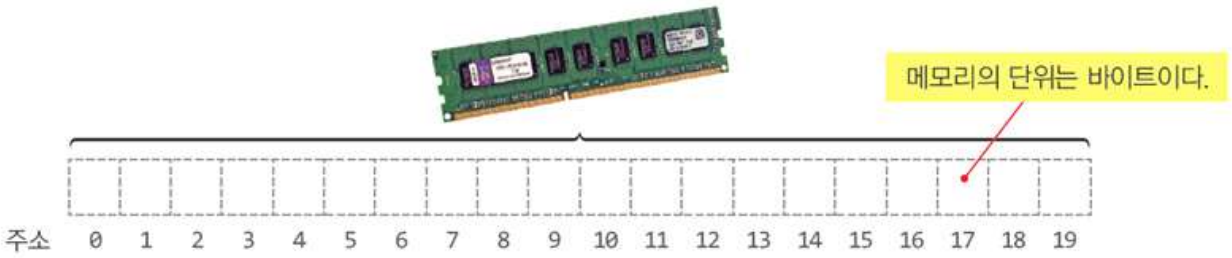
13. 포인터

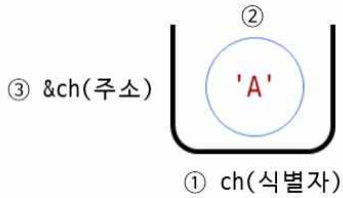
■ 메모리(memory)란? 컴퓨터 내부에서 변수, 명령을 처리한 결과를 기억하는 장치



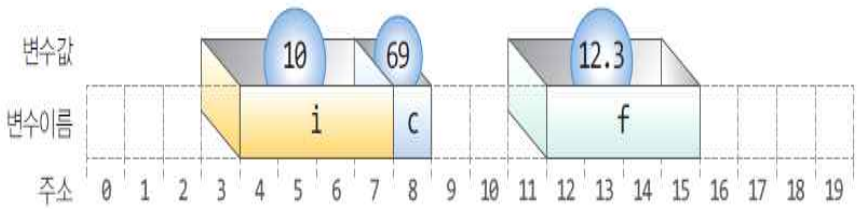
- 변수는 메모리에 저장되고, 바이트 단위로 메모리 주소가 구분된다.
- 메모리 주소 : 변수는 메모리에 저장되며, 메모리에 저장된 위치를 파악하기 위해 번호(일련번호)를 갖는데, 이 번호를 메모리 주소라 함.
- 변수를 이루는 세 가지 요소

```
char ch = 'A';
```

① 이름이 부여된 메모리
② 그 안에 담긴 정보
③ 메모리의 주소




```
int main(void)
{
        int i = 10;
        char c = 69;
        float f = 12.3;
        return;
}
```



- 어떤 변수를 피연산자로 사용한다면 대부분 변수에 저장된 정보가 연산에 참여

ex) f = (float)(i+c);

- 어떤 연산자는 메모리에 저장된 정보가 아니라 변수의 주소, 즉 메모리 주소 자체에 관심이 있음
- 주소번지 연산자 & (변수의 주소를 계산하는 연산자)

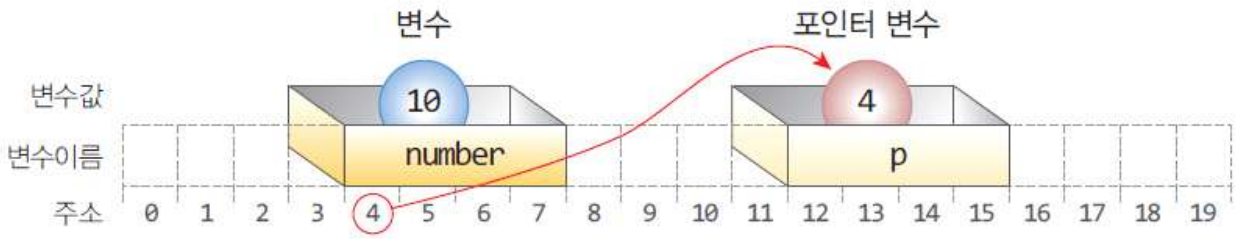
```c
#include<stdio.h>
int main(){
    int i = 10;
    char c = 69;
    double f = 12.3;
    // 변수에 들어있는 값을 출력
    printf("%d %c %f\n", i, c, f);
    // 변수 nData의 메모리 주소를 출력
    printf("%p %p %p\n", &i, &c, &f);
    return 0;
}
```

실행결과

- %p : 주소를 출력하는 서식 문자
- &i, &c, &f 라는 연산은 “이름이 i, c, f인 변수의 메모리 주소는?” 이라는 의미
- 이름이 i인 정수형 변수의 메모리 실제 주소는 0x 0000005A8D31F924이고, 그 변수 안에 저장된 값은 10이다.

