมอ คือ

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

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

ความยาวสตริง (strlen)

นับจำนวนอักขระในสตริง มีรูปแบบดังนี้

```
strlen(สตริง);
```

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    int length = strlen(name);
```

```
    printf("\"%s\" is %d characters string\n", name, length);
```

```
    printf("Length of \"C Language\" is %d\n", strlen("C Language"));
```

```
    return 0;
```

```
}
```

ความยาวสตริง (strlen)

นับจำนวนอักขระในสตริง มีรูปแบบดังนี้

```
strlen(สตริง);
```

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    int length = strlen(name);
```

```
    printf("\"%s\" is %d characters string\n", name, length);
```

```
    printf("Length of \"C Language\" is %d\n", strlen("C Language"));
```

```
    return 0;
```

```
}
```

Out put

```
"Watcharee" is 9 characters string
Length of "C Language" is 10
```

คัดลอกสตริง (strcpy)

คัดลอกสตริงต้นทางไปไว้ปลายทาง มีรูปแบบดังนี้

strcpy(สตริงปลายทาง , สตริงต้นทาง);

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    char a_name[20];
```

```
    strcpy(a_name, name);
```

```
    printf("name: %s\n", name);
```

```
    printf("a_name: %s\n", a_name);
```

```
    return 0;
```

```
}
```

คัดลอกสตริง (strcpy)

คัดลอกสตริงต้นทางไปไว้ปลายทาง มีรูปแบบดังนี้

strcpy(สตริงปลายทาง , สตริงต้นทาง);

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    char a_name[20];
```

```
    strcpy(a_name, name);
```

```
    printf("name: %s\n", name);
```

```
    printf("a_name: %s\n", a_name);
```

```
    return 0;
```

```
}
```

Output

```
name: Watcharee
```

```
a_name: Watcharee
```

เปรียบเทียบสตริง (strcmp)

นำสตริงสองตัวมาเปรียบเทียบกัน มีรูปแบบดังนี้

```
strcmp(สตริง1 , สตริง2);
```

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    if (strcmp(name, "Watcharee") == 0) {
```

```
        printf("Equal");
```

```
    } else {
```

```
        printf("Not equal");
```

```
    }
```

```
    return 0;
```

```
}
```

ผลการเปรียบเทียบ	ค่าที่ส่งกลับ
สตริง1 < สตริง2	จำนวนลบ
สตริง1 = สตริง 2	ศูนย์
สตริง1 > สตริง 2	จำนวนบวก

เปรียบเทียบสตริง (strcmp)

นำสตริงสองตัวมาเปรียบเทียบกัน มีรูปแบบดังนี้

strcmp(สตริง1 , สตริง2);

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

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

```
int main()
```

```
{
```

```
    char name[] = "Watcharee";
```

```
    if (strcmp(name, "stdio") == 0) {
```

```
        printf("Equal");
```

```
    } else {
```

```
        printf("Not equal");
```

```
    }
```

```
    return 0;
```

```
}
```

Out put **Equal**

ผลการเปรียบเทียบ	ค่าที่ส่งกลับ
สตริง1 < สตริง2	จำนวนลบ
สตริง1 = สตริง 2	ศูนย์
สตริง1 > สตริง 2	จำนวนบวก

ต่อสตริง (strcat)

นำสตริงสองตัวมาต่อกัน มีรูปแบบดังนี้

```
strcat(สตริง1 , สตริง2);
```

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

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

```
int main()
```

```
{
```

```
    char sitename[] = "Watcharee ";
```

```
    char tld[] = "Shapphum";
```

```
    strcat(sitename, tld);
```

```
    printf("%s ", sitename);
```

```
    return 0;
```

```
}
```

ต่อสตริง (strcat)

นำสตริงสองตัวมาต่อกัน มีรูปแบบดังนี้

```
strcat(สตริง1 , สตริง2);
```

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

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

```
int main()
```

```
{
```

```
    char sitename[] = "Watcharee ";
```

```
    char tld[] = "Shapphum";
```

```
    strcat(sitename, tld);
```

```
    printf("%s ", sitename);
```

```
    return 0;
```

```
}
```

Output

Watcharee Shapphum

โจทย์

1. ให้นักเรียนเรียงตัวอักษร ชื่อของตัวเองจากน้อยไปหามากโดยรับค่าชื่อจากแป้นพิมพ์ (1) + รับมากกว่า 2 คน โดยใส่จำนวนคนที่ต้องการรับ (3) + อื่นๆ ใช้ฟังก์ชันใดก็ได้ต่อกันเช่นแปลงตัวอักษรพิมพ์เล็กเป็นพิมพ์ใหญ่หรือ นำชื่อเพื่อนมาต่อกัน (4)

Input → watcharee

Out → aaceehrtw

```

#include <stdio.h>

int main() {
    char c[30],a,i,j;

    printf("Enter a character: ");
    scanf("%s", &c);
    for (i = 0; i <= 4; ++i) {
        for (j = i + 1; j <= 4; ++j) {
            if (c[ i ] > c[ j ]) {
                a = c[ i ];
                c[ i ] = c[ j ];
                c[ j ] = a;   }
        }
    }
    for (i = 0; i <=4; ++i) {
        printf("%c \n", c[ i ]);
    }
    return 0;
}

```

โจทย์

2. ให้นักเรียนทำการเปรียบเทียบ ชื่อ นามสกุลของตนเอง โดยรับชื่อ และนามสกุลทางแป้นพิมพ์ ถ้าชื่อตัวอักษรมากกว่านามสกุลให้ใช้คำว่า name more than surname ถ้านามสกุลมากกว่าชื่อให้ใช้คำว่า surname more than name พร้อมแสดงชื่อ นามสกุลของตนเอง แบบกลับหลัง (2) + รับมากกว่า 2 คน (3) + อื่นๆ (4)

```
#include <stdio.h>
#include <string.h>
int main()
{
    char name[30],surname[30];
    printf(" Input name: ");
    scanf("%s", &name);
    printf(" Input surname: ");
    scanf("%s", &surname);
    int length1 = strlen(name);
    int length2 = strlen(surname);
    if (length1 == length2) {
        printf("Equal");
    } else {
        printf("Not equal");
    }
    return 0;
}
```