

Week 06 21.12.04. 07:00 PM

---

+

o

•

# C Programming

Second-half period / CMD-C\_Programming Study

Mentor : Jh9892

# INDEX

- 💧 C Language Personal Concept to Pointer
- 💧 C Language – Structure



# Pointer

*Please explain to Pointer for us :D*

# Pointer

문자열

- char

다음 나

string\_p

```
#inclu
```

```
int ma
```

```
{
```

```
ch
```

```
ch
```

```
pr
```

```
pr
```

```
re
```

```
}
```

실행 결과

```
a
Hello
```

char c1

문자

a

char \*s1

메모리 주소

메모리 주소

문자열

H

e

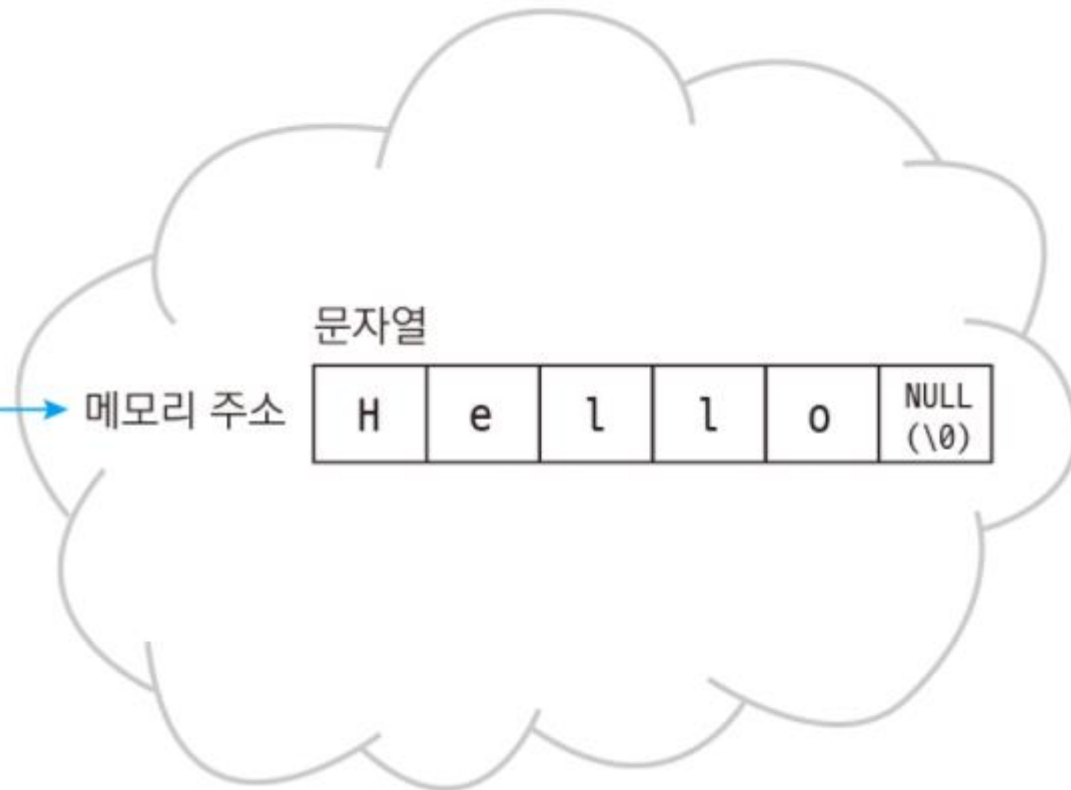
l

l

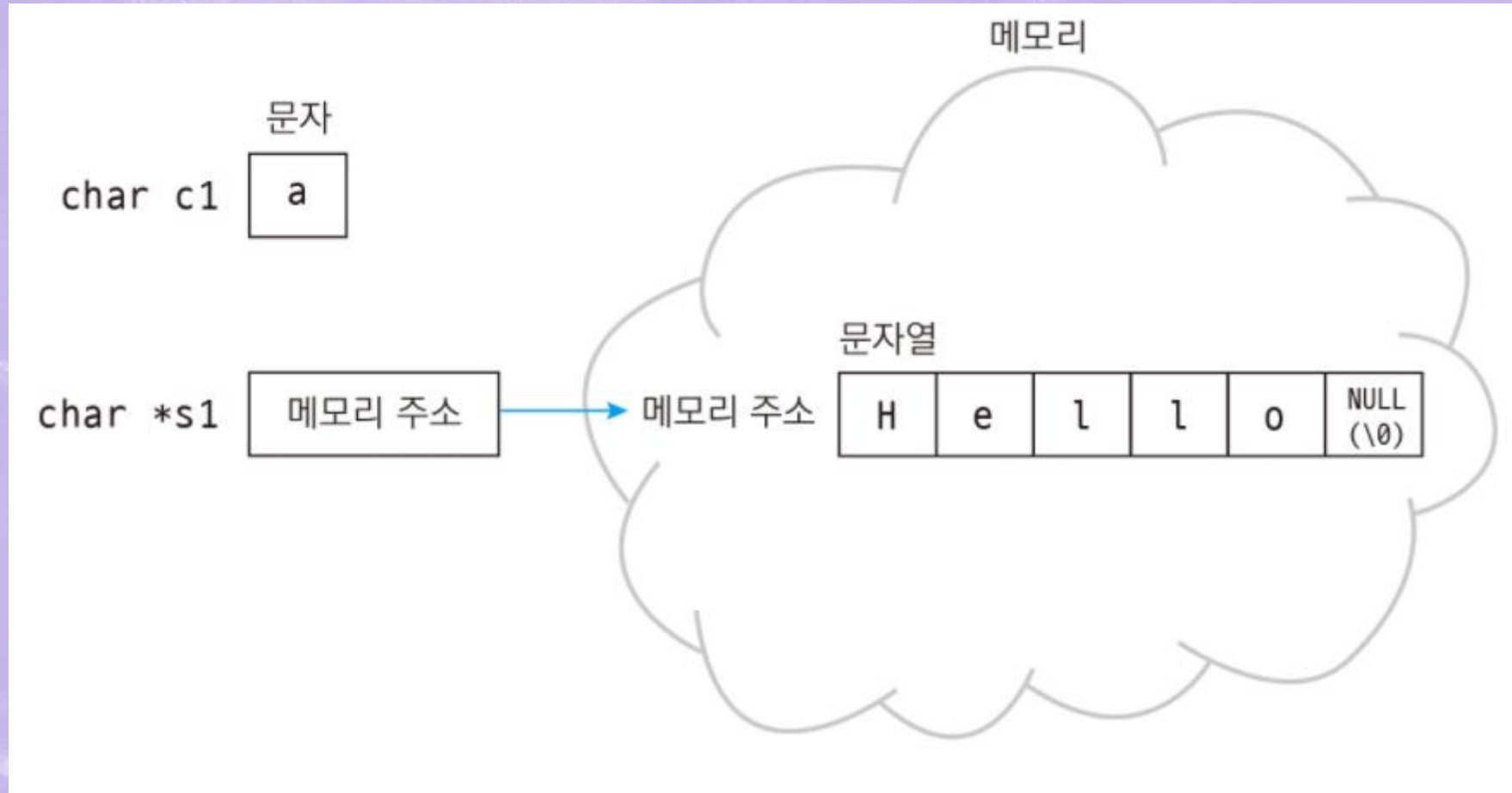
o

NULL  
(\0)

메모리



# Pointer



# Pointer

## string\_pointer\_element.c

```
#include <stdio.h>

int main()
{
    char *s1 = "Hello";    // 포인터에 문자열 Hello의 주소 저장

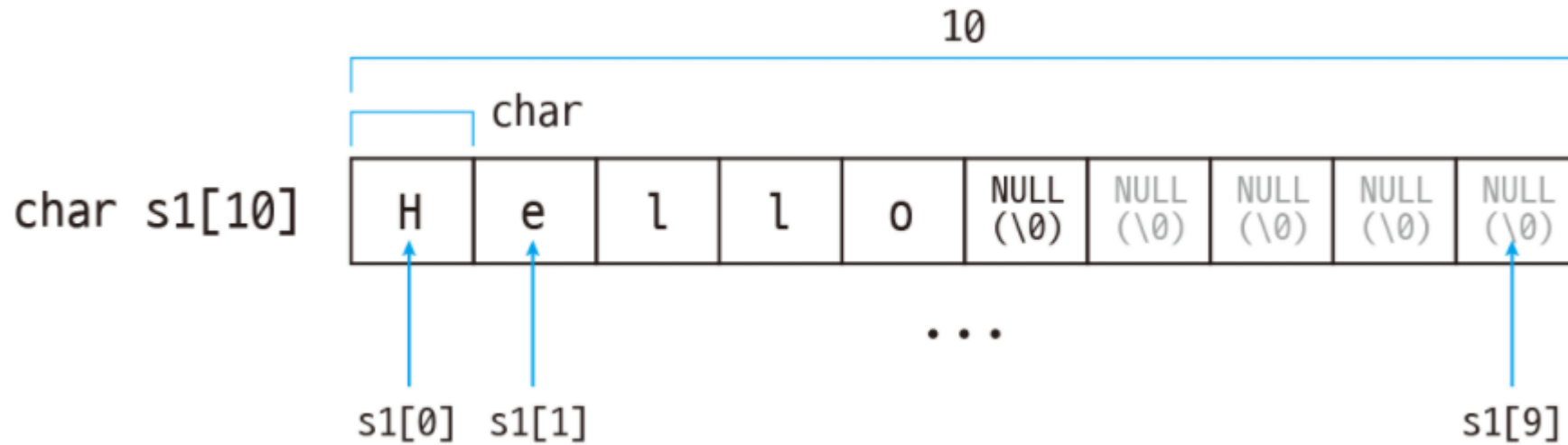
    printf("%c\n", s1[1]);  // e: 인덱스 1(두 번째)의 문자 출력
    printf("%c\n", s1[4]);  // o: 인덱스 4(다섯 번째)의 문자 출력
    printf("%c\n", s1[5]);  // 문자열 맨 뒤의 NULL(\0) 출력. NULL은 화면에 표시되지 않음

    return 0;
}
```