■ 포인터란?

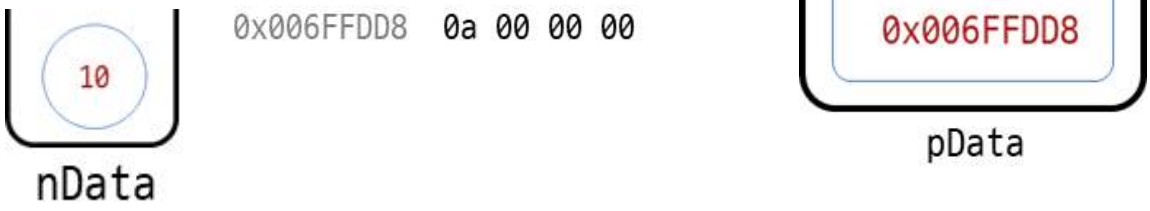
- 포인터 변수 : 변수의 주소를 저장하는 변수



- 포인터 변수 선언

`int * pData;`	● pData : 포인터 변수의 이름 ● int * : 자료형이 int형 변수의 주소 ● pData 는 int형 자료의 주소를 저장하는 포인터 변수이다!

```
int nData = 10;
int* pData = &nData;
```



- 포인터의 형(Type)

```
int *           int 형 포인터
int * pnum1;    int 형 포인터 변수 pnum1
double *        double 형 포인터
double * pnum2; double 형 포인터 변수 pnum2
char *          char 형 포인터
char * pchar;   char 형 포인터 변수 pchar
```

● 포인터 변수 예제(1)

```c
#include<stdio.h>
int main(){
    int nData = 10;
    int* pData = &nData;

    // 변수 nData에 들어있는 값을 출력
    printf("nData : %d\n", nData);
    // 변수 nData의 메모리 주소를 출력
    printf("&nData : %p\n", &nData);
    // 변수 pData에 들어있는 값을 출력
    printf("pData : %p\n", pData);
    return 0;
}
```

실행결과

```
nData :
&nData :
pData :
```

● 포인터 변수 예제(2)

```c
#include<stdio.h>
int main(){
    int num1 = 20;
    int num2 = 30;

    int* pnum1 = &num1;
    int* pnum2 = &num2;

    printf("  num1 value : %8d,  &num1 : %p\n", num1, &num1);
    printf("  num2 value : %8d,  &num2 : %p\n", num2, &num2);
    printf(" pnum1 value : %p, &pnum1 : %p\n", pnum1, &pnum1);
    printf(" pnum2 value : %p, &pnum2 : %p\n", pnum2, &pnum2);
    return 0;
}
```

실행결과

● 포인터 변수 예제(3)

```c
#include<stdio.h>
int main(){
    int num1 = 20;
    int num2 = 30;

    int* pnum1 = &num1;
    int* pnum2 = &num1;

    printf("  num1 value : %8d,  &num1 : %p\n", num1, &num1);

    printf(" pnum1 value : %p, &pnum1 : %p\n", pnum1, &pnum1);
    printf(" pnum2 value : %p, &pnum2 : %p\n", pnum2, &pnum2);

    pnum2 = &num2;
    printf(" pnum2 value : %p, &pnum2 : %p\n", pnum2, &pnum2);
    return 0;
}
```

실행결과

● 포인터 변수 예제(4)

```c
#include<stdio.h>
int main()
{
    printf("sizeof(char)   : %d, sizeof(char*)    : %d\n", sizeof(char), sizeof(char*));
    printf("sizeof(short)  : %d, sizeof(short*)   : %d\n", sizeof(short), sizeof(short*));
    printf("sizeof(int)    : %d, sizeof(int*)     : %d\n", sizeof(int), sizeof(int*));
    printf("sizeof(double) : %d, sizeof(double*)  : %d\n", sizeof(double), sizeof(double*));
    printf("sizeof(long long) : %d, sizeof(long long*) : %d\n", sizeof(long long), sizeof(long long*));
    return 0;
}
```

실행결과