

# SWIFT

## Data Structure Introduction

Bill Kim(김정훈) | [ibillkim@gmail.com](mailto:ibillkim@gmail.com)

# 목차

Introduction

Data Structure

Purpose

Classification

Applications

References

# Introduction

자료(Data) :

수집한 사실이나 개념의 값 또는 이들의 집합, 흔히 가공되지 않는 형태의 데이터를 자료라고 합니다.

정보(Information) :

특정 용도로 사용하기 위하여 자료를 의미 있게 처리 및 가공한 형태의 데이터

# Data Structure

자료구조(data structure)란 자료(data)에 대한 처리를 효율적으로 수행 및 사용할 수 있도록 자료를 자료의 특성에 따라서 조직적, 체계적으로 만들어서 저장하여 자료들의 집합을 표현한 것을 말합니다.



# Introduction

Swift에서의 데이터 타입(Data Type)은 값의 전달 방식에 따라서 아래와 같이 분류됩니다.

## Value Type

- 변수 또는 상수에 할당되었을 때 값을 복사하여 전달
- 오직 하나의 소유 객체를 가짐
- 구조체와 열거형이 해당 타입임
- 모든 스위프트의 기초적인 타입은 구조체에 의해 구현

## Reference Type

- 할당될 때 복사되지 않고 공유
- 즉 복사가 아닌 인스턴의 참조값을 전달
- 즉 메모리 주소를 전달하는 객체
- 여러 소유 객체를 가질 수 있음
- 클래스 및 클로저 등

# Introduction

Swift에서의 데이터 타입(Data Type)은 **형태**에 따라서도 아래와 같이 분류될 수 있습니다.

## Named Types

- 이름이 있는 타입
- 클래스, 구조체, 열거형, 프로토콜 등이 포함
- 배열, 딕셔너리, 집합, 옵셔널 값을 포함
- Extension을 선언하여 행동을 확장할 수 있음

## Compound Types

- 복합 타입, 이름이 없는 타입
- 함수 타입
- 타입의 타입 정의(튜플) (ex> (Int, (Float, Float))

# Purpose

그렇다면 자료구조가 왜 중요할까요?

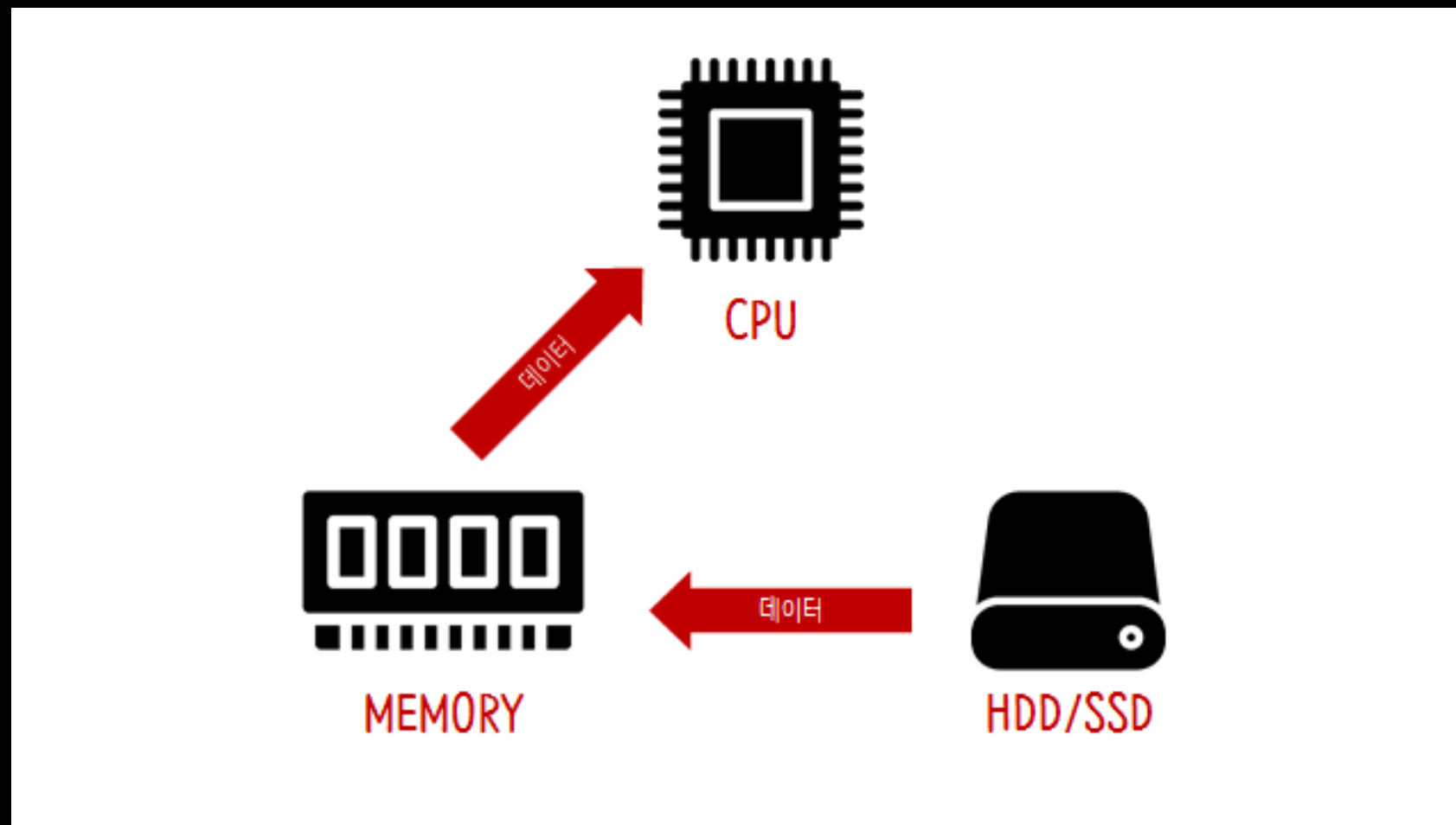
자료구조의 중요성을 알기 전에 우선 우리는 컴퓨터에서 가장 중요한 아래의 요소에 대해서 알 필요가 있습니다.

