→ Pointer

└→ Concept ⟶ variable ⟶ Store Address

⟶ Object ←
Pointer to object
→ int → char
→ double → Float
→ Array → Pointer
→ struct → union

⟶ Function
Pointer to Function

[Pointer to object]

*Definition
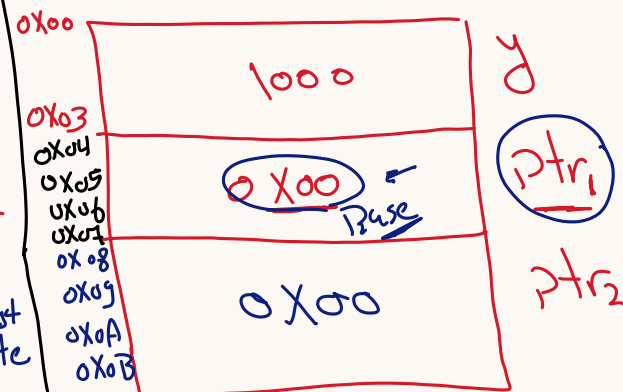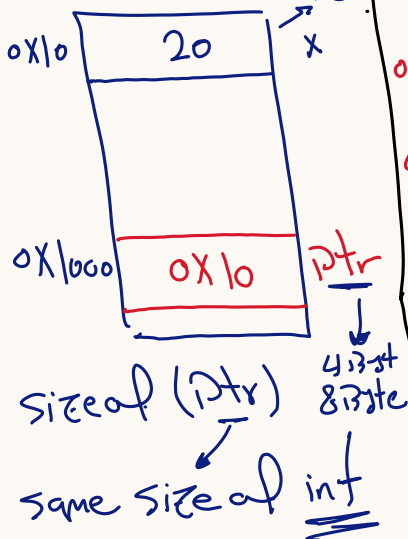
Pointee step

*PtrName = Address; ⟶ First 1 Byte

char x = 20;   int   y = 1000;

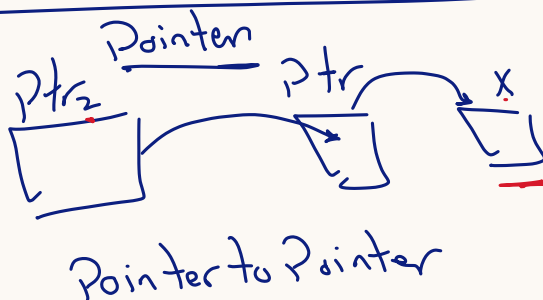char *Ptr = &x;   Short int *Ptr1 = &y;

                    int *Ptr2 = &y;

0X10	20	→ 13 x
0X1000	0X10	Ptr

43-34
8Byte

0X00
0X03
0X04
0X05
0X06
0X07
0X08
0X09
0X0A
0X0B

1000   y

0X00 Base   Ptr1

0X00   Ptr2

Sizeof (Ptr)
same size of int

Float
Float y = 20;   0X00
Float *Ptr = &y;

0X03
0X0L

20   y

0X00

① → Ptr → Base Address 0X07
  └→ step → Access Byte

Pointer
Ptr2   Ptr   X

Pointer to Pointer

NOTE
IN Created by Notein
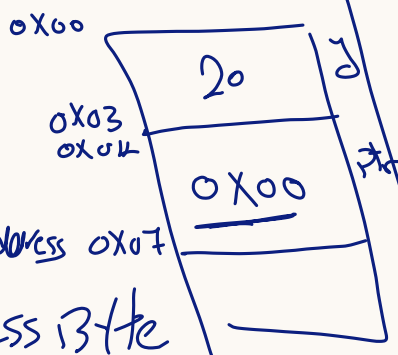
Declartion

└ [Char] * Ptr ; →

└ char x = 20; →

Ptr = & X;

Base Address 0x00
garbage

0x04 0x0C Ptr

0x03
0x04   20   x   → garbage
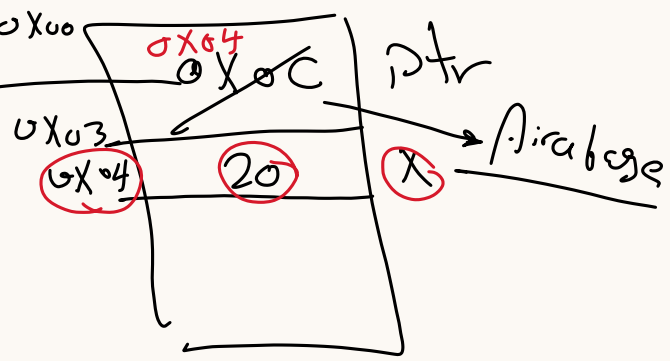
Pointer dereference

int x = 20;

x = 30;

