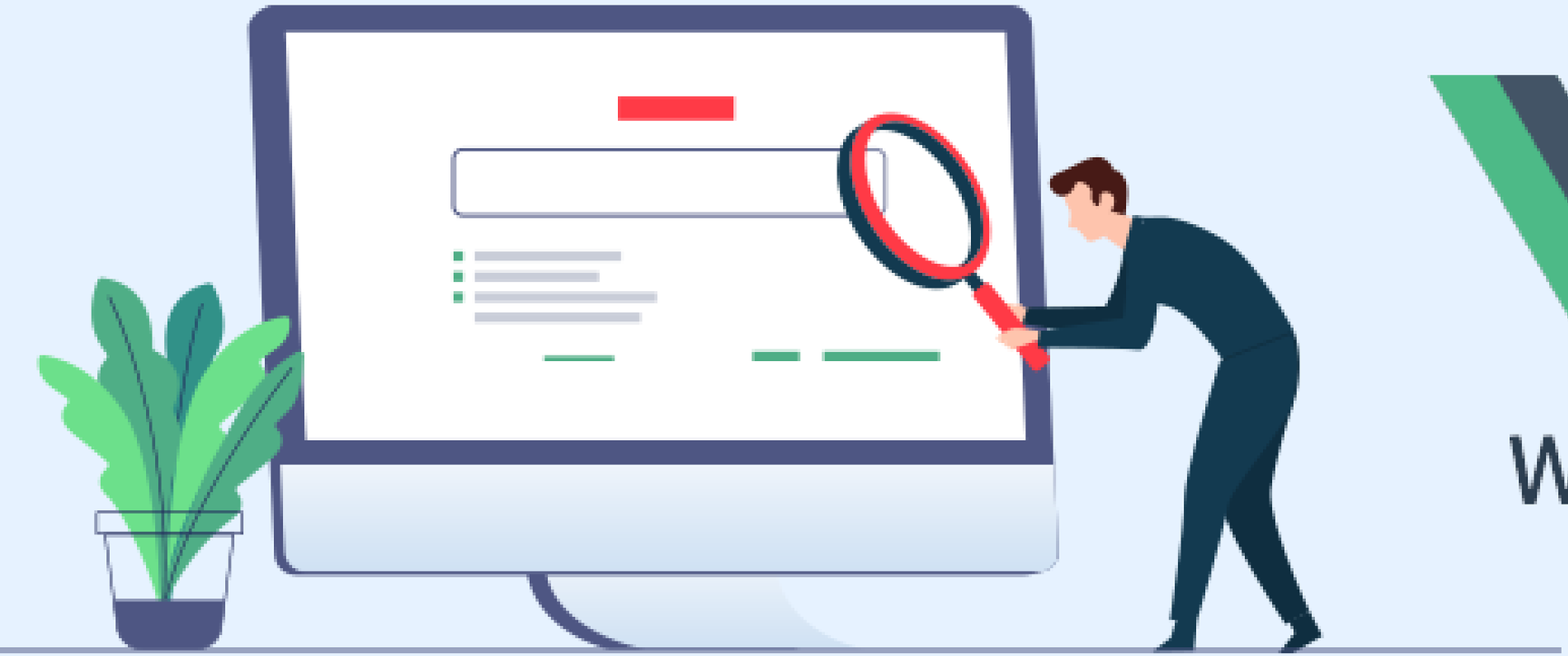


Vue.js

Web Application

CONTENTS

01 정의

Vue.js 란?

- Vue.js는 웹 개발을 단순화하고 정리하기 위해 개발된 대중적인 자바스크립트 프론트엔드 프레임워크이다.

