

THE  
**C**  
PROGRAMMING  
LANGUAGE

13차시 포인터

# 포인터 변수

- 포인터는 메모리의 주소 값을 가리키는 것
- 포인터를 통해 메모리 접근가능
- 메모리의 주소 값을 저장할 수 있는 변수

주소값	100	101	102	103	104
-----	-----	-----	-----	-----	-----

- ```
int num = 10;  
int * pnum;  
pnum = &num;  
printf("%p", pnum); //100  
printf("%d", *pnum); //10
```

|    |  |  |  |  |
|----|--|--|--|--|
| 10 |  |  |  |  |
|    |  |  |  |  |
|    |  |  |  |  |
|    |  |  |  |  |

# 포인터 변수의 참조(값변경)

➤ `int num = 10;`  
`int* pnum = &num;`  
`printf("%d", *pnum);`  
`*pnum = 20;`  
`printf("%d", *pnum);`

주소값

| 100 | 101 | 102 | 103 | 104 |
|-----|-----|-----|-----|-----|
| 10  |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |
|     |     |     |     |     |

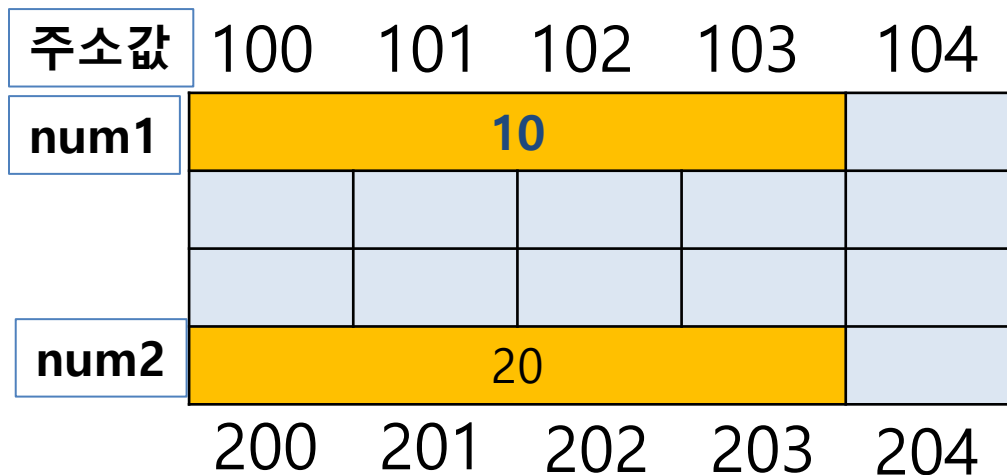
20으로 변경

# 포인터 변수의 참조(주소변경)

➤ `int num1 = 10;`  
`int num2 = 20;`

`int* pnum = &num1;`  
`printf("%d", *pnum);`

`pnum = &num2;`  
`printf("%d", *pnum);`



# 실습

- $a=10$ ,  $b=20$ ,  $c=30$  일 때  $base=100$ 을 각 변수에 더해보시오. 단, 변수  $a$ ,  $b$ ,  $c$ 에 접근 할 때 포인터 변수를 활용하시오

# 배열의 포인터

## ➤ 배열의 이름은 포인터

배열이름 == &배열이름

배열이름+1 != &배열이름+1

➤ 

```
int arr[3] = {10, 20, 30};  
int *p = arr;  
printf("%d %d %d",arr[0], arr[1], arr[2]);  
printf("%d %d %d",p[0], p[1], p[2]);  
printf("%d", *(arr+1)); //포인터는 그대로  
printf("%d", *(p+1));  //포인터는 그대로
```

# 포인터 연산

- 포인터의 값을 선언형의 크기에 따라 증감한다.

```
int* p = &num1;  
++p : 4 byte만큼 포인터 이동
```

```
double *p = &num1;  
++p : 8 byte만큼 포인터 이동
```

```
char *p = arr;  
++p : 1 byte만큼 포인터 이동
```

# 실습

- 문자형 배열에 'b', 'f', 'g', 'y', 'z'를 저장하고 포인터 변수를 이용하여 한 글자씩 화면에 출력하여라



# 문자열의 포인터

- `char str[] = "happy";`  
//가리키는 대상 변경 불가능
- `char *p_s = "happy";`  
//p\_s 에 h의 주소 값을 저장  
//가리키는 주소 값을 변경할 수 있다

# 실습

- 'hello'를 저장하고 포인터변수를 이용하여 'hello!'로 글자를 변경하여라

# 포인터의 배열

- 주소값의 저장이 가능한 배열

```
int * 포인터명[3] = {주소1,주소2,주소3};
```

- 정수형 포인터를 저장하는 예

```
int num1=10, num2=20, num3 =30;  
int * p[3] = {&num1, &num2, &num3} ;
```

- 문자열을 포인터를 저장하는 예

```
char * p[3] = {"happy", "sad", "angry"};
```

# 다차원배열의 포인터

- 2차원 이상의 배열에 대한 포인터
- `char arr[][5] = {"park", "kim", "hong"};`  
`char (*p)[5];`  
//포인터의 선언 형은 가리키는 데이터의 사이즈정보를  
얻기 위함이다.

# 실습

➤ 좋아하는 칼라를 세가지 입력 받아 배열로 저장하고 포인터를 이용해 출력해 보시오

➤ 예시)  
좋아하는 칼라는? red  
좋아하는 칼라는? yellow  
좋아하는 칼라는? black

내가 좋아하는 칼라는 red yellow black

# 함수 매개변수로 포인터 전달

- call by reference 방법
- 매개변수로 값이 아닌 주소를 전달
- 함수에서 포인터를 이용한 연산

```
➤ void add(int *p, int *pSum){  
    *pSum += *p;  
}
```

```
void main(){  
    int num1 = 10; int sum = 0;  
    add(&num1, &sum);  
}
```

# 실습

- 두개의 정수가 주어 졌을 때 두 정수의 값을 바꾸는 함수를 작성하시오  
(함수의 매개변수를 포인터를 이용)

# 실습

- 이름을 입력 받고 화면에 출력하는 프로그램을 작성하시오  
(이름을 입력 받는 함수를 작성)



# main 함수에 인자 전달

➤ 프로그램 실행시에 인자를 전달할 수 있다

➤ 

```
int main(int argc, char * argv[] ){  
    ....  
    return 0;  
}
```

인자의 수

인자의 값

```
D:\data\src\Project1\Debug>Project1 | Like C  
인자의 수 4  
1번째 : Project1  
2번째 : |  
3번째 : Like  
4번째 : C
```