

코멘토 – 1주차 피드백

[툭아보기] – 공통 피드백 모아보기

1. div vs section

<https://cafe.naver.com/codingstudypplace/112>

2. SEO

<https://cafe.naver.com/codingstudypplace/111>

3. 폴더 구조

<https://cafe.naver.com/codingstudypplace/74>

4. git commit

<https://cafe.naver.com/codingstudypplace/118>

5. html 작성 규칙

<https://jeonghakhur.gitbooks.io/frontend-coding-convention/content/html/>

6. 스타일 초기화

<https://cafe.naver.com/codingstudypplace/62>

7. 네이밍 컨벤션

<https://cafe.naver.com/codingstudypplace/102>

8. 클래스 네이밍

<https://cafe.naver.com/codingstudypplace/92>

9. 반응형

<https://cafe.naver.com/codingstudypplace/101>

2. 안유리

[질문1]

코드를 작성할 때, 주석을 최대한 많이 작성하는 것이 좋은가요?

[답변1]

주석은 필요성에 따라 작성해주시면 됩니다. 많다고 좋은 것이 아닌, 의미가 있어야 합니다.

공부하면서 모르는 부분에 대해 주석을 작성할 수 있습니다. 이런 주석은 공부를 해나가며 점차적으로 줄어들게 되는 부분입니다.

현업에서 주석을 작성하는 이유는 협업을 하면서 다른 동료 개발자가 빠르게 파악하고 로직이 복잡할 때 부연 설명을 위해 주로 사용합니다.

좋은 주석의 기준을 정리해보자면, 아래와 같이 정리해볼 수 있습니다.

- 1.코드를 왜 이렇게 짰는지 설명
- 2.복잡한 로직 요약 (복잡한 조건문, 반복문 등에 대한 요약)
- 3.응답 포맷에 대한 부연 설명이 필요할 때
- 4.리팩토링 예정, 향후 작업에 대한 설명이 필요할 때

주석보다 우선시 되어야 할 것은

- 1.명확한 네이밍 (함수/변수명이 역할을 설명하면 주석이 필요 없어짐)
- 2.함수 분리 (복잡한 로직은 함수를 분리하고 이름으로 의도 표현)
- 3.시맨틱 태그 사용 (의미 있는 태그 구조로 주석 의존도 줄이기)

Style.css 코드에서 header, main, footer, 섹션별 시작점에 대한 주석 등을 달아 놓으셨는데 해당 주석들은 불필요한 내용들이 아니기 때문에 작성하셔도 무방합니다.

[질문2]

페이지 디자인을 구성하여 코드를 작성해야할때는, 우선적으로 피그마로 레이아웃 및 디자인을 정하고 코드를 작성하게 되는 것인가요?

[답변2]

디자인 팀에서 디자인 및 디자인 가이드를 작성해드립니다. 보통 가이드를 먼저 만들고 이를 토대로 디자인을 진행합니다.

디자인 가이드란, 공통으로 사용할 부분에 대한 정의입니다.

아이콘, 컬러, 버튼, 로딩화면, 캘린더, 팝업, selector, input, 폰트, 탭메뉴, checkbox, radiobox 등을 정의합니다.

예를 들어보겠습니다.

*버튼 크기별 정의

디자인 타입 (primary, border, highlight 등) / 버튼 크기 (xs, sm, md, lg 등)

*팝업 형태 정의

Top, center, bottom, full 등

정의한 이름은 디자인팀에서 부여하기 때문에 고민할 필요는 없습니다.

프론트팀에서는 가이드를 보고 공통으로 사용할 컴포넌트 스타일을 만듭니다.

반복하여 사용하는 디자인(레이아웃)을 작성해놓고 필요에 따라 불러와 사용하면 중복되는 코드를 줄일 수 있고 편의성 또한 올라갑니다.

결론적으로, 페이지 디자인이 들어가기 전에 디자인팀에서 가이드를 제공해줍니다.

[질문3]

HTML을 작성할 때 div를 너무 많이 사용하게 되는데, class나 id 속성값만 잘 설정해주면 괜찮을까요?

[답변3]

화면 구성을 위해 유의미한 div 태그를 작성한다면 많이 작성한다고 해서 문제가 되지 않습니다.
클래스명, 아이디값 또한 많이 작성한다고 해서 잘못된 코드 작성이 아니니 필요에 따라 작성해주시면 됩니다.
작성하신 코드 중에 불필요한 태그 없이 적절하게 사용되었습니다.

[질문4]

h1과 같은 hn 태그를 사용함에도 폰트 사이즈나 weight을 변경하게 되면, hn 태그 대신 p 태그를 사용하는 것이 나은 방법일까요?

[답변4]

작성하신 코드를 보면 font-size: medium 과 같이 웹 브라우저에서 정의한 사이즈가 적용되게끔 적용하셨습니다.
제목 태그들의 크기 또한 멘티님이 원하는 크기로 지정해놓고 사용하시면 됩니다. 회사에서는 이미 디자인 가이드로 지정합니다.
멘티님이 원하는 사이즈로 지정을 하시고 사용해주시면 되고 제목 태그를 사용해야 하는 부분에 p 태그를 사용할 필요는 없습니다.

[질문5]

박스 여백을 줄 때, padding과 margin 중 선택하는 기준이 있을까요?

