

삼성 청년 SW 아카데미

Vue.js

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

Vue.js

목차

1. [vuex](#)
2. [저장소 - state](#)
3. [저장소 - Getters](#)
4. [저장소 - mapGetters](#)
5. [저장소 - Mutations](#)
6. [저장소 - Actions](#)

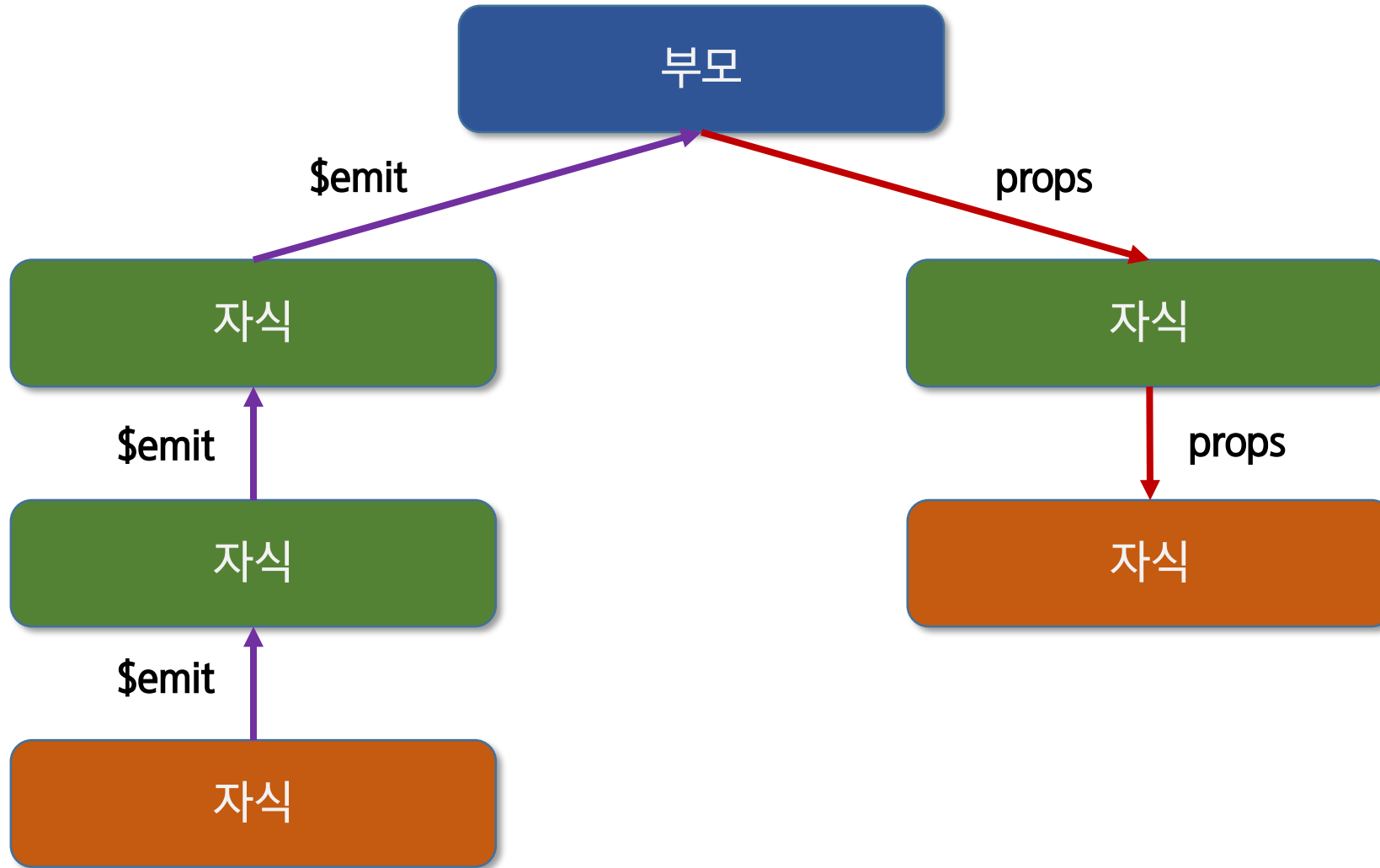
vuex

삼성 청년 SW 아카데미

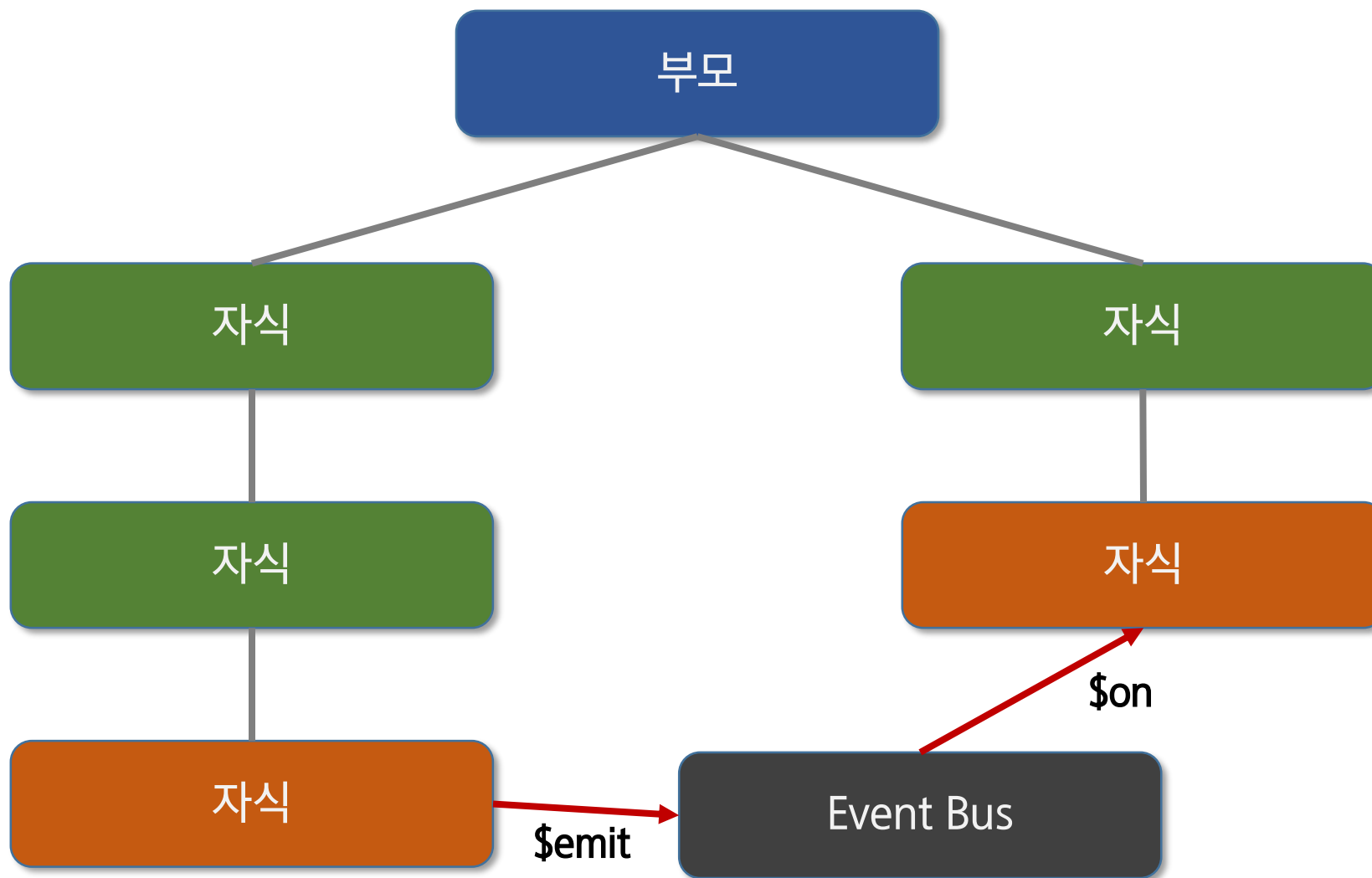
