AXIOS

> 비동기 요청을 간편하게 처리할 수 있도록 하는 JavaScript 라이브러리

- XMLHttpRequest, fetch를 대체하여 사용할 수 있다
- Promise 객체를 활용하고 있다
  - `then()` , `catch()` 를 그대로 사용할 수 있음

Vue Project에 적용

1. vue project 폴더 내에서 npm으로 axios 라이브러리 설치하기

```
$ npm i axios
```

2. main.js에서 전역변수 등록 (**!!주의!! 라이브강의에서 설명한 방법과 다름)**

```
import { createApp } from "vue";
import { createPinia } from "pinia";

import App from "./App.vue";
import router from "./router";
import axios from "axios";

const app = createApp(App);

app.use(createPinia());
app.use(router);

app.provide('axios', axios);

app.mount("#app");
```

3. Vue Componet에서 axios 사용하기

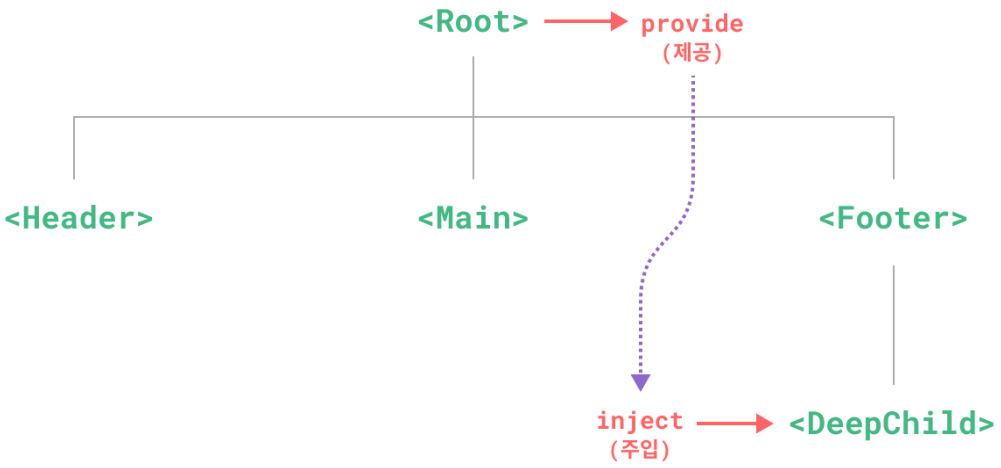
```
<script setup>
import { ref, inject } from 'vue';

const axios = inject('axios');
```

```
const imgUrl = ref('');

const getCatImg = () => {
  imgUrl.value = 'loading';

  axios.get('https://api.thecatapi.com/v1/images/search')
    .then(({data}) => {
      imgUrl.value = data[0].url;
    });
}
</script>

<template>
  <div class="cat">
    <h1>이것은 CatView입니다.</h1>
    <button @click="getCatImg">고양이 사진을 가져올래</button>
    <div class="catImage">
      <p v-if="imgUrl == 'loading'">로딩중... </p>
      <img v-else :src="imgUrl">
    </div>
  </div>
</template>

<style scoped>
  ...
```

Provide / Inject (싸피커리큘럼 상에는 없음)

> 자손 컴포넌트에 의존성을 주입할 수 있도록 해주는 API

- 대부분의 data는 `props / emit` 또는 `Pinia` 의 `store`를 활용하여 공유할 수 있지만, App 전체에 필요한 의존성의 경우는 app수준에서 provide하여 각 컴포넌트마다 inject 해주는 것이 좋다.



AXIOS 사용법

- 요청 보내기

```
axios.get(url);

axios.post(url, requestBody);
```

- 응답 받기
  - AXIOS는 Promise 객체를 활용하고 있으므로 비동기 응답을 받은 후 `then()` 을 사용하여 처리한다

```
  axios.get('https://api.thecatapi.com/v1/images/search')
    .then((response) => {
      imgUrl.value = response.data[0].url;
    });
```

- 응답의 구조

```
{
  data: {},   // `data`는 서버가 제공하는 응답입니다.
  status: 200,   // `status`는 HTTP 상태 코드입니다.
  statusText: 'OK', // `statusText`는 HTTP 상태 메시지입니다.

  // `headers`는 HTTP 헤더입니다.
  // 모든 헤더 이름은 소문자이며, 괄호 표기법을 사용하여 접근할 수 있습니다.
  // 예시: `response.headers['content-type']`
```

```
  headers: {},

  config: {},   // `config`는 요청을 위해 `Axios`가 제공하는 구성입니다.

  // `request`는 이번 응답으로 생성된 요청입니다.
  // 이것은 node.js에서 마지막 ClientRequest 인스턴스 입니다.
  // 브라우저에서는 XMLHttpRequest입니다.
  request: {}
}
```