

# Vue.js 컴포넌트 디자인 패턴

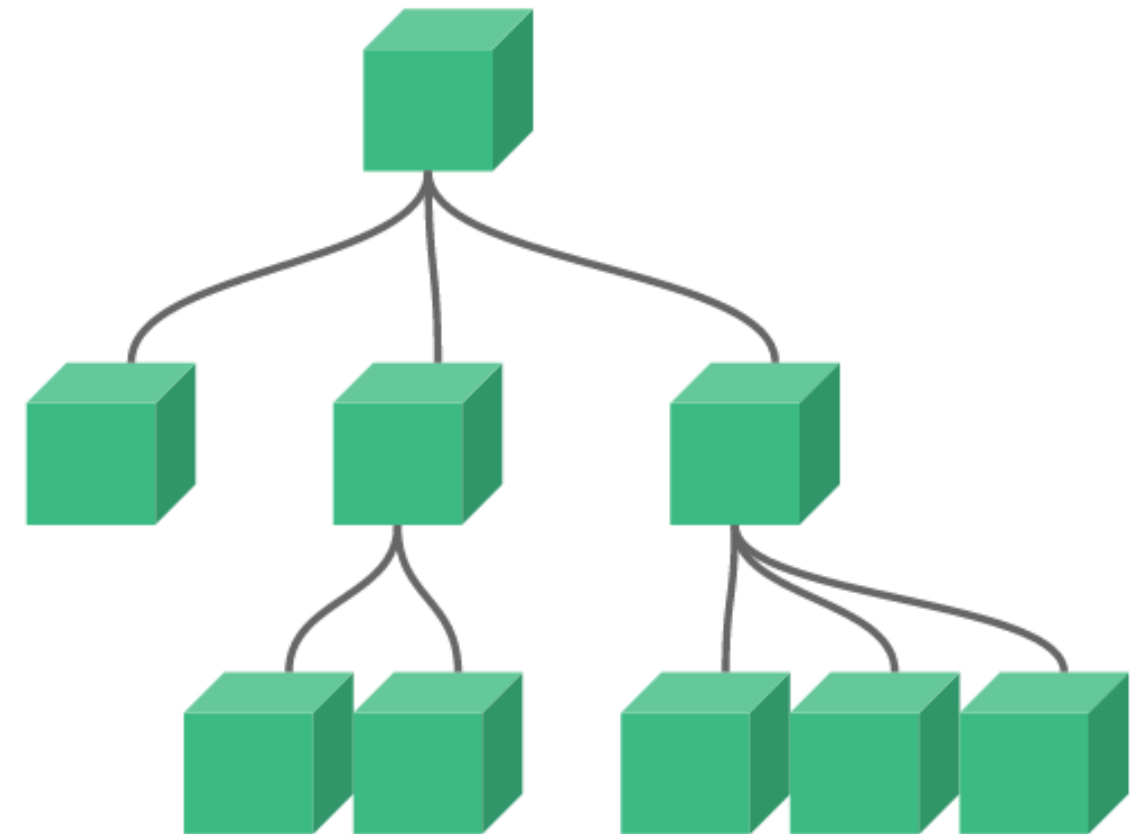
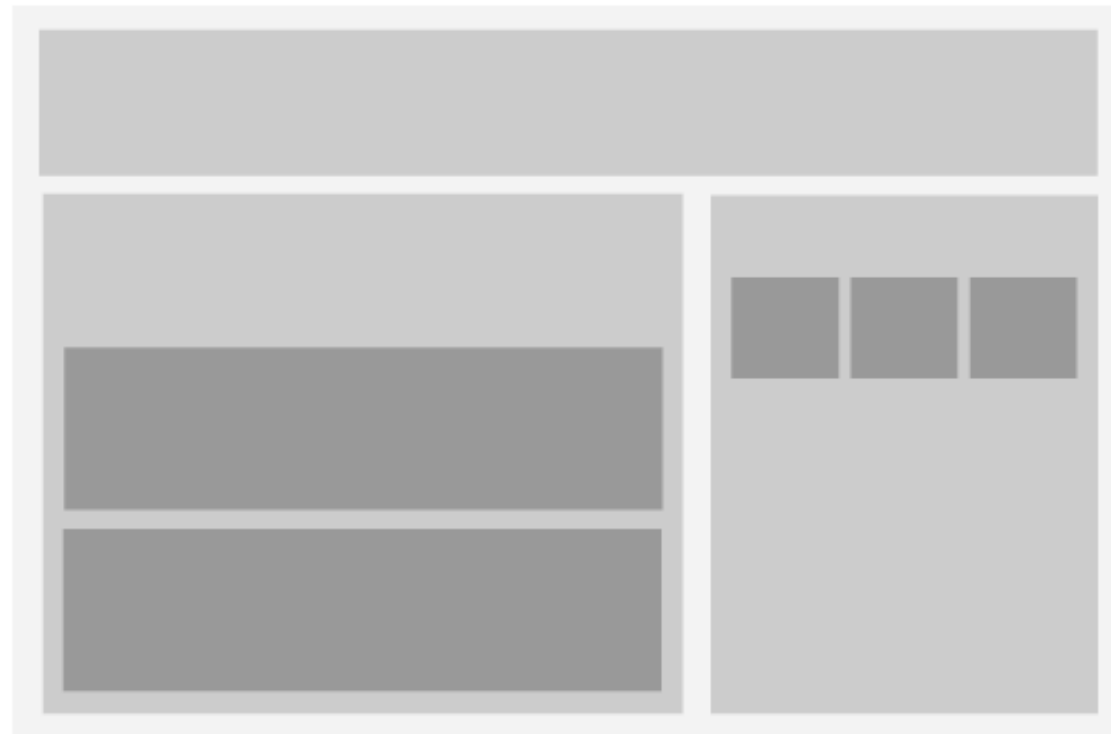
# 목차

