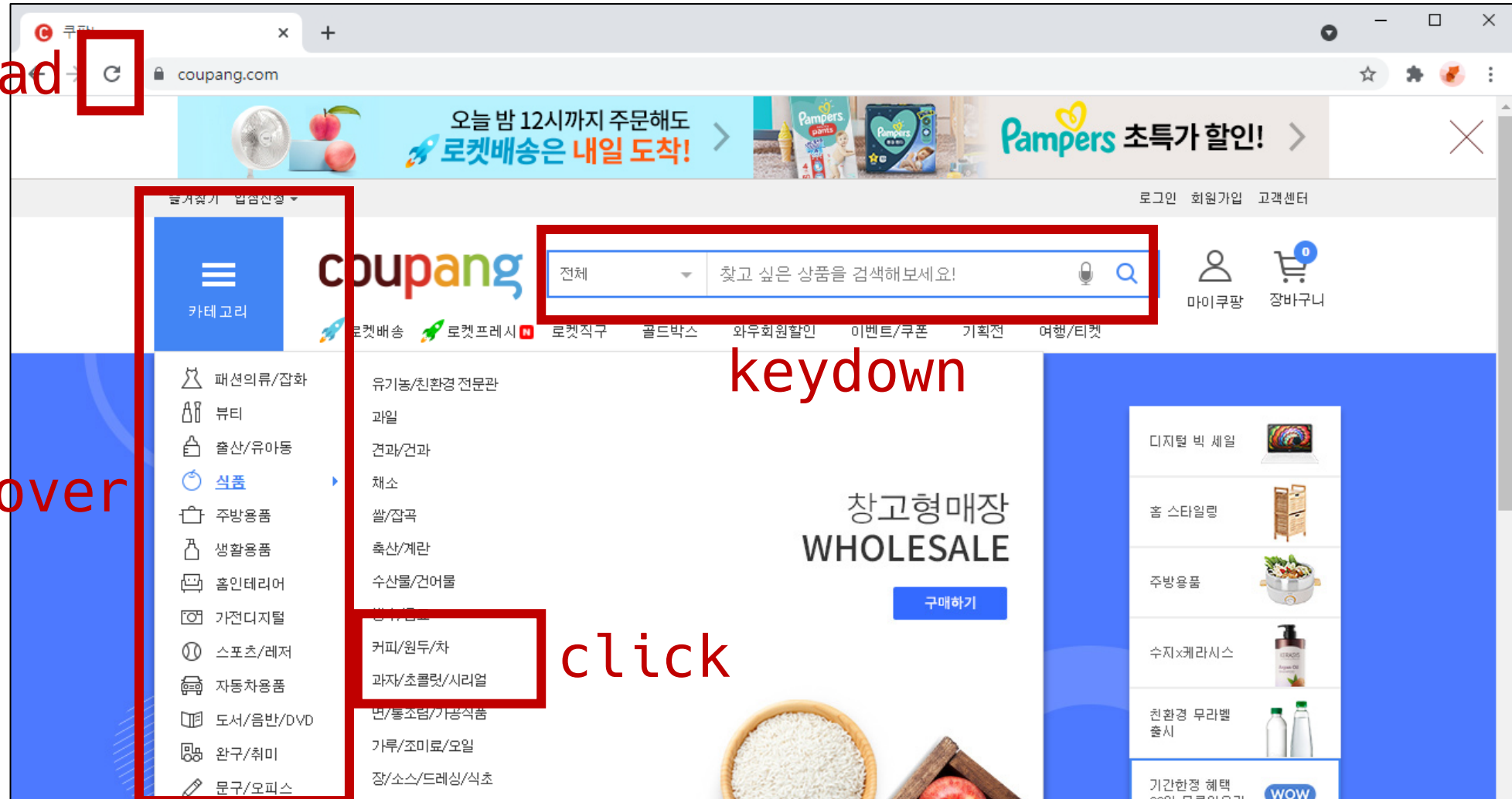


Event (이벤트) 개념

- 네트워크 활동이나 사용자와의 상호작용 같은 사건의 발생을 알리기 위한 객체
- 이벤트 발생
 - 마우스를 클릭하거나 키보드를 누르는 등 사용자 행동으로 발생할 수도 있음
 - 특정 메서드를 호출(`Element.click()`)하여 프로그래밍적으로도 만들어 낼 수 있음



Event의 역할 (1/3)

“~ 하면 ~ 한다.”

