

8강. 풀과 정규 표현식



폼 요소에 접근하는 방법

체크 박스(CheckBox)의 checked 속성

Checkbox: Check Checkbox

```
Checkbox: <input type="checkbox" id="myCheck">

<button onclick="check()">Check Checkbox</button>
<button onclick="uncheck()">Uncheck Checkbox</button>

<script>
function check() {
    document.getElementById("myCheck").checked = true;
}

function uncheck() {
    document.getElementById("myCheck").checked = false;
}
</script>
```

폼 요소에 접근하는 방법

배송정보 자동으로 입력하기

온라인 쇼핑몰에서 상품 주문시
주문한 사람의 정보와 배송을 받
는 사람의 정보가 일치할 경우,
체크 상자만 누르면 주문한 사람
의 정보를 그대로 배송정보에 입
력해 주는 것을 구현.

주문 정보

이름 장그래

연락처 01012345678

주소 서울시 구로구

배송 정보

주문 정보와 배송 정보가 같습니다

이름 장그래

연락처 01012345678

주소 서울시 구로구

결제하기

폼 요소에 접근하는 방법

배송정보 자동으로 입력하기 – order.html

```
<div id="container">
  <form name="order" action="order.jsp" method="post">
    <fieldset>
      <legend>주문 정보</legend>
      <ul>
        <li>
          <label class="field" for="billingName">이름</label>
          <input type="text" class="input-box" id="billingName" name="billingName">
        </li>
        <li>
          <label class="field" for="billingTel">연락처</label>
          <input type="text" class="input-box" id="billingTel" name="billingTel">
        </li>
        <li>
          <label class="field" for="billingAddr">주소</label>
          <input type="text" class="input-box" id="billingAddr" name="billingAddr">
        </li>
      </ul>
    </fieldset>
  </form>
```

폼 요소에 접근하는 방법

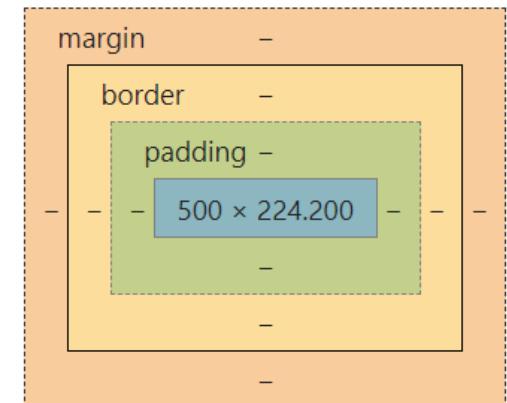
```
<form name="ship">
  <fieldset>
    <legend>배송 정보</legend>
    <ul>
      <li>
        <input type="checkbox" id="shippingInfo" name="shippingInfo">
        <label for="shippingInfo" class="check">주문 정보와 배송 정보가 같습니다</label>
      </li>
      <li>
        <label class="field" for="shippingName">이름</label>
        <input type="text" class="input-box" id="shippingName" name="shippingName">
      </li>
      <li>
        <label class="field" for="shippingTel">연락처</label>
        <input type="text" class="input-box" id="shippingTel" name="shippingTel">
      </li>
      <li>
        <label class="field" for="shippingAddr">주소</label>
        <input type="text" class="input-box" id="shippingAddr" name="shippingAddr">
      </li>
    </ul>
  </fieldset>
  <button type="submit" class="order">결제하기</button>
</form>
</div>
<script src="js/order.js"></script>
```



