* placeholder는 html5에 추가된 기능으로 글자의 기본 색이 회색이다. 이 색을 바꾸면 크롬, 파이어폭스, 오페라, 사파리 최신 버전에서는 잘 적용된다. 그러나 크롬, 파이어폭스, 오페라, 사파리의 구버전과 Edge, IE에서는 적용되지 않는다. 따라서 이를 해결하는 코드를 추가한다.(영상에서 prefix되어있으면 이 과정이 필요 없다고 했다. 오래된 방식이라고 함.)
* text-indent:18px; padding:0 60px 0 0; 이 두 문장을

padding:0 60px 0 18px;

이렇게 줄여 쓸 수 있다. 그러나 아래 문장의 경우 입력 칸을 넘는 입력이 들어올 때 좌측의 글자가 입력 칸을 넘지 않는 입력과 같게 보이기 때문에 사용자는 오버플로가 발생했는지 알기 어렵다. 따라서 위처럼 두 줄로 나누어 쓰는 것이 사용자 친화적이다.

* css에서 .fill의 fill property 값을 변경한 게 적용이 안됐다. html의 button의 class name을 st0에서 fill로 바꿔야 했는데 하지 않아서 fill class를 찾을 수 없었던 것이다.
* js에서 log를 작성할 때 ‘와 “ 모두 사용할 수 있다. html과 js의 구분을 위해 html에서 “, js에서 ‘를 쓴다는 사람이 있는데 이 방식이 마음에 든다. {}가 아니라 ()이다.
* script tag를 body가 아니라 head에 쓰면 main.js에서 출력한 button id log는 null이다. HTML은 top에서 bottom으로 load하기 때문에 head를 load할 때 body의 정보는 존재하지 않는다. button id는 body에 있는 정보이므로 main.js에서 가져올 수 없다. 따라서 script tag는 body의 bottom에 위치하는 것이 좋다. 정말 정말 head에 두고 싶다면 js file의 첫 줄에 window.onload = function() { … }을 작성한다. … 부분에 html body 영역의 정보를 가져오는 코드를 작성하면 정상적으로 동작한다. {} 밖에 log를 출력하는 코드를 적으면 그 결과가 먼저 출력된다. onload는 body의 정보를 다 가져온 후 실행해야 하기 때문이다.
* document.getElementById('add').addEventListener('click', function() {});

두 개의 argument는 각각 event의 종류와 callback 함수이다.

* 아래 두 가지는 같은 역할을 한다.

document. getElementById('add').onclick

document.getElementById('add').addEventListener('click', function() {});

window.onload

window.addEventListener('load', function functionName() { … })

* html minifier를 이용해 손쉽게 newline을 제거할 수 있다.