

Html Portfolio Guide Sheet

정보 전달을 주제로 기업형 사이트를 구현했습니다.

보여주어야 할 정보가 명확한 사이트를 만들고자 직관적이고 명확한 레이아웃,
단순한 동선과 주요 정보를 강조할 수 있는 잡지를 디자인 컨셉으로 선정했으며
사이트에 들어가는 description은 패션 잡지 vogue의 웹사이트에서 영감을 받
아 제작되었습니다.

THE FEATURE OF SITE

- 순수 자바스크립트를 이용한 portfolio
- kakao map Api 활용하여 data 관리
- Youtube Api data 관리
- Flickr Api data를 가져오고 data 검색 기능
- 반응형 style 적용
- 회원가입 시 validation(유효성 검사) 인증 코드 구현

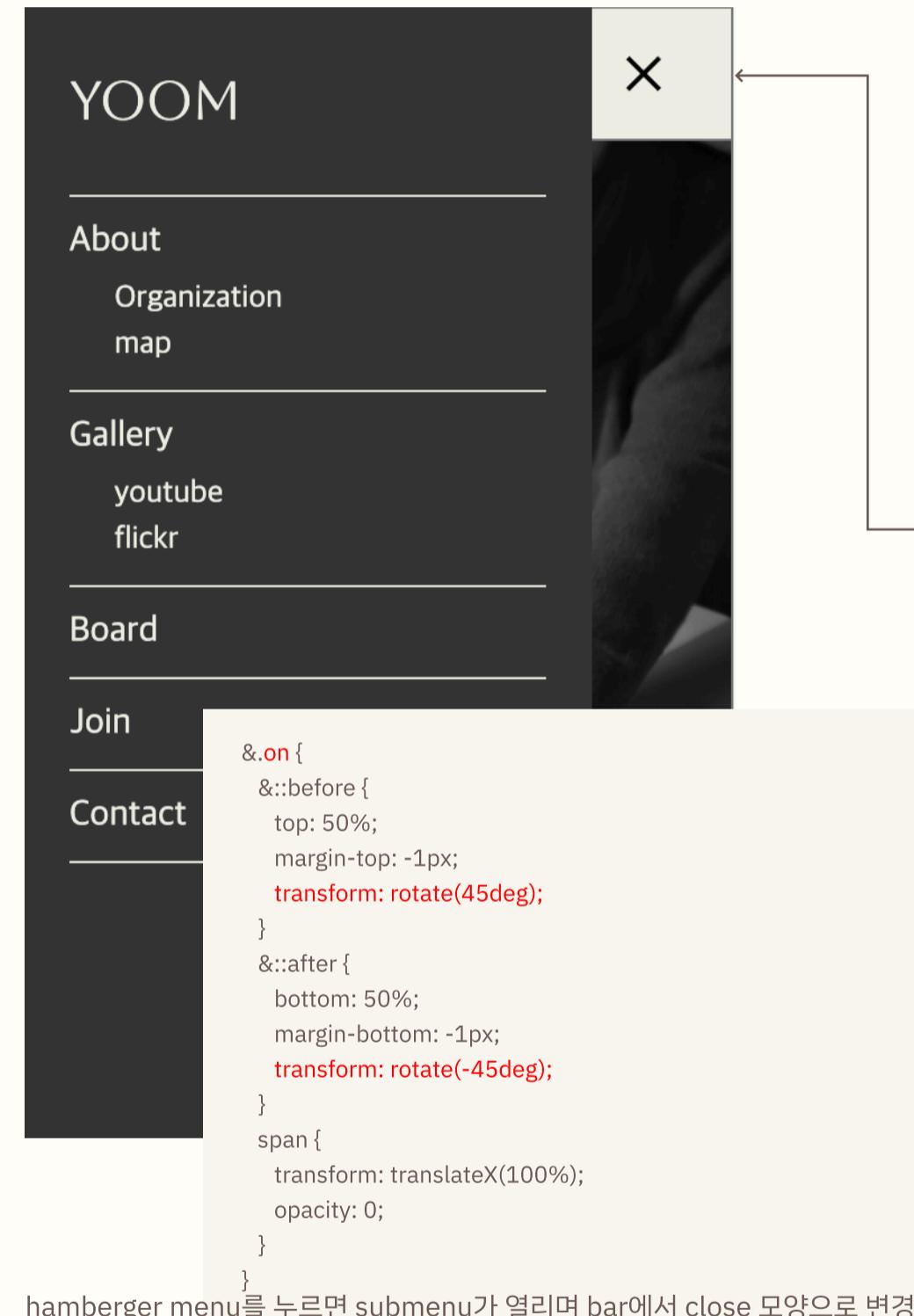


Directory

DIRECTORY

-  css 컴파일된 css 파일 포함
-  img 프로젝트에 사용된 Image 포함
-  js 프로젝트에 사용된 js파일 포함
-  scss scss 파일 포함
-  organization.html 조직도 페이지
-  board.html 자유게시판 페이지
-  contact.html 컨택트 페이지
-  flickr.html flickr 갤러리 페이지
-  youtube.html youtube 갤러리 페이지
-  join.html 회원가입 페이지
-  main.html 메인페이지
-  map.html 지도 페이지
-  sub_page.html sub-page에 사용된 공통 레이아웃