Event의 역할 (2/3)

“클릭하면, 경고창을 띄운다.”

Event의 역할 (3/3)

“특정 이벤트가 발생하면,
할 일(함수)을 등록한다.”

Event handler – addEventListener() (1/4)

- EventTarget **.addEventListener()**
 - 지정한 이벤트가 대상에 전달될 때마다 호출할 함수를 설정
 - 이벤트를 지원하는 모든 객체(Element, Document, Window 등)를 대상으로 지정 가능

Event handler – addEventListener() (2/4)

- target.**addEventListener(type, listener[, options])**
 - type
 - 반응 할 이벤트 유형 (대소문자 구분 문자열)
 - listener
 - 지정된 타입의 이벤트가 발생했을 때 알림을 받는 객체
EventListener 인터페이스 혹은 JS function 객체(콜백 함수)여야 함

Event handler – addEventListener() (3/4)

“대상에 특정 이벤트가 발생하면, 할 일을 등록하자”

Event handler – addEventListener() (4/4)

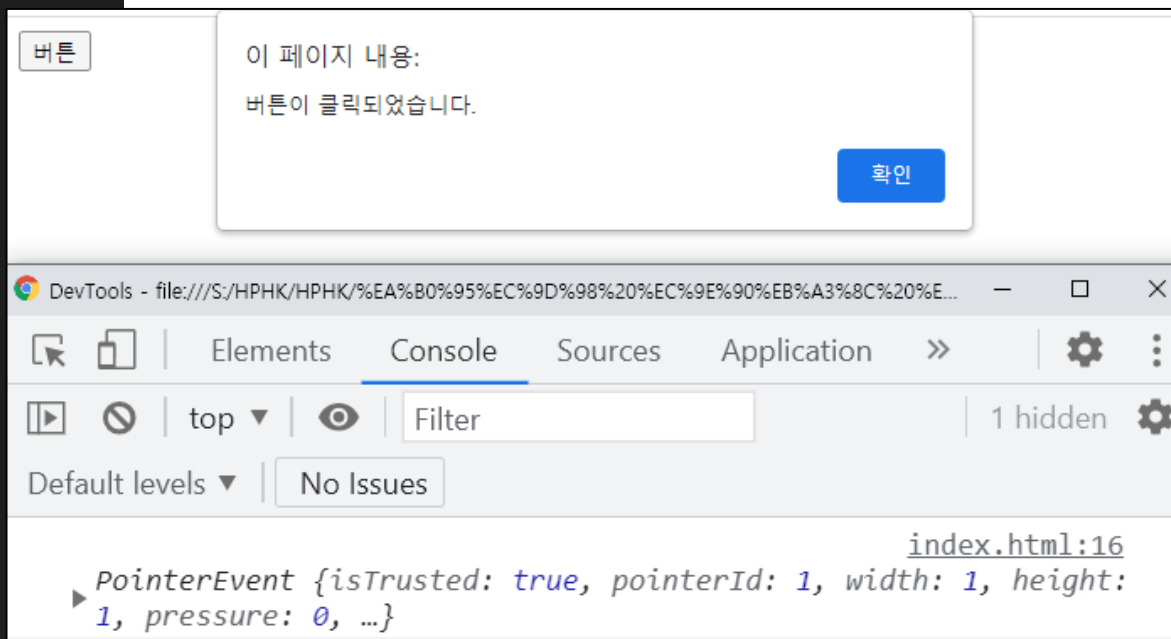
“대상^{대상}에 특정 이벤트^{특정 이벤트}가 발생하면, 할 일^{할 일}을 등록하자”

```
EventTarget.addEventListener(type, listener)
```