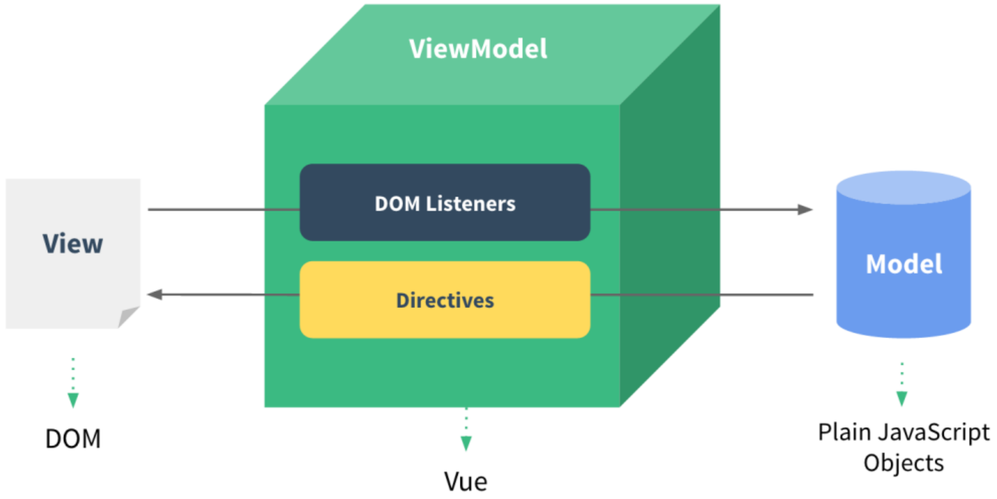
- 수많은 프로젝트에서 AngularJS를 사용하여 구글을 위해 작업하던 Evan You에 의해 개발되었다.

- 라우팅, 상태 관리, 빌드 도구화와 같이 복잡한 애플리케이션에 필요한 고급 기능들은 공식적으로 유지 보수되는 지원 라이브러리와 패키지를 통해 제공된다.

02 특징

Vue.js 특징

1. UI 화면단 라이브러리



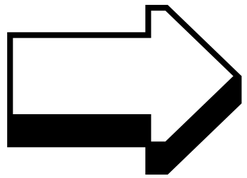
MVVM(Model - View - ViewModel)

02 특징

Vue.js 특징

2. 컴포넌트 기반 프레임워크

화면을 여러개의 작은 단위로 쪼개어 개발



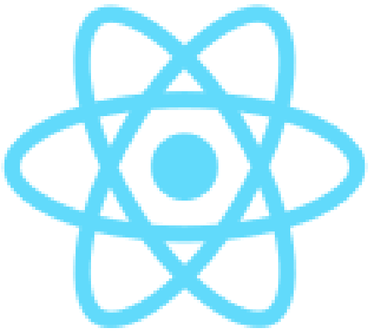
재사용성 ↑ , 구현 속도 ↑

코드 가독성 ↑

02 특징

Vue.js 특징

3. 양방향 데이터 바인딩 + 가상돔 기반 렌더링



화면에 표시되는 값과 프레임워크 모델 데이터 값이 동기화

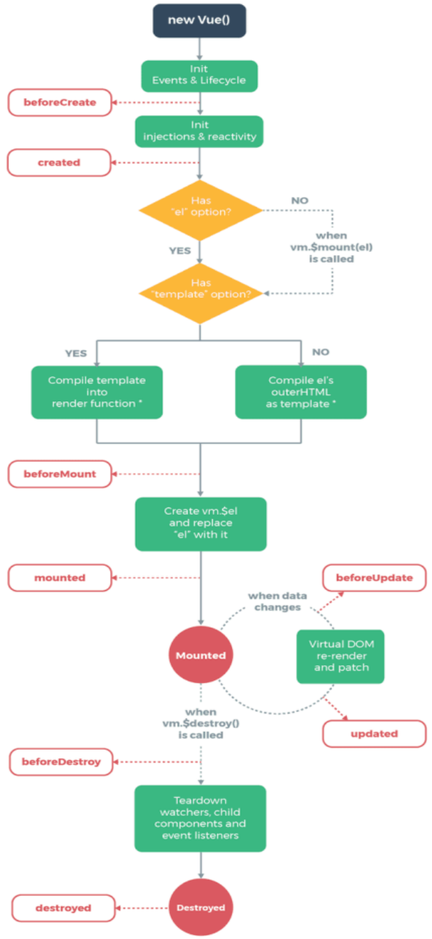
-> 한쪽이 변경되면 다른 한쪽도 자동변경

화면 전체를 다시 그리지 않고 프레임워크에서

정의한 방식에 따라 화면갱신

03 라이프사이클

Vue.js Lifecycle



* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

① Creation
컴포넌트들을 초기화 하는 단계

② Mounting
초기 렌더링 직전에 컴포넌트에 접근

③ Updating
컴포넌트에서 사용되는 반응형 속성들이 변경되거나 재렌더링 발생 시 실행

④ Destruction
해체 직전 및 이후 호출되는 단계

참고

https://velog.io/@hyeonjeong/Vue-%EB%B7%B0-%EB%9D%BC%EC%9D%B4%ED%94%84-%EC%82%AC%EC%9D%B4%ED%81%B4-life-cycle

