



## lesson 33 pointer

A pointer is essentially a simple **integer variable** which holds a **memory address** that points to a value, instead of holding the actual value itself.

But first, what's the **memory address** ?

When a variable is created in C, a memory address is assigned to the variable. The memory address is the **location** of where the variable is stored on the computer.

If we have a variable **x**, and we want to accede to its memory address, we need to use **&**.

Example:

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

```
int main() {
```

```
int x = 5;
```

```
printf("x : %d \n", x);
```

```
printf("address of x : %p ", &x);
```

```
/*
```

here we putted the & to print the address not the value of x itself

```
*/
```

```
}
```



output:

x : 5

address of x : 0113FBB4

**//The address may differ from one device to another**

-**Pointers** are variables are used to store [addresses](#) instead of [values](#), and their syntax is as follows:

```
int *p;
```

```
int* p;
```

```
int * p;
```

How to **assign** address to a pointer:

```
int* p, x;
```

```
x = 2;
```

```
p = &x;
```

//Now p contains the address of x.

If we want to print the value of the pointer (p), it's going to be the same value of x:

```
int* p, x;
```

```
x = 2;
```

```
p = &x;
```

```
printf("%d", *p);
```

Try to code yourself: [click here](#).



We can now say that :

**&**: address of variable

**\*** : value of the variable

That's why we cannot write

```
*p=&x;
```

-How to change the value using **Pointer**:

```
int main() {  
    int* p, x;  
    x = 3;  
    p = &x;  
    x = 1;  
    printf("%d ", x); // output: 1  
    printf("%d", *p); // output: 1  
}
```

Try to code yourself: [click here](#).

If we also change the pointer's value, the variable's value will change too :

```
int main() {  
    int* p, x;  
    x = 3;  
    p = &x;
```



```
*p = 1;  
printf("%d ", x); // output: 1  
printf("%d", *p); // output: 1  
}
```

Try to code yourself: [click here](#).

Another example of pointer and variables :

```
#include <stdio.h>  
  
int main() {  
    int* p, x;  
    x = 10;  
    printf("Address of x: %p\n", &x);  
    printf("Value of x: %d\n\n", x); // 10  
    p = &x;  
    printf("Address of pointer p: %p\n", p);  
    printf("Content of pointer p: %d\n\n", *p); // 10  
    x = 15;  
    printf("Address of pointer p: %p\n", p);  
    printf("Content of pointer p: %d\n\n", *p); // 15  
    *p = 25;  
    printf("Address of x: %p\n", &x);  
    printf("Value of x: %d\n\n", x); // 25
```



```
}
```

output:

Address of **x**: 004FFC38

Value of **x**: 10

Address of pointer **p**: 004FFC38

Content of pointer **p**: 10

Address of pointer **p**: 004FFC38

Content of pointer **p**: 15

Address of **x**: 004FFC38

Value of **x**: 25

Try to code yourself: [click here](#).

Some **mistakes** while using pointers :

```
int main() {
```

```
int x, * p;
```

```
//x refers to the value
```

```
//p to the address
```

```
p=x; // WRONG &x refers to the address
```

```
*p = &x; //WRONG x refers to the value
```

```
*p =x; //CORRECT
```

```
}
```

Try to code yourself: [click here](#).



We want to make a program that changes the values of **variables** but using function:

```
#include <stdio.h>

void fun(int x, int y) {
    int t = x;
    x = y;
    y = t;
}

int main() {
    int x = 5, y = 20;
    fun(x, y);
    printf("x=%d y=%d", x, y);
}
```

output:

**x=5 y=20**

Try to code yourself: [click here](#).

We found that the two values did not change, why?

**y** and **x** inside **main**

**t, y** and **x** inside fun

The problem is that the values are changed inside fun

But when fun was called inside main it didn't change it inside main



So to solve this problem, we send the address of the variable in main to the function to change the value inside the address and calling fun inside main it becomes the values of the variables.

so we can find the values that we want.

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

```
void fun(int* x, int* y) {
```

```
int t = *x;
```

```
*x = *y;
```

```
*y = t;
```

```
//Here, change the values inside the addresses sent to you  
from main
```

```
}
```

```
int main() {
```

```
int x = 5, y = 20;
```

```
fun(&x, &y);
```

```
// هنا أذهب لعنوان المتغيرات
```

```
printf("x=%d y=%d", x, y);
```

```
}
```

output:

x=20 y=5

Try to code yourself: [click here](#).