폼 요소에 접근하는 방법

배송정보 자동으로 입력하기 – order.css

```
#container{  
    width: 500px;  
    margin: 30px auto;  
}  
  
fieldset{  
    border:1px solid #ccc;  
    margin-bottom: 35px;  
}  
  
legend{  
    font-size:1.2em;  
    font-weight:bold;  
    margin-left: 10px;  
}  
  
ul{ list-style: none; }  
li{margin: 20px; }  
  
.field{  
    width: 60px;  
    float: left;  
    line-height: 2em;  
    text-align: right;  
    margin-right: 10px;  
}  
  
.input-box{  
    width: 250px;  
    height: 30px;  
    border:1px solid #aaa;  
    border-radius:5px;  
}  
  
.order{  
    width:100%;  
    padding: 10px;  
    font-size: 1em;  
    border:1px solid #ccc;  
}
```



폼 요소에 접근하는 방법

배송정보 자동으로 입력하기 – order.js

```
//체크 박스 속성에 이벤트 처리하기
var check = document.querySelector("#shippingInfo");

check.addEventListener("click", checkBox);
//체크박스에 click 이벤트 발생했을 때 함수 호출

var billingName = document.querySelector("#billingName");
var billingTel = document.querySelector("#billingTel");
var billingAddr = document.querySelector("#billingAddr");

var shippingName = document.querySelector("#shippingName");
var shippingTel = document.querySelector("#shippingTel");
var shippingAddr = document.querySelector("#shippingAddr");
```

폼 요소에 접근하는 방법

배송정보 자동으로 입력하기 – order.js

```
function checkBox(){
    if(check.checked==true){ //체크박스에 체크 되었다면
        shippingName.value = billingName.value;
        //주문 정보(이름)을 배송 정보(이름)에 대입
        shippingTel.value = billingTel.value;
        shippingAddr.value = billingAddr.value;
    }else{
        shippingName.value = "";
        shippingTel.value = "";
        shippingAddr.value = "";
    }
}
```

선택한 옵션 항목 찾아내기

onchange 속성 – HTML Reference > HTML Events > From Events

HTML Reference

HTML by Alphabet

HTML by Category

HTML Browser Support

HTML Attributes

HTML Global Attributes

HTML Events

HTML Colors

HTML Canvas

HTML Audio/Video

HTML Character Sets

Form Events

Events triggered by actions inside a HTML form (applies to all elements):

Attribute	Value	Description
<u>onblur</u>	<i>script</i>	Fires the moment that the element loses focus.
<u>onchange</u>	<i>script</i>	Fires the moment when the value of the element changes.
<u>oncontextmenu</u>	<i>script</i>	Script to be run when a context menu is opened.

선택한 옵션 항목 찾아내기

onchange 속성 예제.

Select a new car from the list.

When you select a new car, a function is triggered which outputs the value of the selected car.

You selected: BMW

```
<p>Select a new car from the list.</p>

<select id="mySelect" onchange="myFunction()>
  <option value="Audi">Audi
  <option value="BMW">BMW
  <option value="Mercedes">Mercedes
  <option value="Volvo">Volvo
</select>

<p>When you select a new car, a function is triggered which outputs the value of the selected car.</p>

<p id="demo"></p>

<script>
function myFunction() {
  var x = document.getElementById("mySelect").value;
  document.getElementById("demo").innerHTML = "You selected: " + x;
}
</script>
```

선택한 옵션 항목 찾아내기

선택 목록에서 사용자가 선택한 옵션 항목 찾아내기

수 127.0.0.1:5500 내용:
[전기전자공학과]를 선택했습니다.

이름 장그래

학과 산업공학과

신청 과목

이 달에 신청할 과목을 선택하세요.

회화 문법 작문

메일링

메일로 받고 싶은 뉴스 주제를 선택해 주세요

해외 단신 5분 회화 모닝팝스

선택한 옵션 항목 찾아내기

```
<div id="container">
  <h1>수강신청</h1>
  <form name="testForm">
    <fieldset>
      <legend>신청인</legend>
      <ul>
        <li>
          <label class="reg" for="userN...>이름</label>
          <input type="text" id="userN... name="userN... autofocus required>
        </li>
        <li>
          <label class="reg" for="class">학과</label>
          <select name="major" id="major" onchange="displaySelect()">
            <option>---- 학과 선택 ----</option>
            <option value="archi">건축학과</option>
            <option value="mechanic">기계공학과</option>
            <option value="elec">전기전자공학과</option>
            <option value="computer">컴퓨터공학과</option>
            <option value="chemical">화학공학과</option>
          </select>
        </li>
      </ul>
    </fieldset>
```