✓ vuex.

- Vue.js application에 대한 **상태관리패턴 + 라이브러리** ([공식페이지 참조](#)).
- application 모든 component들의 중앙 저장소 역할 (데이터 관리).
- 상위(부모) 하위(자식)의 단계가 많이 복잡해 진다면 데이터의 전달하는 부분이 매우 복잡해 짐.
- application이 여러 구성 요소로 구성되고 더 커지는 경우 데이터를 공유하는 문제가 발생.

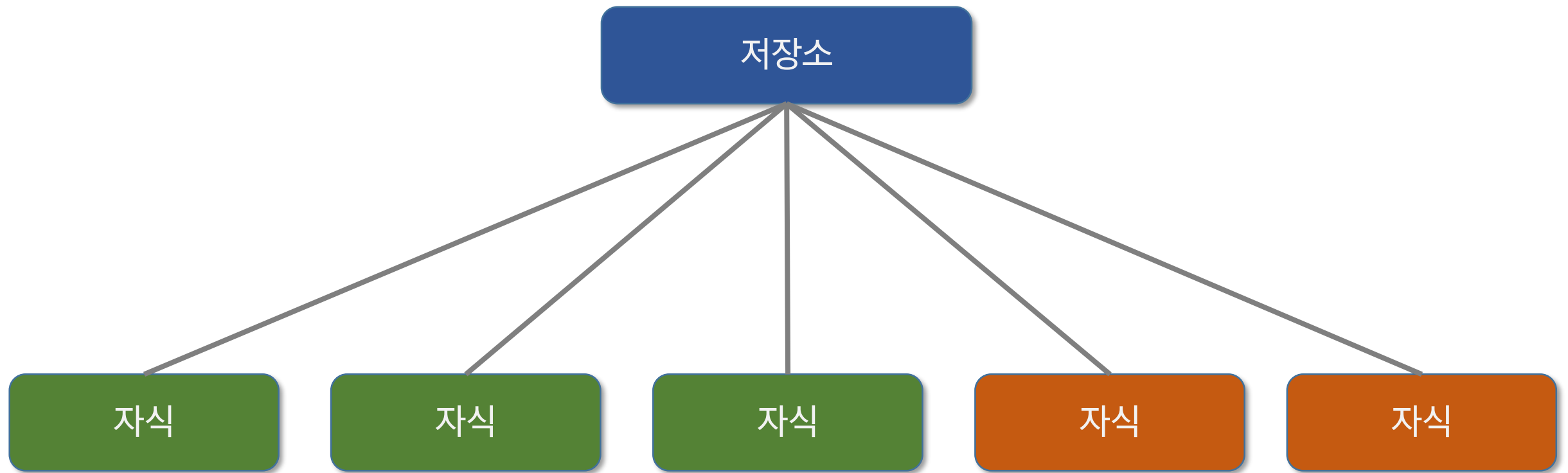
✓ 부모-자식 컴포넌트간의 data 전달.



