



# 팀 프로젝트 잘 마무리하기

🕒 생성일	@2023년 3월 25일 오전 9:20
☰ 태그	문서 배포용



그동안 팀 프로젝트 진행하느라 고생하셨습니다! 우리의 프로젝트를 잘 마무리해 볼까요?

## 프론트엔드 체크리스트

메타 태그

HTML 태그

## 프로젝트 파일 정리

프로젝트 정리하기

[필수] Github README 작성

[필수] 마무리하기

## 프론트엔드 체크리스트

### 메타 태그

- **Doctype:** HTML5 을 사용하며, Doctype이 모든 HTML 페이지의 최상단에 위치함

```
<!-- Doctype HTML5 -->
<!doctype html>
```

- **Charset:** 문자집합(UTF-8)이 올바르게 선언됨

```
<!-- 이 문서에 대한 문자 인코딩을 설정 -->
<meta charset="utf-8">
```

- **Viewport:** Viewport가 제대로 선언됨

```
<!-- 반응형 웹 디자인을 위한 Viewport 설정 -->
<meta name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover">
```

- **Title:** 모든 페이지에 title이 사용됨 (평균적인 글자 길이의 제한은 약 55자)  
페이지가 달라질 때마다 title 태그의 내용(content)이 적절히 변경되어야 합니다.

```
<!-- 문서의 Title -->
<title>55개 이하의 문자로 구성된 페이지 제목</title>
```

- **Description:** description이 제대로 기재됨. (설명값은 고유해야 하며, 150개 이하의 문자로 구성되어야 함) 해당 페이지에 대한 설명을 2~3줄 정도 작성.

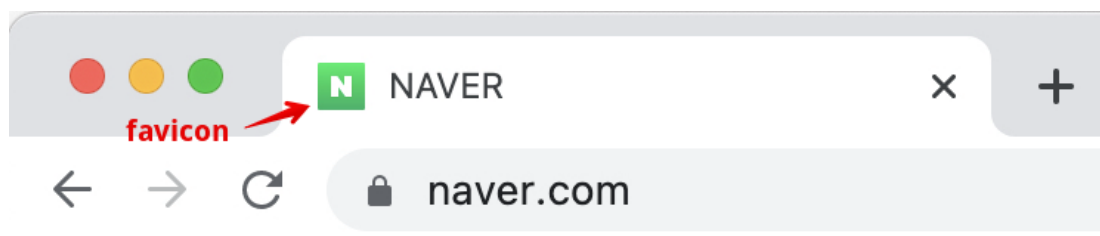
```
<!-- Meta Description -->
<meta name="description" content="페이지에 대한 설명 (150개 이하의 문자)">
```

- **파비콘:** 각각의 파비콘이 제대로 생성되었고 올바르게 보여지는가? 만약 `favicon.ico` 파일만 가지고 있다면, 해당 내용을 페이지의 상단부에 추가해야 함. 일반적으로는 해당 태그를 사용할 필요는 없지만, 아래의 예시를 포함하는 것이 좋은 습관임. 오늘날에는 `.ico` 포맷보다 **PNG 포맷의 아이콘 사용을 추천함**(크기: 32x32px).

```
<!-- 표준 파비콘 -->
<link rel="icon" type="image/x-icon" href="https://example.com/favicon.ico">
<!-- 추천 파비콘 포맷 -->
<link rel="icon" type="image/png" href="https://example.com/favicon.png">
```

#### ▼ 파비콘이란?

favicon이란 웹 페이지에 접속했을 때, **상단 탭에 보여지는 아이콘**을 의미합니다.



#### ▼ 파비콘 생성 사이트

사이트1: <https://www.favicon-generator.org/>

사이트2: <https://realfavicongenerator.net/>

## HTML 태그

- **언어 속성:** 현재 페이지 내의 언어에 알맞게 속성 값이 부여됨  
한국어 콘텐츠로 서비스를 제공하고 있다면, html 태그의 lang 속성 값을 한국어(`ko`)로 변경해주세요!

## 프로젝트 파일 정리

- 코드 주석 지우기
  - 또는 주석 코드를 한 파일에 모아놓기
  - 코드를 설명하는 주석이 아닌 **동작하지 않는 코드에 대한 주석**을 지워주세요!
- 코드 포매팅
  - 탭 간격, 엔터 개수 등 코드 정렬 예쁘게 수정
  - 코드가 예쁘게 정렬되어 있어야 보기 편하겠죠?
- html 파일 없애기
  - `views/` 폴더 안에는 `ejs` 파일만 남겨놓기
- `package.json` 수정하기

(`package.json` 파일은 누군가가 `node.js` 프로젝트를 볼때 가장 먼저 열게되는 파일, 즉 해당 프로젝트의 첫 인상입니다. 깔끔하게 정리해두면 좋겠죠?)

```
{
  "name": "first-node", // 프로젝트 이름
  "version": "1.0.0",
  "description": "이 프로젝트는 어찌고 저찌고..", // 프로젝트에 대한 간단한 설명 추가
  "main": "app.js", // 진입점 파일명 index.js -> app.js 변경겨
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Z팀", // 팀명 추가
  "license": "ISC",
  "dependencies": {
    "ejs": "^3.1.8",
    "express": "^4.18.2"
  }
}
```

- 버그 최종 수정
  - 오류가 없어야 한다!!

- 사용하지 않는 branch 지우기
  - 작업이 완전히 끝난 local branch, remote branch들은 지워주세요!

## 프로젝트 정리하기

### [필수] Github README 작성

#### ▼ README란?

Github에 프로젝트를 올릴 때, 프로젝트에 대한 설명, 사용 방법 등을 기술하는 파일로 해당 repository에서 가장 먼저 마주치는 파일입니다. 따라서 내용이 **간단하면서도 구체적이어야** 합니다.

README는 왜 필요할까?

즉, 미래의 본인이 프로젝트를 다시 분석/작업해야 하거나 다른 유저(협업자 등)가 readme를 읽고 이 프로젝트에 대한 정보를 파악할 수 있도록 하는 것이 목적이다.

README는 어떻게 작성해야할까?

보통 마크다운(확장자: `.md`)으로 많이 작성한다. `README.md` 파일을 생성해 마크다운 문법에 따라 프로젝트에 대한 설명을 작성한다.

- **방법1. github repository에서 `README.md` 생성**
  - 이 경우 remote에서 변경사항이 생겼으므로 local에 pull 받아야 함
- **방법2. local project 에서 `README.md` 파일 생성**
  - 이 경우 local에서 변경사항이 생겼으므로 remote에 push 해야 함
- **[참고] README 작성법 (마크다운, 이미지, Gif, 코드 블록 사용법)**
- **[참고] README 포함되면 좋은 내용들**
- **[참고] Github README 쓰는 법**

### [필수] 마무리하기

- 사이트 실행 영상 전체 녹화 제출

- 브라우저를 모니터 전체 사이즈만큼 키운 후, 사이트 동작 내용을 동영상으로 녹화해서 **DM으로 제출**해주세요! (분량 1분 내외)
- **프로젝트 회고 작성하기** (4주차 블로깅 내역은 프로젝트 회고를 작성하는 것으로 대체합니다!)
- 개인 블로그에 프로젝트 회고를 작성하여 Slack DM으로 강사에게 전송해주세요~
- 잘한 점
- 아쉬운 점
- 프로젝트 설명
- 내가 담당할 부분
- 자유롭게 작성해주세요!!