04 Router

Vue-Router

Vue.js 에서 페이지 간 이동을 위한 라이브러리

페이지 이동할 때 url 변경되면, 변경된 요소의 영역에 컴포넌트를 갱신

설치하는 방법

① Script를 추가하여 사용

```
<script src="/scripts/vue-router.js"></script>
```

② node package 사용

```
npm install vue-router
```

04 Router

Vue-Router

```
import Vue from 'vue';
import VueRouter from 'vue-router';
import Home from '../views/Home.vue';

//Vue와 VueRouter 연결
Vue.use(VueRouter);

//우리가 사용할 route 생성 및 설정
const routes = [
  {
    path: '/',
    name: 'Home',
    component: Home,
  },
  {
    path: '/about',
    name: 'About',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/About.vue'),
  },
];

//VueRouter에 route를 등록하고 설정한다.
const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes,
});

//설정한 VueRouter 내보낸다.
export default router;
```

04 Router

Router 주소이동

① router-link

```
<router-link :to="/">Home<router-link>
<router-link :to="/siteList">SiteList<router-link>
```

② methods

$router.push()

```
var router = this.$router;

router.push('/siteList') // 이동 위치를 입력
router.push({ name: 'SiteList' }) // 해당하는 라우터 이름으로 이동
router.push({ path: '/siteList' }) // 해당하는 pathname을 입력하여 이동
```

05 Vuex

Vue-vuex

Vue.js의 상태 관리를 위한 패턴이자 라이브러리

여러 컴포넌트 간의 데이터 전달과 이벤트 통신을 한곳에서 관리

Vuex 구조

State : 상태

Getters: state값을 쉽게 접근

Mutations : state 값을 변경

Actions : 비동기 처리

05 Vuex

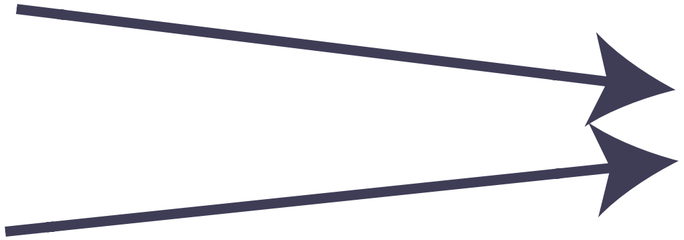
Vuex 접근

① State

$store.state

② Getters

$store.getters



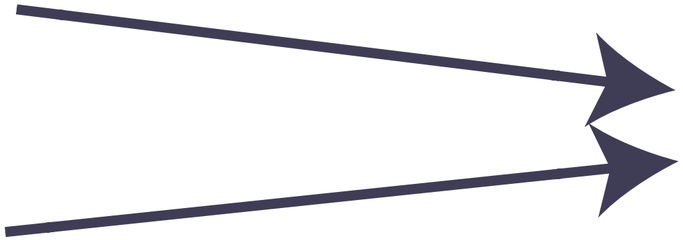
Computed

③ Mutation

$store.commit()

④ Actions

$store.dispatch()



Methods

05 Vuex

Vuex Helper

① state -> mapState

② getters -> mapGetters

③ mutations -> mapMutations

④ actions -> mapActions

```
import { mapState } from 'vuex';
import { mapGetters } from 'vuex';
import { mapMutations } from 'vuex';
import { mapActions } from 'vuex';

export default {
  computed() {
    ...mapState(['num']),
      ...mapGetters(['countedNum'])
  },
  methods: {
    ...mapMutations(['clickBtn']),
    ...mapActions(['asyncClickBtn'])
  }
}
```

Vuex-persisedstate

Vuex-persisedstate

Vuex의 상태는 메모리에 저장되는 것이기 때문에 새로 고침시 초기화

상태를 새로고침 해도 유지하기 위해서는 vuex-persisedstate 라이브러리를 설치 하여 해결

감사합니다