**1. 배열**

* **배열(array)**
* **동일한 자료형**을 여럿 담을 수 있는 자료 구조
* 배열 안에 있는 각 **데이터를 요소(element)**라고 부름
* 몇 개의 데이터를 담을지 결정한 뒤에는 그 수(길이)를 바꿀 수 없음  
  - 단 내용 변경은 가능
* 반복문과 함께 사용하면 매우 훌륭한 자료 구조가 됨
* **배열의 선언**
* <자료형>[]: 자료형에 대한 배열을 담음을 의미
* **배열의 선언과 동시에 초기화**

텍스트이(가) 표시된 사진

자동 생성된 설명

* **요소에 접근**
* [] 안에 접근(access)하고자 하는 데이터의 색인(index)을 넣음
* []를 첨자(subscript) 연산자라고 함
* 데이터가 들어가 있는 순서  
  - 제일 앞(왼쪽 끝)에서부터 셈  
  - 메모리에 들어가 있는 순서와 동일함 (즉, 데이터가 선형적으로 메모리를 구성함)
* 접근한 후에는 그 요소를 변수와 똑같이 사용 가능
* **배열의 색인**
* 대부분의 프로그래밍 언어에서 배열의 색인은 0부터 시작함  
  - 즉, 첫 번째 요소는 0번째로 시작함
* 이유: 메모리의 구조  
  - 한 배열의 모든 요소는 메모리에서 연속적으로 연결되어 있음  
  - 첫 번째 요소(시작)으로부터 **오프셋(offset) 0**만큼 떨어져 있음
* **char 배열과 string**
* 문자열(string)은 여러 개의 문자가 줄줄이 모인 집합체
* char 배열은 여러 개의 문자 데이터를 담은 배열
* char 배열과 string은 **기본적으로 같은 데이터 방식**  
  - 그래서 일부 언어에서는 문자열을 지원 안 함  
  - 그냥 char 배열을 이용하면 되기 때문임
* 문자열은 그저 char 배열보다 좀 더 많은 기능을 가짐
* 배열의 문제점(고정 길이)을 해결
* **char 배열은 반복문과 함께 프로그래머 면접 문제로 매우 자주 나옴**