major.html

선택한 옵션 항목 찾아내기

major.html

```
<fieldset>
    <legend>신청 과목</legend>
    <p>이 달에 신청할 과목을 선택하세요.</p>
    <label><input type="radio" name="subject" value="speaking">회화</label>
    <label><input type="radio" name="subject" value="grammar">문법</label>
    <label><input type="radio" name="subject" value="writing">작문</label>
</fieldset>
<fieldset>
    <legend>메일링</legend>
    <p>메일로 받고 싶은 뉴스 주제를 선택해 주세요</p>
    <label><input type="checkbox" name="mailing1" value="news">해외 단신</label>
    <label><input type="checkbox" name="mailing2" value="dialog">5분 회화</label>
    <label><input type="checkbox" name="mailing3" value="pops">모닝팝스</label>
</fieldset>
</form>
</div>
```

선택한 옵션 항목 찾아내기

```
#container {  
    width:500px;  
    margin:10px auto;  
}  
form fieldset{margin-bottom:25px;}  
form legend{  
    font-size:15px;  
    font-weight:700;  
    margin-left: 10px;  
}  
form ul li{  
    list-style:none;  
    margin: 15px 0;  
    font-size:14px;  
}  
label.reg {  
    width:80px;  
    float:left;  
    text-align:right;  
    margin-right:10px;  
    font-size:15px;  
    color:#390;  
    font-weight:bold;  
}
```

