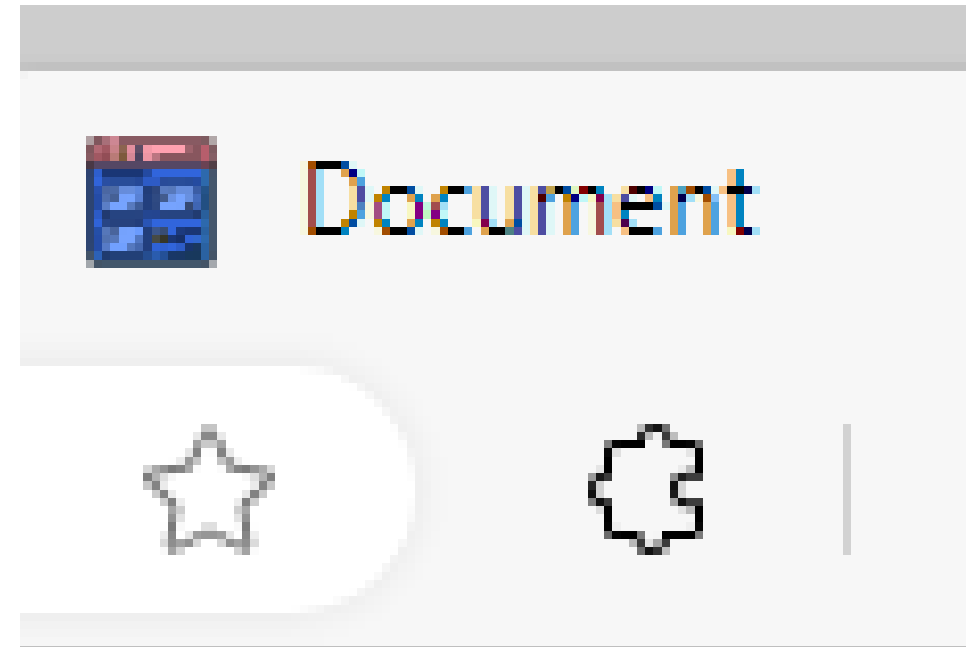


# HTML/CSS 4일차

# favicon

- 브라우저에서 탭 옆에 보이는 로고
- favicon에서 맘에 드는 이미지를 찾아서 16\*16으로 다운로드
- head안에 아래 문장을 추가한다.
- `link rel="icon" href="operating-system.ico">`



