

1) package.json 에 관하여 <https://just-do-it-man.tistory.com/33> 추가로 읽을 것!!

- npm run build 는 vue 로 개발한 파일들을 최종적으로 운영환경에 빌드하기 전에 사용할 수 있는 파일들로 변환하는 명령어다.
- 우리가 npm 프로젝트를 실행할때 npm run serve를 입력했다. 여기서 serve를 run 한다는 것은 vue-cli-service serve 를 run하는 것과 같다.
- dependencies 부분은 운영환경에 배포할때 필요한 vue 관련 모듈
  - package-lock.json에서 볼 수 있다. 여기에 있는 모든 모듈들은 node\_modules에서 볼 수 있다.
- idea 디렉토리는 인텔리제이가 자동 생성해주지만 웹에 올릴때는 필요하지 않다.
- node\_modules 도 웹에 배포시 불필요
- Test\_Vue 에서 npm init : package.json 파일 생성
  - 노드와 모듈러등과 관련/관리할 내용에 대한 모든 파일 설정
  - "main": "index.js", : 메인을 index.js로 하겠다
  - package.json : 노드가 사용할 공간이 생겼다.
  - 순서 : npm init > 설정내용 선택 > package.json 자동 생성 > 노드 사용공간 생긴것
- npm install : package-lock.json 파일 생성 (package.json의 상세 버전)
  - package.json이 추가될수록 node가 업데이트한다?
  - package-lock.json : 노드가 프로그램을 설치하면서 의존성과 관련된 추가 설정
- Vue 설치 : npm install -g vue
  - 삭제 : npm uninstall -g @vue/cli
- CLI가 있어야 원활히 진행됨, Vue CLI 도 설치해야 됨 : npm install -g @vue/cli
- 현재 우리는 node 위에서, 자바가 브라우저가 아닌 런타임환경(node)에서 진행중
- npm i -g@vue/cli-init : 2.x 의 템플릿을 가져올때 사용

## 2) [Vue Instance 속성 \(method, filter, computed, watch\)](#)

### 2-1) [Vue method](#)

- Vue Instance : Vue 생성 관련된 data 및 method의 정의 가능
- method 안에서 data를 this.데이터이름 으로 접근 가능

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Vue.js</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
</head>
<body>
  <div id="app">
    <div>data : {{message}}</div>
    <div>method kor : {{helloKor()}}</div>
    <div>method eng : {{helloEng()}}</div>
  </div>
</body>
</html>

new Vue({ //vue 생성
  el: "#app",
  data: { // 사용할 data 모음
    message: "Hello",
    name: "여러분",
  },
  methods: { // 함수(method)의 모음
    helloEng() {
      return "Hello " + this.name;
    },
    helloKor() {
      return "안녕 " + this.name;
    },
  },
});
```

### 2-2) [Vue filter](#) : 뷰의 문법체계

- 화면에 표시되는 텍스트의 형식을 쉽게 변환해주는 기능
- filter를 이용하여 표현식에 새로운 결과 형식을 적용
- 중괄호 보관법 {{}} 또는 v-bind 속성에서 사용 가능
- 여러 개 사용 가능