Common Layout



hamberger menu를 누르면 submenu가 열리며 bar에서 close 모양으로 변경
transform,transition을 이용하여 on class가 붙었을 때 적용

Header 공통 Header

position fixed로 header가 scroll 시에도 계속 따라다니도록 구현

```
#header {
  position: fixed;
  top: 0;
  left: 0;
  ...생략
}
```

tablet 디바이스부터는 bar menu로 submenu로 나타내며 bar icon을 클릭 시
모바일 서브메뉴 패널이 on class로 활성화 되도록 구현

VOGUE

```

hamMenu.addEventListener("click", (e) => {
  e.preventDefault(); // a 태그로 만들어진 버튼의 기존 a태그 이벤트 막기
  subMenu(hamMenu); // 매번수 자리에 활성화 이벤트가 붙는 요소를 넣어줌
  subMenu(menuMo);
});

function subMenu(val) {
  // on Class가 붙는 활성화 코드가 여러번 재사용 되므로 재사용 가능하도록 함수화
  val.classList.toggle("on");
}

```

SHIN YUMI
FRONT-END DEVELOPER

CONTACT ME
If you are interested in my portfolio, please email me!

view CONTACT

SOCIAL
Instagram Blog

display flex가 적용된 footer를 mobile 화면에서
는 flex-direction column을 적용하여 반응형 디자
인 완성

github notion tistory
Copyright © 2023 Yoom All Right reserved

VOGUE

About Gallery Board Join Contact

Contents

sub page에 사용되는 공통 layout page를 만들어 관리

VOGUE

Contact me Social
If you are interested in my portfolio, please email me! instagram
View blog

