

---

김도은

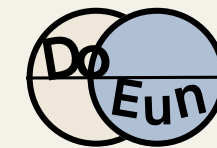
# 포트폴리오 사이트를 소개합니다.

---

사이트 바로가기

<https://kde5813930.github.io/Portfolio-Doeun/>

kde5813@gmail.com



# 목차

01 소개

---

02 메인 페이지

---

03 ABOUT 페이지

---

04 PROJECT 페이지

---

05 STUDY 페이지

---

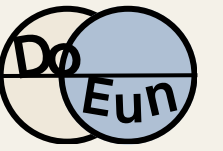
06 CONTACT 페이지

---

07 끝 마침

---

# 소개



안녕하세요!

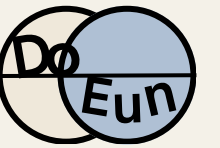
저는 **김도은**입니다.

함께 협력하며 맡은 업무에 최선을 다합니다.  
배움에 가치를 느끼고 발전함을 중요시 합니다.

저의 포트폴리오 사이트의  
주요 기능들을 정리하였습니다 :)



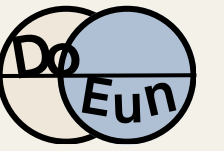
# 메인 페이지



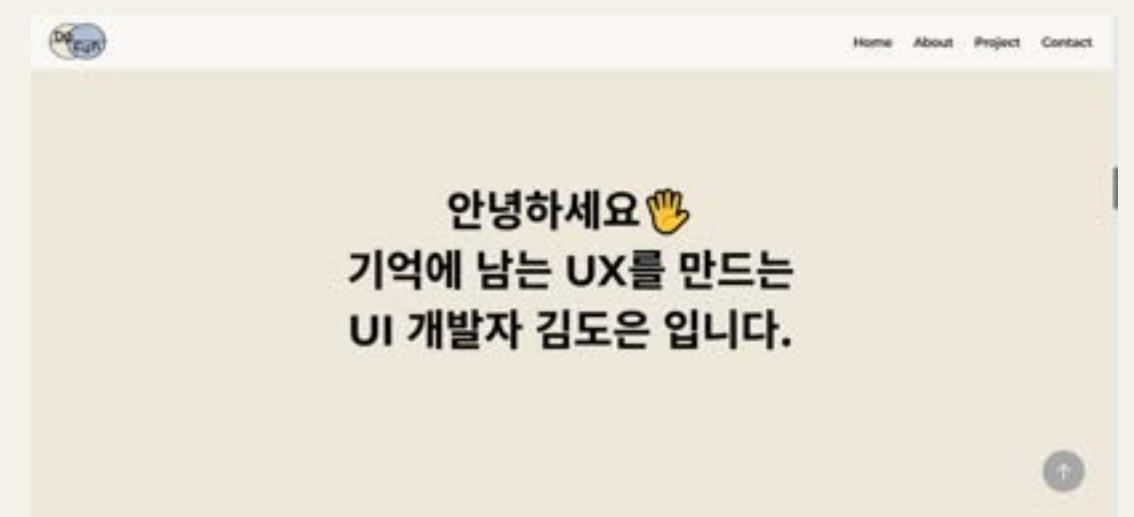
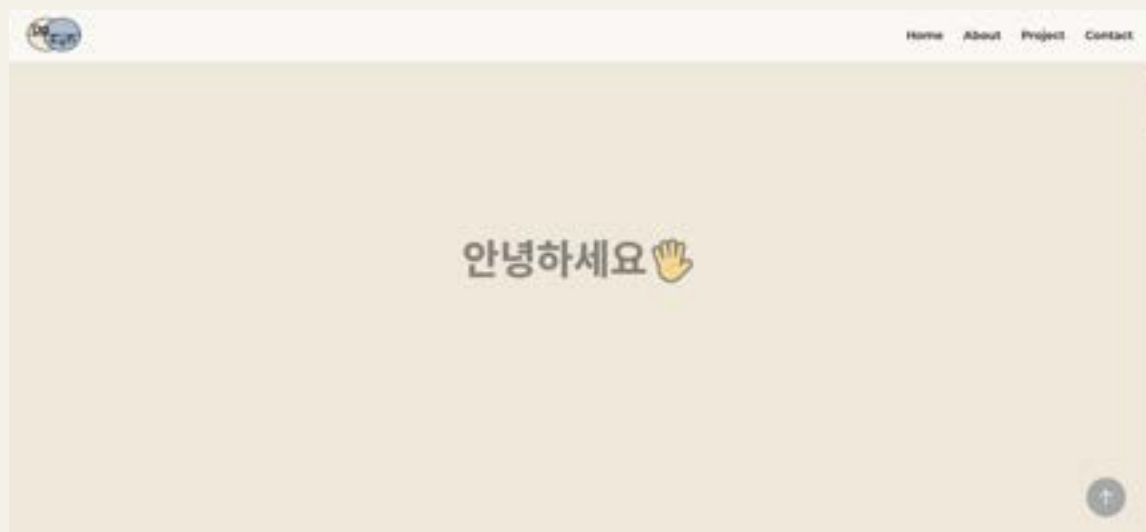
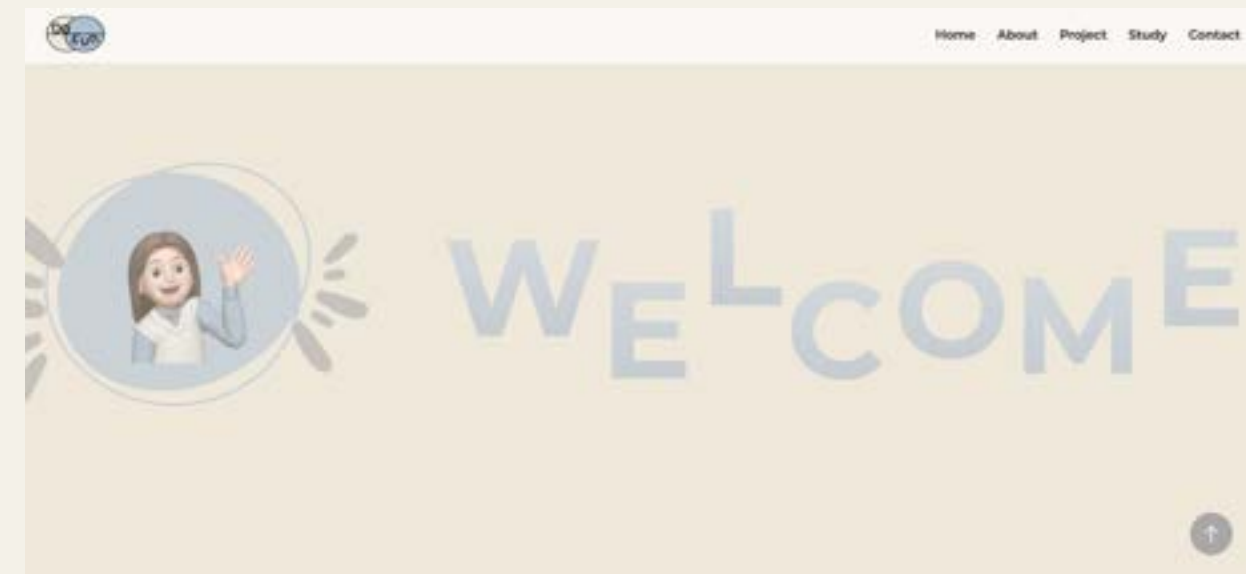
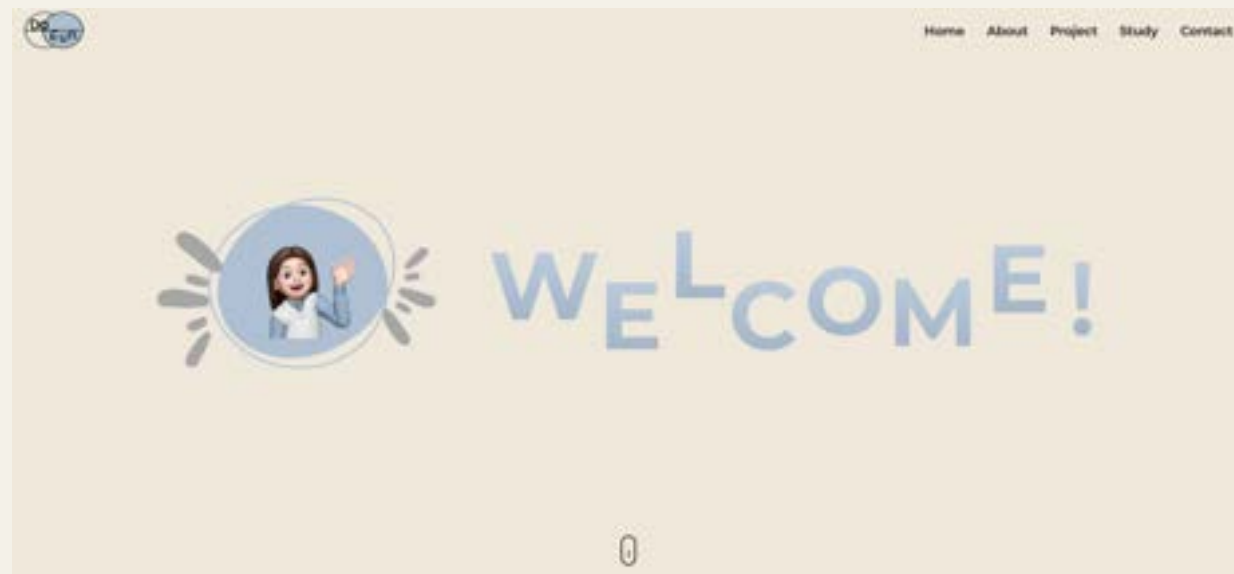
```
.main-tit-wrap .main-tit span{position:absolute;bottom:-248px;left:0;font-size:13rem;font-weight:500;display:inline-block;width:100%;  
-webkit-background-clip: text;  
color: transparent;  
background-color:var(--blue);  
background-image: linear-gradient(  
    180deg,  
    #d9d9d9 9%,  
    #d9d9d9 22%,  
    #e8e2c8 39%,  
    #e8d8b7 54%,  
    #c3d1e1 69%,  
    var(--blue) 86%,  
    #a3b3c6  
);  
background-size: 100% 300%;  
background-position-y: 0;  
animation-fill-mode: forwards, both;  
animation-name: cu-fadeUp, gradient;  
animation-delay: 0s, .5s;  
animation-duration: .5s, 3s;  
animation-timing-function: ease-in-out, ease-in-out;
```

메인 페이지 처음 접속 시 이미지와 타이틀 문구(welcome)를 css 애니메이션을 이용하여 모션을 적용하였습니다.  
이미지가 먼저 fade in 되고, 텍스트를 등장 시킨 후 텍스트의 color를 그라데이션 처리 해주었습니다.

