

이벤트

# 이벤트, 이벤트 핸들러, 이벤트 리스너

- 이벤트

사용자의 동작이나 브라우저의 상태 변화와 관련하여 발생하는 사건을 의미

- 이벤트 핸들러

특정 이벤트가 발생했을 때 실행되는 함수

- 이벤트 리스너

JavaScript에서 지원하는 `addEventListener()` 메소드를 의미함

# 이벤트의 종류

- 이벤트의 종류

이벤트	설명
click	사용자가 요소를 클릭할 때 발생하는 이벤트
mouse	마우스와 관련된 동작에 대한 이벤트
keyboard	키보드 입력에 대한 이벤트
focus	포커스를 받거나 잃을 때 발생하는 이벤트
submit	폼 요소가 제출될 때 발생하는 이벤트
scroll	페이지가 스크롤될 때 발생하는 이벤트
load	웹 페이지가 완전히 로드 되었을 때 발생하는 이벤트

# 이벤트 핸들러 예시

## • 이벤트 핸들러 예시

```
<button id="myButton" onclick="alert('이벤트 핸들러')">클릭해주세요</button>  
<p id="output"></p>
```

```
<body>  
  <button id="myButton">클릭해주세요</button>  
  <p id="output"></p>  
  <script>  
    const myButton = document.getElementById('myButton');  
    myButton.onclick = function() {  
      alert("이벤트핸들러");  
    }  
  </script>  
</body>
```

# 이벤트 리스너 예시

## • 이벤트 리스너 예시

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myButton = document.getElementById('myButton');

    myButton.addEventListener('click', function(event) {
      alert("이벤트 리스너");
    });
  </script>
</body>
```

이벤트  
(click)

# 클릭 이벤트 - 1

- 클릭 이벤트

사용자가 요소를 클릭할 때 발생하는 이벤트

## 클릭 이벤트 - 2

- 클릭 이벤트 예시(이벤트 리스너 - 콜백 함수)

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
</body>
</html>
```

```
<script>
  const myButton = document.getElementById('myButton');
  const output = document.getElementById('output');

  myButton.addEventListener('click', function(event) {
    myButton.textContent = '클릭되었습니다!';
    output.textContent = '버튼이 클릭되었습니다.';
  });
```

함수명



## 클릭 이벤트 - 3

- 클릭 이벤트 예시(이벤트 리스너 - 일반 함수)

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myButton = document.getElementById('myButton');

    myButton.addEventListener('click', btnClick);

    function btnClick() {
      alert("이런식으로도 활용 가능합니다.");
    }
  </script>
</body>
... ..
```

# 클릭 이벤트 활용하기

## • Q1. 버튼 클릭할 때 마다 버튼의 이름과 내용을 변경 하세요.

1. 처음 버튼의 이름 : "버튼을 클릭 해주세요."
2. 최초 버튼 클릭 시 아래와 같이 변경됨
  - 버튼명(myButton) : 클릭 되었습니다!
  - 버튼 내용(output) : 버튼이 클릭 되었습니다.
3. 이후 버튼 클릭 시 아래와 같이 변경됨
  - 버튼명(myButton) : 취소 되었습니다!
  - 버튼 내용(output) : 버튼이 취소 되었습니다.
4. 이후 클릭과 취소가 번갈아 가며 변경

이벤트  
(mouse)

# 마우스 이벤트 - 1

## • 마우스 이벤트

마우스와 관련된 동작에 대한 이벤트로써 여러가지가 있지만, 클릭을 제외한 3개만 실습

1. click  
→ 대표적으로 사용이 많이 되어 클릭 이벤트를 따로 빼놓았지만 크게 보았을 때는 마우스 이벤트에 속함
2. dblclick
3. mouseover
4. contextmenu

## 마우스 이벤트 - 2

- **dblclick**

마우스가 더블 클릭 됐을 때 발생하는 이벤트

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myButton = document.getElementById('myButton');

    myButton.addEventListener('dblclick', function(event) {
      output.textContent = '마우스가 더블클릭을 했습니다.';
    });
  </script>
</body>
</html>
```

## 마우스 이벤트 - 3

- mouseover

마우스가 요소 위로 이동했을 때 발생하는 이벤트

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myButton = document.getElementById('myButton');

    myButton.addEventListener('mouseover', function(event) {
      output.textContent = '마우스가 버튼 위로 이동했습니다.';
    });
  </script>
</body>
</html>
```

## 마우스 이벤트 - 4

- contextmenu

마우스가 우클릭 되었을 때 발생하는 이벤트

\* `event.preventDefault();` 를 사용해 보세요.

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myButton = document.getElementById('myButton');

    myButton.addEventListener('contextmenu', function(event) {
      output.textContent = '마우스 오른쪽 버튼이 클릭되었습니다.';
      // 기본 컨텍스트 메뉴 표시 방지
      // event.preventDefault();
    });
  </script>
</body>
```