**CPU** : 중앙정보처리장치, 레지스터, 데이터를 연산/처리하는 가장 중요한 부분, 아주 빠름

**메모리** : RAM(Random Access Memory), CPU에서 직접 접근이 가능, 사용자가 자유롭게 내용을 읽고 쓰고 지울 수 있습니다. 휘발성 기억장치, 빠름

**기억장치** : HDD, SSD등의 저장 장치, 비휘발성, 용량이 아주 크지만 느림, 데이터를 저장

# Purpose





# Purpose

자료구조의 목적은 아래와 같습니다.

1. 데이터를 효율적으로 저장 및 관리
2. 데이터를 찾고 실행(처리)하는 시간을 단축
3. 효율적인 메모리 사용 및 저장

적절한 자료구조를 선택하여 자료(데이터)를 효율적으로 처리하여 성능과 안정성이 좋은 프로그램으로 만들 수 있습니다.

# Purpose

자료구조는 자료의 처리를 보다 효율적으로 하기 위해서 아래와 같은 사항을 고려하여 선택 및 사용해야 합니다.

- 자료의 처리 시간
- 자료의 크기(양)
- 자료의 활용 빈도(횟수)
- 자료의 갱신 정도
- 프로그램의 용이성
- 알고리즘의 복잡도

# Purpose

자료구조는 크게 아래와 같은 3가지의 특징을 가지고 있으며 적절하게 상황에 맞게 사용하여야 합니다.

1. 효율성
2. 추상화
3. 재사용성

# Purpose

## 효율성

적절한 자료구조를 선택하여 사용한다면 프로그램의 효율이 올라갑니다.

한가지 예를 들어보자면 검색에 대한 알고리즘을 구현할때, 데이터의 양이 많다면 순차 검색(Linear Search)를 사용하는 것 보다 이분 검색(Binary Search)를 활용하는것이 더 효율 적일 것입니다.

왜냐하면 학생이라는 테이블에 학생에 대한 데이터가 100만개 있다고 할때, 순차 검색으로 데이터를 검색하게 되면 운이 좋을때는 1번의 연산으로 찾을 수 있겠지만, 운이 없을 경우에는 100만번의 연산을 거쳐야 할 것입니다. 이에 반해 이분 검색은 연산의 횟수가 훨씬 줄어듭니다.

