

프로그래밍	11. 문자열의 사용	일자: 2024. . .
학습 목표	<input type="checkbox"/> 문자와 문자형 배열, 문자열 개념 학습 <input type="checkbox"/> 문자열의 사용	하이다:

1. 문자와 문자형 배열, 문자열

1) 문자(Character)

- ① ASCII 코드 방식으로 컴퓨터의 특정 숫자에 문자를 대응시켜 나타낸 것이다.
 - 예를 들어, 'A'(65), 'a'(97), '0'(48) 등이 있다.

2) 문자형 배열(Character Array)

- ① 문자들을 여러 개 모아둔 집합, 다시 말해 자료형이 char인 1차원 배열이다.
 - 예를 들어, { 'A', 'a', '0' }, { '+', '-', '*', '/' } 등이 있다.

3) 문자열(String)

- ① 단순히 문자들의 모임이 아니라, 그 자체로 하나의 시퀀스, 문장을 이루는 문자 배열이다.
 - 예를 들어, "Aa0", "+-*/", "Hello" 등이 있다.
- ② 문자형 배열과는 다르게 마지막 배열 요소로 NULL 문자('\0')가 들어간다.
 - 위의 문자열 예들을 배열의 요소로 쪼개면, { _____ }, { _____ }, { _____ }로 볼 수 있다.
- ③ 문자열을 준말로 생각하여, 그 본말이 문자형 배열이라 말하는 경우도 종종 있지만 이는 엄밀히 보면 틀린 말이다. 문자열은 문장을 구성할 요소()을 가진 문자형 배열만을 뜻한다.

2. 문자열의 사용

1) 문자열의 선언

```
char1) str2)[5]3);
```

- ① 타입: 문자열은 항상 문자형 배열이므로, 배열 타입은 ____형으로 명시한다.
- ② 배열 이름: 배열을 구분하고 접근하기 위해 사용된다.
- ③ 배열의 크기(길이): 해당 배열이 몇 개의 엘리먼트를 가지게 될지 명시한다. 단, 문자열은 항상 널 문자('\0')를 포함해야 하므로 일반적으로 읽는 글자 수보다 +1 해야 한다는 것을 명심하라!

2) 문자열의 초기화 : 선언과 동시에 초기화

```
char str1[10] = { 'H', 'e', 'l', 'l', 'o', '\0' }1);  
char str2[6] = "Hello"2);  
char str3[]3) = "Hi!"; // 배열 크기는 4로 자동 설정된다 [3 + 1(null)]
```

- ① 초기화 리스트: 중괄호를 사용하여 초깃값을 순서대로 나열한다. 문자열의 경우, 배열의 맨 마지막 요소로 '\0'이 들어가는 것을 잊지 마라.
 - 초기화 리스트의 크기가 인덱스 크기보다 작아, 값으로 초기화되지 않은 요소들은 전부 정수 _으로 자동 초기화된다.
 - 이때 정수 _에 대응되는 ASCII 코드값은 무엇인가? 바로 __이다!
- ② 문자열 상수: 큰따옴표로 묶이는 문자열 상수로 배열을 초기화하는 경우이다.
 - 문자열 상수는 당연히 '\0'을 포함한다. 예를 들어 "A"는 { 'A', '\0' }이다.
- ③ 선언과 동시에 배열을 초기화하는 경우, 배열 크기를 명시하지 않아도 된다. 이때 배열 크기는 초기화되는 문자열의 크기로 자동 설정된다.

3) 문자열의 초기화 : 개별 요소들을 초기화

```
char str[3];
str[0] = 'H', str[1] = 'i';
str[2] = '\0'; // str[2] = 0;
```

- ① 배열의 각 인덱스에 직접 접속해서 값을 대입하면 된다.
- ② 계속해서 강조하지만 널 문자('\0')를 마지막에 대입하는 것을 잊지 마라.

