

→ Pointer Datatype:

→ functions and pointer

→ Dynamic variables.

No name / exist between function calls

→ Dangling pointer ]

→ memory leak.

1) array name  
constant  
pointer

2) Access of  
data and  
processing is  
similar

int \* ptr = &arr[0]; →  
1 Store address of  
single variable  
Static / Dynamic.  
2 Store the base address  
of an array.  
1 → Subscript operator.  
2 → offset notation.

or address of  
any element  
from arr.

## Array and pointers:

(collection  
of data)

Name of array is a constant  
pointer. (can only point to one  
location / Address)

int arr[5];

Cout << arr;      Address base      012

Cout << 8arr[0];      012

Cout << 8arr[1];

int arr2[5];

[ arr = arr2;  
arr2 = arr; ]      compile time  
Error

int \* ptr = arr;      base address

Cout << ptr;

Cout << arr[0];      1/5

Cout << \*(arr + 0);      1/5

arr[1]

8

\*(arr + 1)

8

\*(arr + 3)      9

Cout << ptr;      base Address, 1st element

Cout << \*ptr;      "S

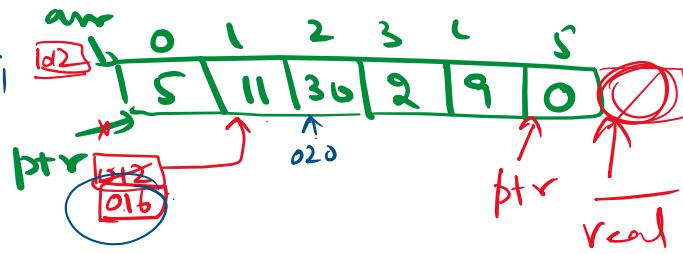
Cout << ptr[0];      "S

Cout << \*(ptr + 0);      "S

Subscript  
→ ptr[1];      \*(ptr + 1);  
→ ptr[3];      \*(ptr + 3);

int arr[6] = {5, 11, 30, 2, 9};

int \*ptr = arr;



for(int i=0; i<6; i++)

{ cout << ptr[i]; }

cout << \*(ptr+i);

} [ cout << \*(arr+i); ] II  
arr = arr + 1; Error

5 5 11 11 30 30 2 2 9 9 0 0

Cannot modify arr pointer constant.

[ptr = ptr + 1;] can modify normal pointer:

{ cout << ptr[0]; } 11  
cout << arr[0]; 5  
cout << arr[1]; 11

cout << \*(ptr+0); cout << \*ptr;  
cout << ptr[0]; 30

ptr = arr;  
ptr = 8 arr[5];

012  
020

032

cout << arr[6];

Runtime error.

arr[6] = 10 ✓

cout << ptr[1]; cout << \*(ptr+1); JUNK

→ [ptr[1] = 50; Runtime Error] (unallocated memory)

Array index out of bound.

int arr[4];

{ arr[4];  
arr[5]; }