Event

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <button type="button">버튼</button>

  <script>
    const btn = document.querySelector('button')
    btn.addEventListener('click', function (event) {
      alert('버튼이 클릭되었습니다.')
      console.log(event)
    })
  </script>
</body>
</html>
```



addEventListener 실습 (1/4)

```
<button onclick="alertMessage( )">나를 눌러봐!</button>
```

```
const alertMessage = function () {  
  alert('메롱!!!')  
}
```

| | | |
|---|----------|---|
| 나를 눌러봐! | 나를 눌러봐!! | 이 페이지 내용: 메롱!!! <div>확인</div> |
| 내용을 입력하세요. <input type="text"/> | | |
| <h3>Change My Color</h3> | | |
| 원하는 색상을 영어로 입력하세요. <input type="text"/> | | |

addEventListener 실습 (2/4)

```
<button id="my-button">나를 눌러봐!!</button>
```

```
const myButton = document.querySelector('#my-button')  
myButton.addEventListener('click', alertMessage)
```

| | | |
|---|----------|---|
| 나를 눌러봐! | 나를 눌러봐!! | 이 페이지 내용: 메롱!!! <div>확인</div> |
| 내용을 입력하세요. <input type="text"/> | | |
| <h3>Change My Color</h3> | | |
| 원하는 색상을 영어로 입력하세요. <input type="text"/> | | |

addEventListener 실습 (3/4)

```
<p id="my-paragraph"></p>

<form action="#">
  <label for="my-text-input">내용을 입력하세요.</label>
  <input id="my-text-input" type="text">
</form>
```

```
const myTextInput = document.querySelector('#my-text-input')

myTextInput.addEventListener('input', function (event) {
  // console.log(event)
  const myPtag = document.querySelector('#my-paragraph')
  myPtag.innerText = event.target.value
})
```

나를 놀려봐! 나를 놀려봐!!

안녕

내용을 입력하세요.

Change My Color

원하는 색상을 영어로 입력하세요.

addEventListener 실습 (4/4)

```
<h2>Change My Color</h2>
<label for="change-color-input">원하는 색상을 영어로 입력하세요.</label>
<input id="change-color-input"></input>
<hr>
```

```
const colorInput = document.querySelector('#change-color-input')

const changeColor = function (event) {
  const h2Tag = document.querySelector('h2')
  h2Tag.style.color = event.target.value
}

colorInput.addEventListener('input', changeColor)
```

나를 눌러봐! 나를 눌러봐!!

안녕

내용을 입력하세요.

Change My Color

원하는 색상을 영어로 입력하세요.

Event 취소

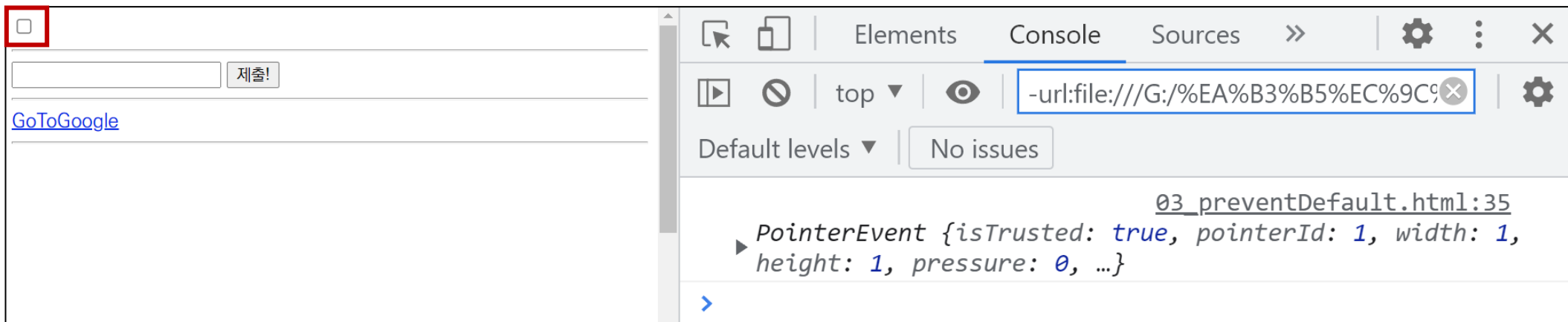
- `event.preventDefault()`
- 현재 이벤트의 기본 동작을 중단
- HTML 요소의 기본 동작을 작동하지 않게 막음
 - ex) a 태그의 기본 동작은 클릭 시 링크로 이동 / form 태그의 기본 동작은 form 데이터 전송
- 이벤트를 취소할 수 있는 경우, 이벤트의 전파를 막지 않고 그 이벤트를 취소

Event 취소 실습 (1/5)