4) 문자열의 사용

```
char str1[5];
scanf("%s", str1); // "GOOD" 입력
printf("%s", str1);
```

```
char str2[11] = "안녕하세요"; // 한글은 2Bytes 크기
for (int i = 0; str2[i] != '\0'; i++)3)
    printf("%c", str2[i]);4)
```

	char	str1[5]
str1 == str1[0] →	0x10	'G'
str1[1] →	0x11	'O'
str1[2] →	0x12	'O'
str1[3] →	0x13	'D'
str1[4] →	0x14	'\0'

- ① 문자열은 __ 형식지정자를 이용해 입출력할 수 있다.
- ② 입력할 때는 주소 연산자(&)를 사용하지 않는다. **배열의 이름 자체가 문자열의 시작 주소를 가리키고 있기 때문이다.**
- ③ 문자열의 처음부터 끝까지를 사용하는 반복문은 0번 인덱스부터 널 값을 가지는 인덱스 전까지 반복함으로써 쉽게 구현할 수 있다.
- ④ 문자열은 결국 문자형 배열이므로, 배열 요소를 하나하나 뽑아 문자로 사용할 수도 있다.

3. 실습

1) LAB11_1 (문자열)

최대 길이 50의 문자열을 입력받아, 입력된 문자열의 길이를 구하는 프로그램을 작성하라.

힌트는 아래와 같다.

- 문자열의 맨 마지막 요소에는 반드시 '\0'이 들어가 있다.

```
문자열을 입력(<=50): qwerty
길이는 6
```

```
문자열을 입력(<=50): 123456789
길이는 9
```

2) LAB11_2 (문자열)

최대 길이 50의 문자열을 입력받아, 입력된 문자열에서 대문자만을 뽑아 새로운 문자열을 만들어 출력하라.

```
문자열을 입력(<=50): ComputerScience
CS
```

3) LAB11_3 (문자열, 배열)

주어진 문자열(≤ 50)에서 대소문자에 관계없이 영자 모음이 몇 번씩 등장하였는가를 출력하라. if문은 사용하지 않는다.

힌트는 아래와 같다.

- 알파벳에서 모음은 'a', 'e', 'i', 'o', 'u'가 있다.

```
문자열을 입력(<=50): WeAreShiningLittleStar
a or A: 2 개
e or E: 3 개
i or I: 3 개
o or O: 0 개
u or U: 0 개
```

```
문자열을 입력(<=50): WhereverYouAre
a or A: 1 개
e or E: 4 개
i or I: 0 개
o or O: 1 개
u or U: 1 개
```

4) LAB11_4 (문자열, 배열)

주어진 문자열(≤ 50)에서 대소문자에 관계없이 각 영문자가 몇 번 등장했는지 체크하는 프로그램을 구현하라.

힌트는 아래와 같다.

- 알파벳의 개수는 26개이다.

```
문자열을 입력(<=50): LetMeGo
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 0 0 0 2 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0
```

```
문자열을 입력(<=50): ABBCCDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 2 3 4 31 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

5) LAB11_5 (문자열)

문자열(≤ 50)과 찾을 문자(key)를 입력받아, 입력된 문자열에서 가장 처음으로 key가 등장하는 인덱스를 찾아 출력하라. key가 문자열에 없다면 -1을 출력한다.

```
KEY를 입력 : L
문자열을 입력(<=50): appleAndLemon
3
```

```
KEY를 입력 : Z
문자열을 입력(<=50): abcdefg
-1
```

6) LAB11_6 (문자열, 배열)

문자열(≤ 50)을 입력받고, 대소문자에 관계없이 각 영문자가 가장 처음으로 입력된 문자열에서 등장하는 인덱스를 출력하는 프로그램을 구현하라. 등장하지 않은 경우에는 -1을 출력한다.

```
문자열을 입력(<=50): AbcDE12345
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

```
문자열을 입력(<=50): 01Z3g5Yu
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
-1 -1 -1 -1 -1 -1 4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 7 -1 -1 -1 6 2
```

```
문자열을 입력(<=50): ZyWXvUtSRqpOnMLkJihGFedcBAz
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 2 3 1 0
```