1. **ARC(automatic reference counting)**

* 자동으로 메모리를 관리해주는 방식
* 컴파일러가 retain/release를 통해 reference counting을 자동으로 관리
* 가비지컬렉션 과의 큰 차이는 참조를 계산하는 시점
* ARC는 인스턴스가 언제 메모리에서 해제되어야 할지를 컴파일과 동시에 결정
* 작동규칙을 모르고 사용하면 Retain Cycle에 걸릴수가 있음

1. 강한참조 (Strong)

* 기본적으로 객체를 가르키는것은 강한참조이다. strong이 생략되고 있었던것
* 최소한하나의 객체가 있다면 메모리는 해제 되지 않음
* 강한참조의 규칙을 모르고 사용하게 되면 Retain Cycle에 걸릴수 있음
* 메모리 누수 (memory leak) 이 발생할수 있음

1. 약한참조 (weak)

* 자신이 참조하는 인스턴스의 참조횟수를 증가시키지 않음
* weak 키워드 사용, 메모리에서 해제 될수도 있다는것을 예상해야함
* 메모리가 해제된후 자동으로 nil이 되므로 반드시 optional 형태여야 한다

1. 약한참조 (unowned)

* weak와 비슷하지만 optional형태로 쓸수없다.
* 해제된 메모리 영역을 접근하지 않는다는 확실한 경우만 사용해야한다.
* 해당 변수가 가리키는 객체의 메모리가 해제된 이후에는 해당 영역을 가리키지 않는다는 확신이 있다면 사용하는것이다

1. Retain Cycle 시나리오
2. Delegate

* delegate를 설정한 부모뷰컨트롤러에서 자식 delegate를 대리자로 자기 자신을 가르키면서 retain cycle이 발생한다.
* 따라서 delegate는 weak로 선언해야함

1. Closure

클로저도 클래스와 마찬가지로 reference 타입이므로 강한참조를 한다

capture list를 해서 해결 [weak self]

클로저 내부에서 참조 타입을 획득하는 규칙을 제시해줄수 있는 기능