- ▶ 패턴 1 - Common Approach
- ▶ 패턴 2 - Component with Slots
- ▶ 패턴 3 - Controlled Component
- ▶ 패턴 4 - Renderless Component
- ▶ 패턴 5 - Mixins
- ▶ 패턴 6 - High Order Component

# 패턴 1 - Common Approach

# 패턴 1 - Common Approach

화면의 영역별로 컴포넌트를 구분하는 방법



# 패턴 1 - Common Approach

UI 영역 기준으로 코드를 나눠서 관리



# 실습

# Common Approach - 참고 사항

- ▶ 요구 사항 변경에 대응하기 어려움
- ▶ 작은 UI 변경에도 새로 컴포넌트를 생성하거나 기능 확장 어려움

# 패턴 2 - Component with Slots



# 패턴 2 - Component with Slots

하위 컴포넌트의 **표현부(태그, 스타일)**를 유연하게 **확장**할 수 있는 방법

```
<error-modal>
  <h1 slot="title">Network Error!</h1>
  <p slot="content">
    Server is not available
  </p>
</error-modal>
```

실습

# Component with Slots - 참고 사항

- ▶ 모달, 툴팁, 토스트 등의 공통 UI 컴포넌트에 활용
- ▶ 카드, 테이블 등의 그리드 컴포넌트에도 사용 가능

# 패턴 3 - Controlled Component

# 패턴 3 - Controlled Component

상위 컴포넌트에서 하위 컴포넌트를 더 쉽게 제어할 수 있는 방법

```
<date-picker v-model="selectedDate"></date-picker>
```

실습

# Controlled Component - 참고 사항

- ▶ 날짜 선택기, 체크 박스 등의 UI 컴포넌트에서 활용도 ↑

# 패턴 4 - Renderless Component



# 패턴 4 - Renderless Component

UI 로직을 제외하고 데이터 로직만 구분하여 관리하는 방법

```
<script>
export default {
  render() {
    return this.$scopedSlots.default({
      // ...
    });
  },
}
</script>
```

# 시연

# 패턴 5 - Mixins

# 패턴 5 - Mixins

중복되는 **컴포넌트 로직**을 **재활용**하기 위한 첫 번째 방법

```
var DataTableMixin = {  
  // 인스턴스 & 컴포넌트 옵션  
  created: function() { ... },  
  methods: {  
    setupPagination: function() { ... },  
    onRowClass: function() { ... },  
  }  
}  
  
new Vue({  
  mixins: [DataTableMixin]  
})
```

실습

# Mixins - 참고 사항

- ▶ 믹스인을 사용하는 컴포넌트의 로직과 충돌하지 않게 **팀내 코딩 컨벤션** 선정

```
data: { mixinModalStatus : '' },  
methods: {  
  mixinOpenModal: function() {},  
}
```

- ▶ 믹스인과 컴포넌트의 로직이 충돌하면 **컴포넌트의 로직**이 우선 순위를 가짐

# 패턴 6 - High Order Component

# 패턴 6 - High Order Component

중복되는 **컴포넌트 로직**을 **재활용**하기 위한 두 번째 방법

▼ `<news-feed> = $vm1`

▼ `<list-view> = $vm0`

`<ListItem> = $vm3`

▼ `<job-board>`

▼ `<list-view> = $vm4`

`<FeedItem>`



# 시연

# High Order Component - 참고 사항

- ▶ 컴포넌트의 레벨이 깊어져서 데이터 전달이 번거로워지지 않게 주의

끝