**--Html, Css, JavaScript 에 대하여--**

Html은 웹페이지를 만드는 뼈대이고 css는 html의 외형을 디자인하는 근육 같은 역활을 수행함 그리고 몸을 움직이거나 생각을 해 상호작용,서버와 통신, 데이터 처리 등을 하는 웹(html)에 동적인 움직주는 것은 javascript의 역할임. Html은 기본적으로 오류가 존재하지 않지만 정해진 규칙대로 입력하지 않으면 에러는 뜨지 않지만 html이 읽지 못함.

**--html—**

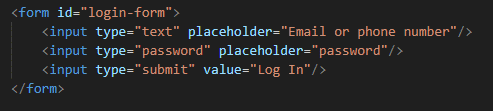
**Id**와 **class** 의 차이

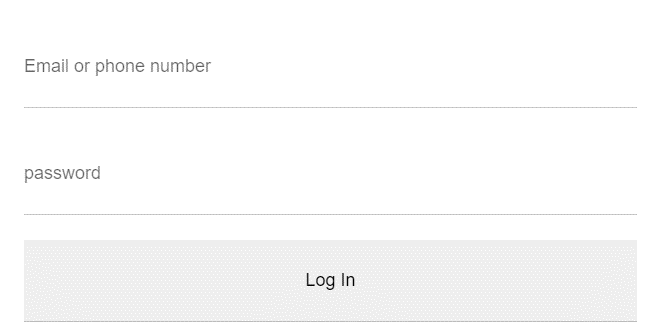
**id**는 고유성이고 class선택자보다 특정성이 높아 스타일 적용되는 우선순위가 높음 하지만 재사용이 불가능해 여러 요소에 적용하는데 부적합하고 한 요소에만 적용이 되서 유연성이 떨어짐

**class**는 여러 요소에 동일하게 적용할 수 있어 재사용성이 좋고 유연성도 뛰어남.

**div** 는 각 구역을 나누기 위해 사용되어 css를 지정하거나 레이아웃 구성하는데 사용이 됨.

**Form**은 주로 데이터를 서버로 제출하는 데 사용이 됨.

<<=(form)

<<=(form)

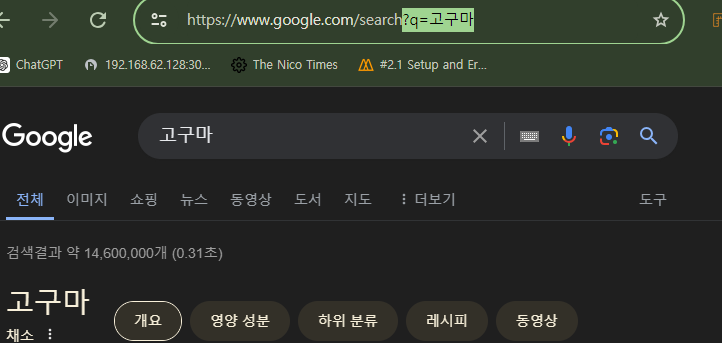
Form은 중요한 2가지 속성 action 와 method 가 있다.

**Action**은 어떤 페이지로 data를 보낼지 지정할 수 있음

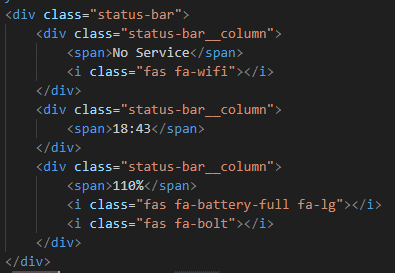
**Method**는 POST 와 GET인 2가지 방법으로 사용이 가능함

**POST** : 백엔드 서버에 정보 전송하는 방식

**GET** : 보안 취약하므로 가벼운 정보를 GET방식으로 공유 /업로드할 때 사용

 🡸GET 방식 사용

<<=(dive)

<<=(dive)

