

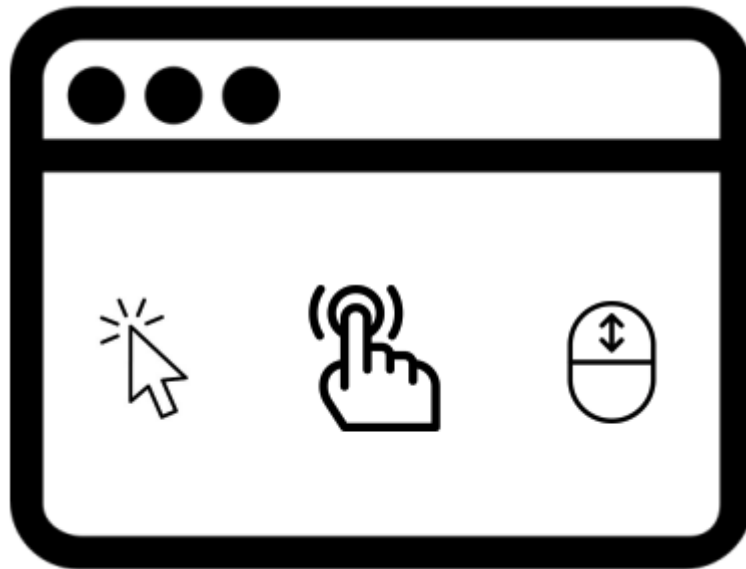
이벤트 흐름과 위임

👤 배정	
📌 상태	
🕒 속성	@2021년 8월 19일 오후 7:58
🗨 언어	

브라우저가 사용자의 입력을 받으면 어떤일이 일어날까??

1. 브라우저 화면에서 이벤트가 발생한다.

이때의 브라우저 관점의 이벤트란?



| 마우스의 click , 휠의 움직임, 포인터 이동, 화면 터치 등..

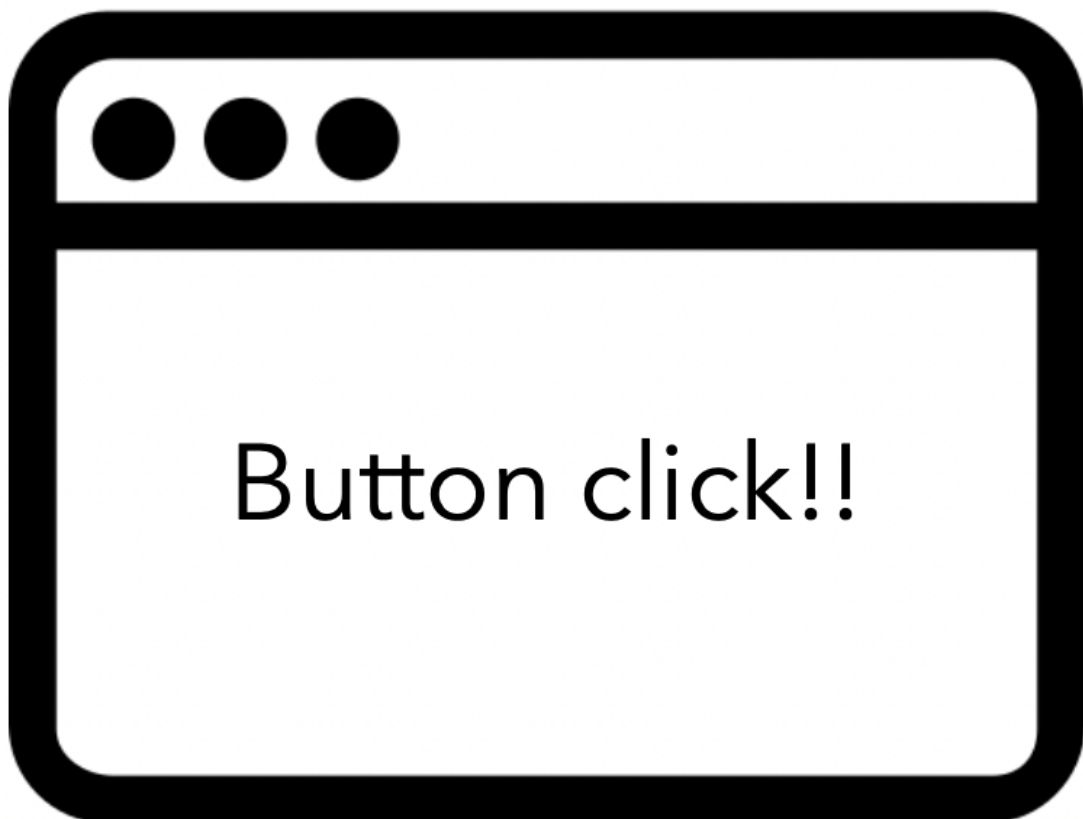
2. 이벤트가 발생했다! 그러면 브라우저는 제일 먼저 무슨일을 할까?



