

# Array

---

배열은 메모리에 연속적으로 나열되는 좋은 특성을 가지고 있어서 구체적인 데이터의 메모리 레이아웃을 준비하는데 유용하게 사용된다. 하지만 주로 슬라이스의 내부 데이터 구조로 사용되는 경우가 대부분이다.

Go와 C의 배열이 다른 점

- 배열은 **값**이다. 어떤 배열을 다른 배열 타입의 변수에 할당하면 모든 요소들이 **복사**된다.
- 특히 함수에 배열을 전달하면 포인터가 아닌 복사된 배열이 전달된다.
- 배열의 크기는 타입 정보의 일부이다. 즉 `[10]int`와 `[20]int`는 서로 다른 배열 타입이다.

## 선언, 초기화, 그리고 반복처리

### ArrayBasic

## 배열의 크기도 타입 정보의 일부

2개의 배열이 있을때 서로 타입이 같다고 말할 수 있으려면 즉, 배열의 타입 아이덴티티는 배열요소의 타입이 같고 배열 요소의 숫자도 같아야 한다.

### ArrayTypeIdentity

## 배열 요소들은 메모리에 연속적으로 할당

예를 들어 `int`를 요소 타입으로 하는 배열은 64bit 주소를 갖는 아키텍처에서 8 바이트마다 요소를 가리키는 값이 할당된다.

- $8 \text{ (bits per byte)} \times 8 \text{ bytes} = 64 \text{ bit}$

`string`이 요소 타입인 경우, `string`은 2 워드 데이터 구조이므로 64bit 주소를 갖는 아키텍처에서는 16 바이트 마다 값이 할당된다.

### ArrayContiguousAllocation

## range를 사용한 배열 요소의 반복처리

- range를 사용해 얻은 인덱스를 통해 요소를 접근할 때는 포인터의 의미로 range를 사용하는 것
- range를 사용해 얻은 값으로 요소를 접근할 때는 값의 의미로 range를 사용하는 것. 값의 의미라는 말은 요소의 실제 값이 range의 짧은 선언문에서 복사되어 사용되는 것을 의미한다.

### ArrayRangeMechanics

## 참고자료

- Effective Go - Arrays