

231121(비번유효성검사,모달창,투두)

자바스프링

2023/11/21 17:48

<http://blog.naver.com/sophia2164/223270934477>

비밀번호 유효성 검사

html Copy code

비밀번호 유효성 검사

asef@ef

- ☒ 소문자
- ☐ 대문자
- ☐ 숫자
- ☒ 특수 문자 (!@#\$%^&*)
- ☐ 8자 이상

유효성 검사를 충족하지 못합니다.

```

Password validation > Password.html > html > body > script
47 <!-- 비밀번호 유효성 검사 컨테이너 -->
48 <div class="title"><span>비밀번호 유효성 검사</span></div>
49 <div class="password">
50 <!-- 비밀번호 입력란 -->
51 <input type="text" class="input-password">
52 </div>
53 <ul>
54 <!-- 각 비밀번호 유효성 항목 -->
55 <li class="lowercase"><div class="checkbox"></div><span class="label">소문자</span> </li>
56 <li class="uppercase"><div class="checkbox"></div><span class="label">대문자</span> </li>
57 <li class="number"><div class="checkbox"></div><span class="label">숫자</span> </li>
58 <li class="special"><div class="checkbox"></div><span class="label">특수 문자 (1@#%&* )</span> </li>
59 <li class="character-length"><div class="checkbox"></div><span class="label">8자 이상</span> </li>
60 </ul>
61 <div><p class="passok">비밀번호 사용이 가능합니다</p></div>
62 <div><p class="passno">비밀번호 조건이 일치하지 않습니다</p></div>
63 </div>
64 <script>
65 // JavaScript 비밀번호 유효성 검사 로직
66 const inputElem = document.querySelector('input'); // 입력란 엘리먼트 가져오기
67 const lowercaseElem = document.querySelector('.lowercase'); // 소문자 요소 가져오기
68 const uppercaseElem = document.querySelector('.uppercase'); // 대문자 요소 가져오기
69 const numberElem = document.querySelector('.number'); // 숫자 요소 가져오기
70 const specialElem = document.querySelector('.special'); // 특수문자 요소 가져오기
71 const characterLengthElem = document.querySelector('.character-length'); // 길이 요소 가져오기
72
73
74 //정규 표현식
75 //값을 통해 출력할 것? true와 false가 나온다
76 const isValidLowercase = (password) => {
77   return password.search(/[a-z]/g) >= 0; // 소문자 포함 여부 확인
78 }
79 const isValidUppercase = (password) => {
80   return password.search(/[A-Z]/g) >= 0; // 대문자 포함 여부 확인
81 }
82 //g는 특수문자 구분하라는 것
83 const isValidNumber = (password) => {
84   return password.search(/[0-9]/g) >= 0; // 숫자 포함 여부 확인
85 }
86
87 //g는 특수문자 구분하지 말라는 것(전역변수)
88 const isValidSpecial = (password) => {
89   return password.search(/[~!@#$%^&*()_+<?>:{}]/g) >= 0; // 특수문자 포함 여부 확인
90 }
91 const isValidCharacterLength = (password) => {
92   return password.length >= 8 // 최소 8자 이상 여부 확인
93 }
94
95 inputElem.addEventListener('input', (e) => {
96   const password = e.target.value;
97
98   isValidLowercase(password) ? lowercaseElem.classList.add('active') : lowercaseElem.classList.remove('active'); // 소문자 유효성 표시 여부 업데이트
99   isValidUppercase(password) ? uppercaseElem.classList.add('active') : uppercaseElem.classList.remove('active'); // 대문자 유효성 표시 여부 업데이트
100   isValidNumber(password) ? numberElem.classList.add('active') : numberElem.classList.remove('active'); // 숫자 유효성 표시 여부 업데이트
101   isValidSpecial(password) ? specialElem.classList.add('active') : specialElem.classList.remove('active'); // 특수문자 유효성 표시 여부 업데이트
102   isValidCharacterLength(password) ? characterLengthElem.classList.add('active') : characterLengthElem.classList.remove('active'); // 길이 유효성 표시 여부 업데이트
103
104   const ok = document.querySelector('.passok');
105   const no = document.querySelector('.passno');
106   if(isValidLowercase(password) && isValidUppercase(password) && isValidNumber(password) && isValidSpecial(password) && isValidCharacterLength(password)) {
107     // document.querySelector('.passok').innerHTML;
108     ok.style.display="block"
109     no.style.display="none"
110   }
111   else{
112     // document.querySelector('.passno').innerHTML;
113     ok.style.display="none"
114     no.style.display="block"
115   }
116
117 })
118 </script>
119
120 </body>
121 </html>