```
<div v-bind:id="rwald | formatId"></div>
```

### 2-3) Vue computed : 계산된 결과값을 반영

- 특정한 데이터의 변경사항을 실시간으로 처리
- 캐싱을 이용하여 데이터의 변경이 없을 경우 캐싱된 데이터를 반환 (method와의 차이점)
- **Setter**와 **Getter**를 직접 지정할 수 있음
- 작성은 method 형태로 작성하지만 Vue에서 proxy 처리하여 property처럼 사용

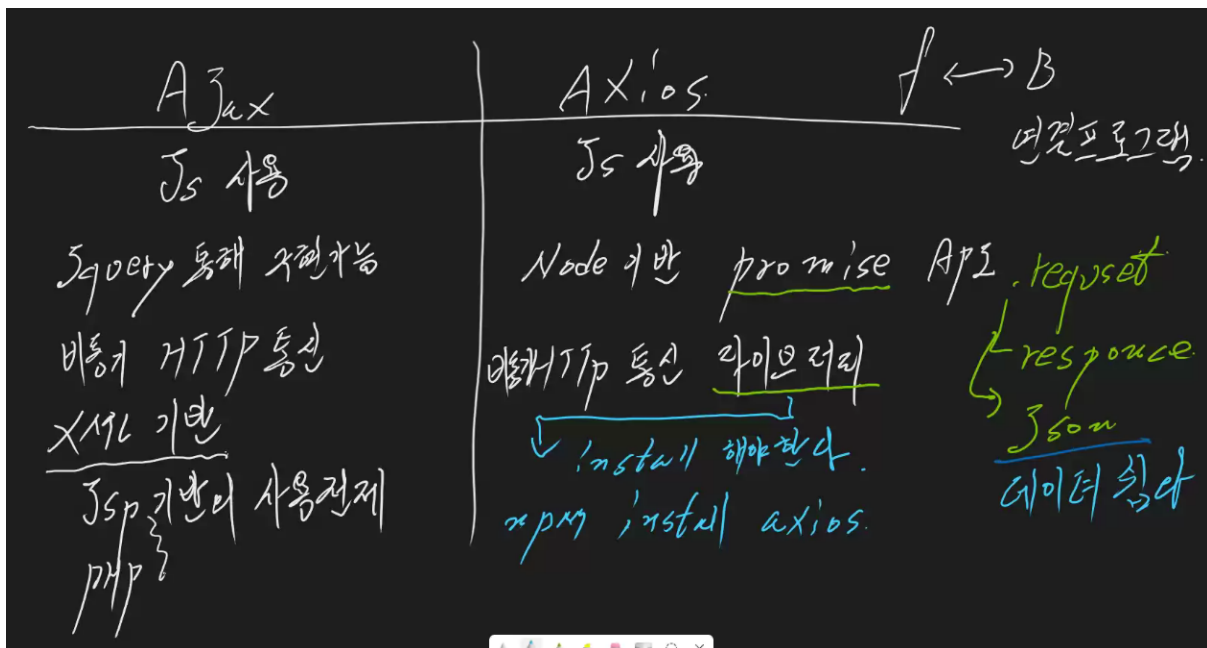
### 2-4) Vue watch

- Vue Instance의 특정 property가 변경될 때 실행할 콜백 함수 설정
- Computed는 종속된 data가 변경되었을 경우 그 data를 다시 계산하여 캐싱 (자기 자신)
- Watch의 경우는 data가 변경되었을 경우 다른 data를 (변경하는) 작업을 함

추후 이해도가 증진된다면 다음 스텝으로

- Event, Binding
- Component

### 3) Axios 에 관하여 <https://github.com/axios/axios>



[개발상식] Ajax, axios, fetch 차이점 장단점

- 설치 : `npm install axios`
- HTML 에 cdn 붙여야 가동됨
  - `<script src="https://cdn.jsdelivr.net/npm/axios@1.1.2/dist/axios.min.js"></script>`
  - ES5 : 표준자바스크립트 문법
- axios API
  - post 와 get 할때 필요
  - url 설정이 없으면 IP 주소로 반환
- node를 통해 install 과 uninstall 로 설치/삭제가 편한것이 node의 장점

## 금일 진행과정 정리

1. AWS에 EC2 인스턴스 생성

2. 리눅스 선택

3. 탄력적 아이피 연결

4. 푸티 보안키 접속 설정

5. 마리아디비 설치 및 환경설정

6.nvm 설치

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.36.0/install.sh | bash
```

```
$ source ~/.bashrc
```

7.nvm 노드 설치

```
nvm install 18.16.1
```

```
node -v 버전확인
```

8.jre 설치

AWS EC2 에서의 Java 설치(ubuntu) 29page 참고

9. product 파일의 dev\_class 의 파일과 일치하는 테이블 생성

- 20p : EC2 외부에서 db 접속을 위한 보안규칙 수정 3개 실행
  - mariadb 3306 (자동)
  - vi 편집기로 0.0.0.0
  - 접속할 아이디,비번 생성과 권한
- heidi > public ip 와 접속할 아이디 비번 넣고 접속
  - 파일 > SQL 파일 불러오기 > dev\_class\_2021-02-04 불러 > ▶ 실행
  - 재접속해서 db 확인 : 호스트 눌러 db 클릭하면 하부 테이블 확인 가능

차주 월요일 : db 생성 완료 상태에서 axios 연동진행

- [postman](#) 설치
- mockup server 구동 <https://antdevhistory.tistory.com/3>
- express 설치(web server) - node 기반
- nginx (web server) - node 기반