CS193p(Lecture2) 2023

강의에서는 슈퍼 클래스의 개념을 부정한다. 구조체가 :로 다른 뷰를 채택하면 단지 슈퍼 클래스처럼 행동하는 것이라고 한다.

**View(let vs var)**

카드게임을 예시로 들어보자. 카드의 형태나 카드가 가지고 있는 문양들은 변할 필요가 없으니 let으로 선언해 상수화 하고 카드가 뒤집히는 경우에 바뀌는 문양은 var을 통해 Bool 변수로 설정한다. 카드는 형태를 바꾸지 않고 변수를 통해 상태만 변화시킬 수 있게 된다.

**타입 추론**

선언과 동시에 값을 할당해 초기화 하는 경우 타입 추론을 이용해 타입을 명시하지 않아도 된다.

스위프트는 굉장히 강력한 타입 추론 언어이다. 자바스크립트의 느슨한 타입추론으로 인해 발생하는 문제들은 스위프트에서 거의 일어나지 않는다.(강력한 타입추론으로 인한 트레이드 오프?)

.onTapGesture

뷰에 이 메서드를 추가해 자바스크립트의 쿼리 셀렉터처럼 탭 발생 시 특정 이벤트를 발생시킬 수 있다.

**View는 Immutable이다.**

카드 게임을 위해 탭 제스처가 발생할 때마다 var로 선언한 뷰의 불리언 속성 값을 반전시키려 했지만 뷰는 불변객체라는 문구와 함께 에러가 났다.

이때 사용 가능한 것이 var 앞의 @State 어노테이션이다. @State는 포인터를 생성한다. 뷰는 변하지 않기에 @State로 선언한 포인터의 주솟값도 변하지 않는다. @State var istrue가 가리키는 값의 대상만 변할 뿐이다.

또 이런 불 방식의 상태 변환은 .toggle()로 가능하다.

**Internal VS external parameter names**

**1강 참조**

**Opacity VS if-else in @ViewBuilder**

**Opaque type**

불투명한 타입을 반환해 공용 인터페이스가 다룰 필요가 없는 구체적인 타입의 파라미터를 작성할 필요를 없앤다. 공용 인터페이스의 사용자는 해당 인터페이스의 역할을 사용할 뿐이지 내부 내용을 알 필요가 없기 때문에 이런 식으로 설계했다.