The background of the slide is a collage of four images. Top-left: Hands holding a tablet. Top-right: A person's hands on a laptop keyboard. Bottom-left: Hands typing on a laptop with a coffee cup and food nearby. Bottom-right: A laptop on a desk with a smartphone and a blue square graphic element.

[4주차 1강] 문자열(1)

학습 내용

📍 10.1 문자열 개요

📍 10.2 문자열 저장 및 기본 입출력 (1)





학습 목표

- 📍 10.1 문자열이란 무엇인지 이해한다.
- 📍 10.2 문자열의 기본 사용법을 익힌다.





1. 문자열 개요
2. 문자열 저장 및 기본 입출력 (1)
3. 문자열과 포인터
4. 문자열의 배열
5. 문자열 및 문자 처리 함수
6. 문자열 및 문자 입출력

1. 문자열 개요



문자열(string): 연속적으로 나열된 문자들의 묶음

- 문자열은 기본적으로 **문자 배열**을 사용하여 저장
 - ✓ 문자 단위로 초기화 하고 출력하는 코드(이미 학습)

```
char str[8] = {'H','e','l','l','o'}; // 문자로 초기화
int i;

for (i=0 ; i<5 ; i++)
    printf("%c", str[i]); // 문자 출력
```

- 이름이나 주소처럼 하나의 묶음으로 처리되는 정보를 매번 문자 단위로 처리하는 것은 매우 불편
 - ➔ C언어에서는 문자들을 **문자열 단위**로 처리할 수 있는 기능 제공

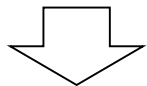


1. 문자열 개요

맛보기 : 앞의 예제 프로그램을 문자열 처리 방식으로 바꾸면?

```
char str[8] = {'H','e','l','l','o'}; // 문자 단위 초기화
int i;

for (i=0 ; i<5 ; i++)
    printf("%c", str[i]); // 문자 단위 출력
```



```
char str[8] = "Hello"; // 문자열로 초기화
printf("%s", str); // 문자열 출력
```



1. 문자열 개요

문자열 표현

- 큰 따옴표로 감싸서 나타냄
- 예) "Hello" , "abc" , "123"
- 비교) 문자는 작은 따옴표로 감싸서 표현 'a' , '1'

문자열 입출력

- scanf, printf에서 문자열 단위 입출력 지원
- 문자열 입출력의 위한 서식 문자: %s
- 인자: 보통 문자열을 저장하는 문자 배열의 이름

```
char str[8] = "Hello"; // 문자열로 초기화  
printf("%s", str);    // 문자열 출력
```



※ 실습하기



[예제 10.1]

다음프로그램을 작성하시오.

- ✓ 크기가 10인 문자 배열 str을 선언과 동시에 문자열 "Hello"로 초기화
- ✓ 문자열 str을 화면에 출력
- ✓ 사용자로부터 문자열 "World"를 입력 받아 str에 저장
- ✓ 문자열 str을 화면에 출력



2. 문자열 저장 및 기본 입출력 (1)

문자열 표현 : 큰 따옴표 사용

- 문자열 예시 : "Hello" , "A" , "123"
- 공백 하나로 구성된 문자열 : " " ⇨ 큰따옴표 사이에 공백
- 큰따옴표 하나로 구성된 문자열 : "\"" ⇨ 큰따옴표 문자 상수 사용 (2.5절 참조)
- 길이가 0인 문자열 : "" ⇨ 큰따옴표 사이에 아무것도 없음

(비교) "A"와 'A' :

- "A" 는 문자열
- 'A' 는 문자
- 자세한 차이점은 잠시 뒤에 설명

2. 문자열 저장 및 기본 입출력 (1)



문자열 저장 및 초기화

- C언어에서는 **문자 배열**에 문자열 저장
- 문자 배열 선언 및 초기화에

```
char str1[8] = "Hello" ;   ⇒ 배열 크기 지정  
char str2[ ] = "Hello" ;   ⇒ 배열 크기 미 지정  
                           (초기화 값에 의해 크기 결정)
```

- 문자열로 초기화하는 것은 선언 시에만 가능

```
char str[20];  
  
str[0] = 'a';           ⇒ 0번 원소에 문자 'a' 대입 (가능)  
str = "Hello";          ⇒ 컴파일 오류 : 배열 초기화는 선언 시에만 가능  
str = {'H','e','l','l','o'}; ⇒ 컴파일 오류 : 위와 동일
```



2. 문자열 저장 및 기본 입출력 (1)