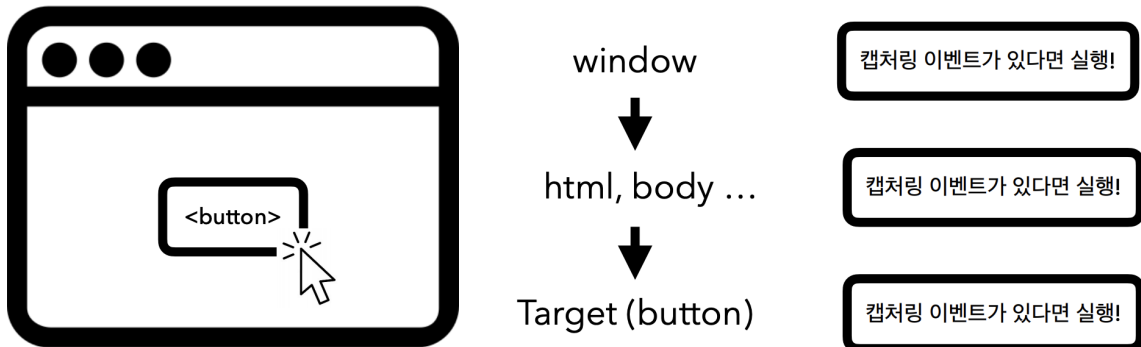
hmmteresting...

- 이벤트가 발생한 좌표에 무엇이 있는지 확인하기 위해 렌더링 과정중 하나인 페인트 기록을 찾는다.

3. 캡처링(capturing)



- 페인트 기록을 통해 좌표를 알아낸 브라우저는 해당 좌표에 위치한 요소의 이벤트 리스너를 실행한다.

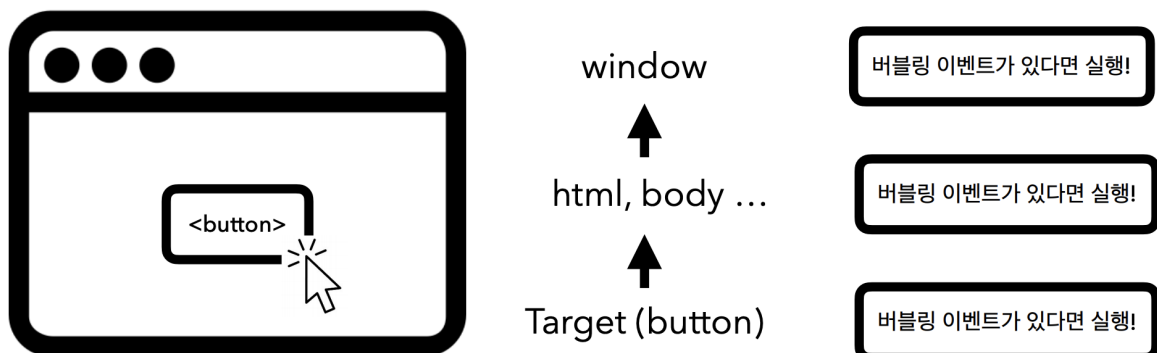


- 타겟 요소의 가장 최상위 window 객체로부터 캡처링 단계의 이벤트 리스너의 등록 여부 확인, 있다면 실행!
- 계속 자식 요소로 전파되며 만나는 캡처링 이벤트 리스너 실행하여 타겟 요소까지 이동한다.



