



배열

배열: 같은 형태의 데이터를 메모리에 연속적으로 저장하는 것

선언

자료형 요소 개수
↓ ↓
int arr[5];
 ↑
 배열명

⇒ int형 변수 5개를 선언하는 것과 전체 공간의 크기는 같다 (20 Bytes)

사용

★ 배열을 선언할 때 대괄호 속 숫자는 배열 요소를 사용할 때의 숫자와 의미가 다르다.

선언 : int arr[5];
 ↳ 배열 요소 개수 5개

사용 : arr[0] = 10;
 ↳ 첫번째 배열 요소

초기화

- case 1: int arr[5] = {1, 2, 3, 4, 5};
case 2: int arr[5] = {1, 2, 3};
 ⇒ 이 경우, arr[3]과 [4]는 0으로 채워질
case 3: int arr[1000] = {0};
 ⇒ 모든 요소를 0으로 초기화
case 4: int arr[] = {1, 2, 3};
 ⇒ 초기값 개수만큼만 공간이 할당될

★ 배열 초기화는 선언 시 최초 한번만 가능

sizeof와 배열

sizeof 연산자를 사용해 요소 개수를 확인할 수 있다.

$$\text{배열요소개수} = \text{sizeof}(\text{배열명}) / \text{sizeof}(\text{배열평}[0])$$

문자를 저장하는 배열

문자열을 저장할 때는 주로 char형 배열이 사용된다.

선언

★ 문자열의 길이보다 크게 배열을 선언해야 한다! ★

↳ why? Null문자(\0)를 지정해야 하기 때문

Null문자

↳ 아스키 코드 값이 0인 문자, 문자 생성은 \0
문자열의 끝을 표시하기 위해 사용된다.

ex) char ary[6] = "apple"

ary	a	p	p	l	e	\0
-----	---	---	---	---	---	----

scanf로 문자열을 입력 받으면 Null문자가 마지막에 추가된다.

대입

대입연산자의 왼쪽에는 배열명이 올 수 없다.

↳ why? 배열명은 컴파일 시 주소 값으로 바뀌기 때문

ex) str = "apple" // Error

Then. 문자열은 어떻게 대입하는가? → strcpy 함수 사용

strcpy()

↳ char형 배열에 새로운 문자를 저장하는 함수

strcpy(저장될 배열명, 저장할 문자열)

ex) strcpy(str, "apple");
strcpy(str1, str2);

문자열 전용 함수

gets()

↳ 빈칸(공백)을 포함하여 문자열을 입력받는 함수

ex) gets(str);
↳ 인수로 char형 배열명

puts()

↳ 문자열을 출력하는 함수

printf와 무엇이 다른가?

↳ puts()는 문자열 출력 후 자동으로 줄바꿈 실행