이와같이 목적에 맞는 자료구조를 사용하는것이 효율적입니다.

# Purpose

## 추상화

추상화란 복잡한 자료, 모듈, 시스템 등으로 부터 핵심적인 개념만 간추려 내는 것입니다.

자료구조를 구현할 때 중요한 것은 어느 시점에 데이터를 삽입할 것이며, 어느 시점에 이러한 데이터를 어떻게 사용할것인지에 대해서 초점을 맞출수 있기 때문에 구현 외적인 부분에 더 시간을 쏟을 수 있습니다.

알고리즘 및 자료구조 내부의 구현은 중요하지 않습니다.  
어떻게 구현했는지 보다 어떻게 사용해야 하는지를 알고 있어야 합니다.

예를들어 스택(Stack)의 경우 먼저 들어간것이 나중에 나오는 FILO(First In Last Out)의 형태를 가지고 있습니다. 그리고 push() 함수를 이용해서 데이터를 삽입할 수 있고, pop() 함수를 이용해서 데이터를 추출할 수 있습니다.

그 함수 내부 구현이 어떻게 되었는지는 중요하지 않습니다. 사람마다 다른 코드를 작성할 것이고, 사용 언어, 개발 툴등 환경적인 변수에 의해서 다른 코드가 나올 것이기 때문에 추상적인 개념에 대해서만 이해하고 있다면 사용할 수 있습니다.

# Purpose

## 재사용성

자료구조를 설계할때 특정 프로그램에서만 동작하게 설계하지는 않습니다.

다양한 프로그램에서 동작할 수 있도록 범용성 있게 설계하기 때문에 해당 프로젝트가 아닌 다른 프로젝트에서도 사용할 수 있습니다.

# Classification

자료구조는 크게 아래와 같이 분류할 수 있습니다.

1. **단순구조** : 프로그래밍에서 사용되는 기본 데이터 타입
2. **선형구조** : 저장되는 자료의 전후관계가 1:1 (리스트, 스택, 큐 등)인 구조
3. **비선형구조** : 데이터 항목 사이의 관계가 1:n 또는 n:m (트리, 그래프 등) 형태의 구조
4. **파일구조** : 서로 관련된 필드들로 구성된 레코드의 집합인 파일에 대한 자료구조

# Classification

## - 선형 구조

- 1) 배열(Array)
- 2) 연결리스트(Linked List)
- 3) 스택(Stack)
- 4) 큐(Queue)

