자료구조

Memory 구조

STACK(스택)	<< 지역변수, 매개변수 데이터가 변경이됨, 값타입
HEAP(힙)	<< 동적 할달을 위한 영역 인스턴스가 만들어지는 영역 객체, 주소타입
DATA(데이터)	<< 전역변수(프로그램이 끝날때까지 있는변수) 정적변수(스태틱, 프로그램 종료될때까지 유지, 변함없음)
CODE(코드)	<< 프로그램 code 저장

Pointer(포인터)
- 프로그래밍 언어에서 다른 변수, 혹은 그 변수의 메모리 공간주소를 가리키는 변수를 말한다.

포인터 사용 이유
- 메모리에 있는 정보를 쉽게 다루기 위하여
- 복잡한 자료 구조를 효율적으로 처리하기 위해
- 메모리 공간을 효율적으로 사용하기 위해

에스터 리스크 (*)
- 변수 선언시 에스터 리스크(*)는 주소값을 가지는 변수를 나타낸다.
- 변수 사용시 에스터 리스크는 변수의 값을 가르키며, 엔드(&)는 주소값을 나타낸다.
- Objective-C에서 사용되는 모든 Reference변수에는 (*)가 붙어 있다.

Value & Reference

value Type vs Reference Type
- 실질적인 값 저장 vs 참조하고 있는 주소값의 저장(pointer)

Class 내용보기
- command + class이름 클릭
- NSString, NSInteger 등등

구조체(값타입)
- 서로 다른 타입의 데이터를 묶여 있는 데이터 구조
- 메서드 x, 상속 x

구조체 선언

구조체에 이름짓기

```
#import <Foundation/Foundation.h>
//인풀트 아래에 작성
//타입디프로 스트럭 구조체 생성
typedef struct character
{
    CGFloat height;
    CGFloat weight;
}charactor;

@interface Person : NSObject

@property NSString *name;
@property NSInteger age;

//구조체
@property charactor info;

@end
```

구조체 사용

```
#import "Person.h"

@implementation Person

//클래스 안에서 구조체 사용
- (instancetype)initWithName:(NSString*)name
{
    self = [super init];
    if (self) {
        self.name = name;
    }
    return self;
}

//구조체
- (void)setHeight:(CGFloat)height weight:(CGFloat)weight
{
    //_info.height = height;
    //_info.weight = weight;

    charactor tempInfo = {height,weight};
    self.info = tempInfo;
}

//구조체
- (charactor)personInfo
{
    return _info;
}

@end
```

typedef 별명 짓기
- 타입에 별명을 지어주는 키워드
- typedef <데이터타입> <이름>;

구조체 vs 클래스
- 구조체는 할당 시 복사됩니다. 구조체가 새 변수에 할당되면 모든 데이터가 복사되고, 새 복사본을 수정해도 원래 복사본의 데이터는 변경되지 않습니다.
- 구조체는 값 형식이고 클래스는 참조 형식입니다.
- 클래스와 달리 구조체는 alloc을 사용하지 않고 인스턴스화 합니다.
- 구조체는 다른 구조체 또는 클래스에서 상속될 수 없으며, 클래스의 기본 클래스가 될 수도 없습니다.
- Objective-C에서 사용되는 클래스는 구조체 안에서 사용할 수없다.(ARC영향)
- 구조체 대신 class사용 하는 것을 지향.

배열

Array
- 배열은 번호(인덱스)와 번호에 대응하는 데이터들로 이루어진 자료 구조를 나타낸다. 일반적으로 배열에는 같은 종류의 데이터들이 순차적으로 저장되어, 값의 번호가 곧 배열의 시작점으로부터 값이 저장되어 있는 상대적인 위치가 된다. 0번부터 시작된다.

Array 문법
- 선언 : type name[array range];
  NSInteger intList[5] = {1,2,3,4,5};
  CGFloat floatList[10] = {3.1, 3.2, 3.3};
  Char name[9] = “younmin\0”; 0은 종료를 뜻함 9번째
- 사용 : name[index];
  NSLog(@“%ld”, intList[3]);
  NSLog(@“%f”, floatList[0]);
  NSLog(@“my name is %s”, name);

Array 구조
- NSInteger list[5] = {0}; >> 데이터의 크기가 0으로 5까지의 리스트를 만들어라
  list[0] = 3; >> 인덱스 0번에 3을 넣어라
  list[3] = 10; >> 인덱스 3번에 10을 넣어라
  list[2] = 20; >> 인덱스 2번에 20을 넣어라
  NSInteger age = list[0];

요소	list[0]	list[1]	list[2]	list[3]	list[4]
index	0	1	2	3	4
값	3	0	20	10	0

Array 특징
- 선언 당시 데이터 사이즈를 정적으로 만들어야 한다.(동적생성불가)
- index를 통해 데이터에 접근한다.
- 하나의 타입만 저장된다.
- Objective-C에서는 NSArray, NSMuableArray Class를 사용한다.