

## 배열 Array

- 배열에는 같은 종류의 데이터들이 순차적으로 저장되어, 값의 번호가 곧 배열의 시작점으로 부터 값이 저장되어 있는 상대적인 위치가 된다.

### Array 문법

```
//Array 문법
NSInteger intList[5] = {1,2,3,4,5};
NSLog(@"%ld", intList[3]);

CGFloat floatList[10] = {3.1, 3.2, 3.3};
NSLog(@"%lf", floatList[2]);

char name[8] = "geonhui\0";
NSLog(@"%s", name);
return 0;
```

## 링크드 리스트

- 링크드 리스트는 각 노드가 데이터와 포인터를 가지고 한 줄로 연결되어 있는 방식으로 데이터를 저장하는 자료 구조이다, 이름에서 말하듯이 데이터를 담고 있는 노드들이 연결되어 있는데, 노드의 포인터가 다음이나 이전의 노드와의 연결을 담당하게 된다.

### 링크드 리스트의 종류

- Node
  - 링크드리스트는 노드들의 집합
  - 노드의 위치는 메모리상 어디에도 있을 수 있다.
  - 다음 노드로 이동하기 위해서는 노드가 가지고 있는 포인터를 이용한다.
  - 데이터와 포인터를 가지고 있다.
  - 어디에 있는지는 링크를 통해 알수있다.

Data	Link
------	------

### 단일(싱글) 연결 리스트의 특징

- 가장 단순한 구조로 되어있다.
- 다음 노드를 찾기 쉽지만, 선행 노드(이전노드)를 찾는것은 어렵다
- 노드 탐색시 head pointer에서 부터 시작해야만 한다.

### 이중(더블) 연결 리스트의 특징

- 시작이 정해지지 않았다
- head를 내 마음대로 변경할 수 있다.
- 마지막에 add할 경우 단일 연결에 비해 쉽다.

### 원형(서클) 연결 리스트의 특징

- 노드가 양방향의 링크를 모두 가지고 있으므로 탐색이 다른 리스트에 비해 용이하다.
- 다른 리스트에 비해 코드가 복잡하고, 메모리 공간을 많이 차지한다.

## Dictionary

### Associating Data With Arbitrary Keys(임이의 키와 데이터의 연결) : Dictionaries

- Hash Table를 사용한 데이터 구조인 Dictionary는 **순서가 보장되지 않는** 컬렉션으로 key값으로 데이터에 access 가능하다.
- 정적 데이터를 저장하는 NSDictionary와 동적 데이터 저장이 가능한 NSMutableDictionary Class를 사용할수있다.
- 대괄호@{key:value}를 리터널 문법으로 사용하며 콤마로 데이터를 구분해준다.(ex : @{@"**name**":@"too",@"age":@"20"})
- key는 한 Dictionary에서 **유니크**한 값을 가진다.
- key의 값은 어떤 객체도 가능하나 일반적으로 string으로 사용한다.

### 해쉬맵

- 키를 값에 매핑할 수 있는 구조인, 연관 배열 추가에 사용되는 자료 구조이다. 해시 테이블은 해시 함수를 사용하여 색인(index)를 버킷(bucket)이다 슬롯(slot)의 배열로 계산한다.