Copyright © 2023 Yoom All Right reserved

Footer 공통 Footer

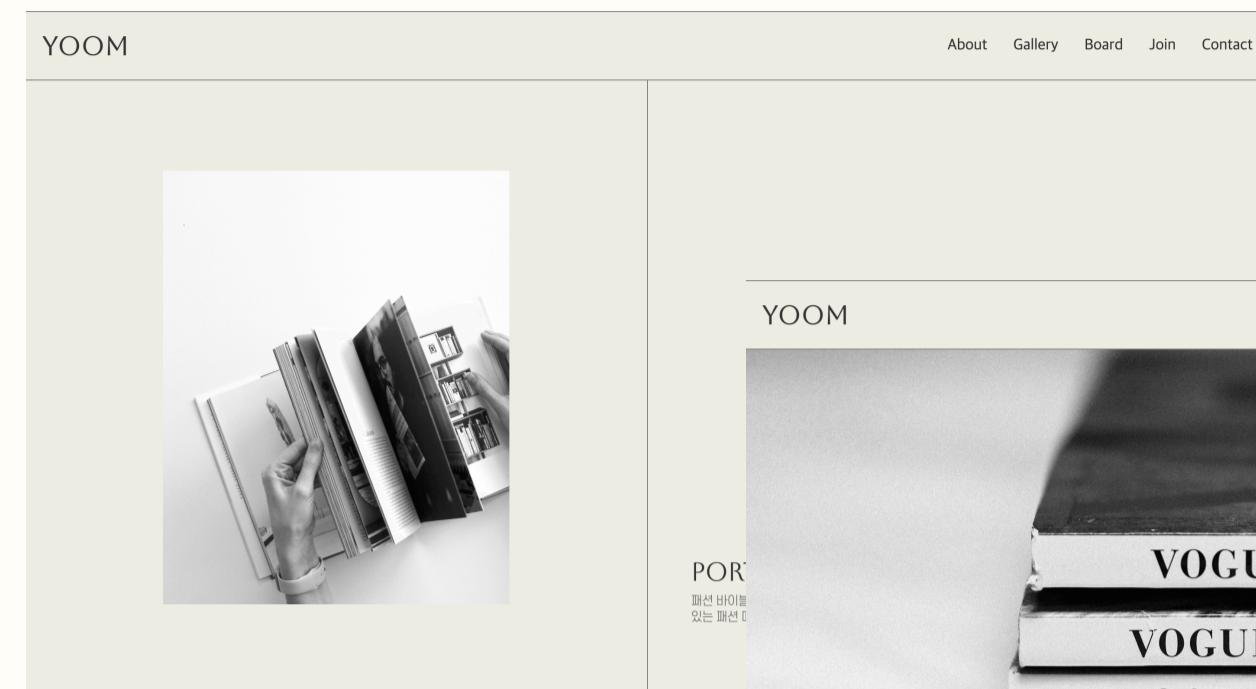
Information from the Html Portfolio

Main page

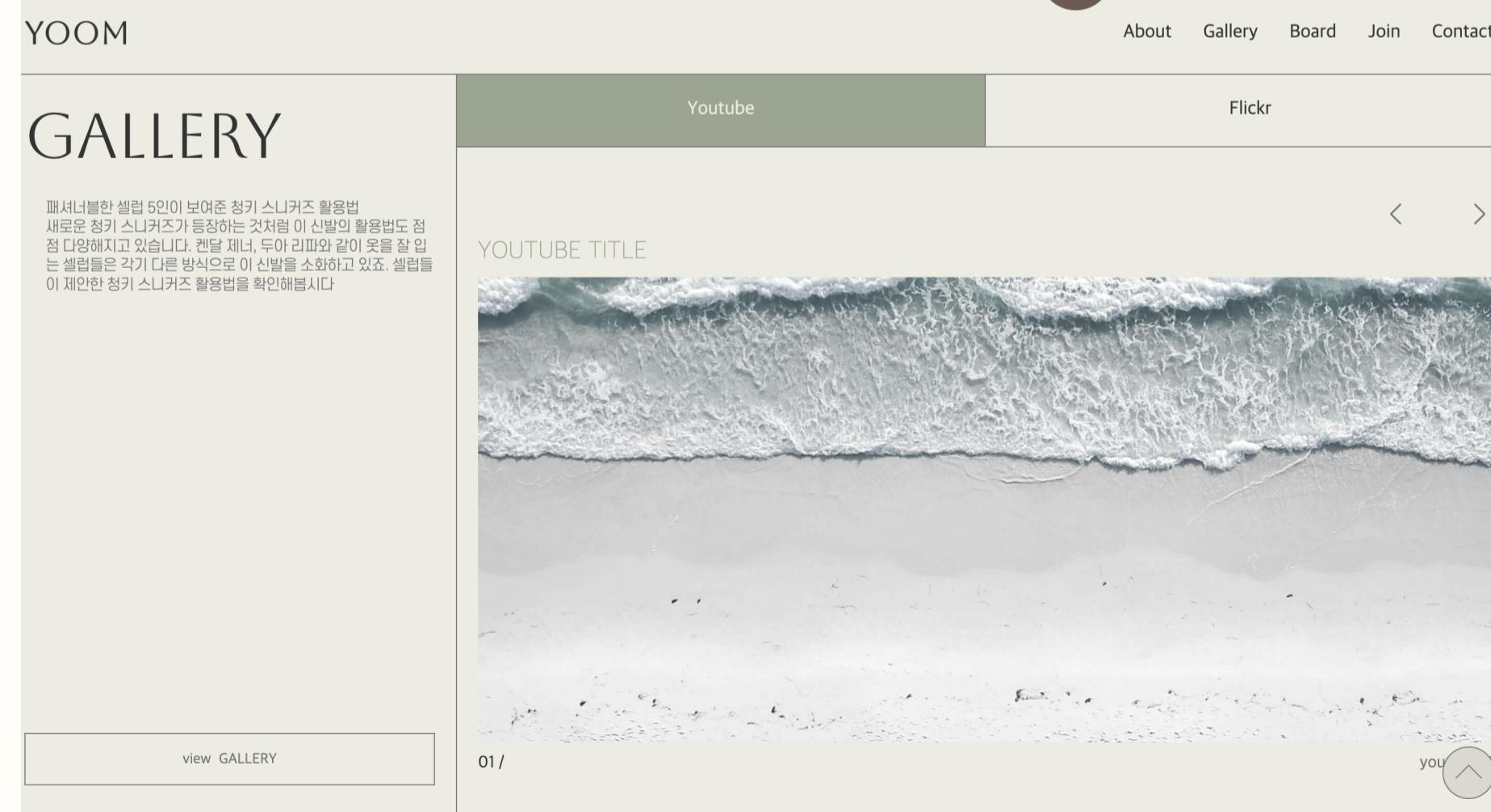
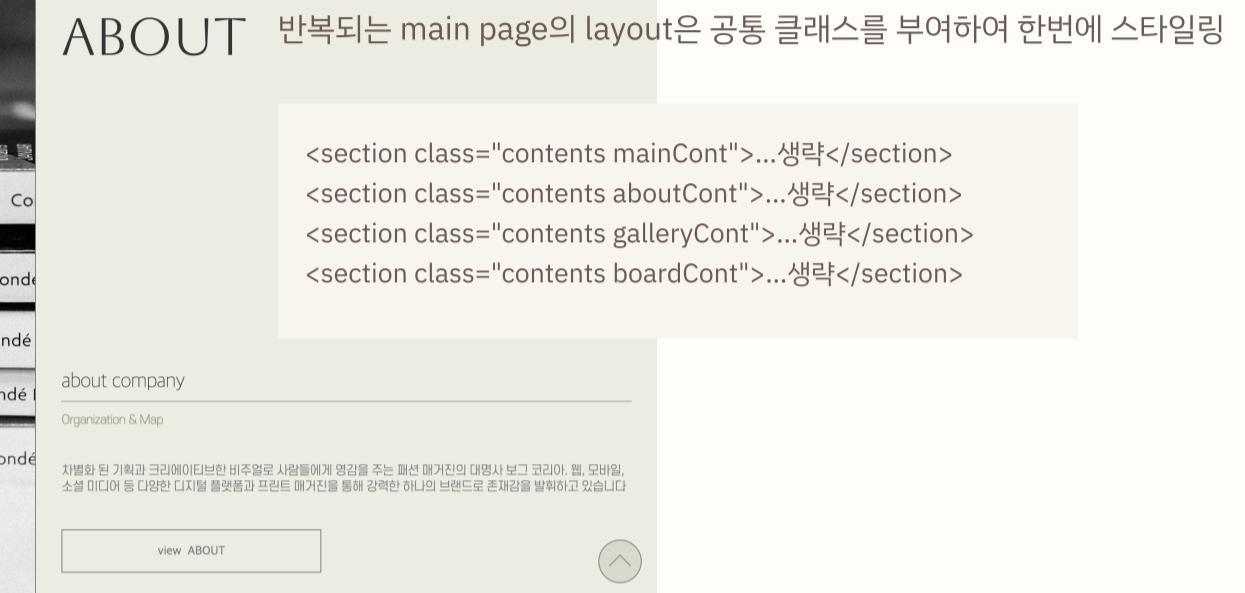
js
main.js