최상위에서부터 이벤트 발생 지점으로 내려가면서 찾는다.

4. 버블링(bubbling)

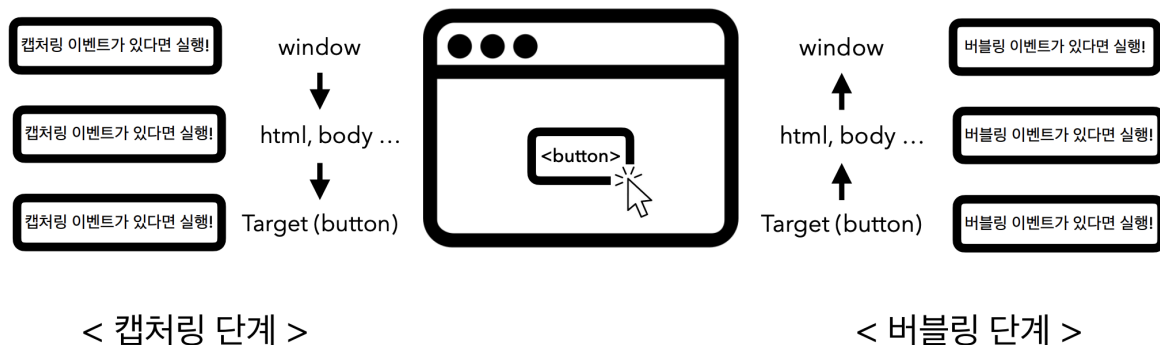


- 캡처링이 끝나고 최초에 이벤트가 발생했던 요소에 버블링 이벤트 리스너의 등록 여부 확인, 있다면 실행!
- 그 다음 직계 부모 요소에 버블링 이벤트 리스너이 등록 여부 확인 후 실행, 이를 가장 최상위 window 객체까지 계속 실행시킨다.



이벤트 발생 지점에서 최상위로 올라가면서 찾는다.

이벤트 캡처링과 버블링을 그림 하나로 정리한다.



왠지 연어의 일생이 떠오른다.(feat. 흐르는 강물을 거꾸로 거슬러 오르는 연어..)

요소에 이벤트를 등록하고 싶다면?

- 일반적으로 `addEventListener()` 를 이용한다. 그런데 100개 넘는 요소에 이벤트를 등록하고 싶다면..100개의 `addEventListener()` 는 가독성을 저하시키는건 안봐도 뻔하다.
- 이벤트 흐름을 잘 이용한다면 단 1개의 이벤트 리스너로 수많은 요소의 이벤트를 처리할 수 있다.
- 이벤트를 발생시키고 싶은 요소를 이벤트 리스너가 설치된 부모 요소의 자식으로 배치한다면 몇 개든 상관없이 이벤트 등록 가능
- 요소가 동적으로 생성되어 계속 추가되어도 같은 기능 유지

이벤트 위임(delegate)



이벤트 흐름을 활용하여 단일 이벤트 리스너가 여러개의 이벤트 대상을 처리할 수 있게 하는 프로그래밍

결론

1. 이벤트 캡처링 : 최상위에서부터 이벤트 발생 지점으로 내려가면서 찾는다.
2. 이벤트 버블링 : 이벤트 발생 지점에서 최상위로 올라가면서 찾는다.
3. 이벤트 흐름 : 특정 이벤트가 발생시 해당 이벤트가 발생한 요소를 찾는 과정에서 만나는 모든 이벤트 리스너를 실행한다.
4. 이벤트 위임 : 이벤트 흐름을 활용하여 단일 이벤트 리스너가 여러개의 이벤트 대상을 처리할 수 있게한다.