Pointers continued :→ data

```
int x = 10;
```

- `int` → data type
- `x` → var → mem loc ⇒ address ⟶ CPU (0x1000x0) Compiler
- `10` → val
- `;` → [Compiler]

1. Address Operator ⇒ (& x) Address
2. Pointers → It is a va to store address of another variable

```
int *ptr  = &x
int* ptr  = &x
int *ptr  = &x
```

10 → x, 0x1000

x = 10 (1000) v1

ptr = &x = 1000 ⇒ *ptr (2000) v2 | P = address

dptr = &ptr = 2000 (3000) = * dptr (address) = ** dptr = 10

```
int
main() x  y
       20 30
```

```
void findMax ( int *x , int *y ) {
    *y = 30 is larger than *x = 20 ;
}
```

Arrays :→ Collection of Homogeneous Data

Any one particular data type: * Numbers * Decimals * Characters * Strings

data type ↑ `int` (data is fixed) ↓

name of array ↑ `arr[ ]` ↓ variable

```
int arr[ ] = { 9, 6, 8, 1, -2, 4 };
```

positions: 1 2 3 4 5 6 → position

elements: 9 6 8 1 -2 4 → elements

indexes: 0 1 2 3 4 5 → index

6 − 1 = 5 (len − 1) last index

{ ind = pos − 1 ; pos = ind + 1 }

(0 − based index)

* Max ① & Min ② in an array :
* Average ③

1, 2, 3, 4, 5

= 15 → Sum / 5 → (size)

= 3 (size)