SCSS
_main.scss

main.html



Layout 공통 class 적용



Gallery section

gallery section에 tab menu, swiper 적용

tab menu 구현

계속 반복되는 활성화 코드를 onActive라는 함수를 만들어 적용. forEach 문을 사용하여 tab들을 반복으로 돌면서 해당되는 index에 on class를 붙여 활성화

```
galleryTabs.forEach((tab, i) => {
  tab.addEventListener("click", (e) => {
    e.preventDefault();
    onActive(galleryTabs, i);
    onActive(galleryCont, i);
  });
});

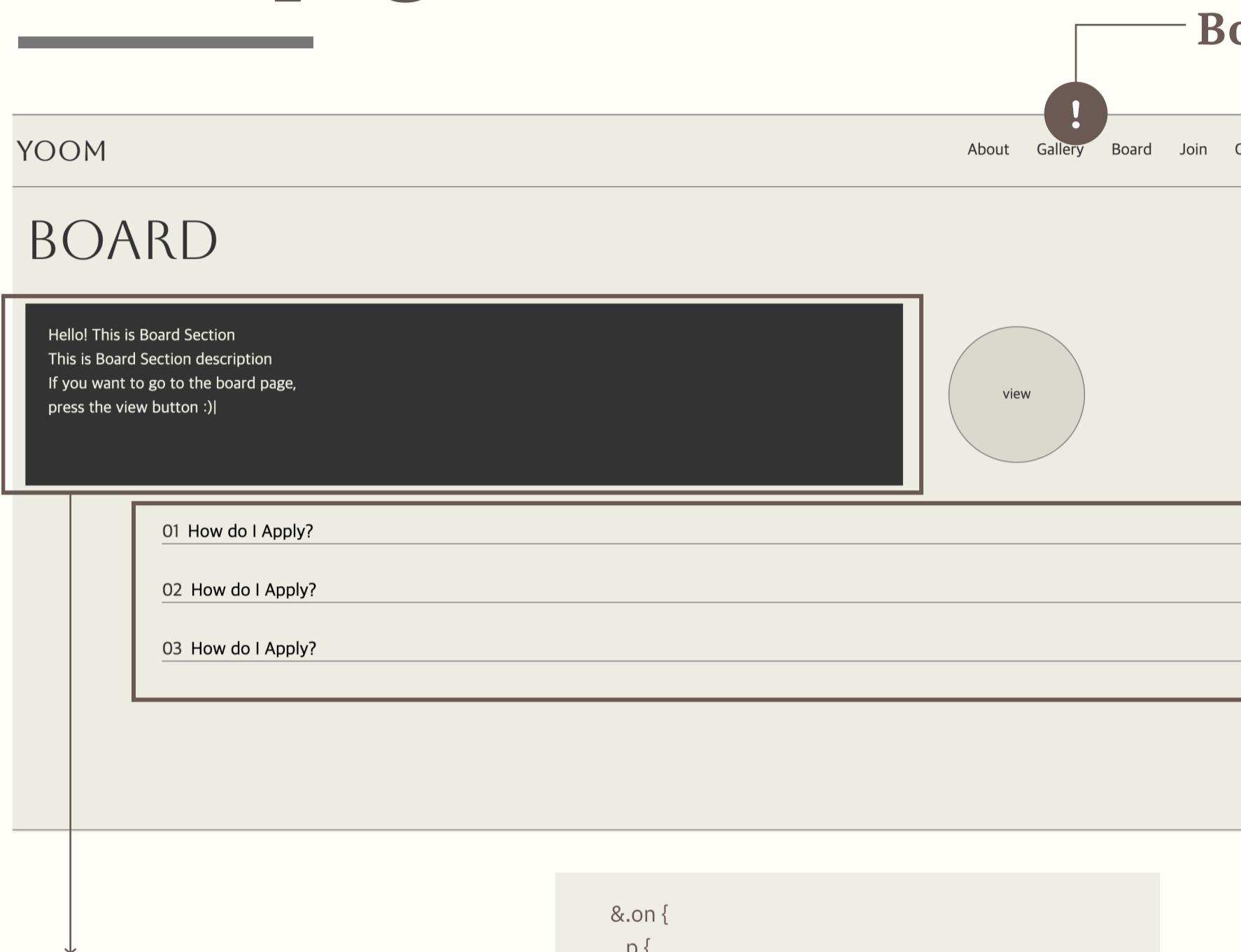
function onActive(arr, idx) {
  for (let el of arr) {
    el.classList.remove("on");
  }
  arr[idx].classList.add("on");
}
```

slider 구현

swiper plugin을 사용하여 slider를 구현
main.html에 swiper cdn을 적용한 후 사용

```
var swiper = new Swiper(".gallerySlide", {
  loop: "auto", //무한 반복
  grabCursor: true, // cursor를 잡아 슬라이더 조작 가능
  slidesPerView: 1, //한번에 보여질 slide의 갯수
  autoplay: { //자동 재생
    delay: 2000, //2초마다재생
    disableOnInteraction: false,
    //스와이프 후 자동 재생 비활성화 방지
  },
  navigation: { //navigation 버튼
    nextEl: ".swiper-button-next",
    prevEl: ".swiper-button-prev",
  },
});
```

