

[WEB UI BASIC]

JavaScript 4탄

NHN NEXT 우재우

본 자료는 NHN NEXT 윤지수 교수님의 자료를 바탕으로 제작했습니다.

Previously JavaScript...

특정 Element를 잡아와서

Event를 걸 수 있다!

오늘은 Event를 더 자세히 봅시당~!

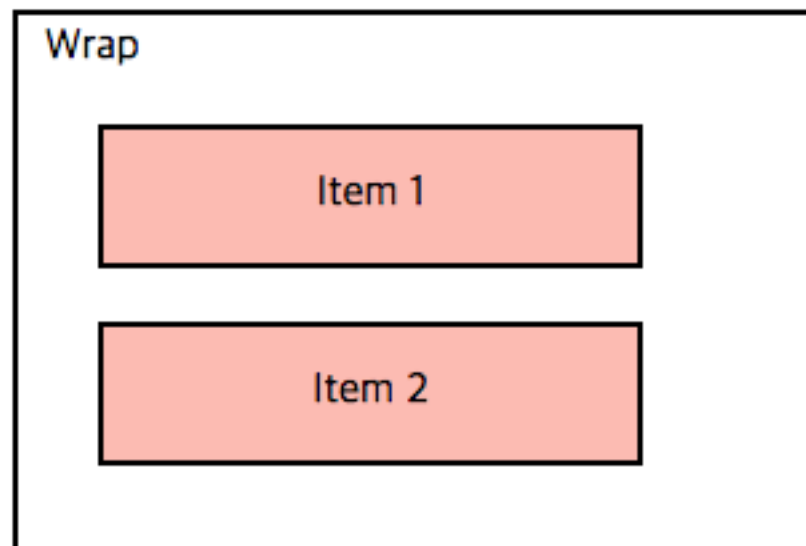
addEventListener(event, callback, useCapture);

addEventListener()

Item1과 Item2에 이벤트를 걸고 싶으면 앞에서 배운대로...

```
item1.addEventListener( ' click ' , function, false);
```

```
item2.addEventListener( ' click ' , function, false);
```



그런데 만약 Item이 100개라면?!?!



이렇게 하면 되죵 ^^

```
item1.addEventListener( ' click ' , function, false);  
item2.addEventListener( ' click ' , function, false);  
item3.addEventListener( ' click ' , function, false);  
item4.addEventListener( ' click ' , function, false);  
item5.addEventListener( ' click ' , function, false);  
item6.addEventListener( ' click ' , function, false);  
item7.addEventListener( ' click ' , function, false);  
item8.addEventListener( ' click ' , function, false);  
item9.addEventListener( ' click ' , function, false);  
...  
  
item99.addEventListener( ' click ' , function, false);  
item100.addEventListener( ' click ' , function, false);
```

와~!!!

코딩 많이 한 것 같아서 좋네요 :)

사실 이렇게 하면 폭망입니다 ㅋㅋㅋ

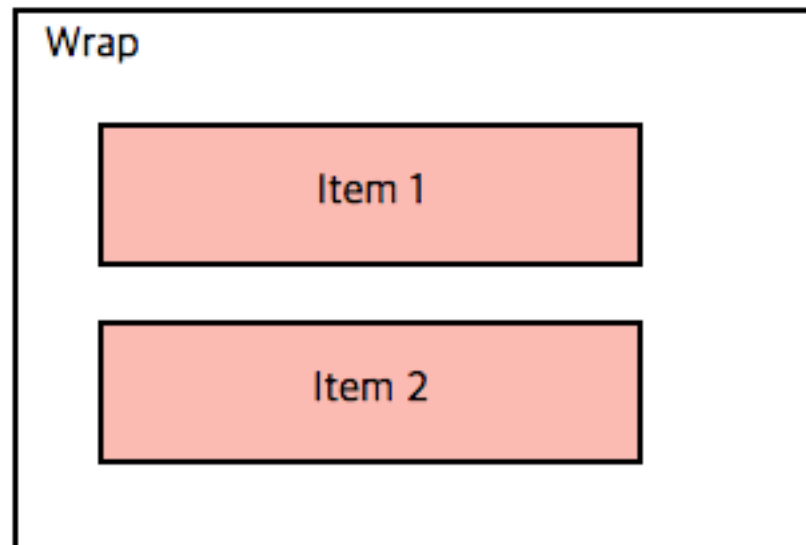
이벤트 핸들러가 많을 수록

페이지 성능이 떨어져요ㅠ

이벤트 위임(Event Delegation)

