

메모리 주소 메모리는 바이트(Byte) 단위로 나뉜다. 각 바이트에는 주소가 지정됨

int 즉, 정수형 메모리는 4Byte로 임의의 위치에 장소를 잡음.

변수가 위치 하는 곳 = '주소' → 변수의 주소를 알려면 "&변수명" 으로 알수있다.

```
int aa[3] = { 1, 2, 3};
```

aa[0]의 주소 (&aa[0]) = 주소변지

aa[1]의 주소 (&aa[1]) = 주소변지 + 4번지

aa[2]의 주소 (&aa[2]) = 주소변지 + 8번지

→ 전체 주소변지 시작 이므로 즉, &aa[0] == aa  
↓  
주소변지를 보듬

&aa[0]	aa+0
&aa[1]	aa+1
&aa[2]	aa+2

→ 같다.

정수 형태의 주소 값을 저장하는 목적으로 선언되는 변수가 포인터 변수이다.

포인터 선언 : \*을 붙여줌

char ch; → 문자형 변수 ch 선언

char\* pch; → 문자형 포인터 변수 pch 선언

ch = 'A'; → 변수 ch에 'A' 값 대입

pch = &ch; → 포인터 변수 pch에 ch 변수의 주소를 대입.

주소값 출력은 printf("%p", pch);

변수와 변수의 주소를 참조할 포인터 변수는 데이터 형이 같아야 한다.

포인터 변수 값 정리는 "\*포인터 변수"로 정글하면 된다.

```
int * pch = NULL ... int * pch = 0
```

배열의 이름은 배열의 시작 주소 값을 의미하는 (배열의 첫 번째 요소를 가리키는) 포인터이다.

```
arr[i] == *(arr+i)
```

```
Ptr[i] == *(Ptr+i)
```

```
char s[8] = "Basic-C";
```

```
char* p;
```

$p = s \rightarrow$   $\&s[0]$  과 같다.

```
printf("&s[3] => %s", *p[3])
```

$\rightarrow$  문자열은 시작부분으로 설정되어 끝까지 출력됨  $\Rightarrow$  ic-c

```
printf("s[3] => %c", *(p+3))
```

$\rightarrow$  문자로 1개만 됨  $\Rightarrow$  ;

```
char str1[] = "my string"
```

$\rightarrow$  |m|y| |s|+|r|+|i|+|n|g|\0|

```
char *str2 = "your string"
```

$\rightarrow$  |yourstring|

$\rightarrow$  문자열로 저장된 배열

$\rightarrow$  주소값 저장