Main page



typing animation

줄마다 p 태그로 감싸준 후 style에서 활성화 되면 animation을 적용

```
<div class="boardTextArea">
  <p>Hello! This is Board Section</p>
  <p>This is Board Section description</p>
  <p>If you want to go to the board page,</p>
  <p>press the view button :)<span>|</span></p>
</div>
```

```
&.on {
  p {
    opacity: 1;
    animation: type 4s steps(60, end);
  }
  &:nth-child(2) {
    animation: type2 8s steps(60, end);
  }
  &:nth-child(3) {
    animation: type3 12s steps(60, end);
  }
  &:nth-child(4) {
    animation: type4 18s steps(60, end);
  }
  span {
    animation: blink 1s infinite;
  }
}
```

Board section

css만으로 accordion menu 구현, typing animation 적용

accordion menu

input radio type을 이용하여 accordion menu를 구현 appearance: none; 속성으로 기존 radio 박스 요소를 숨기고 check 되었을 때의 스타일을 적용한다

01 How do I Apply?

```
input[type="radio"] {
  appearance: none;
}
```

```
&:checked + label::after {
  transform: rotate(-45deg);
}
```

style 적용

```
<div class="tab">
  <input type="radio" name="acc" id="acc1" />
  <label for="acc1">
    <h2>01</h2>
    <h3>How do I Apply?</h3>
  </label>
  <div class="tabContents">
    <p>...내용생략</p>
  </div>
</div>
```

```
input[type="radio"] {
  appearance: none;
}
```

```
&:checked ~ .tabContents {
  max-height: 80vh;
}
```

```
&:checked ~ label h3 {
  padding: 2px 10px;
  border-radius: 2px;
}
```

animation은 scroll 값이 board section의 scrollTop보다 커질 경우 활성화 되도록 구현

```
window.addEventListener("scroll", () => {
  if (window.scrollY > boardTop) {
    boardTextArea.classList.add("on");
  }
});
```

map page

교통정보. 로드뷰

kakao에서 제시하는 코드를 사용하여 교통 정보, 로드뷰를 on/off할 수 있도록 구현

```
traffic_on.addEventListener("click", (e) => {
  e.preventDefault(); //a 태그 기존 이벤트 막기
  if (traffic_on.classList.contains("on")) return; //재이벤트 방지 구문
  map.addOverlayMapTypeId(kakao.maps.MapTypeId.TRAFFIC);
  traffic_on.classList.add("on");
  traffic_off.classList.remove("on");
});

traffic_off.addEventListener("click", (e) => {
  e.preventDefault();
  map.removeOverlayMapTypeId(kakao.maps.MapTypeId.TRAFFIC);
  traffic_off.classList.add("on");
  traffic_on.classList.remove("on");
});
```

kakao map api를 사용하여 구현

```
var mapContainer = document.getElementById("map"), // 지도를 표시할 div
mapOption = {
  center: new kakao.maps.LatLng(37.5665256, 127.0092219), // 지도의 중심좌표
  level: 3, // 지도의 확대 레벨
};
// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);
```

- js
 - map.js
- SCSS
 - _about.scss

- map.html

다른 명소 더보기

명소들의 정보를 배열로 담고 명소 클릭 시 for문으로 반복을 돌면서 지도에서 각 명소가 표시 되도록 구현

```
for (let i = 0; i < markerPoints.length; i++) {
  new kakao.maps.Marker({
    map: map,
    position: markerPoints[i].latlng,
    title: markerPoints[i].title,
    image: new kakao.maps.MarkerImage(markerPoints[i].imgSrc, markerPoints[i].imgSize, markerPoints[i].imgPos),
  });
}
```

```
markerPoints[i].button.addEventListener('click', (e) => {
  e.preventDefault();
  for (let k = 0; k < markerPoints.length; k++) {
    markerPoints[k].button.classList.remove('on');
  }
  markerPoints[i].button.classList.add('on');

  moveTo(markerPoints[i].latlng);
  panTo(markerPoints[i].latlng);
});
```

```
let markerPoints = [
  {
    title: "DDP",
    latlng: new kakao.maps.LatLng(37.5665256, 127.0092219),
    imgSrc: "img/marker-border.png",
    imgSize: new kakao.maps.Size(25, 35),
    imgPos: { offset: new kakao.maps.Point(12, 69) },
    button: location_btns[0],
  },
  ...생략
]
```

about > map page

Yoom

ABOUT

Map / 지도

차별화된 기획과 크리에이티브한 비주얼로 사람들에게 영감을 주는 패션 매거진의 대명사 보그 코리아의 위치를 확인하세요



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Similique culpa dolorum, consequatur eius labore cupiditate voluptibus obsecrati earum aliquam deserunt veniam hic nam! Ut animi dolorum expedita repellendus quod sapientia?