부모 Element에 이벤트 리스너를 걸어주는 것이 바로 이벤트 위임(Event Delegation)입니다.

```
wrap.addEventListener( ' click ' , function, false);
```



그리고, target!!!

`addEventListener()`의 콜백함수는 기본적으로 `event` 정보를 `parameter`로 가집니다. 그리고 `event`에는 `target`이라는 속성이 있는데, `event.target`이라고 쓰면 `event`가 발생한 `element`가 무엇인지 알 수 있습니다.

```
wrap.addEventListener( ' click ' , hello, false);  
function hello(event) {  
    console.log(event.target);  
}
```

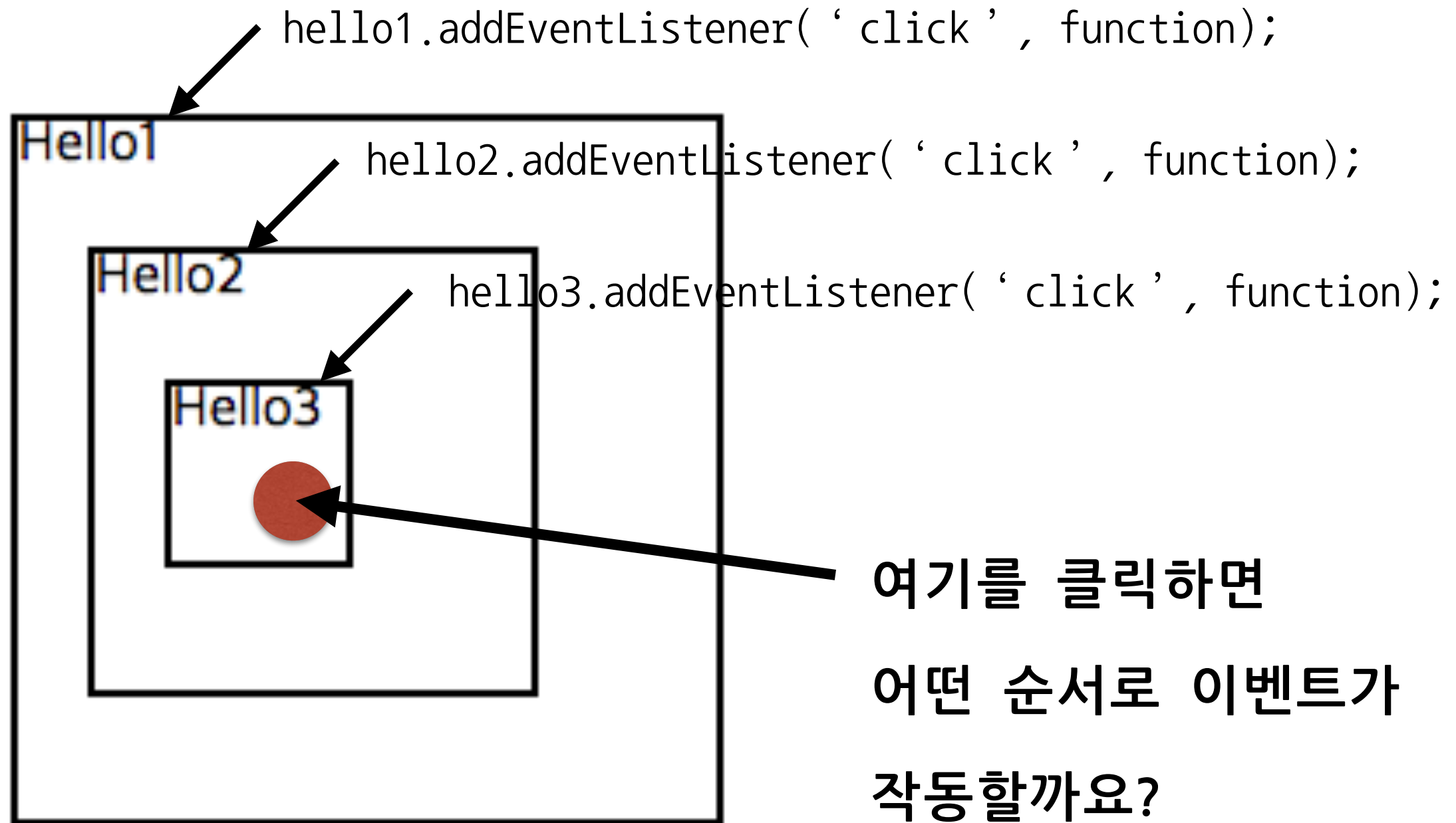
익명함수로 바로 쓸 때는

```
wrap.addEventListener( ' click ' , function(event) {  
    console.log(event.target);  
}, false);
```