```

```
Go Run Terminal Help
Password validation > Password validation.html > html > body > script
39     }
40     </style>
41 </head>
42 <body>
43 <div class="signup-container">
44 <!-- 비밀번호 유효성 검사 컨테이너 -->
45 <div class="title"><span>비밀번호 유효성 검사</span></div>
46 <div class="password">
47 <!-- 비밀번호 입력란 -->
48 <input type="text" class="input-password">
49 </div>
50 <ul>
51 <!-- 각 비밀번호 유효성 항목 -->
52 <li class="lowercase"><div class="checkbox"></div><span class="label">소문자</span> </li>
53 <li class="uppercase"><div class="checkbox"></div><span class="label">대문자</span> </li>
54 <li class="number"><div class="checkbox"></div><span class="label">숫자</span> </li>
55 <li class="special"><div class="checkbox"></div><span class="label">특수 문자 (!@#$%^&*)</span> </li>
56 <li class="character-length"><div class="checkbox"></div><span class="label">8자 이상</span> </li>
57 <span id="checkPass"></span>
58 </ul>
59 </div>
60 <script>
61 // JavaScript 비밀번호 유효성 검사 로직
62 const inputElem = document.querySelector('input'); // 입력한 엘리먼트 가져오기
63 const lowercaseElem = document.querySelector('.lowercase'); // 소문자 요소 가져오기
64 const uppercaseElem = document.querySelector('.uppercase'); // 대문자 요소 가져오기
65 const numberElem = document.querySelector('.number'); // 숫자 요소 가져오기
66 const specialElem = document.querySelector('.special'); // 특수문자 요소 가져오기
67 const characterLengthElem = document.querySelector('.character-length'); // 길이 요소 가져오기
68 const checkPass = document.getElementById('checkPass'); // id 불러올 때는 , 안붙여도 된
69                                     Elements (class)나 다른것을 불러올 때에는 , 붙여야 함
70
71 const isValidLowercase = (password) => {
72     return password.search(/[a-z]/g) >= 0; // 소문자 포함 여부 확인
73 }
74 const isValidUppercase = (password) => {
75     return password.search(/[A-Z]/g) >= 0; // 대문자 포함 여부 확인
76 }
77 const isValidNumber = (password) => {
78     return password.search(/[0-9]/g) >= 0; // 숫자 포함 여부 확인
79 }
80 const isValidSpecial = (password) => {
81     return password.search(/[!@#$%^&*()_+|<?>:;{}]/gi) >= 0; // 특수문자 포함 여부 확인
82 }
83 const isValidCharacterLength = (password) => {
84     return password.length >= 8; // 최소 8자 이상 여부 확인
85 }
86
87 // 비밀번호 입력 시 유효성 검사 상태 업데이트
88 inputElem.addEventListener('input', (e) => {
89     const password = e.target.value;
90     if(isValidCharacterLength(password) && isValidLowercase(password) && isValidNumber(password) && isValidSpecial(password) && isValidUppercase(password)){
91         checkPass.innerHTML="유효성 검사를 충족합니다."
92     } else {
93         checkPass.innerHTML = "유효성 검사를 충족하지 못합니다.";
94     }
95
96     isValidLowercase(password) ? lowercaseElem.classList.add('active') : lowercaseElem.classList.remove('active'); // 소문자 유효성 표시 여부 업데이트
97     isValidUppercase(password) ? uppercaseElem.classList.add('active') : uppercaseElem.classList.remove('active'); // 대문자 유효성 표시 여부 업데이트
98     isValidNumber(password) ? numberElem.classList.add('active') : numberElem.classList.remove('active'); // 숫자 유효성 표시 여부 업데이트
99     isValidSpecial(password) ? specialElem.classList.add('active') : specialElem.classList.remove('active'); // 특수문자 유효성 표시 여부 업데이트
100     isValidCharacterLength(password) ? characterLengthElem.classList.add('active') : characterLengthElem.classList.remove('active'); // 길이 유효성 표시 여부 업데이트
101 })
102 </script>
103 </body>
104 </html>
```

```

// 전체 todos의 check 여부 (isCompleted)
const checkIsAllCompleted = () => {
  if (getAllTodos().length === getCompletedTodos().length) {
    setIsAllCompleted(true);
    completeAllBtnElem.classList.add('checked');
  } else {
    setIsAllCompleted(false);
    completeAllBtnElem.classList.remove('checked');
  }
}

const onClickCompleteAll = () => {
  if (!getAllTodos().length) return; // todos배열의 길이가 0이면 return; 총배열개수가 몇인지? todos는할일의목록
  // !가 없으면 >0 0보다 크다(존재). 그런데 !를 붙여서 부정(0이면)

  if (isAllCompleted) incompleteAll(); // isAllCompleted가 true이면 todos를 전체 미완료 처리
  else completeAll(); // isAllCompleted가 false이면 todos를 전체 완료 처리
  setIsAllCompleted(!isAllCompleted); // isAllCompleted 토글처리 된다
  paintTodos(); // 새로운 todos를 렌더링
  setLeftItems()
}

const appendTodos = (text) => {
  const newId = id + 1; // 기존에 i++ 로 작성했던 부분을 setId()를 통해 id값을 갱신하였다.
  setId(newId)
  const newTodos = getAllTodos().concat({ id: newId, isCompleted: false, content: text })
  // const newTodos = [...getAllTodos(), {id: newId, isCompleted: false, content: text }]
  setTodos(newTodos)
  setLeftItems()
  checkIsAllCompleted();
  paintTodos();
}

