

## Day17. iOS Programming\_1

전미정

### ARC(Automatic Reference Counting)

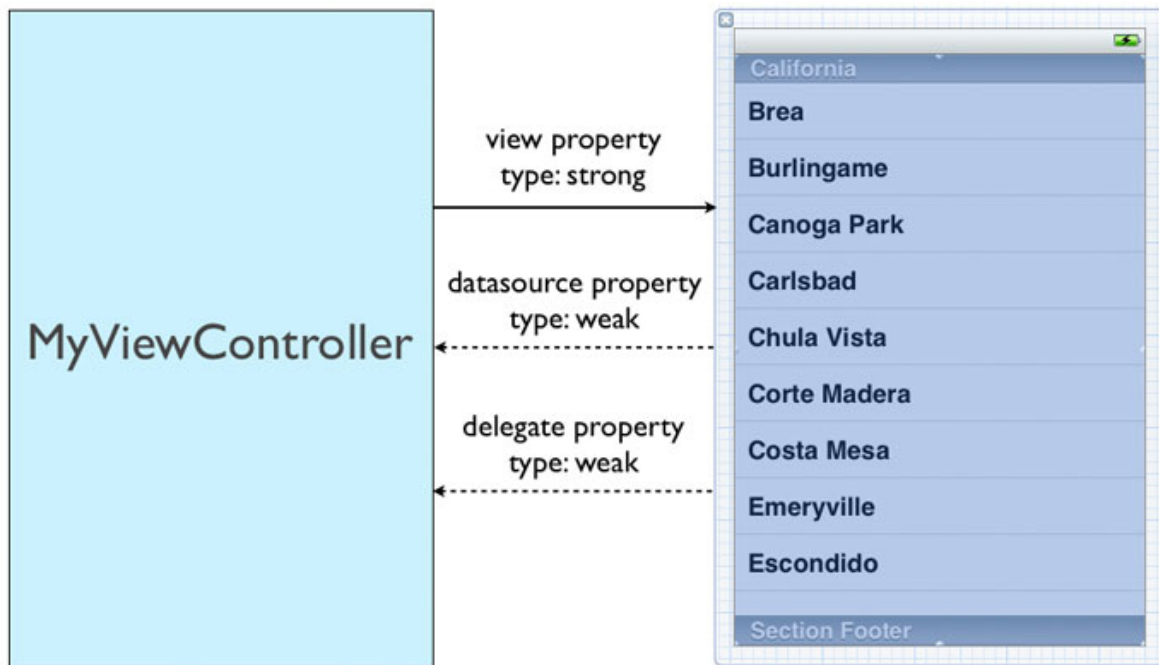
**ARC란?** ARC란 Automatic Object memory management로, 개발자가 수동적으로 retain/release 등의 코드를 이용해 메모리 관리를 하는데서 발생하는 문제점을 해결하기 위해 애플에서 도입한 메모리 관리 시스템이다. 기존의 방식은 메모리 해제를 위한 코드를 직접 작성해야 하므로 코드 복잡도가 증가할 뿐만 아니라, 앱 비정상 종료의 많은 부분이 메모리 문제로 인한 것이었기 때문에 더 나은 개발환경을 위해 ARC의 도입이 필요했다. ARC는 모든 메모리 영역을 관리하는 것이 아니라, Objective-C의 Object를 관리 대상으로 하며, GC(Garbage Collector)와 다르게 런타임이 아닌 컴파일 단계에서 처리된다.

ARC는 클래스에서 인스턴스가 생성되면 ARC가 이를 추적하며, 더 이상 어플에서 사용되지 않으면 ARC가 이를 해제해 다른 곳에 사용할 수 있도록 해준다.

컴파일러의 자동 메모리 관리 기능을 위해서는 C언어 형식이 아닌, object 형식으로 코드를 작성해야한다.

**ARC의 장점** GC는 런타임 중에 메모리를 검사하기 때문에 앱 퍼포먼스에 악영향을 줄 수 있지만, ARC는 개발자가 수동으로 코딩했던 retain/release를 컴파일러가 자동으로 코드에 삽입하기 때문에 결과적으로 MRC와 동일하므로 성능 저하를 유발하지 않는다. ARC를 이용할 시 기존에 비해 3배 정도 빠른 NSObject retain/release가 가능하며, @autoreleasepool 속도는 6배 정도 향상된다.

**Strong and Weak** ACR을 이용하면서 개발자가 고려해야할 사항은 strong/weak 수식자이다. 기본으로 사용되는 수식자는 strong으로, 현재 event cycle에 object를 retain할때 사용한다. Weak 수식자는 zeroing weak reference라고도 알려져있는데 object를 retain할 필요가 없을때 사용한다. Weak 수식자는 특히, 부모-자식 객체 사이에서 많이 사용된다. 부모 객체는 자식 객체를 strong 으로 참조하며, 반대로 자식 객체는 부모객체를 weak으로 참조한다. (만약 서로 strong으로 참조하면 circular reference 구조를 형성하므로 주의해야한다.)

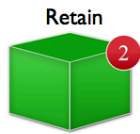


**MRC(Manual Reference Counting)** iOS 5부터 적용된 ARC가 등장하기 전에는 개발자가 직접 객체의 참조 상황을 추적하고 메모리를 해제해야했다. 메모리 해제가 적절히 시행되지 않으면 memory leak, 비정상 종료 등의 문제가 발생하게 된다. 뿐만 아니라 메모리 관리는 매우 복잡하고 다루기 힘든 부분이다. 기존의 retain/release는 아래와 같은 방식으로 이루어졌다.

## Manual Reference Counting



```
MyClass *obj1 = [[MyClass alloc] init];
```



```
MyClass *obj2 = [obj1 retain];
```



```
[obj2 release];
```



```
[obj1 release];
```

수동 메모리 관리에서 사용되는 주요 명령어에는 아래와 같은 것이 있다.

Alloc – create an object and claim ownership of it

Retain – claim ownership of an existing object

Copy – copy an object and claim ownership of it

Release – relinquish ownership of an object and destroy it immediately

Autorelease – relinquish ownership of an object but defer its destruction

## Reference site

ARC <http://goo.gl/JLOrzV>

MRC <http://goo.gl/PPUFhL>

