

## 자바스크립트 form

자바스크립트에서 HTML 폼(form)을 다루는 것은 웹 개발의 중요한 부분입니다. HTML 폼은 사용자로부터 정보를 수집하고, 이를 서버에 제출하기 위한 방법을 제공합니다. 자바스크립트를 사용하여 폼 데이터를 조작하고 검증할 수 있으며, 사용자 경험을 개선하고 더 동적인 웹 페이지를 만들 수 있습니다.

### 자바스크립트 폼 프로퍼티 및 메서드

종류	이름	설명
프로퍼티	elements	폼 내의 모든 입력 필드(예: 텍스트 상자, 체크박스, 라디오 버튼 등)에 접근
프로퍼티	action	폼 데이터가 전송될 서버의 URL
프로퍼티	method	폼 데이터를 서버로 전송하는 방법(GET 또는 POST)
메서드	submit()	프로그래밍적으로 폼을 제출
메서드	reset()	폼의 모든 필드를 초기 상태로 재설정

### 폼 관련 이벤트

이벤트 카테고리	이벤트 이름	설명
Change & Input 이벤트	change	필드의 값이 변경되고, 포커스를 잃었을 때 발생
Change & Input 이벤트	input	필드에 입력이 이루어질 때마다 발생 (타이핑 시 발생)
Cut, Copy, Paste 이벤트	cut	사용자가 필드의 텍스트를 잘라내려고 할 때 발생
Cut, Copy, Paste 이벤트	copy	사용자가 필드의 텍스트를 복사하려고 할 때 발생
Cut, Copy, Paste 이벤트	paste	사용자가 클립보드의 내용을 필드에 붙여넣으려고 할 때 발생
Submit 이벤트	submit	폼이 제출될 때 발생 (데이터 검증 및 기본 동작 막기 가능)

### 폼 제출 방식

방식	설명
표준 제출	<code>&lt;form&gt;</code> 태그의 <code>action</code> 속성에 지정된 URL로 사용자 입력 데이터를 전송. 페이지 새로고침 발생
자바스크립트를 통한 제출	<code>submit</code> 이벤트 리스너를 사용하여 폼 제출을 가로채고, 자바스크립트로 데이터 처리. Ajax를 사용하여 서버와 통신 가능

## form 예제

- HTML 폼 생성  
텍스트 필드, 체크박스, 라디오 버튼, 버튼 등을 포함한 HTML 폼을 만듭니다.
- 자바스크립트 이벤트 리스너 추가  
폼 요소들의 change, input, cut, copy, paste, submit 이벤트에 대한 리스너를 추가합니다.
- 폼 프로퍼티 사용  
elements, action, method 등의 프로퍼티를 사용하여 폼 요소에 접근하고 정보를 표시합니다.
- 폼 메서드 사용  
submit()와 reset() 메서드를 사용하여 폼을 제출하거나 초기화합니다.
- 자바스크립트로 폼 제출 처리  
submit 이벤트를 가로채어 Ajax를 사용하여 서버와 통신하거나 폼 데이터를 처리합니다.

```
<!DOCTYPE html>
<html>
<head>
<title>폼 예제</title>
</head>
<body>

  <form id="myForm">
    <input type="text" name="textField" placeholder="Enter Text">
    <input type="checkbox" name="checkBox">
    <input type="radio" name="radioButton" value="Option1">Option 1
    <input type="radio" name="radioButton" value="Option2">Option 2
    <button type="button" id="submitForm">Submit</button>
    <button type="button" id="resetForm">Reset</button>
  </form>

  <script>
    document.addEventListener('DOMContentLoaded', function () {
      let form = document.getElementById('myForm');

      // 'change' 이벤트 핸들러: 폼 필드 값이 변경되고 포커스가 이동했을 때 실행됩니다.
      form.addEventListener('change', function(event) {
        console.log('Change event on:', event.target);
      });

      // 'input' 이벤트 핸들러: 사용자가 폼 필드에 데이터를 입력할 때마다 실행됩니다.
      form.addEventListener('input', function(event) {
        console.log('Input event on:', event.target);
      });

      // 'cut' 이벤트 핸들러: 사용자가 폼 필드의 텍스트를 잘라낼 때 실행됩니다.
      document.addEventListener('cut', function(event) {
        console.log('Cut event on:', event.target);
      });

      // 'copy' 이벤트 핸들러: 사용자가 폼 필드의 텍스트를 복사할 때 실행됩니다.
```

```

document.addEventListener('copy', function(event) {
    console.log('Copy event on:', event.target);
});

// 'paste' 이벤트 핸들러: 사용자가 폼 필드에 텍스트를 붙여넣을 때 실행됩니다.
document.addEventListener('paste', function(event) {
    console.log('Paste event on:', event.target);
});

// 'submit' 이벤트 핸들러: 폼 제출 버튼이 클릭되었을 때 실행됩니다. 기본 동작은 방지됩니
다.
form.addEventListener('submit', function(event) {
    event.preventDefault();
    console.log('Form Submitted!', 'Form Data:', new FormData(form));
    // 여기서 추가적인 처리를 할 수 있습니다, 예를 들어 AJAX 요청을 보내기
});

// 폼의 프로퍼티 출력
console.log('Form elements:', form.elements); // 폼 내의 모든 요소
console.log('Form action:', form.action); // 폼의 'action' 속성 값
console.log('Form method:', form.method); // 폼의 'method' 속성 값

// 'submitForm' 버튼 클릭 이벤트 핸들러: AJAX 요청을 전송합니다.
document.getElementById('submitForm').addEventListener('click',
function() {
    let formData = new FormData(form);
    let jsonObject = {};

    // 폼 데이터를 JSON 객체로 변환
    for (const [key, value] of formData.entries()) {
        jsonObject[key] = value;
    }

    // AJAX 요청 전송
    fetch('https://jsonplaceholder.typicode.com/posts', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(jsonObject)
    })
    .then(response => response.json())
    .then(data => {
        console.log('Success:', data);
    })
    .catch((error) => {
        console.error('Error:', error);
    });
});

// 'resetForm' 버튼 클릭 이벤트 핸들러: 폼을 초기 상태로 재설정합니다.
document.getElementById('resetForm').addEventListener('click',
function() {
    form.reset();
});

```

```
});  
</script>  
</body>  
</html>
```