✓ 동위 컴포넌트간의 data 전달.



✓ vuex.



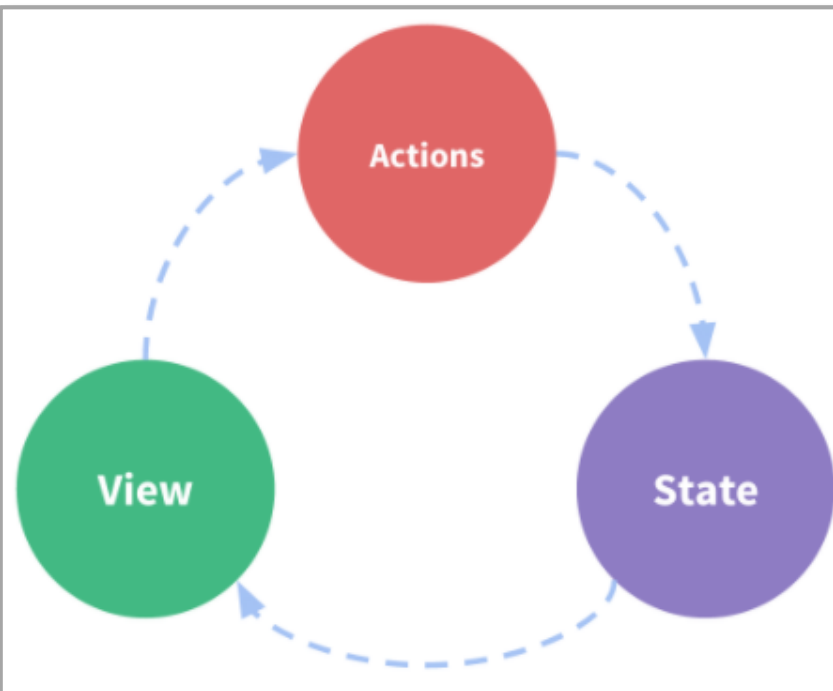
✓ 상태관리패턴.

```
new Vue({  
  // 상태  
  data () {  
    return {  
      count: 0  
    }  
  },  
  // 뷰  
  template: `

{{ count }}

`,  
  // 액션  
  methods: {  
    increment () {  
      this.count++  
    }  
  }  
});
```

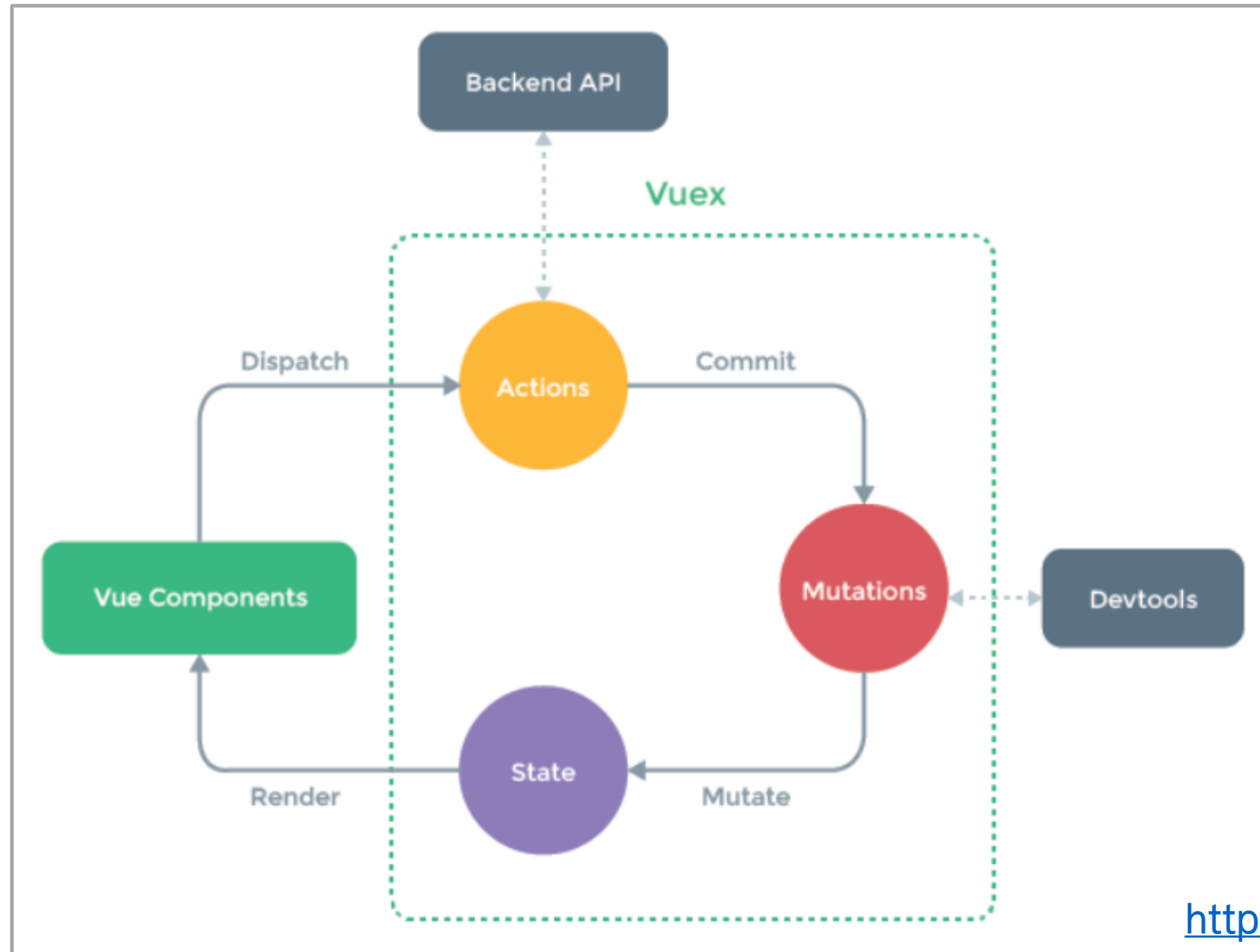
- **상태** 는 앱을 작동하는 원본 소스. (= data)
- **뷰** 는 상태의 선언적 매핑. (= template)
- **액션** 은 뷰 에서 사용자 입력에 대해 반응적으로 상태를 바꾸는 방법. (= method)