[더 살펴보기](#)

[교통상황](#)

[교통정보 보기](#) [교통정보 보기](#)

[로드뷰](#)

[로드뷰 보기](#) [로드뷰 보기](#)

[다른 명소 더 보기](#)

[DDP\(동대문디자인플라자\)](#) [경복궁](#) [남산 N 타워](#) [반포한강공원](#)

SHIN YUMI
FRONT-END DEVELOPER

CONTACT ME
If you are interested in my portfolio, please email me!

SOCIAL
Instagram
Blog

[view CONTACT](#)

github notion tistory

Copyright © 2023 Yoom ALL Right Reserved

gallery page

- js
 - youtube.js
 - flickr.js

- SCSS
 - _gallery.scss

- youtube.html
- flickr.html

GALLERY

[youtube](#) [flickr](#)

Youtube / 유튜브 갤러리
Vogue의 새로운 video를 Youtube에서 만나보세요



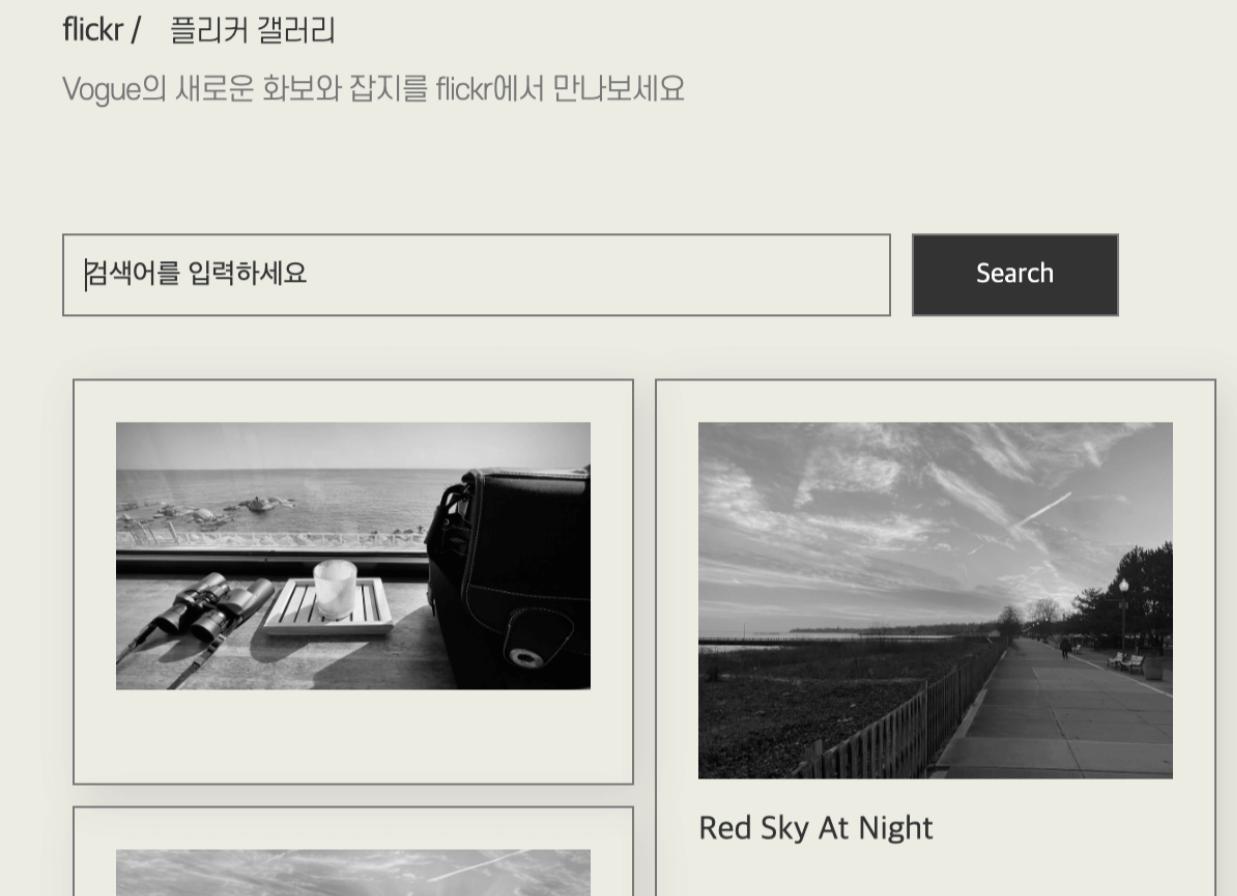
Pamela Anderson & Her So...
2023-02-07T08:33:55Z



Gwendoline Christie Brea...
2023-02-07T08:34:17Z

Gallery page

!