# 저번 과제 정렬하고 꾸미기

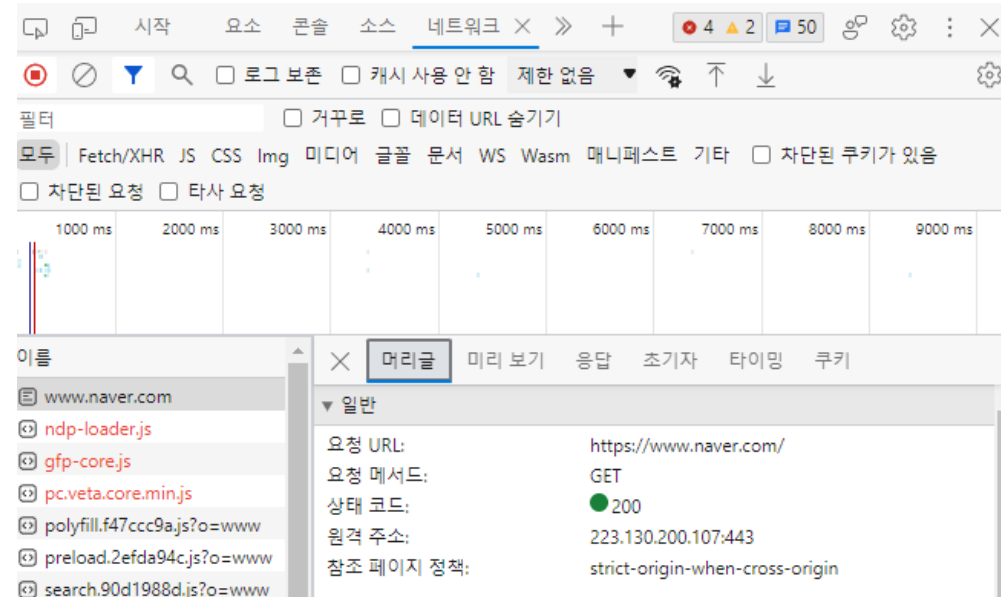
- 다른 페이지도 이런 느낌 만들기



The image shows a login form mockup. It features a light gray background with a white rectangular form box in the center. Inside the box, the word "로그인" (Login) is centered at the top. Below it are two input fields: the first is labeled "아이디" (ID) and the second is labeled "비밀번호" (Password). A yellow button with the text "로그인" (Login) is positioned below the input fields. At the bottom of the form box, there are three links: "아이디 찾기" (Find ID), "비밀번호 찾기" (Find Password), and "회원가입" (Sign Up).

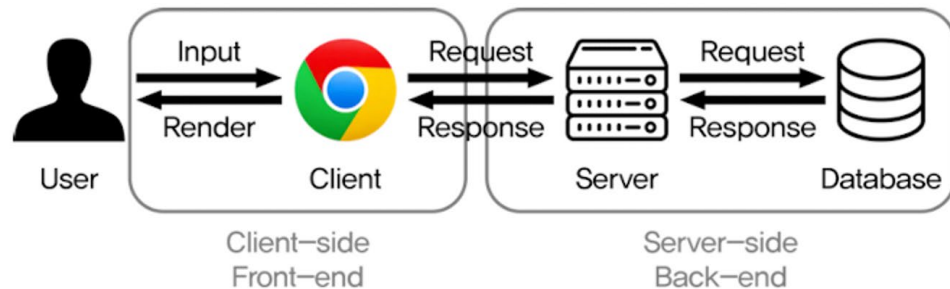
# 웹페이지 배포 원리

- 아무 웹페이지나 방문해서 개발자 도구로 확인해보자
- [www.naver.com](https://www.naver.com)에서 우리가 보는 페이지와 각종 필요한 파일들을 전송해준다.



# 웹 페이지

- 따라서 우리가 인터넷으로 웹페이지를 보면 해당 경로에서 html을 보내주어야 사용하는 것이 가능하다.
- 보통 데이터를 요청하는 쪽을 client, 데이터를 제공하는 쪽을 server라 한다.
- client가 보는 영역은 사용자가 보는 앞쪽이라 frontend,
- server에서 처리하는 영역은 사용자가 보지 못하는 뒤쪽이라 backend 라 한다.

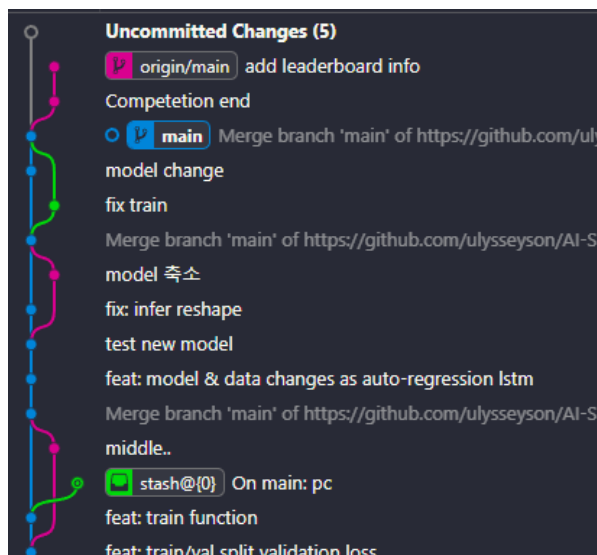


# Git

- 버전 관리 시스템
- 특정 파일을 지속적으로 변경할 때 이전 버전을 쉽게 확인하기 위해 새 파일을 만들어서 하는 경우
- 파일은 그대로 두고 파일을 특정 순간마다 기록하여 파일의 버전을 쉽게 기록가능하다.

이름

결산보고서.hwp  
결산보고서\_남사원 수정.hwp  
결산보고서\_남사원 수정\_과장님 피드백 반영.hwp  
결산보고서\_제출용(수정) - 최종본.hwp  
결산보고서\_제출용(수정).hwp  
결산보고서\_제출용.hwp  
Final 결산보고서\_제출용(수정) - 최종본.hwp  
RealFinal 결산보고서\_제출용(수정) - 최종본.hwp



# GIT add

- 변경 사항이 일어났을 때, git에 그것을 올린다.
- add로 추가한 file들에 한해서만 이후 commit이 이루어진다.