이벤트  
(keyboard)



# 키보드 이벤트 - 1

## • 키보드 이벤트

키보드와 관련된 이벤트

\* 회원 가입에서 정규식을 확인할 때 주로 사용됨

이벤트	설명
keydown	키보드의 키가 눌렸을 때 발생하는 이벤트
keypress	키보드의 키가 눌렸을 때 발생하는 이벤트 (한글 등 일부 문자 인식 불가) * 최근 웹 표준에서는 사용하지 않는 것을 권고
keyup	키보드의 키가 눌려진 상태에서 손을 뗄 때 발생하는 이벤트

## 키보드 이벤트 - 2

- keydown

키가 눌렸을 때 발생

\* event.keyCode : 눌린 키보드의 코드를 알려줌

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <hr>
  <input id="myInput" maxlength="5"></input>
  <script>
    const myInput = document.getElementById('myInput');

    myInput.addEventListener(keydown, function(event) {
      output.textContent = "키가 눌렸습니다. 눌린 키보드 : " + event.keyCode;
    });
  </script>
</body>
```

## 키보드 이벤트 - 3

- keyup

키가 눌렸을 때 발생

\* event.keyCode : 눌린 키보드의 코드를 알려줌

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <hr>
  <input id="myInput" maxlength="5"></input>
  <script>
    const myInput = document.getElementById('myInput');

    myInput.addEventListener('keyup', function(event) {
      output.textContent = "키가 눌렸습니다. 눌린 키보드 : " + event.keyCode;
    });
  </script>
</body>
```

# keydown, keyup 확인해보기

## keydown, keyup 확인해보기

keydown과 keyup 이벤트가 발생 하는걸  
오른쪽 코드를 활용하여 확인해보세요

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <hr>
  <input id="myInput" maxlength="5"></input>
  <script>
    const myInput = document.getElementById('myInput');
    let keyDownTime;

    myInput.addEventListener('keydown', function(event) {
      // 키를 처음 누른 순간 기록
      if (!keyDownTime) {
        keyDownTime = new Date();
        output.textContent = "키를 누른 순간:" + keyDownTime;
      }
    });

    myInput.addEventListener('keyup', function(event) {
      // 키를 놓은 순간과 키 다운 시간 간의 차이 계산
      const keyUpTime = new Date();
      const duration = keyUpTime - keyDownTime;

      output.textContent = "키를 누르고 있던 시간(ms):" + duration;

      // 다음 키 입력을 위해 초기화
      keyDownTime = null;
    });
  </script>
</body>
</html>
```

# keyup 이벤트 사용 예시 - 1

## • 비밀번호 정규표현식을 검증하는 이벤트

1. html 태그 속성에서 onkeyup으로도 keyup 이벤트를 발생시킬 수 있음
2. test 함수 : 정규 표현식을 사용하여 문자열과 일치 하는지의 여부를 확인하는 내장 함수

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <hr>
  <input id="myInput" onkeyup="validatePassword(this.value)"></input>
  <script>
    let isValid;
    const output = document.getElementById("output");

    function validatePassword(password) {
      // 비밀번호 정규식 패턴
      // 소문자, 대문자, 숫자를 포함한 6자리 이상 12자리 이하
      const pattern = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{6,12}$/;

      // 비밀번호와 패턴을 비교하여 유효성을 검증
      isValid = pattern.test(password);

      // 출력 업데이트
      output.textContent = "입력한 비밀번호가 유효한가요? " + isValid;
    }
  </script>
</body>
</html>
```

# 자바스크립트 트러블슈팅 예제

- Q1. 아래의 스크립트가 동작하지 않는 이유를 찾아보세요.

아래의 스크립트가 동작하지 않는 이유를 찾아보세요.

```
<body>
  <button id="myButton">클릭해주세요</button>
  <p id="output"></p>
  <script>
    const myInput = document.getElementById('myInput');

    myInput.addEventListener('keydown', function(event) {
      output.textContent = "키가 눌렸습니다. \n 눌린 키보드 : " + event.keyCode;
    });
  </script>
  <hr>
  <input id="myInput"></input>
</body>
</html>
```