`addEventListener(event, callback, useCapture);`



이 녀석을 알아보시다!

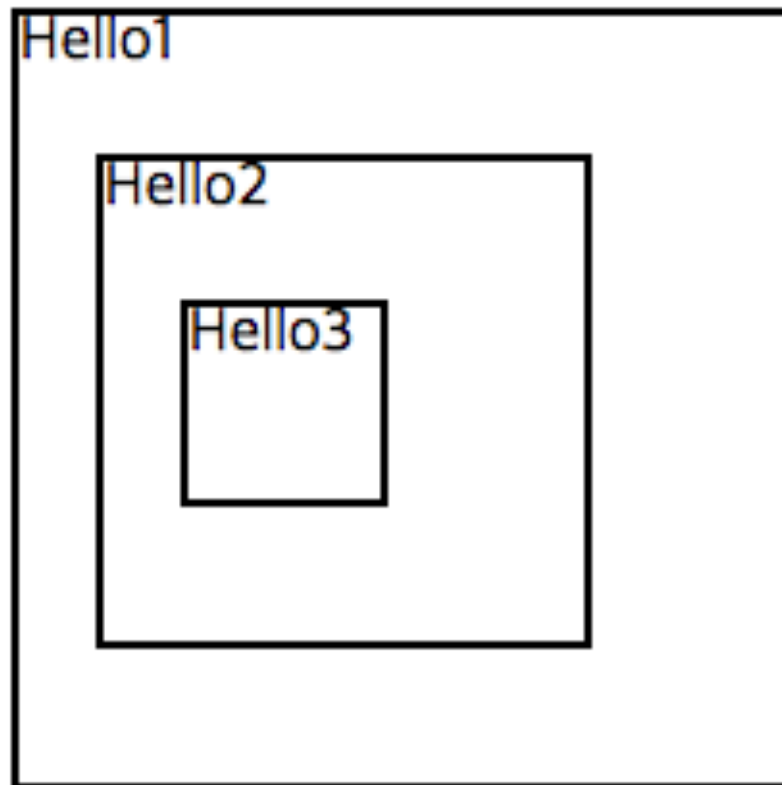


true / false

캡처링(Capturing) / 버블링(Bubbling)

