- En
- 기차프로젝트 리액트 전환
- 기존 JS구조에서 Jquery로 전환
- Html 기반 구조에서 컴포넌트 형태로 전환
- 이미지 문자정보 등 데이터화



프론트엔드 과정 3차 프로젝트

문승준

3차 프로젝트 소개

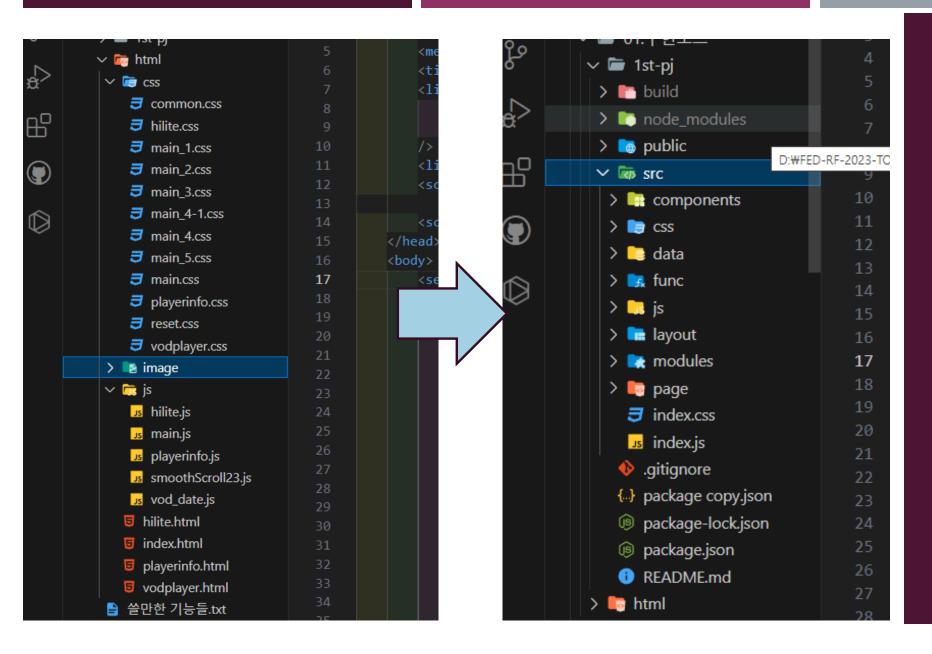
3차 프로젝트는 1차프로젝트의 리액트화를 중점으로 두고 개발. 앞으로의 작업의 편의성과 추가적인 기능추가에 용이하기에 이러한 목표를 설정.

변화된점

전체적으로 부드러워진 기능들.

제이쿼리 사용으로 함수제작 간편화됨

리액트 사용으로 컴포넌트별로 관리가 용이함.



기존 HTML 구조에서 리액트 형식으로 전환

장점

컴포넌트별 관리가 용이 추가페이지 제작에 유리

데이터 JS관리로 원하는 내용들을 관리

모듈 및 기능함수관리에 용이

```
addevt(ele, "mouseover", overln);
 addevt(ele, "mouseout", outFn);
function overFn() {
 let posL = this.offsetLeft;
 console.log("위치", posL);
 mbc.style.opacity = 1;
 mbc.style.left = posL + "px";
function outFn() {
 console.log("아웃", this);
 mbc.style.opacity = 0;
const before = dqs(".before");
let hcode = "";
for (let i = 0; i < 10; i++) {
 hcode +=
   <img src="./image/logo/tlogo${i</pre>
   hcode =
```

```
import $ from 'jquery';
window.jQuery = $;
require('jquery-ui-dist/jquery-ui');
```



```
function outFn() {
  const mlogo = $('.mlogo');
  const cube = $('.cube');
  $('.mlogo').removeClass('on');
  cube.removeClass('on');
  $('.big').removeClass('big');
  cbtn.hide();
```

JS 문법에서 Jquery 문법으로 전환

For문 으로 뿌려지던 이미지를 객체와 하여 맵핑으로 전달

장점 객체 데이터가 늘어나도 추가적인 수정 불필요

기존에 구현하기 힘들었던 기능을 쉽게 구현 가능



기존 왼쪽으로 이동하는 함수를 사용하는 방식에서 swiper 모듈을 활용한 미리보기 게시판



제이쿼리 작업으로 애니매이션 추가가 자유로워짐. 자연스러운 화면전환

3차 프로젝트를 마치며...

리액트로 전환하는 과정에서 배운점이 많음. 제이쿼리를 사용함으로써 제이쿼리의 편리함을 배움.

감사합니다.