fetch method를 이용한 data fetching

fetch method를 이용하여 data를 가져오며 데이터를 가져오는데 성공하면 동적으로 list를 생성하여 화면에 보여지게 함

```

fetch(url)
  .then((data) => {
    return data.json();
  })
  .then((json) => {
    let items = json.items;
    let result = "";

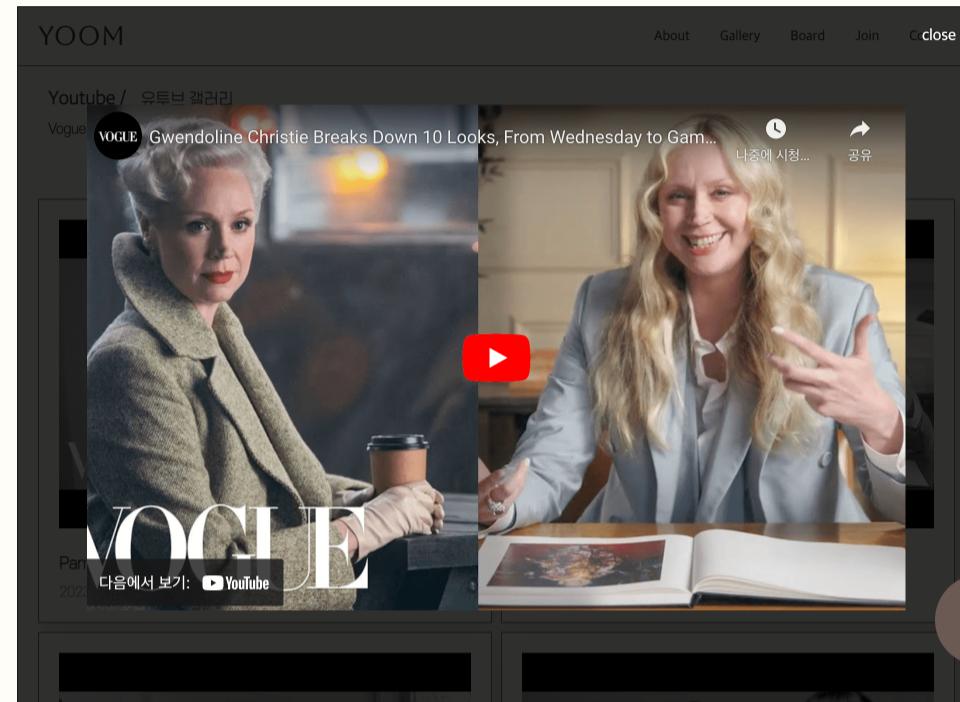
    items.map((el) => { // map을 사용하여 반복 돌기
      let title = el.snippet.title;
      if (title.length > 24) {
        title = title.substr(0, 24) + "...";
      } // title의 길이가 24 글자가 넘어갈 시 잘라낸 후 말 줄임표를 붙임

      let date = el.snippet.publishedAt;
      data = date.split("T")[0];
      // date를 가져올 시 초단위까지 다 표기되기 때문에 일까지만 잘라줌
      result += `
        <article class='youtubeCont'>
          <a href="${el.snippet.resourceId.videoId}" class="pic">
            
          </a>
          <div class="con">
            <h2>${title}</h2>
            <span>${date}</span>
          </div>
        </article>
      `;
    });
    youtubeList.innerHTML = result;
    // 동적으로 생성된 리스트들을 html 안에 뿌려줌
  });
}

```

Information from the Html Portfolio

gallery page



동적으로 popup 생성하기

list 하나 클릭시 각 list에 해당하는 영상/이미지가 popup으로 도출되도록 하며 getAttribute method를 사용하여 필요한 요소를 가져오고 append method를 이용하여 동적으로 생성된 팝업창을 화면에 넣어줌

```
youtubeList.addEventListener("click", (e) => {
  e.preventDefault();

  if (!e.target.closest("a")) return; // e.target이 맞지 않다면 return

  const vidId = e.target.closest("a").getAttribute("href");
  // getAttribute로 a tag의 href 값을 가져옴

  let pop = document.createElement("figure");
  pop.classList.add("pop");

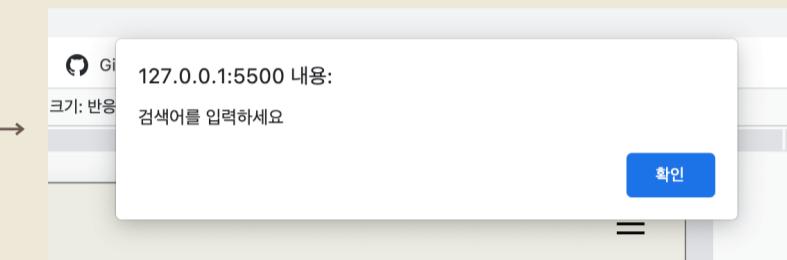
  pop.innerHTML = `
    <iframe ...생략></iframe>
    <span class='btnClose'>close</span>
  `;
  // popup 창을 append method를 이용하여 밀어넣기
  youtube.append(pop);
});
```

검색 기능 구현

입력한 검색어에 따라 list 변경, enter 시에도 검색되도록 하며 빈값, 없는 값 검색 시 alert 창을 띄워 예외 처리해줌

```
btn.addEventListener("click", () => {
  let tag = input.value;
  tag = tag.trim(); // trim method를 이용하여 띄어쓰기가 있을 시 빈값 제거
  const url = `url`;
```

```
if (tag !== "") { // tag가 빈값이 아닐 시 데이터 불러옴
  ...fetch로 가져온 데이터 불러옴
} else { // tag가 빈값이면 alert 창으로 경고
  alert("검색어를 입력하세요");
}
});
```



```
input.addEventListener("keypress", (e) => {
  if (e.keyCode == 13) { //enter(keycode 13) 시 tag가 들어간 url로 변경하여 값 도출
    let tag = input.value;
    const url = `url`;
  }
});
```