**Header**은 도입부에 해당하는 콘텐츠 정보,인사말 등을 만드는데 사용

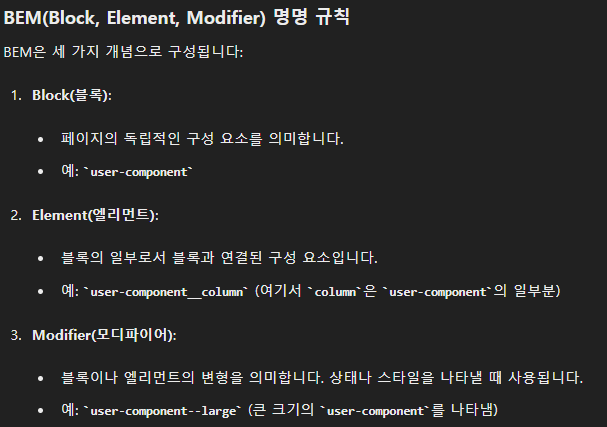
**Nav**는다른 웹 페이지로 연결하거나, 현재 웹 페이지의 콘텐츠 내부로 연결되는 탐색(navigation)을 위한 링크(links)가 있는 영역

**--css--**

테그는 (테그명){ (스타일) } 으로 시작함

Class는 .(클레스 명){ (스타일) } 으로 시작함

Id는 #(아이디 명){ (스타일) } 으로 시작함



가상 클래스 선택자(pseudo element)인 placeholder

@import “주소.css“;를 통해 css끼리 불러와 css파일을 분할해서 정리하며 사용할 수 있음

**Padding**은 안쪽 공간의 공백을 설정하는 스타일이고 margin은 바깥쪽 공간을 설정하는 css 스타일이므로 주로 요소와 요소의 간격을 만들고 싶을 때 사용됨.

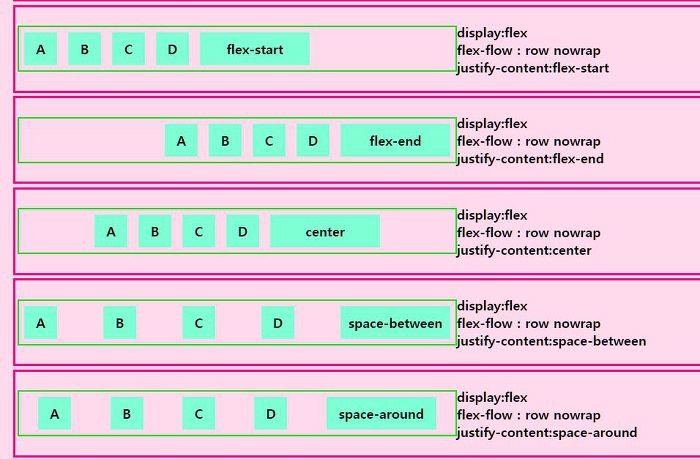
**box-sizing: border-box**을 사용하면 박스 크기를 고정시킴.

Ex) 너비가 200px이고 패딩이 20px, 테두리가 10px인 요소의 차지하는 너비는 260px( 200 + (20 \* 2 )+ ( 10 \* 2 ) )이다. 하지만 box-sizing을 사용하면 너비 200px이 된다.

**position: fixed** 는 스크롤을 내려도 요소를 고정시켜주는 css 속성.

**position: absolute**는 요소의 위치를 자유롭게 배치할 수 있도록 하는 속성 가장 가까운 position: relative를 가진 부모를 기분으로 지정됨.

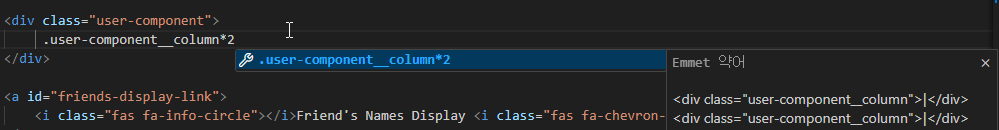
justify-content는 주축에서 자식요소 배치 결정



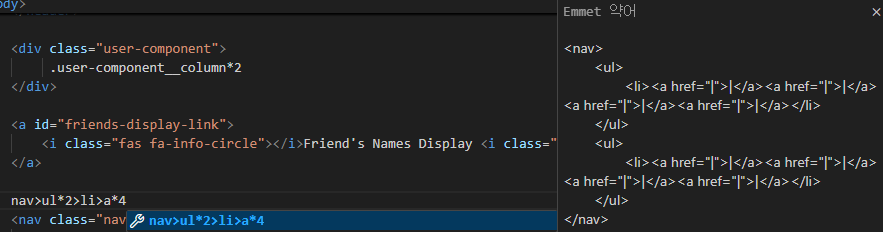
align-items는 세로측에서 자식요소 배치 결정



**단축키/단축어(꿀팁)**



이런형식으로 여러 클레스 지정 가능



중첩반복 지정