```

```

<script>
// JavaScript 비밀번호 유효성 검사 로직
const inputElem = document.querySelector('input'); // 입력란 엘리먼트 가져오기
const lowercaseElem = document.querySelector('.lowercase'); // 소문자 요소 가져오기
const uppercaseElem = document.querySelector('.uppercase'); // 대문자 요소 가져오기
const numberElem = document.querySelector('.number'); // 숫자 요소 가져오기
const specialElem = document.querySelector('.special'); // 특수문자 요소 가져오기
const characterLengthElem = document.querySelector('.character-length'); // 길이 요소 가져오기

//정규 표현식
//값을 뭘로 출력할 것? true와 false가 나온다
const isValidLowercase = (password) => {
  return password.search(/[a-z]/g) >= 0;
  // 입력값의 전체 주소에서 소문자가 포함된 주소값을 출력한다. (0번째~ n번째까지)소문자 포함 여부 확인 ()
}
const isValidUppercase = (password) => {
  return password.search(/[A-Z]/g) >= 0; // 대문자 포함 여부 확인
}
//g는 특수문자 구분하라는 것
const isValidNumber = (password) => {
  return password.search(/[0-9]/g) >= 0; // 숫자 포함 여부 확인
}
//gi는 특수문자 구분하지 말라는 것(전역변수)
const isValidSpecial = (password) => {
  return password.search(/[~!@#$%^&*()_+|<>?:{}~]/gi) >= 0; // 특수문자 포함 여부 확인
}
const isValidCharacterLength = (password) => {
  return password.length >= 8 // 최소 8자 이상 여부 확인
}

inputElem.addEventListener('input', (e) => {
  // 이벤트 인풋 일어날때, (인력값, (e)) 이값을 뒤에 중괄호로 집어넣을것이다.
  const password = e.target.value;

  isValidLowercase(password) ? lowercaseElem.classList.add('active') : lowercaseElem.classList.remove('active'); // 소문자 유효성 표시 여
  isValidUppercase(password) ? uppercaseElem.classList.add('active') : uppercaseElem.classList.remove('active'); // 대문자 유효성 표시 여
  isValidNumber(password) ? numberElem.classList.add('active') : numberElem.classList.remove('active'); // 숫자 유효성 표시 여부 업데이트
  isValidSpecial(password) ? specialElem.classList.add('active') : specialElem.classList.remove('active'); // 특수문자 유효성 표시 여부 안
  isValidCharacterLength(password) ? characterLengthElem.classList.add('active') : characterLengthElem.classList.remove('active'); // 길

  const ok = document.querySelector('.passok');
  const no = document.querySelector('.passno');