# 메인 페이지



이후 페이지를 스크롤 하면 메인 문구인 welcome 이 화면에 꽉 차게 확대 되면서 점점 사라지고 서브 문구가 한 문단씩 나오게끔 만들었습니다.



// 메인 스크롤 이벤트

```
const mainTitle = document.querySelector(".main-tit-wrap");
const subTitle_1 = document.querySelector(".sub-tit .no-1");
const subTitle_2 = document.querySelector(".sub-tit .no-2");
const subTitle_3 = document.querySelector(".sub-tit .no-3");
const scrollWrap = document.querySelector(".scroll-wrap");
```

```
mainTitle.classList.add("active");
```

```
window.addEventListener("scroll", () => {
```

```
    let hei = window.pageYOffset;
```

```
    let textOpacity = hei * -1/1000 + 1;
```

```
    let titleScale = hei * 1/1000 + 1;
```

```
    let subTextOpacity = hei * 1/500;
```

```
    let textScroll = hei * -1/5
```

```
    if (hei <= 0) {
```

```
        mainTitle.style.opacity = "1";
```

```
        mainTitle.style.scale = "1";
```

```
        scrollWrap.style.opacity = "1";
```

```
    }
```

```
    if (hei > 0) {
```

```
        mainTitle.style.opacity = textOpacity;
```

```
        mainTitle.style.scale = titleScale;
```

```
        scrollWrap.style.opacity = "0";
```

```
    }
```

```
    if (hei > 1000){
```

```
        mainTitle.style.opacity = 0;
```

```
        mainTitle.style.scale = 2;
```

```
        mainTitle.classList.remove("active");
```

```
        subTitle_1.style.opacity = subTextOpacity - 2;
```

```
        subTitle_1.style.transform = `translateY(${textScroll + 300}px)`;
```

```
    } else {
```

```
        mainTitle.classList.add("active");
```

```
        subTitle_1.style.opacity = 0;
```

```
    }
```

```
    if (hei > 1500){
```

```
        subTitle_1.style.transform = `translateY(0)`;
```

```
        subTitle_2.style.opacity = subTextOpacity - 3;
```

```
        subTitle_2.style.transform = `translateY(${textScroll + 400}px)`;
```

```
    } else {
```

```
        subTitle_2.style.opacity = 0;
```

```
    }
```

```
    if (hei > 2000){
```

```
        subTitle_2.style.transform = `translateY(0)`;
```

```
        subTitle_3.style.opacity = subTextOpacity - 4;
```

```
        subTitle_3.style.transform = `translateY(${textScroll + 500}px)`;
```

```
    } else {
```

```
        subTitle_3.style.opacity = 0;
```

```
    }
```

```
    if (hei > 2500){
```

```
        subTitle_3.style.transform = `translateY(0)`;
```

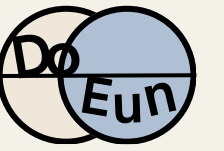
```
    }
```

```
});
```

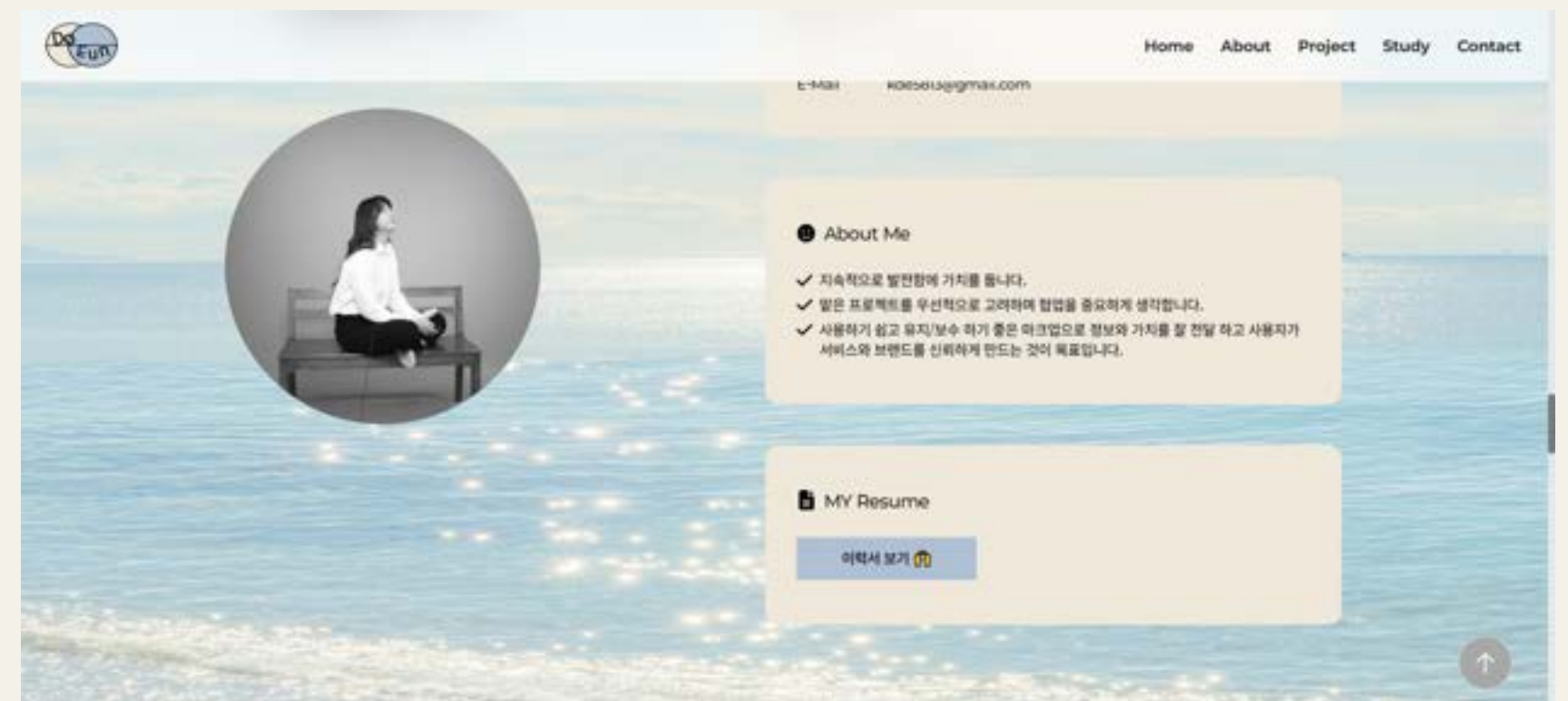
메인 스크롤 이벤트 작업시 방정식을 이용하여 스크롤이 될때마다 opacity 및 scale 등의 스타일의 수치를 조금씩 늘리면서 자연스럽게 애니메이션이 넘어가게 하였습니다.



