

## [디자인패턴 MVC, MVP and MVVM 에 대해서 알아봅시다]



### MVC (Model - View - Controller)

- Controller 에 직접 Input
- View 와 Controller : Many to One 관계
- View 는 Controller 를 참조하지 않음
- Model 은 View 를 간접적으로 참조함

-> Controller 에 입력이 들어오면 Controller 는 Model 에 있는 Data 를 조작하고, View 는 Model 에서 조작된 data 를 참조하여 View 를 수정한다. 이 때 View 가 Model 을 참조하거나 Model 이 View 를 참조하거나 하는 방식으로 변화에 대한 업데이트를 할텐데 결국 View 와 Model 이 참조를 할 수 밖에 없다는 이야기.

### MVP (Model - View - Presenter)

- View 에 직접 Input
- View 와 Presenter : One to One 관계
- View 는 자신의 Presenter 를 참조하고 Presenter 역시 View 를 알고 있음
- View 는 Model 를 참조하지 않아 Presenter 를 통해 Model 을 업데이트함

-> View 에 입력이 들어오면 Presenter 에 data 를 요청하고, Presenter 는 자신이 참조하는 Model 에 업데이트를 요청하는 방식으로 동작한다. 이 경우 View와 Model 은 완벽히 분리되지만 View 와 Code 가 완벽히 분리됐다고 보기는 어렵다.

### MVVM (Model - View - ViewModel)

- View 에 직접 Input
- View 와 ViewModel : Many to One 관계
- ViewModel 은 View 를 참조하지 않음
- View 는 Model 를 참조하지 않아 ViewModel 를 통해 Model 을 업데이트함

-> View 에 입력이 들어오면 View 가 참조하고 있는 ViewModel 에서 Binding 된 객체를 찾아 업데이트를 한다. MVP 패턴에서는 Presenter 는 전적으로 View 의 형태에 따라 달라지지만, MVVM 패턴에서는 ViewModel 이 View 를 참조하지 않으므로 Model 의 형태를 따른다고 할 수 있다. View 는 Model 과 완벽히 분리되며 ViewModel 과도 Binding 을 통해 자동 업데이트 되므로 data 와도 완벽히 분리된다.

