

《 C언어: 포인터의 개념과 활용 》

이 문서는 유료용으로 판매되는 강좌의 자료입니다. 무단 복제 및 배포를 금지합니다.

출처: www.youtube.com/@weekendcode , <https://inf.run/zSrvA>

C언어와 포인터

1. 변수가 정수 형태일 때 (int v = 42;)

2. 변수가 문자열일 때 (여러 문자와 '\0')

```
char str[] = "Hello!";
```

3. 변수가 배열일 때 int arr[] = {1, 2, 3, 4, 5}

C언어에서의 메모리 접근 방법

& (앰퍼샌드) : 이 변수의 메모리 상의 주소가 무엇인가요?

* (에스트러스크) : 그 주소가 가리키고 있는 값은 무엇인가요?

```
int answer = 42;
```

```
printf("%d\n", answer);
```

```
printf("%d\n", *answer);
```

```
#include <stdio.h>
```

```
int main() {  
    int n = 42;  
    printf("%d\n", n);  
    printf("%d\n", &n);  
}
```

출력값: 42

100 (저장된 값과 주소는 아무 상관없음. 여기서 100은 임의의 숫자를 적어둔 것)

```
#include <stdio.h>
```

```
int main() {  
    int n = 42;  
    int *p = &n;  
  
    printf("%d\n", p);  
    printf("%d\n", &n);  
}
```

출력값: 100 (주소가 100이라고 가정한 것)

100

문제가 있다. C언어 만든 사람이 헷갈리게 만들어두었다.

```
#include <stdio.h>
```

```
int main() {  
    int n = 42;  
    int *p = &n;  
  
    printf("%d\n", &n);  
    printf("%d\n", p);  
}
```

이게 무슨 소리냐

```
#include <stdio.h>
```

```
int main() {  
    int n = 42;  
    int *p = &n;  
  
    printf("%d\n", &n);  
    printf("%d\n", p);  
    printf("%d\n", *p);  
}
```

주소 출력과 값 출력 나눠보기

int n = 42	값을 출력하고 싶을 때: printf("%d", n); ↔ printf("%d", *p);
int *p = &n;	주소를 출력하고 싶을 때: printf("%d", &n); ↔ printf("%d", p);

주의! 헷갈리기 딱 좋음!!

```
#include <stdio.h>
```

```
int main() {  
    int n = 50;  
    int *p = &n;  
  
    printf("%d\n", &n);  
    printf("%d\n", *p);  
}
```

* (별표)는 상황에 따라 해석이 달라진다.

1. 포인터 변수를 선언할 때 (태어나게 할 때) int *p = &n;
2. 그 주소가 담고 있는 값을 가져오고 싶을 때 (사용할 때) printf("%d", *p);
3. 숫자끼리의 곱할 때 (3 * 4)

문자열과 배열을 포인터로 가리킨다는 것

```
#include <stdio.h>
```

```
int main() {  
    char *str1 = "Hello, World!";  
    char str2[] = "Hello, World!";  
}
```

```
#include <stdio.h>
```

```
int main() {  
    int array[] = {10, 20, 30, 40, 50};  
    int *ptr = array; // 포인터로 배열을 가리킴  
  
    printf("배열의 첫 번째 요소: %d\n", array[0]);  
    printf("포인터가 가리키는 값: %d\n", *ptr);  
  
    if (*ptr == array[0]) {  
        printf("배열의 첫 번째 요소와 포인터가 가리키는 값은 같습니다.\n");  
    } else {  
        printf("배열의 첫 번째 요소와 포인터가 가리키는 값은 다릅니다.\n");  
    }  
}
```

출력값:

배열의 첫 번째 요소: 10

포인터가 가리키는 값: 10

배열의 첫 번째 요소와 포인터가 가리키는 값은 같습니다.

아래 3개는 같은 동작을 하는 코드
모든 코드의 출력값: Hello, World!

```
#include <stdio.h>

int main() {
    char str[] = "Hello, World!";
    char *ptr = str;
    while (*ptr != '\0') {
        printf("%c", *ptr);
        ptr++;
    }
    printf("\n");
    return 0;
}
```

```
#include <stdio.h>

int main() {
    char str[] = "Hello, World!";
    printf("%s\n", str);
    return 0;
}
```

```
int main() {
    char str[] = "Hello, World!";
    int length = 0;
    // 문자열의 길이 계산
    while (str[length] != '\0') {
        length++;
    }
    // for문을 사용하여 인덱스로 순회
    for (int i = 0; i < length; i++) {
        printf("%c", str[i]);
    }
    printf("\n");
    return 0;
}
```

다음 주소로 넘어가는 방법

```
#include <stdio.h>

int main() {
    char *s = "Hello, World!";
    printf("%c", *s);
    printf("%c", *(s+1));
    printf("%c", *(s+2));
    printf("%c", *(s+3));
    return 0;
}
```

아래 2개는 같은 동작을 하는 코드

```
#include <stdio.h>

int main() {
    char *s = "Hello, World!";
    printf("%c", *s);
    printf("%c", *(s+1));
    printf("%c", *(s+2));
    printf("%c", *(s+3));
    return 0;
}
```

```
#include <stdio.h>

int main() {
    char *s = "Hello, World!";
    printf("%c", s[0]);
    printf("%c", s[1]);
    printf("%c", s[2]);
    printf("%c", s[3]);
    return 0;
}
```

배열(Array)은 어떨까?

```
#include <stdio.h>
```

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr = arr;  
    int size = sizeof(arr) / sizeof(arr[0]);  
  
    printf("배열의 요소: ");  
    for (int i = 0; i < size; i++) {  
        printf("%d ", *ptr);  
        ptr++;  
    }  
    printf("\n");  
    return 0;  
}
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main() {  
    char str1[5] = {'H', 'e', 'l', 'l', 'o'};  
    char str2[] = "Hello";  
  
    printf("str1의 길이: %lu\n", sizeof(str1));  
    printf("str2의 길이: %lu\n", sizeof(str2));  
  
    printf("str1의 내용: ");  
    for (int i = 0; i < sizeof(str1); i++) {  
        printf("%c", str1[i]);  
    }  
    printf("\n");  
  
    printf("str2의 내용: %s\n", str2);  
  
    return 0;  
}
```