# About 페이지



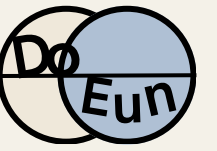
About section은 저에 대한 소개와 정보들을 소개하는 페이지 입니다.



css를 이용하여 스크롤 효과를 적용해 보았습니다.

높이값은 모든 요소를 포함한 높이로 지정하였고, 배경과 이미지를 고정하여 스크롤 될 때 오른쪽 박스만 스크롤시 내려가게끔 만들었습니다.

# About 페이지



## html

```
<div id="About" class="about">
  /* 스크롤시 고정되는 bg */
  <div class="about-bg"></div>
  <div class="about-wrap section-inner">
    <div class="about-content" data-aos="fade-up">
      /* 왼쪽 title 및 img */
      <div class="about-title-wrap">
        <h3 class="main-title">About Me</h3>
        <div class="img-profile-wrap"></div>
      </div>
      /* 오른쪽 description box */
      <ul class="about-info-list" data-aos="zoom-in-left" data-aos-delay="150">
      </ul>
    </div>
  </div>
</div>
```

## CSS

```
/* about */
.about{position:relative;width:100%;height:2150px;}

.about .about-bg{position:sticky;left:0;top:0;z-index:-1;width:100%;height:100vh;background-image:url('./images/about/bg_about.png');
background-repeat:no-repeat;background-size:cover;background-position:center center;}

.about .about-bg:after{content:'';position:absolute;left:0;top:0;width:100%;height:100%;background-color:rgba(255, 255, 255, 0.5)}

.about .about-wrap{position:absolute;left:calc(50% - 640px);top:0;padding:0 20px;}
.about .about-wrap .about-content{position:relative;display:flex;justify-content:space-between;}

.about-wrap .about-content .about-title-wrap{position:sticky;top:0;height:100%;padding:150px 0 18% 0;}

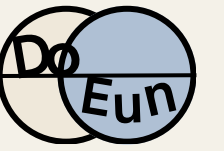
.about-content .about-title-wrap .title{font-size:4rem;}
.about-content .about-title-wrap .img-profile-wrap .img-profile{width:350px;height:350px;border-radius:50%;object-fit:cover;}

.about-wrap .about-content .about-info-list{width:calc(100% - 600px);height:100%;padding-top:250px;}
```

- ① 전체 높이값을 요소 전체 크기에 맞게 지정해 줬습니다.
- ② 해당 섹션에 고정해 놓을 bg를 position:sticky 로 고정해주고, 높이값을 100vh로 뷰포트에 맞게 지정하여 섹션이 끝날때까지 bg를 고정시켜 주었습니다.
- ③ 왼편의 title과 img도 position:sticky 를 적용하여 고정 시켜줬습니다.



