**목차**

**1. HTML**

**기본규칙**

**2. Commit message**

**기본규칙**

**Title**

**Description**

**Branch naming rule**

**3. Function**

**Js function**

**View function**

**4. Etc**

**Doc template**

**1. HTML**

**[기본규칙]**

다음의 요소 규칙을 사용한다

# 페이지는 크게 <navbar>, <section1>, <section2>… , <footer>로 나눈다

# section 아래의 요소 이름 소문자로만 정의하며 다음과 같이 정의한다.

<div class = “section명\_\_div명\_\_요소명”></div>

ex) <div class = “new\_\_chart\_\_title”></div>

# section은 id로 분류하며 나머지 div명은 가급적 class로 정의한다.

**2. Commit message**

**[기본규칙]**

# commit 관련 메시지는 영문으로 작성한다.

**[Title]**

# Title은 되도록 간단하게 40자 이내로 작성하며 첫 글자는 대문자로 시작한다.

# 다수를 수행한 경우 “&” 붙여 작성하며 이 경우 40자 이상의 글자로 작성할 수 있다.

# 다음과 같은 형식으로 작성한다.

1. 수정한 파일 중심의 Title 작성시,

[do list] + [something] + [route] : ex) Add navbar in index.html

1. 구현한 기능 중심의 Title 작성시,

[do list] + [Model / Template / View 명] + [MTV 형식]

ex) Update login view

# Dolist는 다음을 참조한다.

**2. Commit message**

**[Do list]**

# add: add new function or section in existed file

# delete: delete contents in existed file

# update: update in existed file

# create: create new file

# remove: remove file

# rename: rename file

# transfer: change files directory

# fix: fix bug

# test: test issue

# etc: etc

**2. Commit message**

**[Description]**

# Description은 관련된 이슈 넘버를 최상단에 작성하며 “어떻게” 가 아닌 “무엇을”, “왜” 를 중심으로 작성한다.

**[branch naming rule]**

마스터 브랜치로부터 다음과 같이 브랜치를 생성하여 사용한다.

# own: 프로젝트에 사용될 개인 브랜치

ex) own/user/year & date

# hotfix: master 브렌치 내에서만 사용하며 심각한 오류가 있는 경우 사용

ex) hotfix/issue number

# feature: develop 브렌치로부터 생성하는 브렌치로 실험용 브렌치로 자유롭게 사용

ex) feature/issue name

**3. Function**

**[js function]**

**4. Etc**

**[Doc template]**

doctemplate

# doctemplate

# doctemplate

**[Doc template]**

doctemplate

# doctemplate

# doctemplate