```

모달창 열고 닫기

Open Modal

Open Modal

This is Modal

닫기

Password.html

index.html X

Event.stopPropagation() modal > index.html > html > body.container > script

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Event Propagation</title>
6      <link rel="stylesheet" href="style.css">
7
8  </head>
9
10 <body class="container" >
11 <div class="open-modal-btn" onclick="openModal();">Open Modal</div>
12
13
14 <script>
15     // HTML 문서에서 '.container' 클래스를 가진 요소를 찾아 bodyElem에 할당
16     const bodyElem = document.querySelector('.container');
17
18     // 모달 닫기 함수
19     const closeModal = (event) => {
20         // 모달 래퍼 엘리먼트를 찾아 제거
21         const modalWrapperElem = document.querySelector('.modal-wrapper');
22         bodyElem.removeChild(modalWrapperElem);
23     }
24
25     // 이벤트 전파 중지 함수
26     const stopPropagation = (e) => {
27         e.stopPropagation();
28     }
29
30     // 모달 열기 함수
31     const openModal = () => {
32         // 모달 래퍼 엘리먼트 생성
33         const modalWrapperElem = document.createElement('div');
34         // 모달 래퍼에 클릭 이벤트 리스너 추가하여 모달 닫기 함수 호출
35         modalWrapperElem.addEventListener("click", closeModal);
36         modalWrapperElem.classList.add('modal-wrapper');
37
38         // 모달 엘리먼트 생성
39         const modalElem = document.createElement('div');
40         modalElem.classList.add('modal');
41         // 모달에 클릭 이벤트 리스너 추가하여 이벤트 전파 중지 함수 호출
42         modalElem.addEventListener('click', stopPropagation);
43
44         // 모달 콘텐츠 엘리먼트 생성
45         const modalContentElem = document.createElement('div');
46         modalContentElem.classList.add('modal-content');
47         // 모달 콘텐츠에 내용 추가
48         modalContentElem.innerHTML = `
49             <span>This is Modal</span>
50         `
51
52         // 닫기 버튼 엘리먼트 생성
53         const closeBtn = document.createElement('div')
54         closeBtn.classList.add('close-modal-btn')
55         closeBtn.innerHTML = "닫기";
56         // 닫기 버튼에 클릭 이벤트 리스너 추가하여 모달 닫기 함수 호출
57         closeBtn.addEventListener("click", closeModal);
58
59         // 생성한 엘리먼트들을 계층적으로 추가
60         modalElem.appendChild(modalContentElem)
61         modalElem.appendChild(closeBtn)
62
63         modalWrapperElem.appendChild(modalElem)

```

todos

✓	해야 할 일을 입력해주세요.	
<input checked="" type="checkbox"/>	첫번째 할 일	X
<input type="checkbox"/>	두번째 할 일	X
<input type="checkbox"/>	세번째 할 일	X
0 items left All Active Completed Clear Completed		

더블클릭 시 수정 가능!

forEach 함수? 다시 내일 할 예정


```
const num = [1, 2, 3, 4, 5];
```

```
for (i = 0; i < num.length; i++) {  
  console.log(num[i]);  
}
```

```
num.forEach(function(c) { //CODE });
```



Apache 2.4 binaries VS17 Info & Changelog Apache 2.4.58 Win64 다운받기

Apache 2.4.54 Win64 [Apache VC15 Binary] httpd-2.4.54-win64-VC15.zip 다운받기

Apache tomcat 다운받기

이클립스에서 다운받기

[Download Packages](#)

[Eclipse Packages | The Eclipse Foundation](#) - home to a global community, the Eclipse IDE, Jakarta EE and over 350 open source projects...

Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy clic...

www.eclipse.org

웹개발도구

Eclipse IDE 2021-06 R Packages / [Eclipse IDE for Java Developers](#)

[Eclipse IDE for Java Developers | Eclipse Packages](#)

Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy clic...

www.eclipse.org

오라클 다운로드

Name	Download	Release notes
Oracle Database 11gR2 Express Edition for Windows x64	Download	{Note: Unzip the download and run the DISK1/setup.exe}
Oracle Database 11gR2 Express Edition for Windows x32	Download	{Note: Unzip the download and run the DISK1/setup.exe}
Oracle Database 11gR2 Express Edition for Linux x64	Download	{Note: Unzip the download and the RPM file can be installed as normal}

<https://www.oracle.com/database/technologies/xepriorrelease-downloads.html>

XE Prior Release Archive

Oracle Database XE Prior Release Archive Oracle Database Express Edition (XE) Release 11.2.0.2.0 (11gR2).

Name Download Release notes Oracle Database 11gR2 Express Edition for Windows x64 Download { Note:

Unzip the download and run the DISK1/setup.exe} Oracle Database 11gR2 Express Edition for Windo...

www.oracle.com