

property의 Attributes에 관하여 더 알아보기

프로퍼티에 대해서

- objective-c의 프로퍼티가 있기 이전에는 getter/setter가 있었다.
- @property, @synthesize 키워드가 포함되었다. 이 키워드를 이용해서 코드를 작성하면 컴파일러가 getter/setter를 자동으로 생성 시켜 준다.

프로퍼티의 원리

- 프로퍼티는 getter/setter를 작성하는 귀찮은 일을 컴파일러에게 코드를 대신 작성하도록 하는 것이다.
- objective-c의 관례처럼 getter는 같은 이름이고, setter만 set000로 변형됩니다.
- 생성되는 코드는 @property를 사용할때 attributes에 따라서 달라지게 된다.

프로퍼티 옵션들

- 프로퍼티는 @interface와 @end사이에 다음과 정의 합니다.
@property (attribute) type name;
- attribute에 따라서 내부 생성되는 코드들이 달라지기 때문에 원하는 용도에 따라서 적절히 사용해야 합니다.
- 메소드 이름 변경
 - 프로퍼티는 내부적으로 메소드를 정의하게 된다. 이때 만들어지는 메소드의 이름은 관례에 따라서 getter는 프로퍼티이름과 동일 setter는 set을 붙이고 프로퍼티의 앞글자만 대문자로 만들어 사용한다.
 - 보통 BOOL타입 프로퍼티는 getter를 is0000와 같이 is를 붙여 사용한다. 만약 나름의 코딩 스타일이 있다면 이 속성을 사용해서 변경이 가능하니 알아 두도록 하는 것이 좋을 것이다.
- 쓰기여부
 - 프로퍼티가 읽기만 허용해야 한다면 setter함수만 만들어지지 않도록 하면 된다.
 - readwrite : setter / getter 모두 생성(default)
 - readonly : getter만 생성
- setter 함수를 위한 속성들
 - setter는 외부의 값을 저장하는 메소드이다. objective-c는 레퍼런스 카운트에 따라서 객체의 라이프 사이클을 결정하기 때문에 중요한 옵션이다.
 - strong : 입력되는 변수의 소유권을 가져온다. retain count를 1 증가시켜서 내가 소유하고 있는 동안에는 release되지 못하도록 한다.
 - retain : strong과 같이 소유권을 가져온다는 의미
 - weak : weak은 객체의 레퍼런스를 증가시키지 않는다. 따라서 언젠가 객체가 릴리즈 되어서 메모리에서 사라질 수 있다. 이렇게 사라지면 자동으로 nil로 설정된다.
 - copy : 주어진 객체를 복사하는 것으로 strong의 경우 같은 객체를 레퍼런스 카운트를 사용해서 소유권을 가져오는 것이라면 copy는 아예 같은 값을 같은 객체를 새로 생성하는 것이다.
 - assign : 기본 값으로 값을 복사하는 것이다. weak과 유사하나 약간 다른데 weak는 객체를 다룬다. 하지만 assign은 값타입(integer, float등)을 복사한다는 의미이다.
- 스레드 세이프이거나 아니거나
 - nonatomic : setter가 스레드 세이프하면 스레딩 환경에서 아무 걱정 없이 안전하게 사용할 수 있겠지만 이 세상에 공짜가 없듯이 여기에서도 비용이 발생한다.
 - atomic : setter가 스레드 세이프하도록 설정(기본값)

Synthesize와 그 밖에..

- @Synthesize는 프로퍼티 코드 생성을 컴파일러에게 지시한다. 사용되는 위치는 @implementation과 @end 사이에 정의된다.
- @dynamic은 컴파일러에게 setter / getter코드를 생성하지도 말고, 경고도 내지 말라고 지시하는 것이다. runtime으로 setter / getter를 설정하는 메커니즘이 있는 경우에만 사용하게 된다. 현재로는 core data에서 모델 클래스를 정의할 때만 사용되고 있다.