1. **프로퍼티와 메서드**

* 프로퍼티
* 클래스, 구조체 또는 열거형 등에 관련된 값
* 메서드
* 특정 타입에 쓰인 함수

1. 프로퍼티
2. 저장 프로퍼티 - Stored Properties

* 클래스 또는 구조체의 인스턴스와 연관된 값을 저장하는 단순한 개념의 프로퍼티
* var 변수 저장 프로퍼티, let 상수 저장 프로퍼티
* 구조체는 이니셜라이저를 자동으로 생성
* 클래스는 옵셔널이 아니라면 기본값을 지정해 주거나 반드시 이니셜라이져를 해줘야 함

1. 지연 저장 프로퍼티 - Lazy Stored Properties

* 호출이 있어야 값을 저장 lazy 키워드 사용
* let은 인스턴스가 생성되기전에 초기화 해야하므로 var 키워드를 사용
* 인스턴스를 초기화 하면서 저장 프로퍼티로 쓰이는 인스턴스들이 한 번에 생성되어야 하는경우
* 굳이 모든 저장프로퍼티를 사용할 필요가 없을 경우
* **다중스레드 환경에서 지연 저장 프로퍼티에 동시다발적으로 접근할 때는 한번 만 초기화 된다는 보장이 없으므로 주의 해야함**

1. 연산 프로퍼티 - Computed Properties

* 특정 상태에 따른 값을 연산 하는 프로퍼티
* getter, setter의 개념
* why? 인스턴스 외부에서 메서드를 통해 인스턴스 내부 값에 접근 하려면 메서드를 두개 만들어야 한다 (접근자, 생성자) 이는 가독성이 떨어지며 두개를 구현해야 하므로 번거롭다

1. 프로퍼티 감시자 - Property Observers

* 프로퍼티의 값이 새로 할당될 때마다 호출함.
* 프로퍼티 override를 통해 상속받은 저장프로퍼티에서 연산 프로퍼티를 사용할수 있음.
* willSet - 변경될값, didSet - 변경되기전의 값

1. 타입 프로퍼티 - Type Properties (static)

* 인스턴스가 생성되었을때 사용할수있는 프로퍼티가 아닌 타입자체에 속하는 프로퍼티
* static 키워드 사용

1. 메서드
2. 인스턴스 메서드

* 특정 타입의 인스턴스에 속한 함수
* 인스턴스 내부의 프로퍼티 값을 변경하거나 특정 연산 결과를 반환
* 클래스의 인스턴스는 참조타입이라서 self 프로퍼티에 다른 참조를 할당할 수 없다
* 구조체의 경우에는 mutating키워드를 이용하여 self프로퍼티에 자기 자신을 할당할수 있다.

1. 타입 메서드 (static, class)

* 클래스의 타입 메서드는 static, class로 정의
* 구조체의 타입 메서드는 static만 가능
* static으로 정의하면 상속후에 override가 불가함
* class로 정의하면 상속후에 override가 가능
* 타입메서드에서 self를 선언하는것은 타입그자체를 가르키는것