*
Ptr = 50;

(*) (0x00) = 50;

→ dereference

Create

int (*) Ptr = & x;

0x00 → 50 20 30   x

0x04   0x00   Ptr
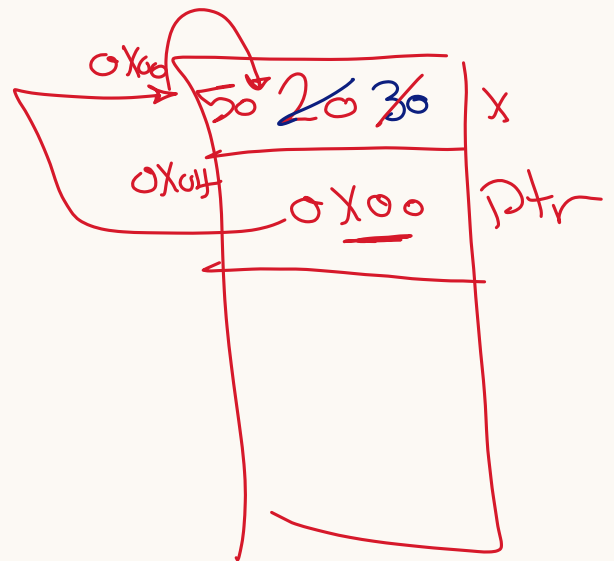
→ int z = 20;

Printf (" z = %d", z); → 20

→ Printf (" z = %d", &z); → Base Address of z

int * Ptr = & z;

* Ptr

* (Base Address)

NOTE
IN Created by Notein
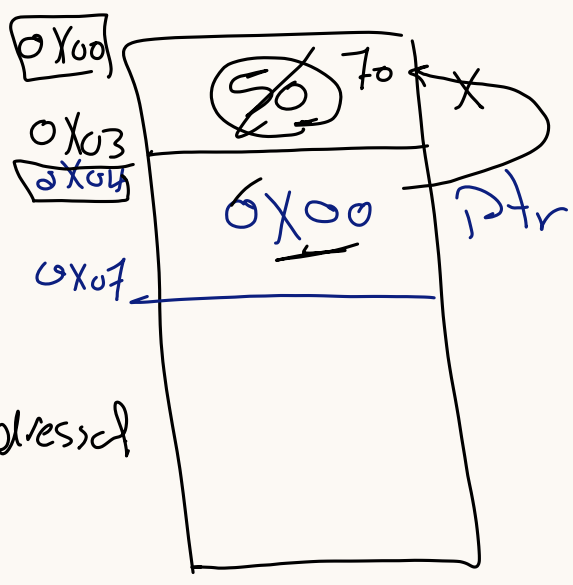
ex
```
int x = 50;
int * ptr = &x;
Print (x);  ⟶ 50 ✓
Print (&x);  ⟶ 0X00 ✓
Print (Ptr);  ⟶ 0X00 ⟶ Base Address of
                                    x
Print (&Ptr);  ⟶ 0X04
Print (*Ptr);  ⟶ 50
```
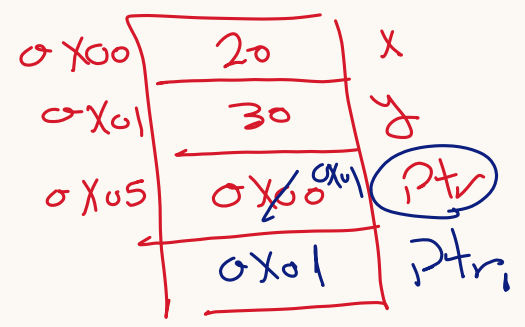
*Ptr = 70;

⎿ operation ⟶ Ptr
    ⎿ Increament   ⎿ decreament   ⎿ sub with step
    ⎿ sub Ptr from Ptr   ⎿ Add with step

Increament

⎿ char x = 20;
   int y = 30;
   char * Ptr = &x;
   ⎿ Ptr ++ ⟶ 0X01
   int * Ptr₁ = &y; ⟶ 0X01
       Ptr₁ ++ ⟶ 0X05

0X00	20	x
0X01	30	y
0X05	0X00 0X01	Ptr
	0X01	Ptr₁

increament by
one step

decreament
    short int x = 20;
    int y = 70;
    int * Ptr = &y;   Ptr --
                       0X02 -4 ⟶   4 Ptr