단방향 데이터 흐름

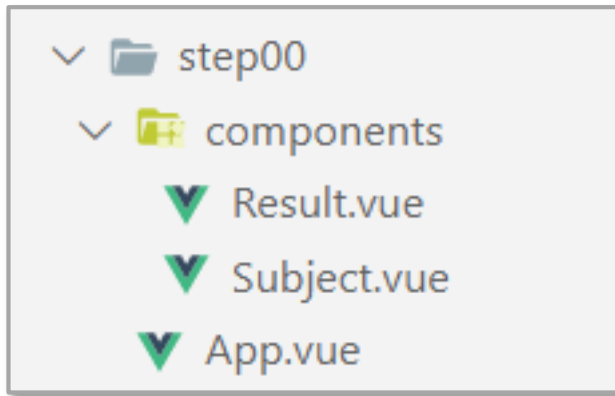
<https://vuex.vuejs.org/kr>

✓ vuex의 핵심 컨셉 이해.



✓ vuex 단계별 이해하기.

- step00 : Vuex 적용전 코드.



✓ vuex 설치.

- CDN

- `<script src="https://unpkg.com/vuex"></script>`

- npm

- `npm install vuex --save`

✓ vuex 저장소 개념.

- State : 단일 상태 트리를 사용. application마다 하나의 저장소를 관리 (data)
- Getters : Vue Instance의 Computed와 같은 역할. State를 기반으로 계산 (computed)
- Mutations : State의 상태를 변경하는 유일한 방법 (동기 methods)
- Actions : 상태를 변이 시키는 대신 액션으로 변이에 대한 커밋 처리 (비동기 methods)

✓ vuex 설정.

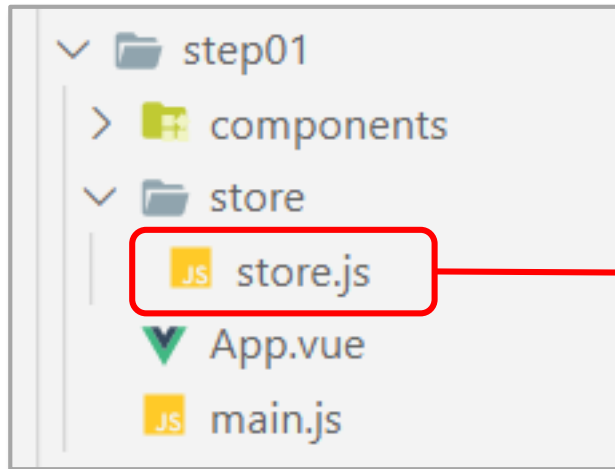
- module 시스템과 함께 사용시 `Vue.use()`를 통해 Vuex 설정.

```
import Vue from 'vue';  
import Vuex from 'vuex';
```

```
Vue.use(Vuex); // Vue Instance에 Vuex를 설정
```

✓ vuex 단계별 이해하기.

- step01 : vuex 사용.



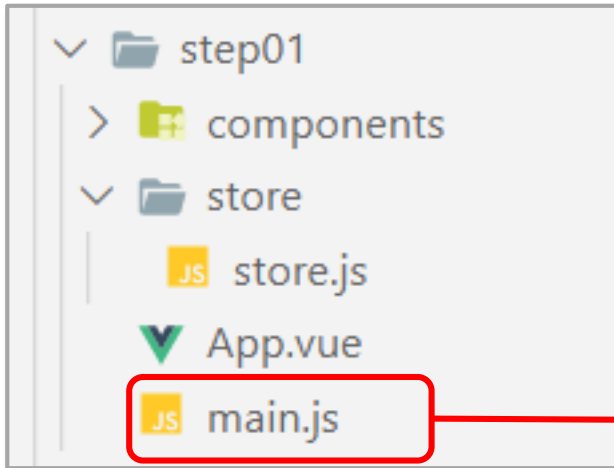
```
import Vue from 'vue';
import Vuex from 'vuex';

Vue.use(Vuex); // Vue Instance에 Vuex를 설정

export default new Vuex.Store({
  state: {
    count: 0,
  },
});
```


✓ vuex 단계별 이해하기.

- step01 : vuex 사용.



```
import Vue from 'vue';
import App from './App.vue';
import store from './store/store';

Vue.config.productionTip = false;

new Vue({
  render: (h) => h(App),
  store,
}).$mount('#app');
```

이어서..