# GIT commit

- 코드 변경을 해서 어떤 기능을 추가 or 수정 or 변경 하였을 때에 기록을 위해 commit을 한다.
- commit은 고유 ID를 가지며 되돌릴 수 있다.
- commit message를 작성하여 변경에 대해 기록한다.
- `git commit -m "commit message"`



# GIT push

- push는 로컬 저장소에서 작업한 내용을 원격으로 업로드 한다.
- 각자 로컬에서 작업 이후 원격 저장소에 업로드 하는 식으로 공유한다.

# GIT pull

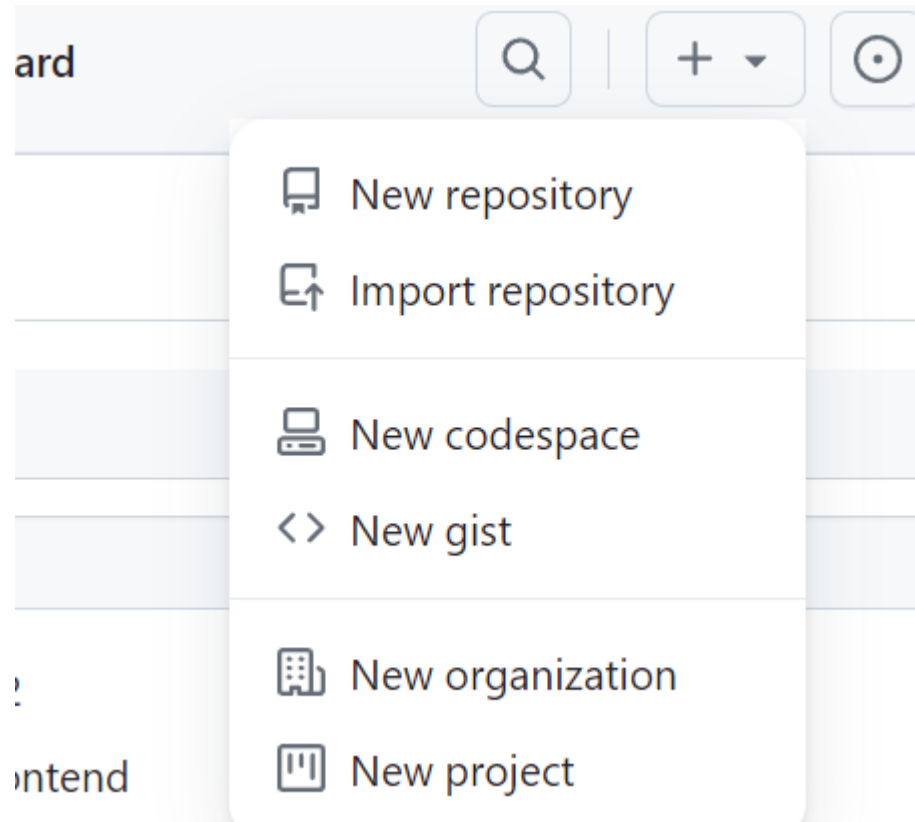
- 원격 저장소에 업로드된 코드를 로컬에 받아서 업데이트 한다.