## - 비선형 구조

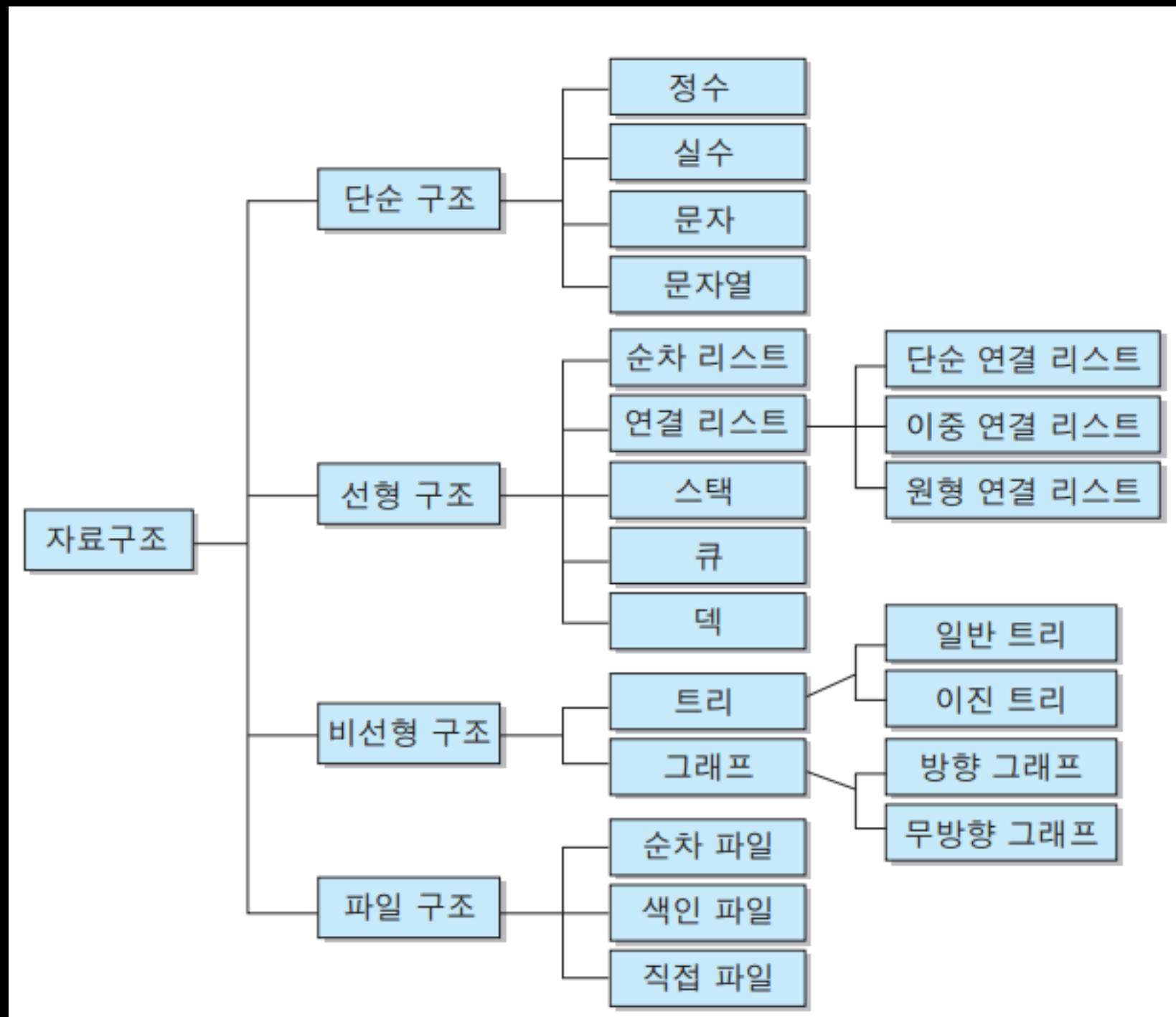
- 1) 트리(Tree)
- 2) 그래프(Graph)

## - 파일 구조

- 1) 순차 파일
- 2) 색인 파일
- 3) 직접 파일



# Classification



# Applications

자료구조는 아래와 같은 분야 등에서 활용할 수 있습니다.

1. 정렬(Sort) : 기억장치 내의 자료를 일정한 순서에 의해 저장
2. 검색(Search) : 기억장치 내의 자료를 최대한 효율적으로 검색
3. 파일 구조 : 자료를 기억 매체에 저장할 때의 파일 구조 형성
4. 인덱스 : 파일에서 특정 자료를 빠르게 찾기 위한 색인표 구성

# References

[1] 자료 구조 : [https://ko.wikipedia.org/wiki/자료\\_구조](https://ko.wikipedia.org/wiki/자료_구조)

[2] 강의노트 17. 알고리즘, 자료구조 개요 : <https://wayhome25.github.io/cs/2017/04/17/cs-18/>

[3] 자료구조란? : <https://m.blog.naver.com/PostView.nhn?blogId=islove8587&logNo=220548856458&proxyReferer=https:%2F%2Fwww.google.com%2F>

[4] 자료구조 : <https://velog.io/@rockjeon/자료구조>

[5] 자료구조 : 자료구조란? (Data Structure) : <https://andrew0409.tistory.com/148>

# References

[6] 프로그래밍을 잘하는 5가지 방법 — 1. 자료구조와 알고리즘 공부하기 : <https://medium.com/@ghilbut/프로그래밍을-잘하는-5가지-방법-1-자료구조와-알고리즘-공부하기-ebf3a3f79ee2>

[7] 자료구조 : <http://www.incodom.kr/자료구조>

[8] Data Structure (자료구조) : <https://opentutorials.org/module/1335>

[9] Data Structure 자료 구조, 데이터 구조 : [http://www.ktword.co.kr/abbr\\_view.php?m\\_temp1=4076](http://www.ktword.co.kr/abbr_view.php?m_temp1=4076)

[10] 자료구조란? 자료구조, 추상 자료형(ADT) : <https://boycoding.tistory.com/32>

Thank you!