삼성 청년 SW 아카데미

저장소 - state

삼성 청년 SW 아카데미

✓ 저장소 (Store) - state.

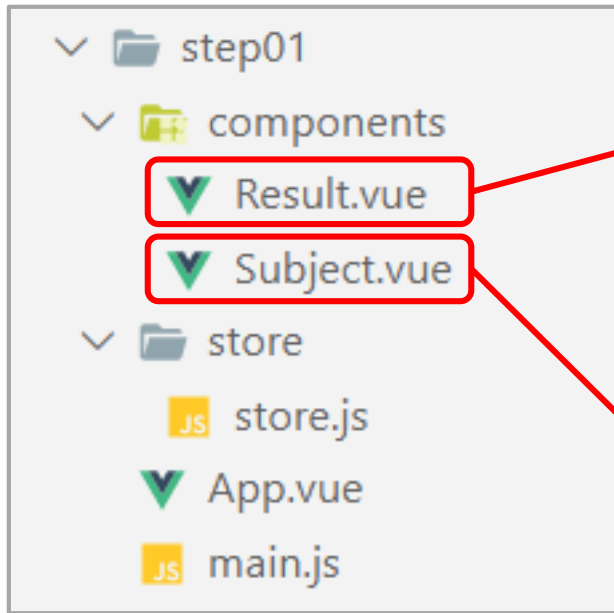
- 저장소에서 data 속성의 역할
- application에서 공유해야 할 data를 관리.
- State에 접근하는 방식 : `this.$store.state.data_name`

✓ Vuex Store - state.



✓ vuex 단계별 이해하기.

- step01 : vuex 사용.



```
computed: {  
  total() {  
    return this.$store.state.count;  
  }  
}
```

```
addCount: function() {  
  this.count += 1;  
  // this.$emit("addtotcount");  
  this.$store.state.count++;  
}
```

이어서..

삼성 청년 SW 아카데미

저장소 - Getters

삼성 청년 SW 아카데미

✓ 저장소 - Getters.

- component가 vuex의 state를 직접 접근하는 코드가 중복된다면??
- 해결 : Store의 state를 참조하는 Getters를 활용.
- 정의

```
getters: {  
  countMsg(state) {  
    state.count += 1;  
  }  
}
```

- 사용

```
this.$store.getters.countMsg;
```

✓ Vuex Store - Getters.



✓ 저장소 (Store) - Getters 등록.

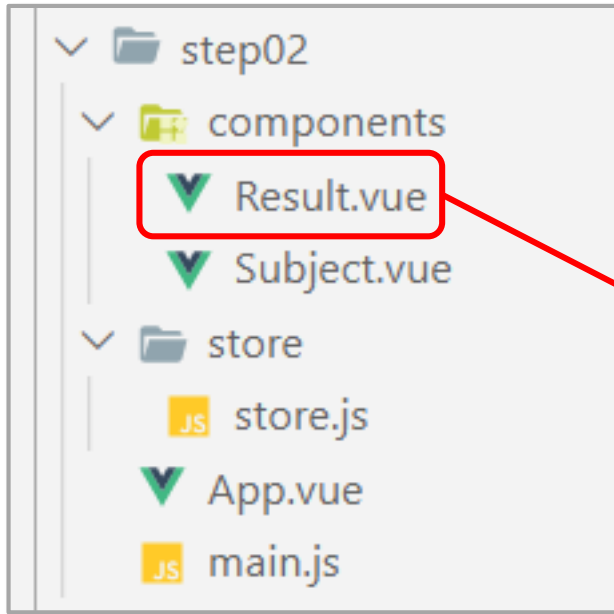
▪ step02



```
export default new Vuex.Store({
  state: {
    count: 0,
  },
  getters: {
    // 복잡한 계산식을 처리 : computed
    countMsg(state) {
      // return state.count + '번 호출됨';
      let msg = '10번보다 ';
      if (state.count > 10) {
        msg += '많이 ';
      } else {
        msg += '적게 ';
      }
      return msg + '호출됨(' + state.count + ')';
    },
  },
});
```

✓ 저장소 (Store) - Getters.

- step02



```
<script>
export default {
  computed: {
    total() {
      return this.$store.state.count;
    },
    countMsg() {
      return this.$store.getters.countMsg;
    }
  }
};
</script>
```

이어서..

삼성 청년 SW 아카데미

저장소 - mapGetters

삼성 청년 SW 아카데미

✓ 저장소 (Store) - mapGetters.

- getters를 조금 더 간단하게 호출.
- 주의 : Babel 관련 에러 발생시 해결

...mapGetters 관련

es6 spread operation 관련 에러 발생.

```
1. npm install --save core-js
2. Updated file babel.config.js :
  presets: [
    [
      '@vue/app',
      {
        useBuiltIns: 'entry',
      },
    ],
  ],
```

✓ 저장소(Store) - mapGetters.