Loading bar

데이터가 불러와질 동안 loading animation을 표기하여 웹 접근성을 높히며 loading은 기본 값이 활성화 였다가 data가 불러와지면 비활성화 되도록 구현

loading bar가 사라지면 list에 걸어둔 인터렉션 on

```
function isoLayout() {
  loading.classList.add("off");
  frame.classList.add("on");
}
```

GALLERY



join page

js
validation.js

SCSS
_join.scss

join.html

The screenshot shows a 'join' form on a website. At the top, there's a navigation bar with links for About, Gallery, Board, Join, and Contact. Below the navigation, the word 'JOIN' is prominently displayed. A large input field for 'User ID' has a red border and a red exclamation mark icon above it, indicating it's a required field. The input field contains the placeholder '아이디를 입력하세요.' (Please enter your ID). To the right of the input field, a red error message box displays the text '아이디를 입력하세요.' (Please enter your ID) and '입력항목을 5글자 이상 입력하셔야 합니다.' (The input field must be at least 5 characters long).

유효성 검사

회원 가입 시 각 input에 필수 값이 들어갔는지, 빈 값은 아닌지 확인할 수 있는 validation 코드를 작성 후 유효성 검사에 적합하지 않으면 에러 메세지를 도출

아이디를 입력하세요.
입력항목을 5글자 이상 입력하셔야 합니다.

text type의 경우 text의 length를 인수로 지정하여 input의 값의 길이가 미달되면 error message를 띄움
element가 되는 Input은 name값으로 구분한다.

```
function isTxt(el, len) {
  if (len === undefined) len = 5;
  let input = form.querySelector(`[name=${el}]`);
  let txt = input.value;

  if (txt.length >= len) {
    const errMsgs = input.closest('.joinInput').querySelectorAll('p');
    if (errMsgs.length > 0) input.closest('.joinInput').querySelector('p').remove();
    return true;
  } else {
    const errMsgs = input.closest('.joinInput').querySelectorAll('p');
    if (errMsgs.length > 0) input.closest('.joinInput').querySelector('p').remove();
    let errMessage = document.createElement('p');
    errMessage.append(`입력항목을 ${len}글자 이상 입력하셔야 합니다.`);
    input.closest('.joinInput').append(errMessage);
    return false;
  }
}
```

```
btnSubmit.addEventListener('click', (e) => {
  if (!isTxt('userid', 5)) e.preventDefault();
  if (!isTxt('name', 2)) e.preventDefault();
  if (!isTxt('comments', 20)) e.preventDefault();
});
```

email type의 경우 input의 value값이 @를 포함하고 있지 않으면 error message를 띄움

```
function isEmail(el) {
  let input = form.querySelector(`[name=${el}]`);
  let txt = input.value;

  if (/@/.test(txt)) {
    ...생략
  } else {
    ...생략
  }
}
```

```
btnSubmit.addEventListener('click', (e) => {
  if (!isEmail('email')) e.preventDefault();
});
```

join page

checkbox type의 경우 check box list 중에 check된 값이 하나라도 없을 시 error message를 띄움

```
function isCheck(el) {
  let inputs = form.querySelectorAll(`[name=${el}]`);
  let isCheck = false;

  for (let el of inputs) {
    if (el.checked) isCheck = true;
  }

  if (isCheck) {
    ...생략
  } else {
    ...생략
  }
}
```

```
btnSubmit.addEventListener('click', (e) => {
  if (!isCheck('gender')) e.preventDefault();
  if (!isCheck('favorite')) e.preventDefault();
});
```

password type의 경우 비밀번호 입력 input과 비밀번호 검증 input의 값을 비교하여 값이 다를 경우, 영문, 숫자, 특수기호가 모두 들어가지 않았을 경우 error message를 띄운다

```
function isPwd(el1, el2, len) {
  let pwd1 = form.querySelector(`[name=${el1}]`);
  let pwd2 = form.querySelector(`[name=${el2}]`);
  let pwd1_value = pwd1.value;
  let pwd2_value = pwd2.value;

  const num = /[0-9]/;
  const eng = /[a-zA-Z]/;
  const spc = /[~!@#$%^&*()_+?<>]/;
  //정규표현식으로 표기
```

```
btnSubmit.addEventListener('click', (e) => {
  if (!isPwd('pwd1', 'pwd2', 5)) e.preventDefault();
});
```

select 박스의 경우 option list 중에 선택한 index가 0일 경우 error message를 띄움

```
function isSelect(el) {
  let sel = form.querySelector(`[name=${el}]`);
  let sel_index = sel.options.selectedIndex;
  let value = sel[sel_index].value;

  if (value !== "") {
    ...생략
  } else {
    ...생략
  }
}
```

```
btnSubmit.addEventListener('click', (e) => {
  if (!isSelect('edu')) e.preventDefault();
});
```

```
if (
  pwd1_value === pwd2_value &&
  pwd1_value.length >= len &&
  num.test(pwd1_value) &&
  eng.test(pwd1_value) &&
  spc.test(pwd1_value)
) {
  ...생략
} else {
  ...생략
}
```