

# DATA STRUCTURE

## Week 5

### - String 1 -



# INDEX

I. 문자열이란?

II. 문자열 입출력과 초기화

III. Null 문자

# I. 문자열이란?

## 문자 배열 (복습)

원소가 문자인 배열

- 각 배열 원소를 하나의 단위로 처리 : 초기화, 입출력

```
int main() {  
    char str[8] = {'H', 'E', 'L', 'L', 'O'}; // 문자로 초기화  
    int i;  
  
    for (i=0; i<5; i++) {  
        printf("%c", str[i]); // 문자 출력  
    }  
}
```

문자는 하나의 문자로 처리되는 경우도 있지만,

문자들의 묶음을 하나로 처리하는 것이 편리한 경우가 많다.

→ C언어의 문자열

# I. 문자열이란?

문자열 (string) : 연속적으로 나열된 문자들의 묶음

문자열 표현

- 큰 따옴표로 감싸서 나타냄
- ex. “Hello”, “A”, “123”
- 주의 : ‘A’는 문자 (작은 따옴표, 문자열 아님 !)

문자열 저장

- C언어에서 문자열은 문자 배열 형태로 저장

문자열 입출력

- scanf, printf에서 문자열 서식 지원 : %s

# I. 문자열이란?

## 문자열 저장 및 초기화

- C에서 문자열을 활용하는 기본 형태는 문자 배열
- 초기화 : 문자열 표현 방법 (큰 따옴표를 사용) 으로 초기화
- 초기화 예

```
char str[8] = "Hello"; // 배열 크기 지정
```

```
char str[] = "Hello"; // 배열 크기 미지정, 초기화에 의해 크기 결정
```

# I. 문자열이란?

scanf, printf를 사용한 문자열 입출력

- 문자열을 하나의 단위로 취급
- 서식에 %s 사용하고, 인자로 배열의 시작 주소를 전달

```
char str[8];  
  
scanf("%s", str); // 문자열 입력  
  
printf("%s", str); // 문자열 출력
```

scanf에서 배열의 이름은 시작 주소이므로 &를 안 붙여도 무방

# 실습 문제 #1

다음 프로그램을 작성하시오.

- ✓ 크기가 10인 문자 배열 str을 선언
- ✓ 위 배열을 선언과 동시에 문자열 “Hello” 로 초기화
- ✓ 문자열 str을 화면에 출력
- ✓ 사용자로부터 문자열 “World” 를 입력 받아 str에 저장
- ✓ 문자열 str을 화면에 출력

# I. 문자열이란?

널 (Null) 문자

- 문자열의 끝을 의미하는 특수 문자로, '\0' 로 표현
- 널문자의 아스키 코드 값은 정수 0, 즉 '\0' == 0
- 문자열은 항상 맨 마지막에 널 문자를 포함하고 있음 (명시하지는 않지만)
- 문자 'A' 와 문자열 "A" 의 차이

'A'



"A"





# I. 문자열이란?

문자 배열의 크기

- 문자 배열에 문자열을 저장하기 위해서는 **배열의 크기가 문자열의 길이보다 하나 더 커야 함**

```
char str1[6] = "Hello"; // 정상 작동
```

```
char str2[5] = "Hello"; // 에러 유발
```

## 실습 문제 #2

사용자로부터 공백을 포함하지 않는 문자열을 입력 받은 후 소문자만 출력하는 프로그램을 작성하시오.

- 입력 받는 문자열 길이는 최대 20이다.

입력 예시 : HelloWorld

출력 예시 : elloorld

Hint 1 : 아스키 코드의 원리를 생각해 보시오.

Hint 2 : 문자열의 길이를 반환하는 함수 `strlen()`을 활용할 수 있다.

만약 주어진 문자열이 “Hello” 이고 변수명이 `str`라면, `strlen(str)` 은 5이다.

**Thank You**