# About 페이지

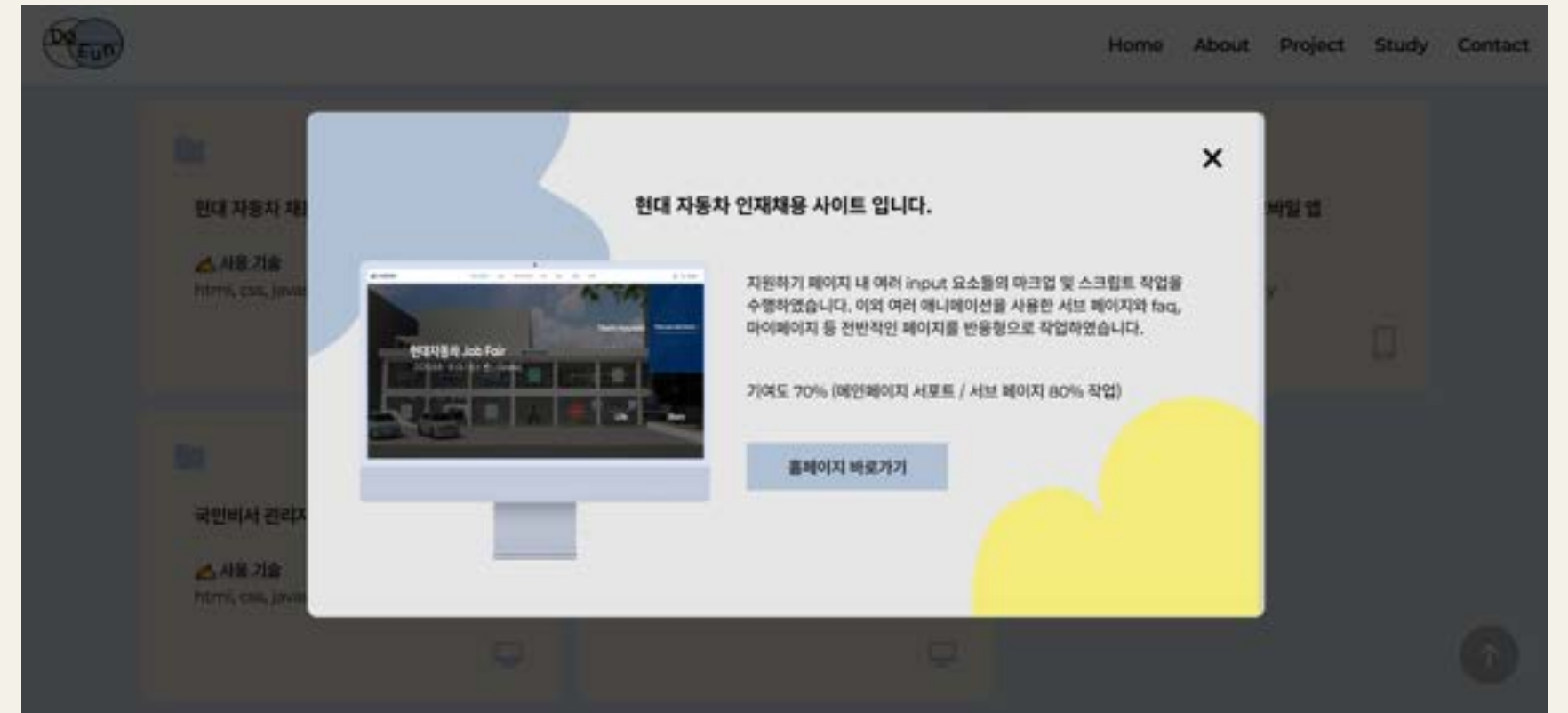
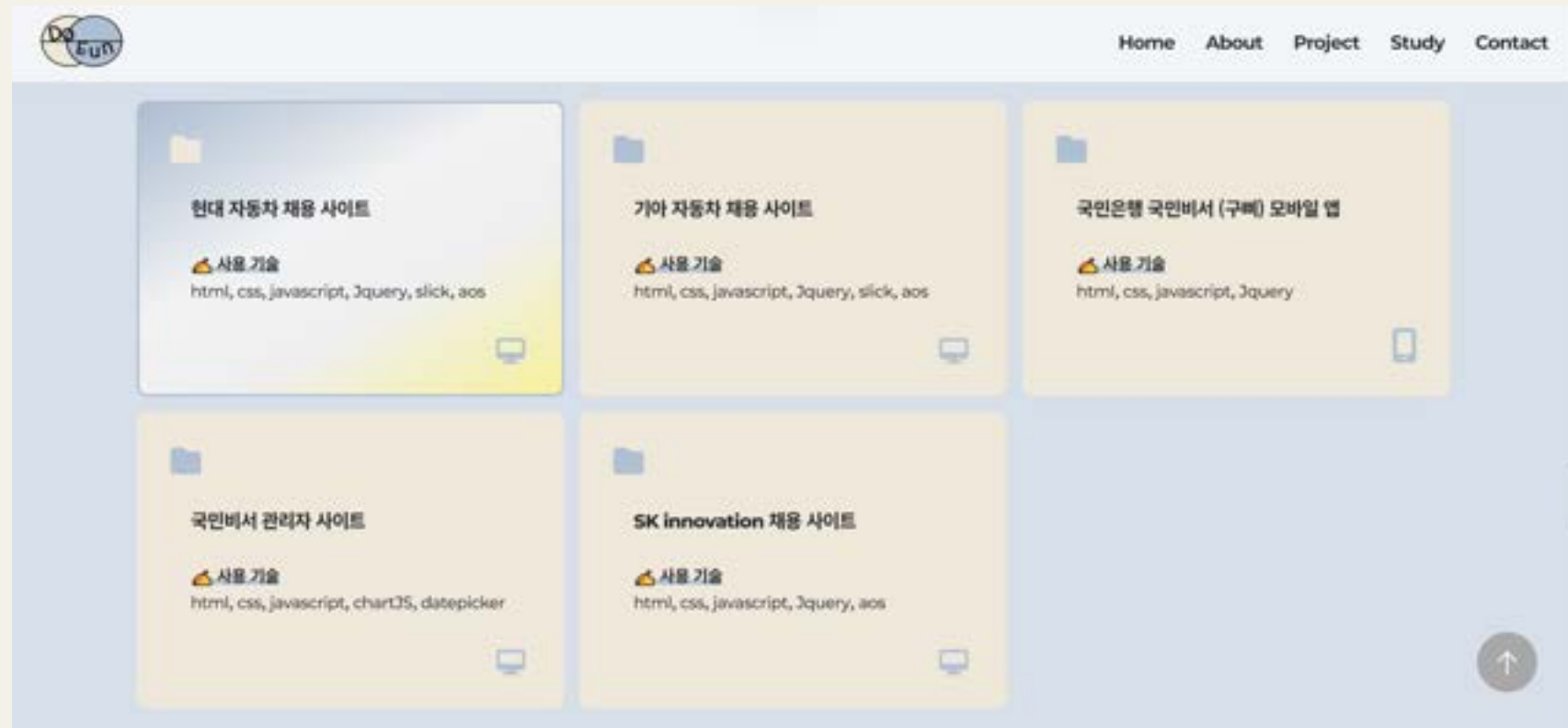
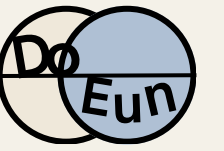


```
/* about section 모바일에서와 슬라이드 불러오기 */  
let ww = $(window).width();  
let mySwiper = undefined;  
  
function initSwiper() {  
  if (ww < 767 && mySwiper === undefined) {  
    mySwiper = new Swiper(".swiper-container", {  
      slidesPerView: 1,  
      spaceBetween: 20,  
      simulateTouch: true,  
      centeredSlides: true,  
      loop: true,  
      pagination: {  
        el: ".swiper-container .swiper-pagination",  
        type: 'bullets',  
      },  
    });  
  }  
  else if (ww >= 767 && mySwiper !== undefined) {  
    mySwiper.destroy();  
    mySwiper = undefined;  
  }  
}  
  
initSwiper();
```

```
$(window).on('resize', function () {  
  ww = $(window).width();  
  initSwiper();  
});
```