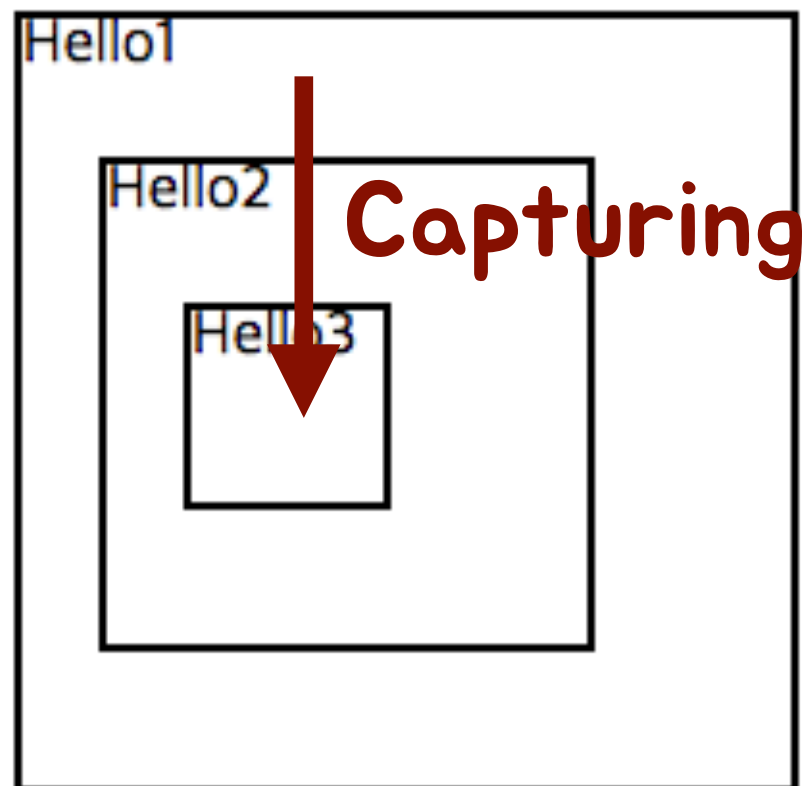
Capturing VS Bubbling

캡처링과 버블링은 이벤트가 전파되는 순서에 대한 설정입니다.
두 가지의 차이점은 전파되는 방향일 뿐입니다.



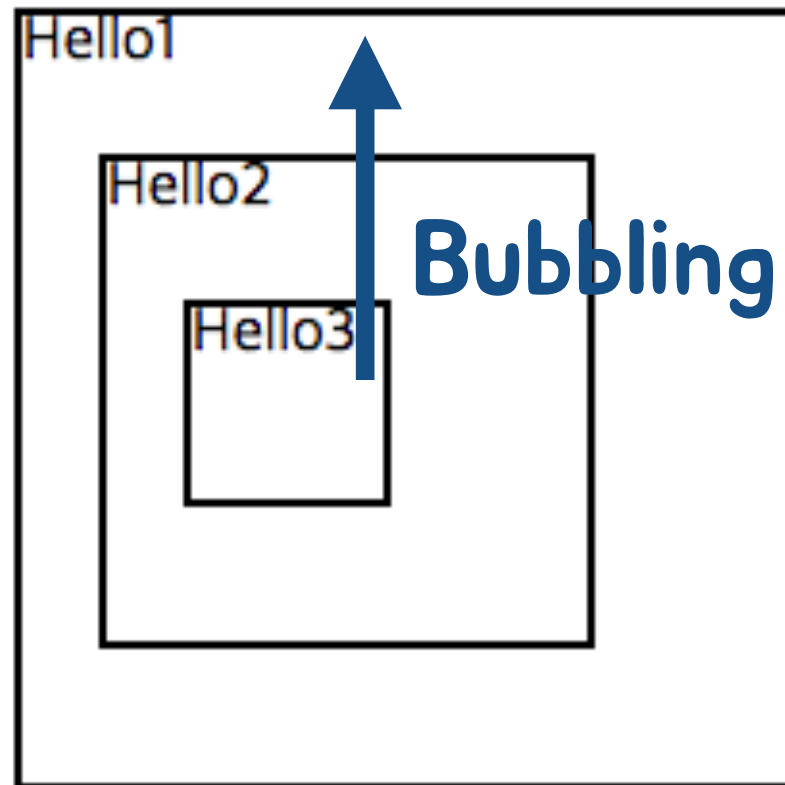
Capturing VS Bubbling

캡처링과 버블링은 이벤트가 전파되는 순서에 대한 설정입니다.
두 가지의 차이점은 전파되는 방향일 뿐입니다.