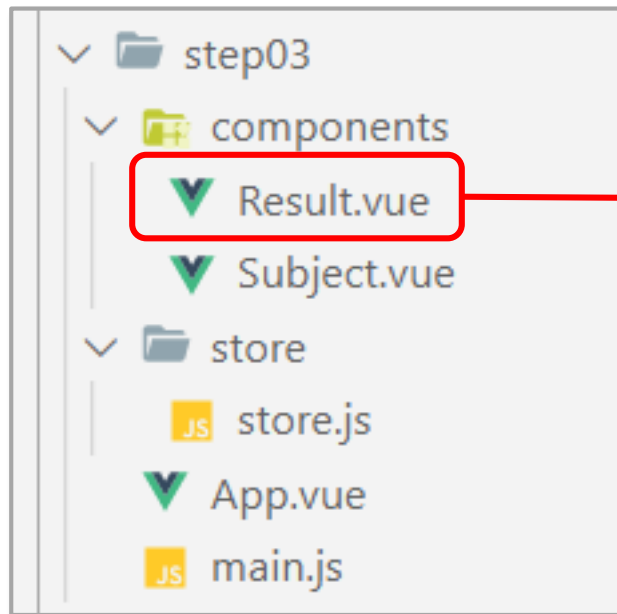
- step03



```
export default new Vuex.Store({  
  state: {  
    count: 0,  
  },  
  getters: {  
    countMsg(state) {  
      return ...;  
    },  
    msg1(state) {  
      return ...;  
    },  
    msg2(state) {  
      return ...;  
    },  
    msg3(state) {  
      return ...;  
    },  
  },  
});
```


✓ 저장소 (Store) - mapGetters.

▪ step03

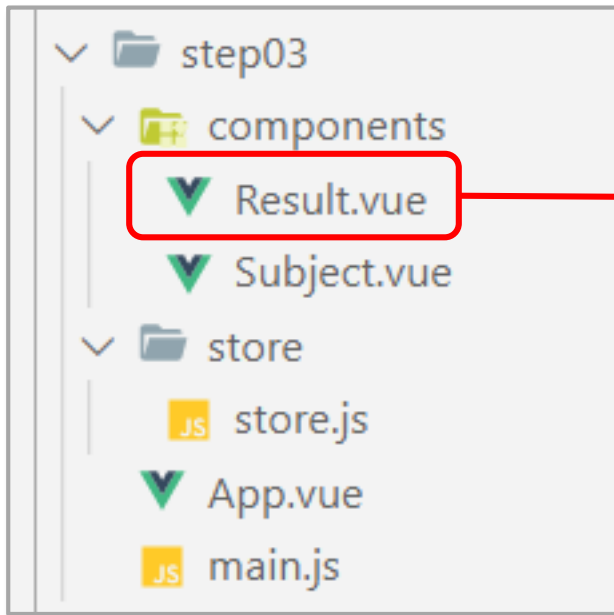


```
<h2>{{ $store.getters.msg1 }}</h2>  
<h2>{{ $store.getters.msg2 }}</h2>  
<h2>{{ $store.getters.msg3 }}</h2>
```

```
countMsg() {  
  return this.$store.getters.countMsg;  
},
```

✓ 저장소 (Store) - mapGetters.

▪ step03



```
<script>
import { mapGetters } from 'vuex';
export default {
  computed: {
    ...mapGetters(['countMsg', 'msg1', 'msg2', 'msg3']),
    total() {
      return this.$store.state.count;
    },
  },
}
```

이어서..

삼성 청년 SW 아카데미

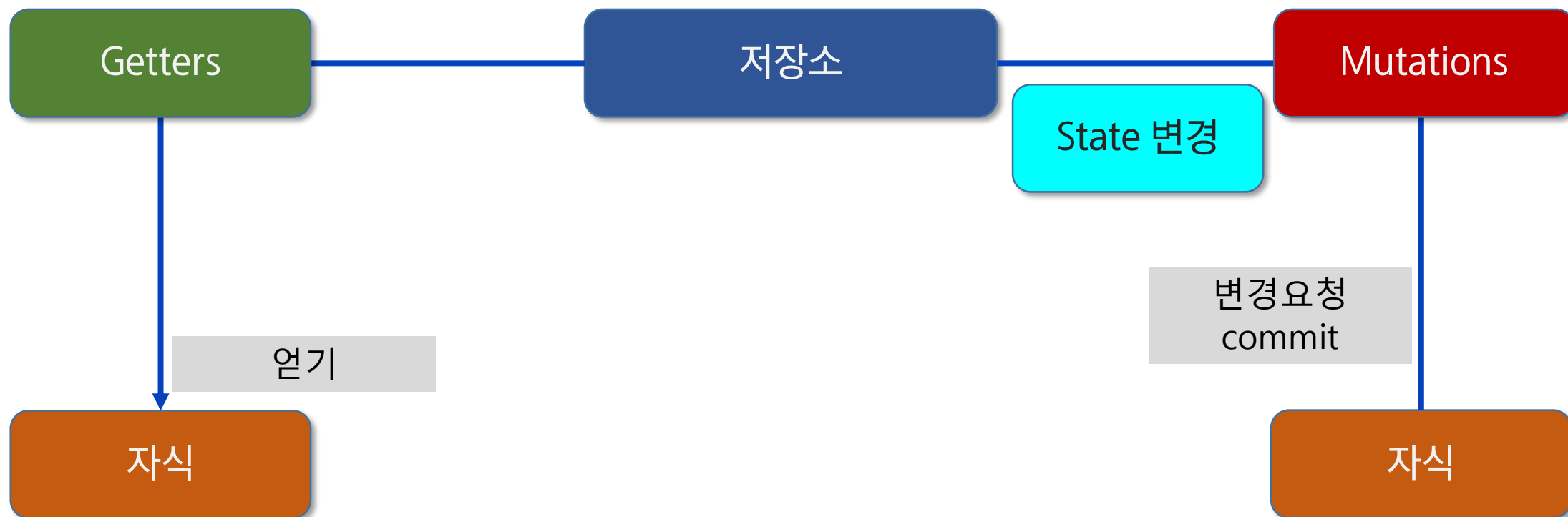
저장소 - Mutations

삼성 청년 SW 아카데미

✓ 저장소 (Store) - Mutations.