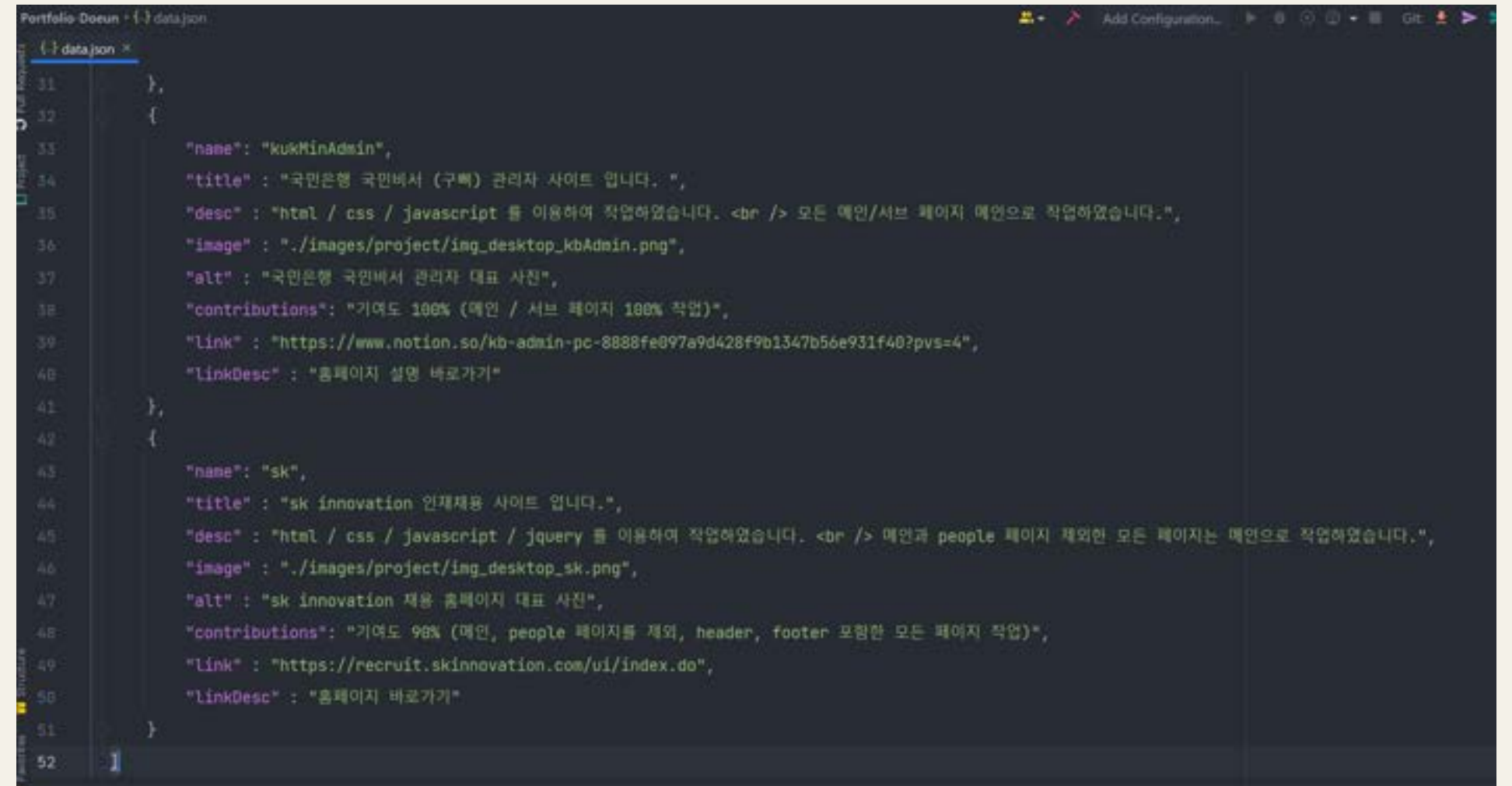
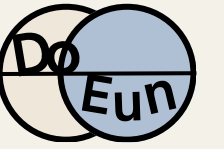
모바일에서는  
좀 더 깔끔하고 눈에 잘 들어오도록 하기 위하여  
오른쪽 박스 리스트를  
모바일 버전에서만 슬라이드를 적용 시켜  
주었습니다.