# GitHub

- git의 원격 저장소 중에서 가장 규모가 큰 사이트

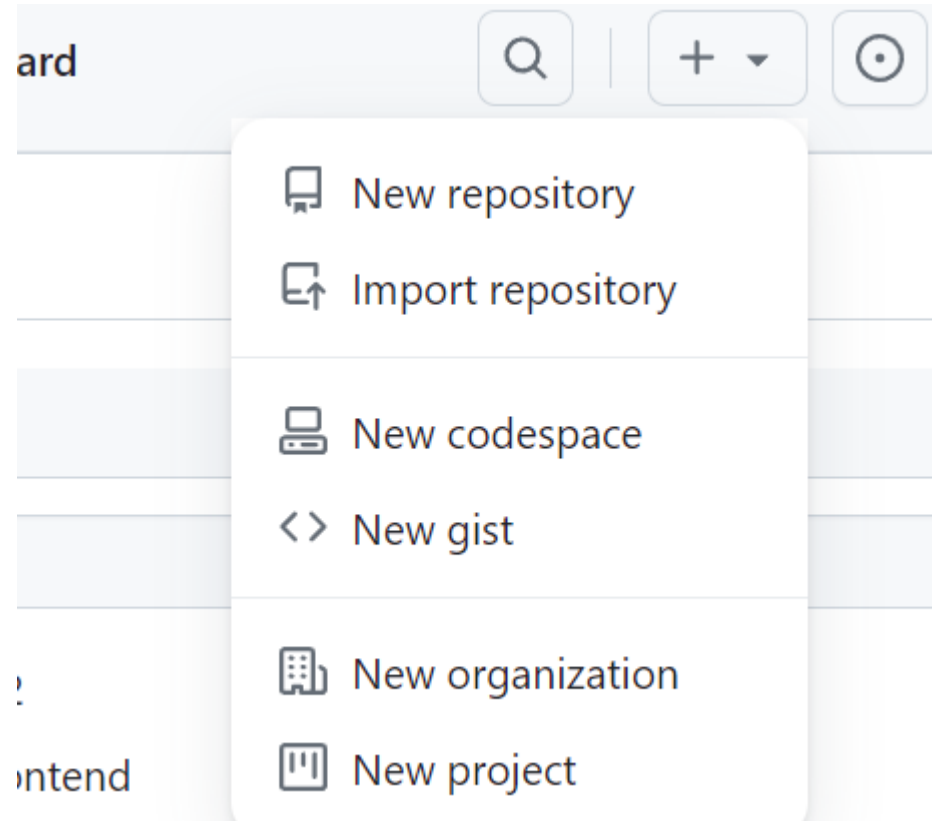
# Github 사용하기

- 새로운 레포지토리 생성



# Github 사용하기

- 해당 레포지토리(저장소) clone
- 작업한 html 코드 추가 및 commit
- 원격 저장소에 push



# Github pages

- github에서 동적 페이지에 대해 무료 퍼블리쉬를 해줌
- 사용자가 직접 서버를 만들 필요를 줄여준다.

# Github pages

- 레포지토리에서 setting -> 코드와 자동화 -> 페이지
- build and deployment에서 branch 설정을 해준다.
- jekyll theme설정을 위해 .nojekyll을 생성

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://ulyseyson.github.io/html/>

Last deployed by  ulyseyson 38 minutes ago

[Visit site](#)



## Build and deployment

### Source

Deploy from a branch ▾

### Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more.](#)

 main ▾

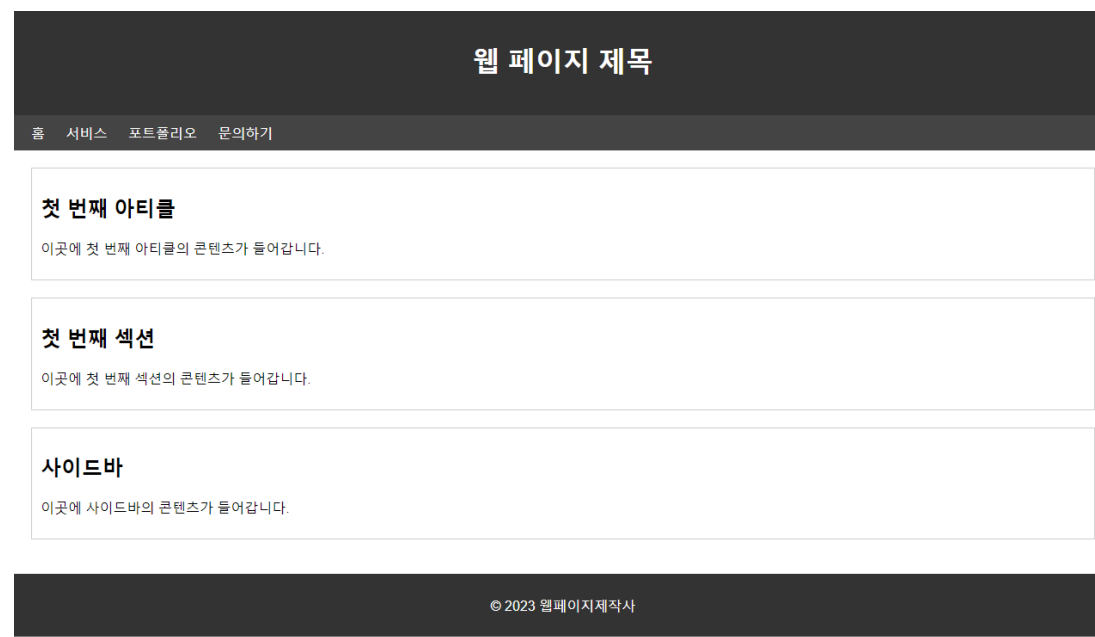
 / (root) ▾

Save

Learn how to [add a Jekyll theme](#) to your site.

