



# #5 Pointer

## What is pointer?

Pointer เป็นตัวแปรประเภทหนึ่ง มีหน้าที่ไว้ชี้ค่าบน memory โดยการใช้ address

## How to use?

Syntax

```
type *variable = &var;
```

- `type *variable` เป็นส่วนของการสร้างตัวแปร pointer ซึ่งจะเป็น pointer ที่ชี้ไปยังค่าชนิดตาม type
- `&var` เป็นการดึงค่า address ของตัวแปรออกมา

Example 1

```
#include <stdio.h> int main() { int x = 5; int *x_ptr = &x; printf("x = %d\n\n", x); printf("x_ptr = %p\n\n", x_ptr); printf("&x = %p\n\n", &x); }
```

- `int *x_ptr` เป็นการสร้างตัวแปร pointer ที่ชื่อ x\_ptr เพื่อไปที่ค่า integer
- `&x` เป็นการดึงค่า address ของตัวแปร x ออกมา

หมายเหตุ เราจะใช้ `type *` ในตอนการสร้างตัวแปรเท่านั้น

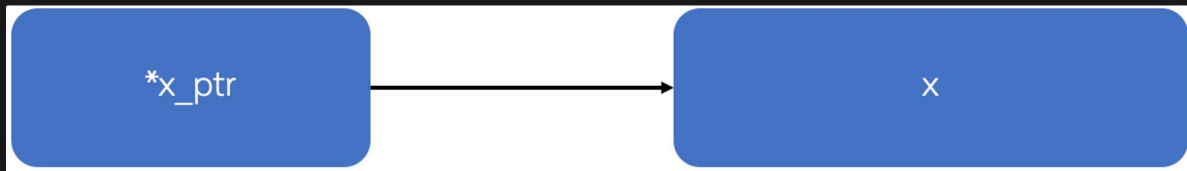
Example 2

```
#include <stdio.h> int main() { int x = 5; int *x_ptr; x_ptr = &x; }
```

- `int *x_ptr;` เป็นการสร้างตัวแปร pointer ที่ชื่อ x\_ptr เพื่อไปที่ค่า integer โดยไม่ได้มีการกำหนดค่าที่ชี้ไป
- `x_ptr = &x;` เป็นการกำหนดค่าของ x\_ptr ให้ชี้ไปที่ address ของตัวแปร x

# Let's see how pointer do

ถ้าหากเรามอง pointer เป็นภาพจะมองได้ดังนี้



ถ้าหากเราลองมองลึกลงไปถึง memory

	Address	Value
x_ptr { x {	1000	5
	1004	1000
	1008	

- ที่ address 1000 - 1003 นั้นคือตัวแปร x ซึ่งเก็บ ค่า 5
- ที่ address 1004 - 1011 นั้นคือตัวแปร x\_ptr เก็บค่า 1000 ซึ่งเป็น address ของตัวแปร x

## How to get the value from pointer?

ตอนนี้เราสามารถที่จะสร้างตัวแปร pointer แล้ว เราสามารถที่จะนำค่าจากตัวแปรที่ pointer นั้นชี้ไปได้โดยการ **Dereference** โดยการใช้ operator **\*** ซึ่งเหมือนกันตอนสร้าง แต่คนละหน้าที่กัน ในการใช้งานนั้นเราจะใช้ **\*** นำหน้าตัวแปร pointer เพื่อที่จะเข้าถึงค่าที่ตัวแปร pointer นั้นชี้อยู่ เพื่อเข้าถึงค่า หรือ แก้ไขค่าได้

### Example 3

```
#include <stdio.h> int main() { int x = 5; int *x_ptr = &x; printf("x = %d\n\n", x); printf("x_ptr = %p\n\n", x_ptr); printf("&x = %p\n\n", &x); printf("=====\n\n"); printf("*x_ptr = %d\n\n", *x_ptr); printf("=====\n\n"); *x_ptr = 10; printf("x = %d\n\n", x); printf("*x_ptr = %d\n\n", *x_ptr); }
```

หมายเหตุ การ Dereference นั้นเปรียบเสมือนการเปลี่ยนตัวแปรนั้นๆไปเป็นตัวแปรที่ Dereference ไป

## Profit of pointer

### Memory Allocation

คือ การขอ allocate พื้นที่ memory เพิ่มซึ่งจะ allocate ในส่วนของ heap ตัวแปรนี้จะมีอายุตลาดการทำงานของโปรแกรม ซึ่งในการ allocate นั้นเราจะต้องใช้ pointer ในการเก็บ address ของพื้นที่ที่ได้ทำการ allocate มา โดยมีหลาย function ที่จะ allocate memory ได้ ในที่นี้จะยกตัวอย่างการใช้ `malloc()` ซึ่งอยู่ใน library `stdlib`

```
void *malloc(size_t size);
```

#### Example 4

```
#include <stdio.h> #include <stdlib.h> int main() { float *x_ptr = (float *)malloc(sizeof(float)); *x_ptr = 5.3; printf("x_ptr = %p\n\n", x_ptr); printf("*x_ptr = %f\n\n", *x_ptr); free(x_ptr); }
```

- `float *x_ptr` ตั้งตัวแปร `x_ptr` เป็นตัวแปร pointer
- `(float *)malloc(sizeof(float))`
  - `malloc` เป็น function สำหรับ allocate memory
  - ในส่วน parameter จะรับขนาดที่ต้อง allocate เป็น byte
  - ใช้ `sizeof(float)` เพื่อเอาขนาด float มา allocate
  - ปกติ function `malloc` นั้นจะ return เป็น `void *` เราจึงต้อง cast `(float *)` เพื่อเปลี่ยนประเภทที่ของตัวแปรที่ return ออกมา

แต่ถ้าแบบนี้อาจจะมองไม่เห็นข้อดีกันลองแปลงเป็นการทำเป็น function กัน

#### Example 5

```
#include <stdio.h> #include <stdlib.h> int *createArray(int size) { int *arr = (int *)malloc(size * sizeof(int)); for (int i = 0; i < size; i++) { arr[i] = i; } return arr; } void printArray(int *arr, int size) { for (int i = 0; i < size; i++) { printf("%d ", arr[i]); } printf("\n"); } int main() { int SIZE = 10; int *arr = createArray(SIZE); printArray(arr, SIZE); free(arr); }
```

- เราสามารถใช้ pointer เพื่อ allocate array ได้เพราะ array นั้นคือ memory ที่อยู่ติดกันได้
- จาก code ดังกล่าวช่วยในการ allocate array เพราะ ปกตินั้นเราไม่สามารถ return array ออกมาได้ถ้าหากไม่ใช่ pointer

## Pass by reference (?)

เราสามารถใช้ pointer เพื่อเลียนแบบการทำ pass by reference ได้

#### Example 6

```
#include <stdio.h> void increaseByTen(int x) { x += 10; } int main() { int  
x = 5; printf("x = %d\n\n", x); increaseByTen(x); printf("x = %d\n\n", x);  
}
```

### Question ?

ทำไม code ข้างต้นจึงใช้ไม่ได้

### Example 7

```
#include <stdio.h> void increaseByTen(int *x) { *x += 10; } int main() {  
int x = 5; printf("x = %d\n\n", x); increaseByTen(&x); printf("x =  
%d\n\n", x); }
```