```
<input type="checkbox" id="my-checkbox">
```

```
const checkBox = document.querySelector('#my-checkbox')

checkBox.addEventListener('click', function (event) {
  event.preventDefault()
  console.log(event)
})
```

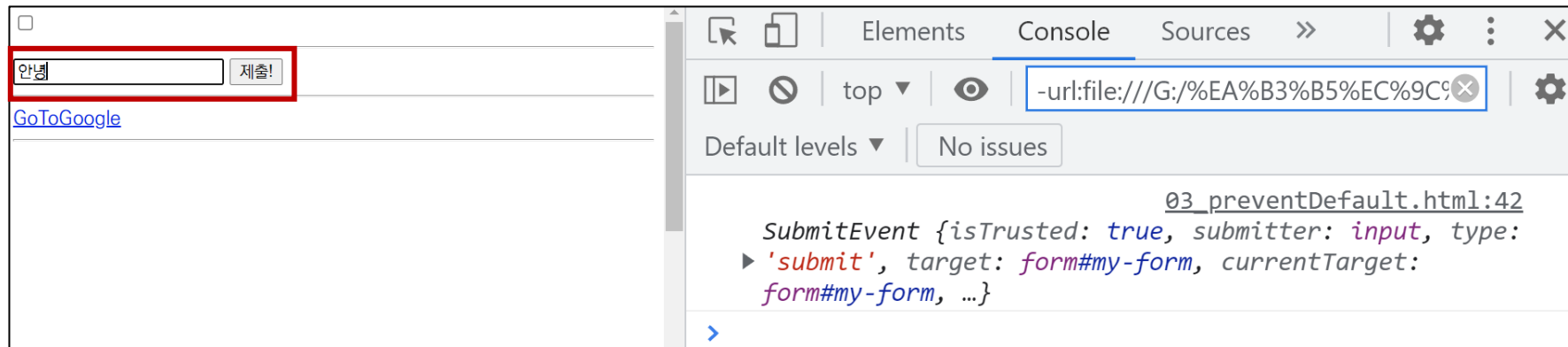


Event 취소 실습 (2/5)

```
<form action="/articles/" id="my-form">
  <input type="text">
  <input type="submit" value="제출!">
</form>
```