[답변5]

padding과 margin은 모두 여백을 조정하는 속성이지만, 용도와 개념이 완전히 다릅니다.

| 속성 | 개념 | 작용 대상 | 영향 범위 |
|---------|-------|-----------------------|----------|
| padding | 내부 여백 | 컨텐츠와 경계(border) 사이 공간 | 요소 내부 변화 |
| margin | 외부 여백 | 요소와 외부 요소 사이 공간 | 요소 바깥 변화 |

| 상황 | 사용 권장 속성 | 이유 |
|---------------------------|----------|---------------------|
| 텍스트나 이미지와 경계선 사이 공간 확보 | padding | 내부 콘텐츠의 가독성 확보 |
| 요소 간 간격 조절 (예: 카드와 카드 사이) | margin | 인접 요소들과 거리 조절 |
| 배경색/테두리와 함께 공간 유지 | padding | 여백도 시각적 스타일의 일부로 포함 |
| 수직 정렬 시 간격 설정 | margin | 요소 간 블록 간격을 명확히 설정 |
| 클릭 영역 확장 | padding | 실제 박스 크기를 늘려 UX 개선 |
| 부모 요소 배경색 안에서 정렬 | padding | 배경 안에서 내부 콘텐츠 위치 조정 |

위아래 마진이 겹칠 수 있는데요 (두 블록 요소가 인접할 때) 아래 방법으로 방지할 수 있습니다.

- 1.overflow: hidden 또는 auto -> 부모 요소에 적용 시 내부 마진 방지
- 2. Padding 사용 -> 두 요소 사이에 실제 물리적 구분을 만들어 방지
- 3. display: flex, grid 사용 -> 플렉스 컨테이너 내부에서는 병합 발생하지 않음

[피드백]

-코드 작성 및 문서 작성, 예제 코드 작성에 고생 많으셨습니다. 내용 정리에 대해서도 정말 열심히 해주신 것이 결과로 보입니다. 문서 작성을 하면서도 아셨겠지만 기본 이론을 정리하며 코드에 적용해보고 여러 갈래로 뻗어가면서 디테일하게 해나가시면 됩니다. 완성도가 매우 높으며 코드 사용에 대한 이해를 많이 하고 계십니다.

-style.css 파일 내 width: 100%; /* 이걸 하지 않으면 오른쪽 정렬이 되지 않습니다. 이유가 궁금합니다 */
=> 답변 : nav 에 display: flex; justify-content: flex-end; 을 작성하셨는데 justify-content: flex-end 해당 속성은 nav 태그 내에 있는 자식 요소들에 적용하겠다는 얘기입니다. 그렇기 때문에 nav 태그 자체에 적용되지 않습니다.
부모 header 태그에 flex 속성을 부여하고 자식 요소인 p, nav 에 대한 정렬은 header 에 해주어야 합니다.
자식 요소인 p, nav 를 양쪽 끝으로 배치하고자 한다면 header 에 justify-content: space-between; 를 추가하면 됩니다.
그러면, 현재 코드 nav에 작성된 *justify-content: flex-end; width: 100%;* 는 없어도 됩니다.

-시맨틱 태그에 대한 이해를 가지고 적절한 태그를 사용하셨습니다.
시맨틱 태그가 중요한 이유는 SEO 와 웹 문서에서 어떤 역할을 하는지 파악하기 위해서입니다.
아래 정리된 내용이 있으니 확인해보세요.
SEO : <https://cafe.naver.com/codingstudyplace/111>

-style 은 reset.css / common.css / 각 페이지별 style 파일로 구성해볼 수 있습니다. 각 파일별 용도에 대한 설명을 살펴보겠습니다.
*reset.css : 각 브라우저들마다 기본 스타일이 다르게 적용되어 있기 때문에 이를 통일하기 위해 작성
*common.css : 모든 페이지들에서 공통으로 사용할 스타일 모음 (예를 들어, header, footer, 버튼 크기별 스타일, 컬러값 정의, 폰트 크기, 팝업 스타일 등)
*각 페이지별 스타일 파일 : 각 페이지별로 스타일 파일 따로 작성 (예를 들어, main.css, sub_page1_style.css 등)
css 스타일 초기화 : <https://cafe.naver.com/codingstudyplace/62>

-제목 태그 중 h1 태그는 해당 페이지를 대표적으로 설명합니다. 보통 header 에서 메뉴와 함께 사용합니다.
작성하신 코드에서도 header 에 있는 INTRODUCTION 이 h1 에 해당합니다. 나머지 section 안에 있는 h1 태그들은 h2 태그로 사용하는 것이 적절한 표현입니다.

-main 태그 안에 article 태그를 사용한 부분 또한 section 태그를 사용하는 것이 조금 더 적절합니다.
article 태그은 독립적이고 재사용하는 콘텐츠가 아닙니다.

-<h4>Phone</h4>, <h4>GitHub</h4>, <h4>E-mail</h4> 는 구성상 h3 로 작성하시면 됩니다.
중간에 건너뛰고 작성하는 것이 아닌 순서 흐름상으로 작성해주시면 됩니다. 중간 레벨을 뛰어넘어서 작성하지 않습니다.

-html 코드에서 코드 구분을 위해 엔터키를 사용하신 것 같습니다.
아직은 공부중이고 연습이니 괜찮습니다. 추후 현업에서는 엔터키가 필요한 상황이 아닌 이상, 사용하지 않으니 참고하시기 바랍니다.

-공통 피드백 [툇아보기] 내용도 같이 확인해주세요.