# Project 페이지



project section은 제가 이제까지 작업하였던 프로젝트들을 설명하는 페이지 입니다.  
각 리스트를 클릭하면 모달창이 나오고, 해당 프로젝트에 관련된 내용들과 링크가 정리되어 있습니다.

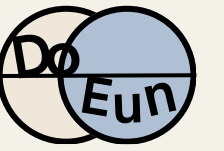
# Project 페이지



modal의 마크업은 동일하지만 각 리스트마다 내용과 이미지가 다르기에 웹사이트 성능을 고려하여 리스트 클릭시 데이터를 불러와 해당 리스트의 내용과 맞게 데이터를 불러오게 하고, 마크업은 동적으로 추가되게 작업하였습니다.  
하여, json파일에 각각 리스트의 데이터들을 정리해 놓았습니다.



# Project 페이지



```
//project 모듈링
let $html = document.querySelector("html");
const container = document.querySelector(".wrap")
const projectBox = document.querySelectorAll(".project-box .box-inner");

fetch("data.json")//json파일 읽어오기
.then((response) => response.json())//읽어온 데이터를 json으로 변환
.then((data) => {
  data.forEach((element) => {
    //이미지 미리 불러오기
    preloading([
      "./images/project/img_desktop_hyundai.png",
      "./images/project/img_desktop_kia.png",
      "./images/project/img_desktop_kbAdmin.png",
      "./images/project/img_mobile_kb.png",
      "./images/project/img_desktop_sk.png",
    ]);

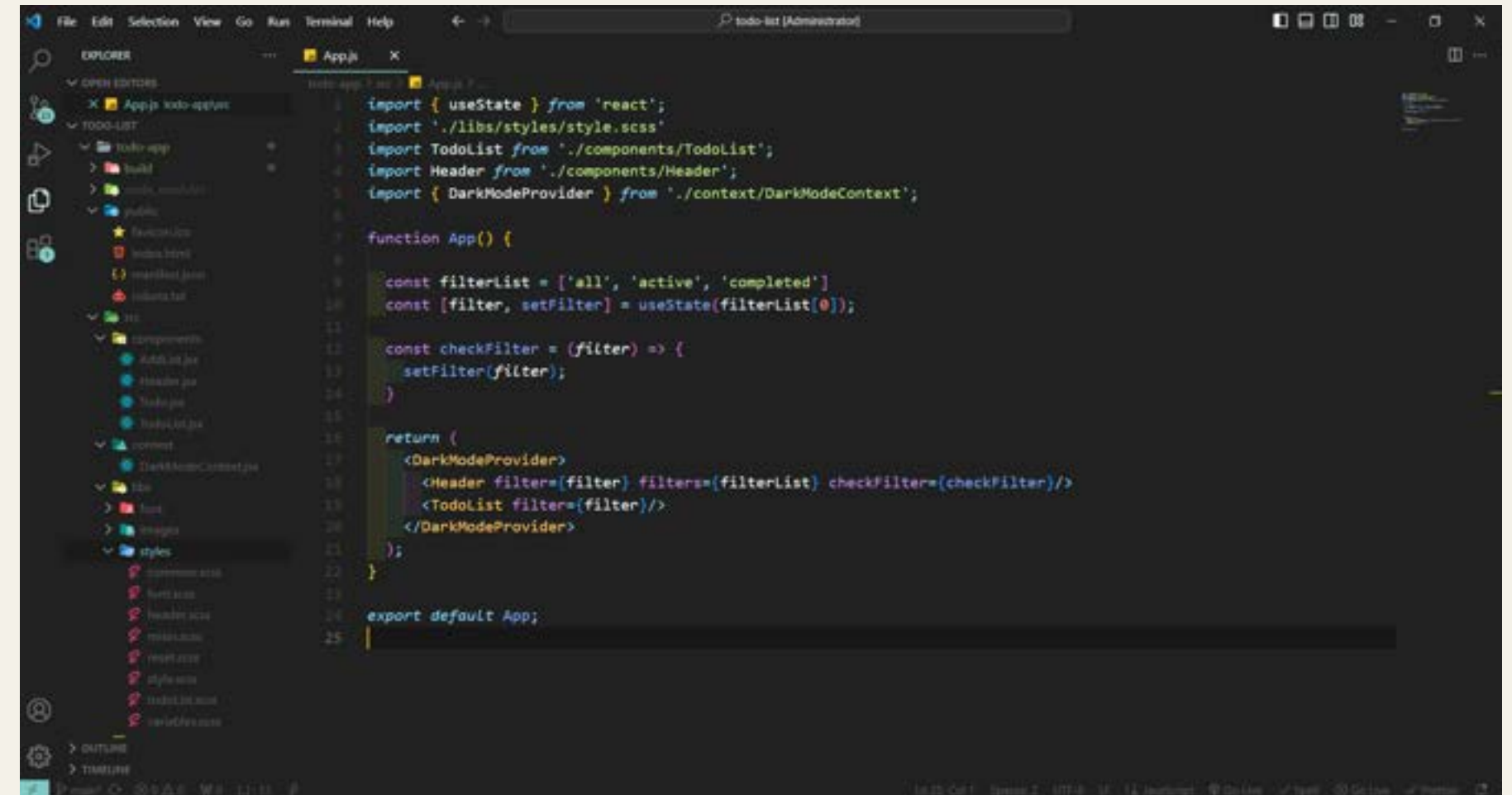
    projectBox.forEach((boxes) => {
      boxes.addEventListener("click", (e) => {
        const target = e.currentTarget;
        let projectName = target.dataset.name;

        if (element.name === projectName){
          $html.classList.add("lock");
        } else{
          return false;
        }
      });
    });
  });
});

$html.classList.add("lock");

container.insertAdjacentHTML("beforeend", `
<div class="modal-wrap">
  <section class="modal ${element.name}">
    <header class="modal-header">
      <h2 class="title">${element.title}</h2>
    </header>
    <div class="modal-content">
      <div class="modal-inner">
        <div class="modal-img-wrap">
          </div>
        <div class="modal-desc-wrap">
          <p class="desc">${element.desc}</p>
          <p class="desc">${element.contributions}</p>
          <a href="${element.link}" target="_blank" class="btn-primary"><span class="btn-underline">${element.linkDesc}</span>
        </div>
      </div>
    </div>
    <div class="btn-wrap"><button type="button" class="btn-modal-close">close</button></div>
  </section>
</div>
`);
```