# Github pages

- index.html 작성
- index.html은 사이트의 루트 / 에 접속했을 때 나오는 html 문서이다.
- 각자 원하는 대로 꾸며보도록 하자





# semantic tag

- div 한 개만 가지고도 원하는 모든 페이지를 만들 수 있다.
- 다양한 tag를 사용해야 하는 이유는?

# semantic tag

- HTML5 부터 등장
- 웹 문서의 구조와 의미를 더 명확히 표현 가능
- 검색엔진에 최적화 가능하고 접근성을 확대

# semantic tag

- <header>: 문서 또는 섹션의 헤더를 정의합니다. 일반적으로 로고, 제목, 네비게이션 등의 요소를 포함합니다.
- <nav>: 네비게이션 링크들을 포함하는 영역을 정의합니다.
- <main>: 문서의 주요 콘텐츠를 정의합니다. 보통 하나의 문서에 하나의 <main> 요소가 있어야 합니다.
- <article>: 문서, 페이지, 사이트 등 독립적인 콘텐츠 조각을 정의합니다.
- <section>: 콘텐츠의 섹션을 정의합니다. 일반적으로 제목을 가집니다.
- <aside>: 사이드바나 콘텐츠 옆에 위치한 추가 정보를 정의합니다.
- <footer>: 문서 또는 섹션의 푸터를 정의합니다. 보통 저작권 정보, 연락처 등이 위치합니다.

# 기존 index.html 변환하기

- 만들어봤던 index.html이 의미가 더 잘 담길 수 있게 변경해보자



# frame

- 기존에 <frame>을 이용해 여러 개의 html 문서를 독립적으로 로드했었다.
- 검색엔진이 페이지를 인식하기 힘들고, 뒤로가기 북마크 등이 페이지에 적용되기 어려워져서 현재는 거의 사용하지 않음

# frame

Top Frame

이것은 상단 프레임에 표시됩니다.

Main Frame

이것은 메인 프레임에 표시됩니다.

# iframe

- HTML5 부터는 iframe을 사용해 외부 페이지를 삽입한다.
- 외부 페이지를 삽입할 때는 안전하지 않은 사이트를 삽입하면 문제가 생길 수 있음을 인지해야함

# iframe

속성	설명
src	삽입할 외부 페이지의 URL을 지정합니다.
width, height	<iframe> 요소의 가로와 세로 크기를 지정합니다.
title	<iframe> 요소에 제목을 지정하며, 웹 접근성에 사용됩니다.
name	<iframe> 요소의 이름을 지정하며, 타겟으로 사용됩니다.
allowfullscreen	<iframe> 요소를 전체 화면 모드로 표시하도록 허용합니다.
sandbox	<iframe> 요소를 보안적으로 격리된 환경에서 로드합니다.
frameborder	<iframe> 요소 주변에 테두리를 생성합니다.
scrolling	<iframe> 요소 내부에 스크롤 바를 표시할지 여부를 지정합니다.
style	CSS 스타일을 사용하여 <iframe> 요소의 스타일을 지정합니다.
기타	allow, referrerpolicy, loading 등의 다른 속성들이 있습니다.



# iframe - youtube

- 오른쪽처럼 원하는 youtube 영상을 iframe에 띄우자.

## YouTube 동영상

아래는 YouTube 동영상이 띄워진 iframe입니다.



# css 의사 클래스

- 특정 요소의 특정 상태를 선택하기 위해 사용되는 선택자
- 의사 클래스는 요소가 어떤 상태에 있을 때 해당 요소에 스타일을 적용하는 데 유용

# 상태 의사 클래스

- :checked 체크된(checked) 상태의 input 요소를 모두 선택함.
- :enabled 사용할 수 있는 input 요소를 모두 선택함.
- :disabled 사용할 수 없는 input 요소를 모두 선택함.

## Checked 상태 의사 클래스 예제

☑ 체크 상태에 따라 스타일이 변경됩니다.