Capturing VS Bubbling

캡처링과 버블링은 이벤트가 전파되는 순서에 대한 설정입니다.
두 가지의 차이점은 전파되는 방향일 뿐입니다.



이렇게 하고 클릭해보세요!

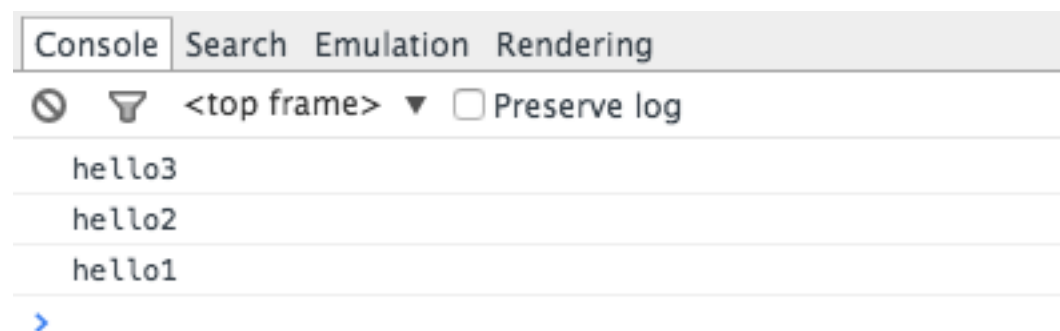
```
function hello(event) { console.log(this.id); }
```

여기서 `this`는 이벤트를 발생시킨 녀석을 말합니다.

```
hello1.addEventListener('click', hello, false);
```

```
hello2.addEventListener('click', hello, false);
```

```
hello3.addEventListener('click', hello, false);
```



이번에는 true로 바꾸고!

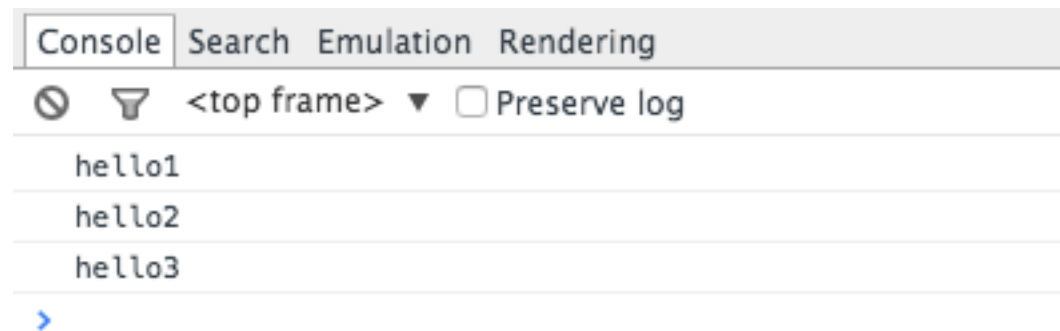
```
function hello(event) { console.log(this.id); }
```

여기서 `this`는 이벤트를 발생시킨 녀석을 말합니다.

```
hello1.addEventListener('click', hello, true);
```

```
hello2.addEventListener('click', hello, true);
```

```
hello3.addEventListener('click', hello, true);
```



지금은 이해가 잘 안 될 수도 있으니

그냥 이렇게 있다더라

정도만 해둡시다 ㅋㅋㅋ

이제 이벤트는 이쯤하고

지난 시간 실습 문제를 같이 해보고 마치죠

[실습]

버튼을 클릭하면 1부터 45 사이의 랜덤한 숫자가 적힌 상자가
중복 없이 7개까지 추가되는 프로그램 짜기

예)

추 가

32

23

1

17

14

41

5

이때까지 배운거 다 써야 짤 수 있을걸요? 후훗