json파일을 불러와 내용에 맞게 각 리스트와 연결해주고, 요소들을 동적으로 불러오도록 스크립트를 작성 하였습니다.



현재 공부하고 있는 react js를 이용하여 만든 프로젝트를 소개하는 페이지 입니다.  
오른쪽 이미지는 todo list 프로젝트 소스이며, 해당 섹션내 프로젝트 링크를 걸어 두었습니다.



## html

```
<section id="Study" class="study section">
  <h2 class="main-title" data-aos="fade-up">Study</h2>
  <div class="section-inner">
    <div class="study-media-wrap" data-aos="fade-right" data-aos-delay="200">
      <div class="media-wrap" data-url="./media/ToDo-List.mp4">
        <div class="img-box"></div>
        <div class="media-box">
          <video class="media" src="" autoplay muted></video>
        </div>
      </div>
    </div>
  </div>
</div>
```

간단한 기능들을 소개하기 위해 동영상을 만들었습니다.

동영상 커버 이미지를 클릭하면 재생 되고 영상이 끝나면 다시 이미지 커버를 덮어 주었습니다.

## javascript

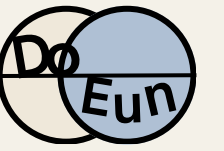
```
// 동영상 커버 클릭시 영상 재생
const media = document.querySelector( selectors: '.media-wrap');

media.addEventListener( type: "click", listener: (e : Event ) => {
  const target = e.currentTarget
  const url = target.getAttribute( qualifiedName: "data-url")
  const video = target.querySelector('video');
  const img = target.querySelector('.img-box');

  video.setAttribute("src", url)
  img.style.display = "none";

  // 동영상 종료 후 섬네일 이미지 다시 보이기
  video.addEventListener("ended", () => {
    img.style.display = "block";
  })
});
```

# Contact 페이지



Do Eun

Home About Project Contact

## Contact

메세지를

이름 \*

이메일 \*

전화번호

문의 내용을 적어주세요.

문의하기

contact section은 emailJS 플러그인을 이용하여 각 input의 value값을 전송하여 문의 내용을 받아볼 수 있도록 만들었습니다.

또한 필수값 체크와 유효성 체크를 통하여  
알맞게 값을 입력 하였는지 판단할 수 있게 작업하였습니다.  
내용 입력 후 toast 메시지를 사용하여 데이터 전송 결과를 나타냈습니다.

이메일 \*

email check

이메일을 올바르게 입력해 주세요.

내용을 모두 입력해 주세요.

## Contact

메세지를

이름 \*

이메일 \*

전화번호

문의 내용을 적어주세요.

문의하기

메세지 전송이 완료 되었습니다.

## Contact

메

이름 \*

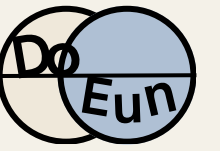
이메일 \*

전화번호

문의 내용을 적어주세요.

문의하기

# Contact 페이지



```
// 이메일 유효성 체크
const fieldWrap = document.querySelector( selectors: ".field-inner");
const emailPattern = /^[A-Za-z0-9_\\.\\-]+@[A-Za-z0-9\\-]+\\. [A-Za-z0-9\\-]+/;

function inputCheck(target, pattern){
  if(pattern.test(target.value) === false){
    target.parentNode.classList.add("parsley-type");
    fieldWrap.classList.add("error");
    Toast( message: "이메일 형식을 확인해 주세요.", type: "fail");
  } else{
    target.parentNode.classList.remove( tokens: "parsley-type");
    fieldWrap.classList.remove( tokens: "error");
  }
}
```

이메일 유효성 체크를 위해 작업한  
스크립트 입니다.

```
// 토스트 메시지
function Toast(message, type) {
  const wrap = document.querySelector( selectors: ".wrap");

  wrap.insertAdjacentHTML( where: "beforeend", html: `<article class="toast-message ${!!type ? type : ''}">${message}</article>`);

  setTimeout( handler: () => {
    const toast = document.querySelector( selectors: ".toast-message");
    wrap.removeChild(toast);
  }, timeout: 2100);
}
```

토스트 메시지는 함수 호출로 마크업을 동적으로 추가 시키고,  
파라미터를 통해 내용과 타입을 정하여 공통으로 사용할 수 있도록 하였습니다.



끝까지 봐주셔서 감사합니다.

제 포트폴리오 사이트가 궁금하시다면,  
아래 url로 접속해 주시기 바랍니다.

감사합니다.

<https://kde5813930.github.io/Portfolio-Doeun/>