```
const formTag = document.querySelector('#my-form')

formTag.addEventListener('submit', function (event) {
  console.log(event)
  event.preventDefault()
  event.target.reset()
})
```



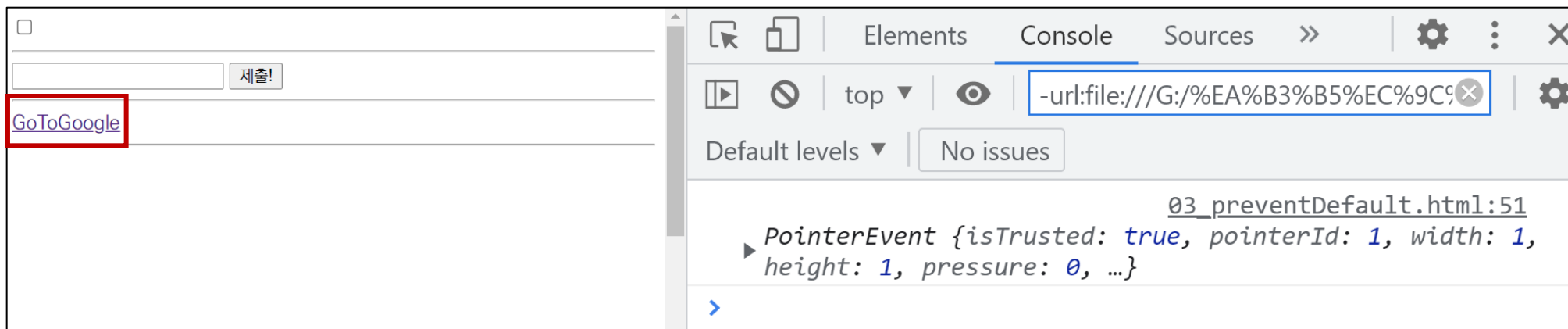
The screenshot shows a web browser window with a form containing a text input field with the text "안녕" and a submit button labeled "제출!". The form is highlighted with a red rectangular box. Below the form is a link "GoToGoogle". The browser's developer console is open, showing a log message for a submit event. The log message is: `SubmitEvent {isTrusted: true, submitter: input, type: 'submit', target: form#my-form, currentTarget: form#my-form, ...}`. The console also shows the file path `03_preventDefault.html:42` and the URL `-url:file:///G:/%EA%B3%B5%EC%9C%`.

Event 취소 실습 (4/5)

```
<a href="https://google.com/" target="_blank" id="my-link">GoToGoogle</a>
```

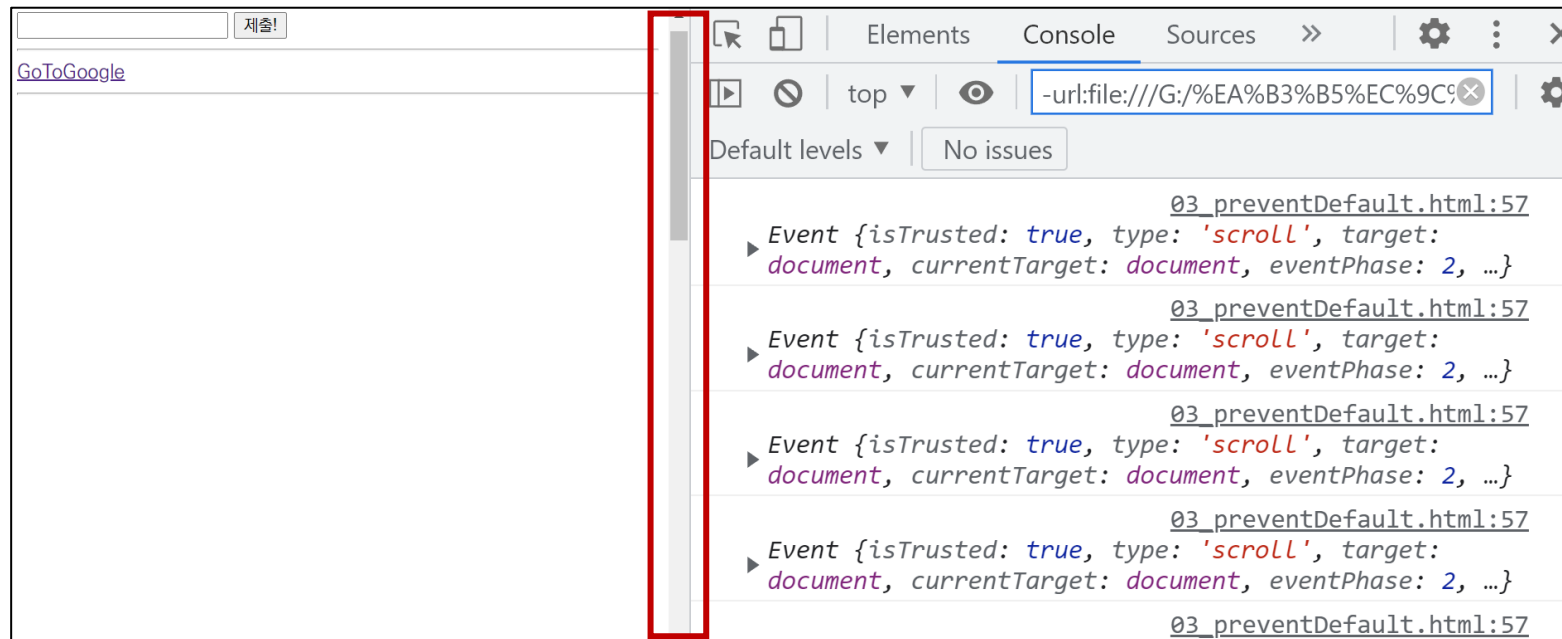
```
const aTag = document.querySelector('#my-link')

