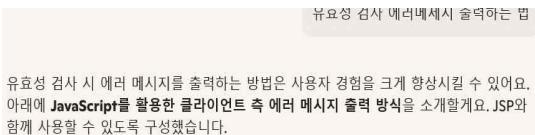


로그인 페이지, 회원가입 페이지, 프로필 페이지 유효성 검사와 에러메세지 출력을 위해 이벤트 리스너 사용, error.textContent로 에러메세지 출력



- DOMContentLoaded: 문서가 완전히 로드된 후 실행
- form.addEventListener("submit", ...): 폼 제출 시 유효성 검사
- event.preventDefault(): 유효하지 않으면 서버로 전송 막기
- 태그에 에러 메시지를 출력하여 사용자에게 즉시 피드백 제공

```
<script>
document.addEventListener("DOMContentLoaded", function() {
  const form = document.getElementById("loginForm");

  form.addEventListener("submit", function(event) {
    const username = document.getElementById("username").value.trim();
    const password = document.getElementById("password").value.trim();
    const usernameError = document.getElementById("usernameError");
    const passwordError = document.getElementById("passwordError");

    // 초기화
    usernameError.textContent = "";
    passwordError.textContent = "";

    let isValid = true;

    if (username === "") {
      usernameError.textContent = "아이디를 입력하세요.";
      isValid = false;
    }

    if (password === "") {
      passwordError.textContent = "비밀번호를 입력하세요.";
      isValid = false;
    }

    if (!isValid) {
      event.preventDefault(); // 폼 제출 막기
    }
  });
});</script>
```

에러메세지가 출력될 때 박스 밑으로 내려가지 않게 line-height를 사용하여 에러메세지 공간을 확보

line-height를 사용하면 텍스트의 줄 간 간격을 조절할 수 있으며, 이를 활용해 에러 메시지 공간을 고정할 수도 있어요. 특히 메시지가 짧거나 없을 때도 일정한 높이를 유지하고 싶을 때 유용합니다.

에러 메시지 공간 확보: line-height 활용 예제

Html

```
<span id="usernameError" class="error">아이디를 입력하세요.</span>
```

Css

```
.error {
  color: red;
  font-size: 0.85em;
  line-height: 20px; /* 줄 간 간격을 고정 */
  display: block; /* 줄바꿈 처리 */
  min-height: 20px; /* 메시지가 없을 때도 공간 유지 */
}
```

```
.error-message{
  color : red;
  text-align : left;
  font-size : 0.8em;
  height: 18px;
  line-height: 1.2;
}
```

한글과 영어를 포함하여 유효성 검사, 이메일 유효성 검사

한글과 영문 유효성 검사

한글과 영문 유효성 검사는 사용자가 입력한 값이 특정 언어 문자만 포함하는지 확인하는 과정입니다. 로그인 폼에서 아이디나 이름 입력 시, 한글 또는 영문만 허용하고 싶을 때 자주 사용됩니다.

▣ 정규표현식 요약

패턴	설명
/^[가-힣]+\$/	한글만 허용
/^[a-zA-Z]+\$/	영문만 허용
/^[가-힣a-zA-Z]+\$/	한글 또는 영문 허용
/^[a-zA-Z0-9]+\$/	영문 + 숫자 허용
/^[가-힣a-zA-Z0-9]+\$/	한글 + 영문 + 숫자 허용

원하시면 비밀번호에 대한 복잡도 검사(특수문자 포함, 최소 길이 등)도 추가해드릴 수 있어요. 어떤 조건을 적용하고 싶으세요?

```
}  
else if(!(/(?=.*[a-zA-Z])(?=.*[0-9])^([a-zA-Z0-9]+$).test(passwdtrim)){  
  passwdError.textContent = "비밀번호는 영문과 숫자를 포함해야 합니다.";  
}  
else if (passwdtrim.length < 8 || passwdtrim.length > 12){  
  passwdError.textContent = "비밀번호는 8자 이상 12자 이하여야 합니다.";  
  valid = false;  
}
```

```
Javascript
```

```
const regex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
```

```
else if(!/^[_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/.test(emailtrim)){
    emailError.textContent = "형식이 올바르지 않습니다.";
    valid = false;
}
```

- `^[\s@]+` → @ 앞에 공백 없이 문자
- `@[\s@]+` → @ 뒤에 공백 없이 문자
- `\.[\s@]+$` → 마지막에 도메인 (.com, .net 등)

로그아웃 버튼에 가져다 대면 마우스 포인터가 손가락 모양으로 변화

마우스 포인터 손가락 모양으로 변화

마우스 포인터를 손가락 모양으로 변화시키려면 CSS에서 `cursor: pointer;` 속성을 사용하면 됩니다. 이 속성은 사용자가 해당 요소 위에 마우스를 올렸을 때 클릭 가능한 상태임을 시각적으로 표시해줍니다.

✓ 기본 사용 예시

```
Html
```

```
<button style="cursor: pointer;">클릭하세요</button>
```

또는 CSS로 지정할 수도 있습니다.

폼 안에 요소들 세로 정렬 후 가운데 정렬

폼 안에 요소들 세로로 정렬하는 법

폼 안의 요소들을 세로로 정렬하려면 HTML과 CSS를 활용해 레이아웃을 수직 방향으로 설정하면 됩니다. 아래에 가장 많이 쓰이는 방법들을 정리해드릴게요.

✓ 방법 1: Flexbox로 세로 정렬

```
Html
```

```
<form style="display: flex; flex-direction: column; gap: 10px;">
    <input type="text" placeholder="이름">
    <input type="email" placeholder="이메일">
    <button type="submit">제출</button>
</form>
```

- `display: flex` → 요소들을 유연하게 배치
- `flex-direction: column` → 세로 방향으로 정렬
- `gap` → 요소 사이 간격 설정

✓ 가운데 정렬까지 하고 싶다면?

```
Css
```

```
form {
    display: flex;
    flex-direction: column;
    align-items: center; /* 가로 가운데 정렬 */
    justify-content: center; /* 세로 가운데 정렬 */
    height: 100vh;
}
```

원하는 정렬 방식이 "세로로 나열"인지, "세로 가운데 정렬"인지에 따라 조금씩 다르니, 구체적인 레이아웃이 있다면 더 맞춤형으로 도와드릴 수 있어요!

```
btn {
    width: 180px;
    padding: 10px 20px;
    margin-top: 10px;
    background-color: black;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}
```

border-radius : 20px;
font-size : 16px;
display: flex;
flex-direction: column;
align-items: center;
background-color : white;
box-shadow: 0 0 10px rgba(0,0,0,0.1);

```
.btnx {
    gap: 20px;
    display: flex;
    justify-content: center;
    flex-direction: row;
    margin-bottom: 100px;
}
```