

# C언어

---

LESSON 11.

**포인터의 이해**



# 1. 포인터

# 1.포인터

## 01 정의



포인터란?

➡ 주기억장치의 특정 주소를 이용하는 방법

포인터 변수

특정 주소가 저장된 변수

# 1.포인터

## 02 형식 및 특징

### 선언 형식

자료형 \*포인터변수명;

자료형은 포인터변수가 연결할 변수와 같은  
자료형이어야 함

# 1.포인터

## 03 예 : 포인터 변수 선언 및 이용

```
#include <stdio.h>

void main()
{
    int a;

    int *pa;

    pa = &a;

    *pa = 10;          // a = 10;

    printf("%d \n", *pa); // printf("%d \n", a);
}
```

# 1.포인터

## 03 예 : 포인터 변수 선언 및 이용

```
#include <stdio.h>

void main()
{
    char ch1 = 'A', ch2;

    char *ptr1, *ptr2;

    ptr1 = &ch1;    ptr2 = &ch2;

    *ptr2 = *ptr1;

    printf("ch1 = %c, ch2 = %c \n", *ptr1, *ptr2);
}
```

# 1.포인터

## 04 예 : 포인터 변수의 이용

```
#include <stdio.h>

void main()
{
    int n1 = 10, n2;
    int *ptr1, *ptr2;
    ptr1 = &n1;      ptr2 = &n2;

    (*ptr1)++;
    *ptr2 = 5;
    *ptr1 = *ptr1 + *ptr2;

    printf("n1 = %d, n2 = %d \n", *ptr1, *ptr2);
}
```

# 1.포인터

## 05 예 : 포인터 변수의 연산

```
#include <stdio.h>

void main()
{
    int n = 123;
    int *pn;

    pn = &n;

    printf("pn+1 = %d \n", pn+1);
    printf("pn+2 = %d \n", pn+2);
    printf("pn+3 = %d \n", pn+3);
}
```



# 1.포인터

## 05 예 : 포인터 변수의 연산

```
#include <stdio.h>

void main()
{
    char n = 'A';
    char *pn;

    pn = &n;

    printf("pn+1 = %d \n", pn+1);
    printf("pn+2 = %d \n", pn+2);
    printf("pn+3 = %d \n", pn+3);
}
```

# 1.포인터

## 06 예 : 포인터 변수의 크기

```
#include <stdio.h>

void main()
{
    char a = 'A';      float b = 12.34;
    char *pa;          float *pb;

    pa = &a;           pb = &b;

    printf("문자형 변수의 크기 = %d \\\n", sizeof(a) );
    printf("포인터 문자형 변수의 크기 = %d \\\n", sizeof(pa) );
    printf("실수형 변수의 크기 = %d \\\n", sizeof(b) );
    printf("포인터 실수형 변수의 크기 = %d \\\n", sizeof(pb) );
}
```

# 1.포인터

## 07 포인터와 배열

배열명

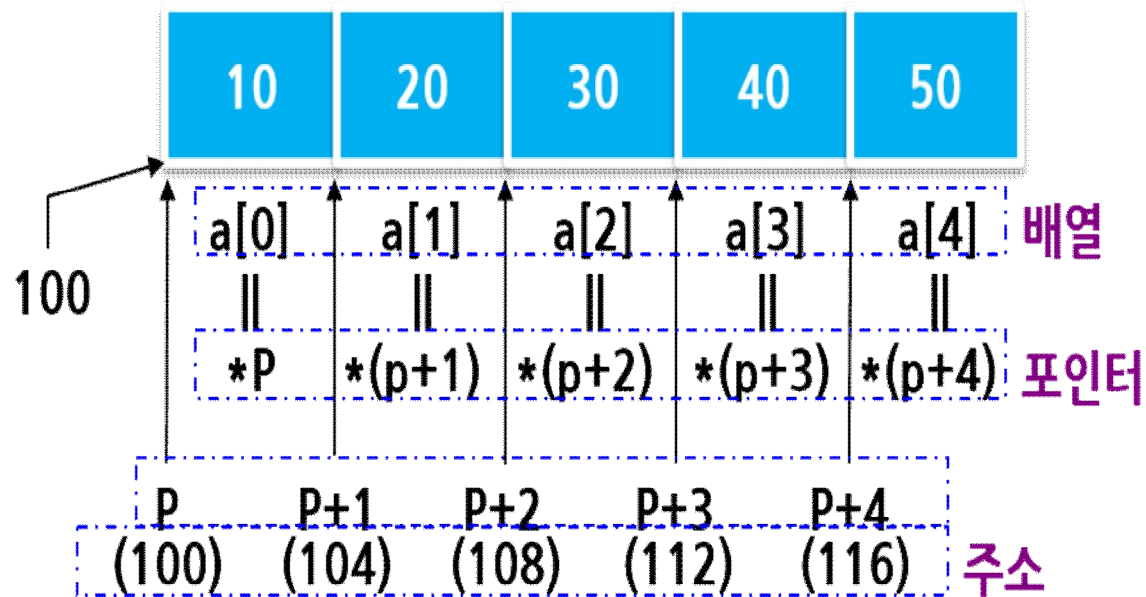


배열의 시작 주소를 의미

- ✓ 포인터의 경우, 배열의 주소를 받아들이므로  
포인터에 배열을 연결하여 사용할 수 있음

# 1.포인터

## 07 포인터와 배열



# 1.포인터

## 08 예 : 배열의 각 인덱스 주소 확인

```
#include <stdio.h>

void main()
{
    int n[5] = {10, 20, 30, 40, 50};
    int i;

    for(i= 0; i<5; i++) {
        printf("배열 %d 인덱스의 주소 = %d \n", i, &n[i]);
    }
}
```

# 1.포인터

## 09 예 : 배열의 각 인덱스와 포인터의 관계

```
#include <stdio.h>

void main()
{
    int n[5] = {10, 20, 30, 40, 50};
    int i;
    int *pn;
    pn = n;
    for(i= 0; i<5; i++) {
        printf("배열 %d 인덱스의 주소 = %d \n", i, &n[i]);
        printf("pn + %d = %d \n", i, pn + i);
        printf("인덱스 %d 값 = %d \n", i, *(pn + i) );
    }
}
```

# 1.포인터

## 10 예 : 문자열과 포인터

```
#include <stdio.h>

void main()
{
    char s[10] = "Seoul";
    int i;

    for(i= 0; i<10; i++) {
        printf("%c \n", *(s + i) );
    }
}
```

# 1.포인터

## 10 예 : 문자열과 포인터

```
#include <stdio.h>

void main()
{
    char s[10] = "Seoul";
    int i;

    for(i= 0; s[i]!=NULL; i++) {
        printf("%c \n", *(s + i) );
    }
}
```



# 1.포인터

## 11 예 : 포인터와 함수 - Call by Value

```
#include <stdio.h>

void ex_func(int sa, int sb) {
    sa += 5; sb += 5;
    printf("값1 : %d, %d \n", sa, sb);
}

void main()
{
    int ma = 100, mb = 200;
    ex_func(ma, mb);
    printf("값2 : %d, %d \n", ma, mb);
}
```

# 1.포인터

## 12 예 : 포인터와 함수 - Call by Reference

```
#include <stdio.h>

void ex_func(int *sa, int *sb) {
    *sa += 5; *sb += 5;
    printf("값1 : %d, %d \n", *sa, *sb);
}

void main()
{
    int ma = 100, mb = 200;
    ex_func(&ma, &mb);
    printf("값2 : %d, %d \n", ma, mb);
}
```

**1. 포인터에 대한  
강의가 끝났습니다.**