

4.5

배열과 문자열

개요

변수는 값을 저장하기 위해 쓰인다는 것을 기억하시나요? 100명의 시험 점수나 1000명의 주소와 같이 여러 값들을 저장하기 위해 여러 개의 변수를 써야 하는 상황이 발생할 수 있습니다. 이러한 경우에 사용할 수 있도록 C에서는 **배열**이라는 자료 구조를 제공합니다. 배열은 같은 자료형으로 된 여러 개의 변수를 연속적으로 저장합니다. 예를 들어 정수 배열은 여러 개의 정수 값을 연속적으로 저장하고 문자배열(**문자열**)은 문자 값을 연속적으로 저장합니다.

핵심개념

- * 배열
- * 문자열
- * 배열의 크기
- * 인덱스
- * 널 종단(w0)

배열

```

1 int ages[5];
2 ages[0] = 28;
3 ages[1] = 15;
4 ages[2] = ages[1];
5 ages[3] = ages[1]-1;
6 ages[4] = 17;

7 for(int i = 0; i < 5; i++)
8 {
9     ages[i] += 1;
10 }
```

The diagram illustrates the state of the `ages` array at different stages of code execution:

- Initial state:** An array of 5 empty slots indexed 0 to 4.
- After line 2:** The value 28 is stored at index 0.
- After lines 3-6:** The array contains [28, 15, 15, 14, 17].
- After line 7-10 (loop):** The array contains [29, 16, 16, 15, 18].

◀ <코드 1>

변수와 마찬가지로, 배열을 선언할 때는 첫째로 배열에 저장되는 자료의 유형을 명시합니다. 그리고 배열의 이름을 지정합니다. 이름 뒤에 오는 대괄호 안에는 배열의 크기가 들어갑니다. **배열의 크기**는 배열에 얼마나 많은 값을 넣어둘 수 있는지를 정의합니다. 예를 들어 <코드 1>의 1행은 다섯 개의 정수 배열을 선언합니다.

배열을 일련의 연속된 칸으로 생각할 수 있는데, 각 칸에는 값이 있고, 숫자 **인덱스**가 있습니다. 숫자 인덱스는 배열 안에 들어있는 특정 값에 접근하기 위해 쓰이며, **몇 번째 값인지를** 가리킵니다. C에서 배열의 인덱스는 **0으로 시작**하므로 배열의 첫 번째 항목은 인덱스 0이고, 두 번째 항목은 인덱스 1입니다.

배열의 특정 값에 접근할 때는 배열의 이름과 괄호 안에 들어가는 특정 인덱스를 사용합니다. 2행은 `ages` 배열의 첫 번째 값(인덱스 0)을 28로 정해주었습니다.

배열 인덱스의 값은 일반 변수와 같이 취급됩니다. 예를 들어, 3-6행 처럼 값을 바꾸고, 연산이나 할당 연산자를 적용할 수 있습니다.

배열의 각 값은 인덱스 숫자로 참조하기 때문에 배열을 반복문으로 돌리기 쉽습니다. 7-10행은 for문인데 배열 전체를 순환하면서 각 나이의 값을 1만큼 올려줍니다.

문자열

```
string s = "cs50";
```

0	1	2	3	4
'C'	'S'	'5'	'0'	'\0'

(= \0)

▲ <코드 2>

C에서 **문자열**은 char 값들의 배열을 나타냅니다. 그러므로 <코드 2>의 예문과 같이 `string s = "CS50";` 라고 썼을 때 이 정보는 char들의 배열로 이루어진 것이고, 각 문자당 하나의 인덱스에 저장됩니다. C에서 **문자열의 마지막 인덱스는 널로 끝나며, '\0'**으로 나타냅니다. **널 종단(null-terminator)**은 문자열에게 문자열이 끝났고 더 이상의 문자가 남아있지 않다고 말하는 문자입니다.

문자열은 하나의 배열이기 때문에, 다른 배열에서 인덱스를 이용해 각 원소에 접근하는 것과 마찬가지로 문자열의 인덱스를 통해 특정 문자 값에 접근할 수 있습니다. 예를 들어, <코드 2>에서, `s[0]`를 부르면 "CS50"의 첫 문자인 'C'를 전달해줄 것입니다.

배열을 사용하므로 문자열에 반복문을 돌려 각 문자에 연산을 하기가 쉽습니다. 보통 반복문의 카운터는 0으로 초기화하여 시작되고, 문자열의 마지막 인덱스까지 반복합니다.