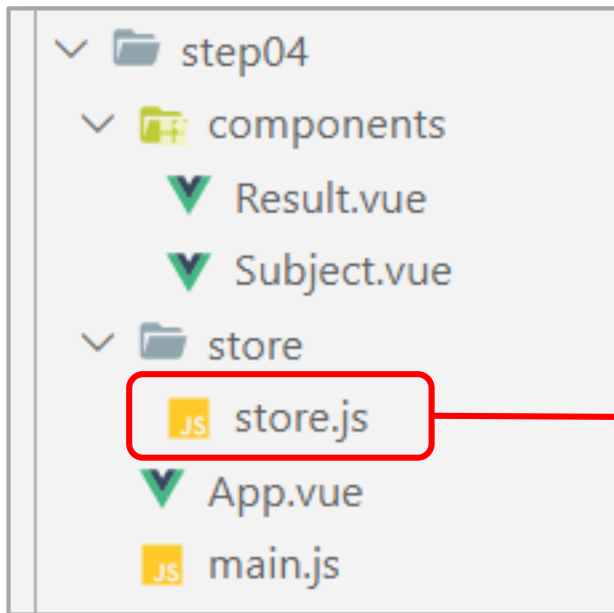
- State의 값을 변경하기 위해서 사용.
- 각 컴포넌트에서 State의 값을 직접 변경하는 것은 권장하지 않는 방식.
- State의 값의 추적을 위해 동기적 기능에 사용.
- Mutations는 직접 호출이 불가능하고 `store.commit('정의된 이름')`으로 호출.

✓ Vuex Store - Mutations.



✓ 저장소(Store) - Mutations 등록.

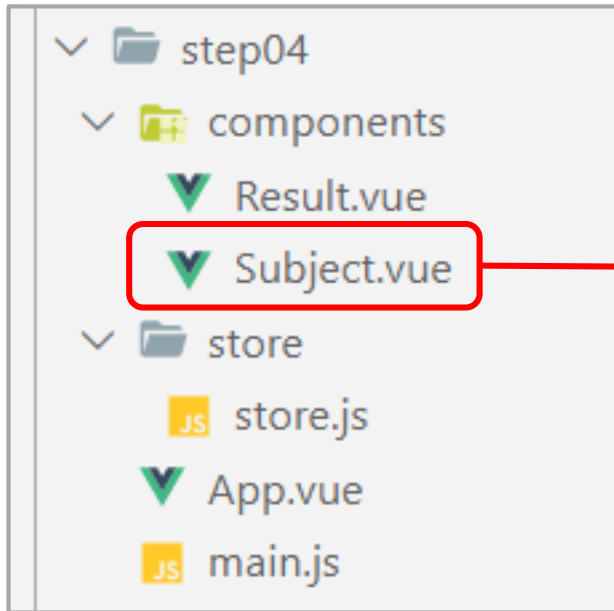
▪ step04



```
export default new Vuex.Store({
  state: {
    count: 0,
  },
  mutations: {
    addOne(state) {
      state.count += 1;
    },
    addCount(state, payload) {
      state.count += payload;
    },
    addObjCount(state, payload) {
      state.count += payload.num;
    },
  },
});
```

✓ 저장소(Store) - Mutations 조회.

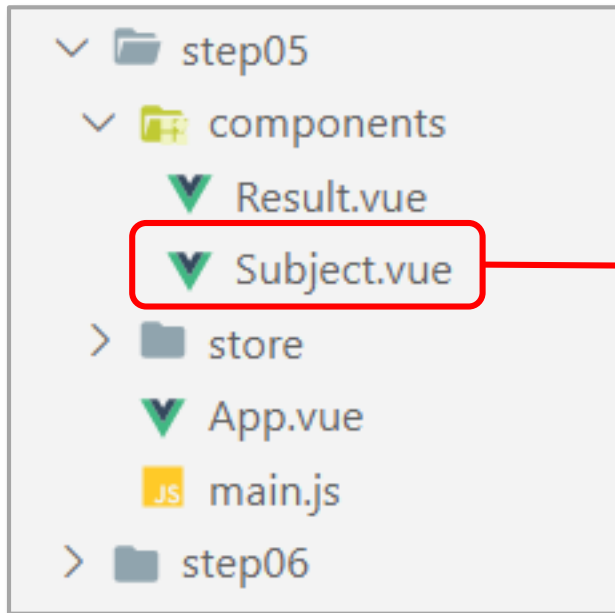
▪ step04



```
methods: {  
  addCount: function() {  
    this.count += 1;  
    this.$store.commit('addOne');  
    // this.$store.state.count++;  
  },  
  addTenCount: function() {  
    this.count += 10;  
    this.$store.commit('addCount', 10);  
  },  
  addObjCount: function() {  
    let num = Math.round(Math.random() * 100);  
    this.count += num;  
    this.$store.commit('addObjCount', { num });  
  },  
},
```


✓ 저장소(Store) - mapMutations.

- step05



```
import { mapMutations } from 'vuex';  
export default {  
  name: 'Subject',  
  props: ['title'],  
  data() {  
    return {  
      count: 0,  
    };  
  },  
  methods: {  
    ...mapMutations({  
      addMOne: 'addOne',  
      addMTenCount: 'addTenCount',  
      addMObjCount: 'addObjCount',  
    }),  
  },  
}
```

이어서..

