

- ## CSS Methodology

SMACSS, OOCSS, BEM 에 대해서 소개한다.

SMACSS(Scalable and Modular Architecture for CSS)

SMACSS 의 핵심은 범주화이며(`categorization`) 스타일을 다섯 가지 유형으로 분류하고, 각 유형에 맞는 선택자(selector)와 작명법(naming convention)을 제시한다.

- 기초(Base)
 - element 스타일의 default 값을 지정해주는 것이다. 선택자로는 요소 선택자를 사용한다.
- 레이아웃(Layout)
 - 구성하고자 하는 페이지를 컴포넌트를 나누고 어떻게 위치해야하는지를 결정한다. `id` 는 CSS 에서 클래스와 성능 차이가 없는데, CSS 에서 사용하게 되면 재사용성이 떨어지기 때문에 클래스를 주로 사용한다.
- 모듈(Module)
 - 레이아웃 요소 안에 들어가는 더 작은 부분들에 대한 스타일을 정의한다. 클래스 선택자를 사용하며 요소 선택자는 가급적 피한다. 클래스 이름은 적용되는 스타일의 내용을 담는다.
- 상태(States)
 - 다른 스타일에 덧붙이거나 덮어씌워서 상태를 나타낸다. 그렇기 때문에 자바스크립트에 의존하는 스타일이 된다. `is-` prefix 를 붙여 상태를 제어하는 스타일임을 나타낸다. 특정 모듈에 한정된 상태는 모듈 이름도 이름에 포함시킨다.
- 테마(Theme)
 - 테마는 프로젝트에서 잘 사용되지 않는 카테고리이다. 사용자의 설정에 따라서 css 를 변경할 수 있는 css 를 설정할 때 사용하게 되며 접두어로는 `theme-` 를 붙여 표시한다.

OOCSS(Object Oriented CSS)

객체지향 CSS 방법론으로 2 가지 기본원칙을 갖고 있다.

- 원칙 1. 구조와 모양을 분리한다.
 - 반복적인 시각적 기능을 별도의 스킨으로 정의하여 다양한 객체와 혼합해 중복코드를 없앤다.
- 원칙 2. 컨테이너와 콘텐츠를 분리한다.
 - 스타일을 정의할 때 위치에 의존적인 스타일을 사용하지 않는다. 사물의 모양은 어디에 위치하든지 동일하게 보여야 한다.

BEM(Block Element Modifier)

웹 페이지를 각각의 컴포넌트의 조합으로 바라보고 접근한 방법론이자 규칙(Rule)이다. SMACSS 가 가이드라인이라는 것에 비해서 좀 더 범위가 좁은 반면 강제성 측면에서 다소 강하다고 볼 수 있다. BEM 은 CSS 로 스타일을 입힐 때 `id` 를 사용하는 것을 막는다. 또한 요소 선택터를 통해서 직접 스타일을 적용하는 것도 불허한다. 하나를 더 불허하는데 그것은 바로 자

손 선택자 사용이다. 이러한 규칙들은 재사용성을 높이기 위함이다.

- Naming Convention
 - 소문자와 숫자만을 이용해 작명하고 여러 단어의 조합은 하이픈(-)과 언더바(_)를 사용하여 연결한다.
- BEM의 B는 "Block"이다.
 - 블록(block)이란 재사용 할 수 있는 독립적인 페이지 구성 요소를 말하며, HTML에서 블록은 class로 표시된다. 블록은 주변 환경에 영향을 받지 않아야 하며, 여백이나 위치를 설정하면 안된다.
- BEM의 E는 "Element"이다.
 - 블록 안에서 특정 기능을 담당하는 부분으로 block_element 형태로 사용한다. 요소는 중첩해서 작성될 수 있다.
- BEM의 M는 "Modifier"이다.
 - 블록이나 요소의 모양, 상태를 정의한다. block_element-modifier, block-modifier 형태로 사용한다. 수식어에는 불리언 타입과 키-값 타입이 있다.

normalize vs reset

브라우저마다 기본적으로 제공하는 element의 style을 통일시키기 위해 사용하는 두 CSS에 대해 알아본다.

reset.css

reset.css는 기본적으로 제공되는 브라우저 스타일 전부를 제거하기 위해 사용된다. reset.css가 적용되면 ~, , `` 등 과 같은 표준 요소는 완전히 똑같이 보이며 브라우저가 제공하는 기본적인 styling이 전혀 없다.

normalize.css

normalize.css는 브라우저 간 일관된 스타일링을 목표로 한다. ~과 같은 요소는 브라우저간에 일관된 방식으로 굵게 표시됩니다. 추가적인 디자인에 필요한 style만 CSS로 작성해주면 된다.

즉, normalize.css는 모든 것을 "해제"하기보다는 유용한 기본값을 보존하는 것이다. 예를 들어, sup 또는 sub와 같은 요소는 normalize.css가 적용된 후 바로 기대하는 스타일을 보여준다. 반면

reset.css를 포함하면 시각적으로 일반 텍스트와 구별할 수 없다. 또한 normalize.css는 reset.css보다 넓은 범위를 가지고 있으며 HTML5 요소의 표시 설정, 양식 요소의 글꼴 상속 부족, pre-font 크기 렌더링 수정, IE9의 SVG 오버플로 및 iOS의 버튼 스타일링 버그 등에 대한 이슈를 해결해준다.