aTag.addEventListener('click', function (event) {
  console.log(event)
  event.preventDefault()
})
```



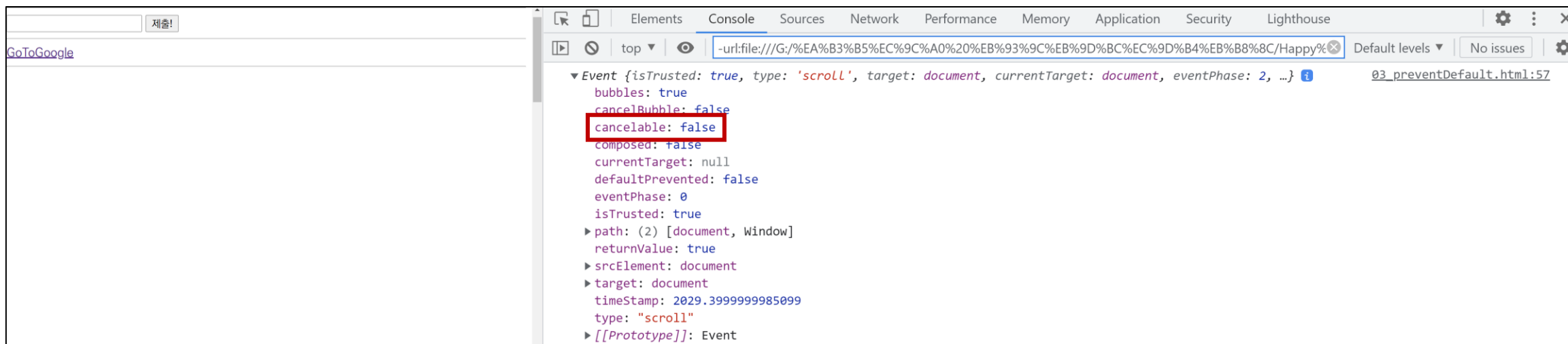
Event 취소 실습 (4/5)

```
document.addEventListener('scroll', function (event) {
  console.log(event)
  event.preventDefault()
})
```



Event 취소 실습 (5/5)

- 취소 할 수 없는 이벤트도 존재
 - 이벤트의 취소 가능 여부는 `event.cancelable`을 사용해 확인할 수 있음



Event 추가 학습

- 다양한 이벤트 유형 참고
- <https://developer.mozilla.org/en-US/docs/Web/Events>