## Enabled 상태 의사 클래스 예제

활성화된 입력 필드

비활성화된 입력 필드

# 구조 의사 클래스

- :first-child: 형제 요소 중 첫 번째 요소
- :last-child: 형제 요소 중 마지막 요소
- :nth-child(n): 형제 요소 중 n번째 요소
- :nth-of-type(n): 동일한 유형의 요소 중 n번째 요소

- 첫 번째 아이템
- 두 번째 아이템
- 세 번째 아이템
- 네 번째 아이템

첫 번째 단락

두 번째 단락

# 동적 의사 클래스

- :link: 방문하지 않은 링크
- :visited: 이미 방문한 링크
- :focus: 포커스를 받은 요소
- :hover: 마우스가 요소 위에 있을 때
- :active: 요소를 활성화했을 때 (예: 클릭한 순간)
- :checked: 체크 박스 또는 라디오 버튼이 선택된 상태

마우스를 올리면 빨간색으로 변경

클릭하면 배경색이 노란색으로 변경

포커스를 받으면 테두리가 파

이미 방문한 링크는 보라색으로 표시

# 의사 클래스 응용

- 옆의 html 문서를 만들어보자
- 상품에 마우스를 올리면 크기 증가!



# 의사 요소

- ::first-letter
- ::first-line
- ::before
- ::after
- ::selection

# ::first-letter

- 텍스트의 첫 글자만을 선택
- block타입의 요소에만 적용 가능

## **::first-letter**를 이용한 선택

아래의 단락에서 첫 글자만을 선택합니다!

**L**orem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent at tellus et neque luctus tincidunt at vitae ligula.  
Donec a accumsan magna. Pellentesque habitant morbi



# ::first-line

- 텍스트의 첫 라인만 선택
- block타입의 요소에만 적용 가능

## ::first-line을 이용한 선택

아래의 단락에서 첫 라인만을 선택합니다!

Lorem ipsum dolor sit amet,

consectetur adipiscing elit. Praesent at tellus et neque luctus

tincidunt at vitae ligula. Donec a accumsan magna.

Pellentesque habitant morbi tristique senectus et netus et

# ::before

- 현재 요소 content 앞에 삽입

## ::before를 이용한 요소의 삽입

아래의 단락에서 콘텐츠 부분 바로 앞에 이미지를 삽입합니다!




Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent at tellus et neque luctus tincidunt at vitae ligula.  
Donec a accumsan magna. Pellentesque habitant morbi

# ::after

- 현재 요소 content 뒤에 삽입

## ::after를 이용한 요소의 삽입

아래의 단락에서 콘텐츠 부분 바로 뒤에 이미지를 삽입합니다!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent at tellus et neque luctus tincidunt at vitae ligula. Donec a accumsan magna. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed dolor nunc, facilisis in felis efficitur, sodales maximus nunc. Integer congue lectus vitae velit aliquam vehicula. Nunc mauris massa, sodales vitae augue ut, pulvinar mollis erat. Sed feugiat, mauris vitae convallis iaculis, nibh magna pretium diam, a sagittis quam urna quis augue. Nam feugiat accumsan ante id porttitor. Morbi tempus accumsan sem vitae venenatis. Duis mattis consequat ipsum nec interdum. Curabitur placerat vulputate faucibus. Cras consectetur elit ut metus consectetur tristique. 

# ::selection

- 해당 요소에서 사용자가 선택한 부분만 선택

## ::selection을 이용한 선택

아래의 단락에서 원하는 부분을 마우스로 선택해 보세요!

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent at tellus et neque luctus tincidunt at vitae ligula.  
Donec a accumsan magna. Pellentesque habitant morbi

# icon 넣기

- fontawesome 사이트를 이용하면 간편하다
- 회원가입 필요
- cdn 설정을 문서 내 head 태그에 넣어줘야 사용가능하다
- `<script src="https://kit.fontawesome.com/cbcad42a26.js" crossorigin="anonymous"> </script>`
- `<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css" rel="stylesheet">`
- 아래 처럼 i태그로 감싸 사용한다.
- `<i class="fas fa-ad" style="font-size:100px; color:red"> </i>`

# ::before를 사용한 input 꾸미기

- 아래 같이 input옆에 icon을 넣어보자

☒ 로그인 상태 유지