널(null) 문자:

- 문자열의 끝을 의미하는 특수 문자로, '**\0**'으로 표현
- 널문자의 아스키 코드 값은 정수 0, 즉 '**\0**' == 0
- 문자열을 처리하는 기준이 되는 매우 중요한 요소
- 문자열은 항상 맨 마지막에 널 문자를 포함하고 있음 (명시하지 않지만)
 - ✓ 문자 '**A**' 와 문자열 "**A**"의 차이

'A' →

A

"A" →

A	\0
---	-----------



2. 문자열 저장 및 기본 입출력 (1)



널(null) 문자

- 예) 선언문의 초기화

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

||

```
char str[] = {'H','e','l','l','o','\0'};
```

str

H	e	l	l	o	\0
[0]	[1]	[2]	[3]	[4]	[5]

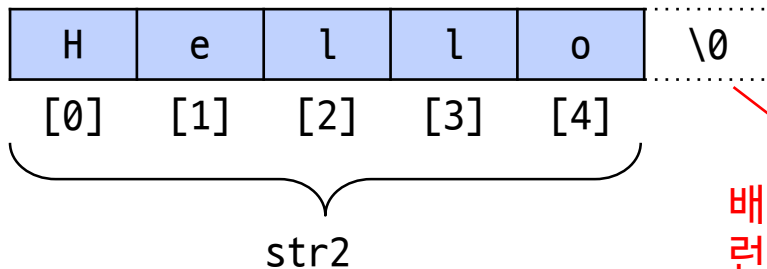


2. 문자열 저장 및 기본 입출력 (1)

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";    // 런타임 오류 유발
```



배열 영역 벗어남
런타임 오류의 원인

2. 문자열 저장 및 기본 입출력 (1)



C언어에서 문자열의 기준

- 널 문자까지의 문자들의 묶음을 지칭
- 주의) 배열의 크기와 관계없음
 - ✓ 배열은 단순히 저장 공간으로서의 역할
- 문자열의 끝은 배열의 크기가 아니라 널 문자에 의해 결정
 - ✓ 입출력을 비롯한 모든 문자열 처리의 기준



2. 문자열 저장 및 기본 입출력 (1)

printf() 함수를 이용한 문자열 출력

- 문자열을 하나의 단위로 취급
- 서식 지정자: %s
- 인자: 문자열의 시작 주소 (보통 문자 배열의 이름)

```
char str[8] = "Hello";    // 문자 배열  
printf("%c", str[2]);     // 문자 출력 (배열 원소 전달)  
printf("%s", str);       // 문자열 출력 (배열 이름 전달)
```



2. 문자열 저장 및 기본 입출력 (1)



출력 시 널 문자의 역할

- 느낌표의 출력 위치에 주목하자

```
char str[20] = "Hello World";  
int i;  
for( i=0 ; i < 20 ; ++i )  
    printf("%c", str[i]);  
printf("!!\n");
```

결과:

Hello World !!

```
char str[20] = "Hello World";  
printf("%s!!\n", str);
```

결과:

Hello World!!

배열의 크기는 20인데, 왜 11자만 출력할까?

배열에서 초기화가 명시되지 않은 원소는 0(즉, '\0')으로 초기화 됨

널 문자는 화면에 공백처럼 출력, but 공백 문자와는 다름



2. 문자열 저장 및 기본 입출력 (1)



출력 시 널 문자의 역할

- 인자로 전달된 주소의 문자부터 널 문자 전까지 출력
 - ✓ 배열 크기만큼 출력하는 것이 아님
 - ✓ printf 함수는 배열의 크기를 모름

```
char str[20] = "Hello World";
int i;
for( i=0 ; i < 20 ; ++i )
    printf("%c", str[i]);
printf("!!\n");
```

결과:

Hello World !!

```
char str[20] = "Hello World";

printf("%s!!\n", str);
printf("%s!!\n", str+5);
```

결과:

Hello World!!
World!!





학습 정리

- **문자열(string)**은 연속적으로 나열된 0개 이상의 문자들의 묶음으로, **큰따옴표**로 감싸서 표현함
- 문자 **'A'**와 문자열 **"A"**는 서로 다름에 주의
- 널 문자 **'\0'**은 문자열의 끝을 의미하는 특수문자로, 아스키 코드 값은 0임
- 문자열은 기본적으로 **문자 배열에 저장**되고, 배열의 크기는 널 문자로 인해 실제 저장할 문자의 개수보다 하나 더 크게 지정해야 함