삼성 청년 SW 아카데미

저장소 - Actions

삼성 청년 SW 아카데미

✓ 저장소(Store) - Actions.

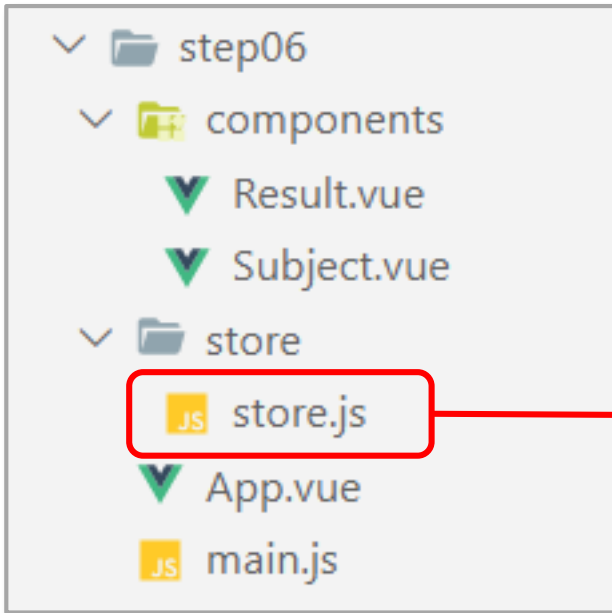
- 비동기 작업의 결과를 적용하려고 할 때 사용
- Mutations는 상태 관리를 위해 동기적으로 처리하고 비동기적인 처리는 Actions가 담당
- Actions는 비동기 로직의 처리가 종료되면 Mutations를 호출

✓ Vuex Store - Actions.



✓ 저장소(Store) - Actions 등록.

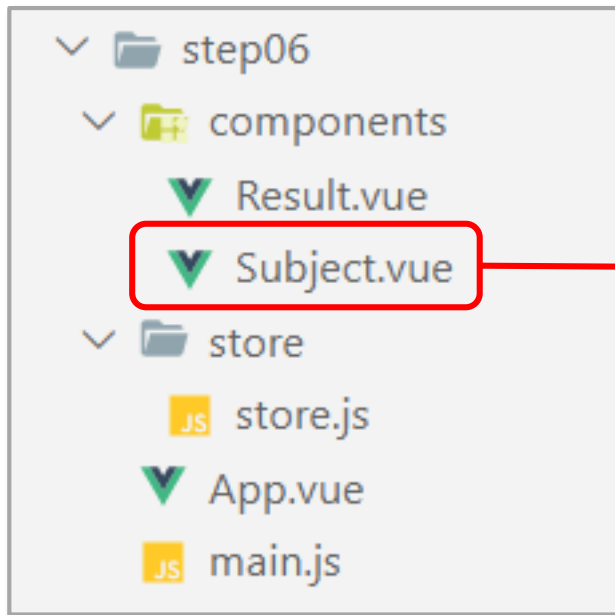
▪ step06



```
export default new Vuex.Store({
  state: {
    count: 0,
  },
  actions: {
    asyncAddOne(context) {
      setTimeout(() => {
        context.commit('addOne');
      }, 2000);
    },
  },
  mutations: {
```

✓ 저장소(Store) - Actions 호출.

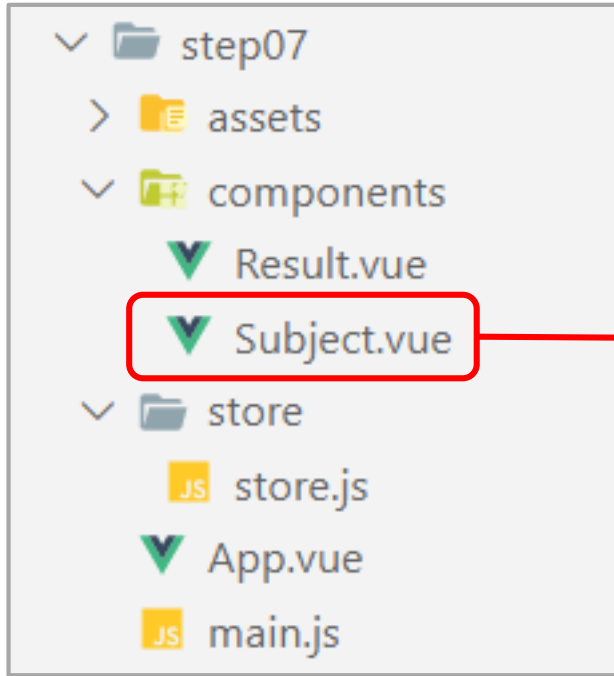
- step06



```
addObjCount: function() {  
  let num = Math.round(Math.random() * 100);  
  this.count += num;  
  this.addMObjCount({ num });  
},  
asyncCount() {  
  this.$store.dispatch('asyncAddOne');  
},
```

✓ 저장소(Store) - mapActions.

▪ step07



```
<script>
import { mapMutations, mapActions } from 'vuex';
export default {
  ...
  methods: {
    ...mapMutations({
      addMOne: 'addOne',
      addMTenCount: 'addTenCount',
      addMObjCount: 'addObjCount',
    }),
    ...mapActions(['asyncAddOne']),
    ...
  },
  asyncCount() {
    this.asyncAddOne();
  },
},
};
</script>
```


내일 방송에서 만나요!

삼성 청년 SW 아카데미