0X00		
0X01	20	x
0X02		
	70	y
0X05		
	0X02	Ptr
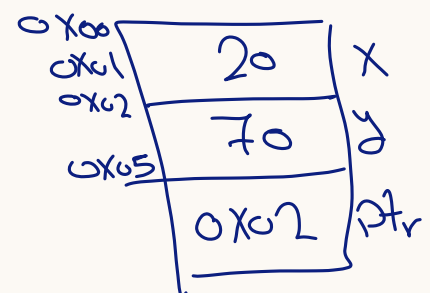
tips & tricks

```
int    x = 20;
int    y = 50;
char   z = 100;
int * Ptr = & x;
```

(*)Ptr ++ ; ──────────→ Print (20)
                                x = 21;
 * ++Ptr; ────────→
 ⎸
) Ptr ++;
   * Ptr ;

try it on Compiler
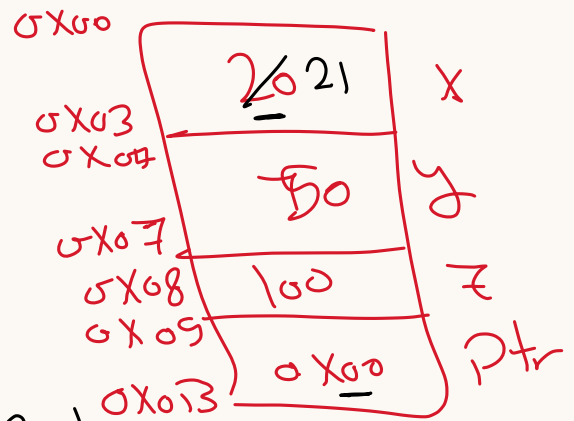
0x00		
	20 21	x
0x03 0x04	50	y
0x07 0x08 0x09	100	z
0x013	0x00	Ptr

⤷ NULL
   ✓
                    Nothing
              ⟋
(Void *)   O

Casting ──────→ NULL ⟹

int x = 0; ──→ Value
              ↘

int * Ptr = (void *) 0;
                        ↓
                  Addressing
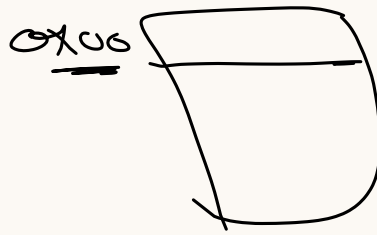
(void *) 0

int * Ptr = NULL;

Casting

int No1 ;

(float) No1 | No2;

Casting

Null ⟵ [(void *)0]

0x00 ▭

Char *Ptr = (Char *)0x0c;

Casting

Char * Ptr = (void *)0 ;

= Null ⟶ Not vailed Address

⤷ Not Access any Byte
in memory

0x0c

0x0f

0x00

1 Byte

X
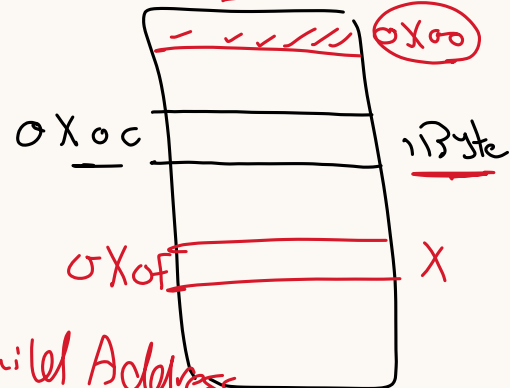
Ptr = X ;

Relation Between Arrary & Ptr

Char Arr[5] = {1,2,3,4,5};

arr [0] ⇒ 1
arr [1] ⇒ 2

$*(arr + 1) \longrightarrow$
$*(0x00 + 1) \longrightarrow$
$*(0x01) \longrightarrow 2$

0x00	1		arr 0
0x01	2		4
0x02	3		2
0x03	4		3
0x4	5		4

arr

0x00

↳ alias Pointer

arr [index] $\longrightarrow$ $*(arr + index)$

No of step

arr [3] → 4

$*(arr+3) \longrightarrow 0x00 + 3 \xrightarrow{*} *(0x03) \Rightarrow (4)$

& arr [4] $\longrightarrow$ (arr + 4)

(arr + 4)

Char arr [5];
Char * Ptr = arr;

Ptr [0]
Ptr [1]
Ptr [2]