## 실행 결과

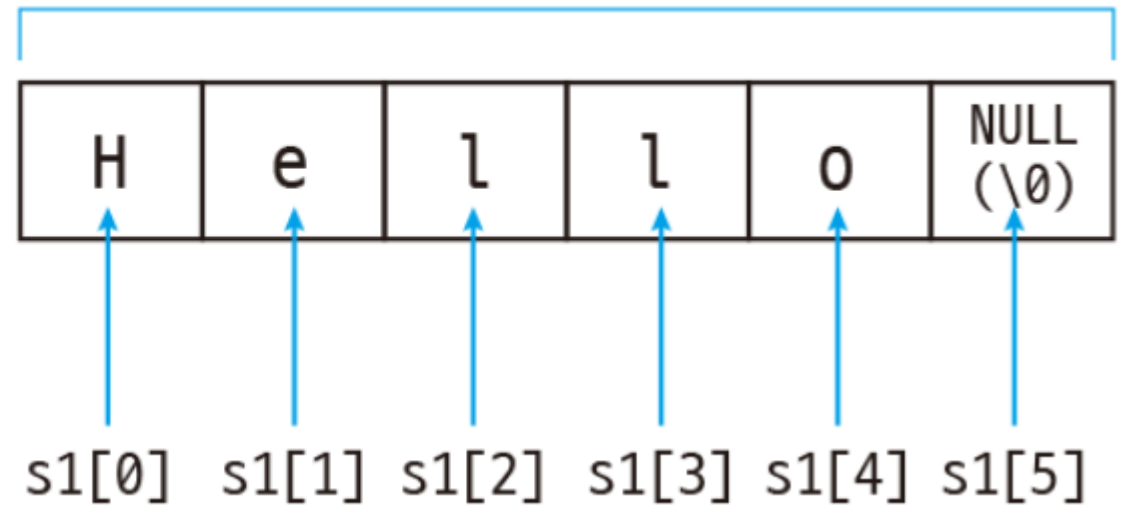
e  
o

# Pointer



6 : Hello 5글자 + NULL 하나

char s1[6]





# Structure

```
struct 구조체이름 {  
    자료형 멤버이름;  
};
```

## typedef\_struct.c

```
#define _CRT_SECURE_NO_WARNINGS    // strcpy 보안 경고로 인한 컴파일 에러 방지  
#include <stdio.h>  
#include <string.h>    // strcpy 함수가 선언된 헤더 파일  
  
typedef struct _Person {    // 구조체 이름은 _Person  
    char name[20];        // 구조체 멤버 1  
    int age;              // 구조체 멤버 2  
    char address[100];    // 구조체 멤버 3  
} Person;                // typedef를 사용하여 구조체 별칭을 Person으로 정의  
  
int main()  
{  
    Person p1;    // 구조체 별칭 Person으로 변수 선언  
  
    // 점으로 구조체 멤버에 접근하여 값 할당  
    strcpy(p1.name, "홍길동");  
    p1.age = 30;  
    strcpy(p1.address, "서울시 용산구 한남동");  
  
    // 점으로 구조체 멤버에 접근하여 값 출력  
    printf("이름: %s\n", p1.name);    // 이름: 홍길동  
    printf("나이: %d\n", p1.age);    // 나이: 30  
    printf("주소: %s\n", p1.address);    // 주소: 서울시 용산구 한남동  
  
    return 0;  
}
```



# Structure

## 참고 | typedef 활용하기

typedef는 자료형의 별칭을 만드는 기능입니다. 따라서 구조체뿐만 아니라 모든 자료형의 별칭을 만들 수 있습니다.

- typedef 자료형 별칭
- typedef 자료형\* 별칭

다음은 int를 별칭 MYINT, int 포인터를 별칭 PMYINT로 정의하는 예제입니다(보통 포인터 별칭은 포인터라는 의미로 앞에 P를 붙임). 별칭으로 변수와 포인터 변수를 선언한다는 점만 다를 뿐 사용 방법은 일반 변수, 포인터와 같습니다.

typedef.c

```
typedef int MYINT;      // int를 별칭 MYINT로 정의
typedef int* PMYINT;    // int 포인터를 별칭 PMYINT로 정의

MYINT num1;            // MYINT로 변수 선언
PMYINT numPtr1;        // PMYINT로 포인터 변수 선언

numPtr1 = &num1;       // 포인터에 변수의 주소 저장
                        // 사용 방법은 일반 변수, 포인터와 같음
```

이처럼 typedef로 정의한 별칭을 사용자 정의 자료형, 사용자 정의 타입이라 부릅니다.

여기서 PMYINT는 안에 \*가 이미 포함되어 있으므로 포인터 변수를 선언할 때 \*를 붙여버리면 이중 포인터가 되므로 사용에 주의해야 합니다.

```
PMYINT *numPtr1;        // PMYINT에는 *가 이미 포함되어 있어서 이중 포인터가 선언됨
int* *numPtr2;          // PMYINT *와 같은 의미. 이중 포인터
```