major.css

선택한 옵션 항목 찾아내기

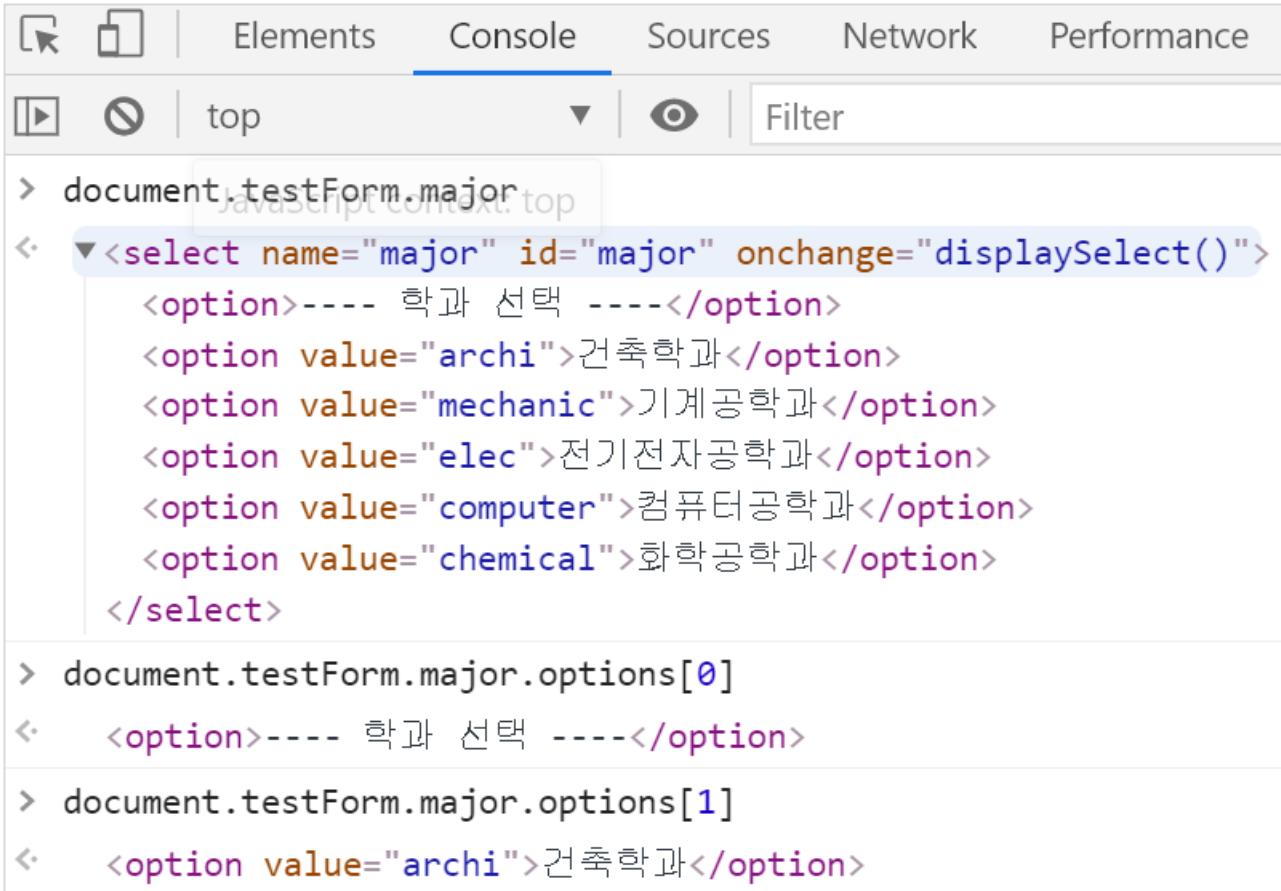
선택 목록에서 사용자가 선택한 옵션 항목 찾아내기

```
//select 태그 이름에 폼 이름으로 접근  
var selectMenu = document.testForm.major;  
  
function displaySelect(){  
    //select option속성의 인덱스로 선택  
    // var selectedText = selectMenu.options[selectMenu.selectedIndex].innerText;  
    // alert(selectedText + "가 선택되었습니다.");  
  
    //id 선택자로 처리  
    var selectedText = document.getElementById("subj").value;  
    alert(selectedText + "가 선택되었습니다.");  
}
```

major.js

선택한 옵션 항목 찾아내기

콘솔 창에서 보기



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console output displays the following code and its execution results:

```
> document.testForm.major
<-- ▼<select name="major" id="major" onchange="displaySelect()>
    <option>---- 학과 선택 ----</option>
    <option value="archi">건축학과</option>
    <option value="mechanic">기계공학과</option>
    <option value="elec">전기전자공학과</option>
    <option value="computer">컴퓨터공학과</option>
    <option value="chemical">화학공학과</option>
</select>

> document.testForm.major.options[0]
<-- <option>---- 학과 선택 ----</option>

> document.testForm.major.options[1]
<-- <option value="archi">건축학과</option>
```

The code demonstrates how to access the options of a select element and inspect individual options.

입력값 검증 프로그램

회원가입 폼 - 입력값 검증하기(유효성 검사)

회원가입

아이디	4~15자리의 영문과 숫자로 입력
이메일	<input type="text"/>
비밀번호	8자리 이상
비밀번호 확인	<input type="text"/>
메일링 수신	<input type="radio"/> 예 <input type="radio"/> 아니오
<input type="button" value="가입하기"/> <input type="button" value="취소하기"/>	

입력값 검증 프로그램

register.html

```
<div id="container">
    <h2>회원 가입</h2>
    <form action="processJoin.jsp" method="post">
        <ul>
            <li>
                <label for="user-id" class="field">아이디</label>
                <input type="text" id="user-id" name="id"
                       placeholder="4~15자리의 영문과 숫자로 입력" required>
            </li>
            <li>
                <label for="email" class="field">이메일</label>
                <input type="email" id="email" name="email" required>
            </li>
            <li>
                <label for="user-pw1" class="field">비밀번호</label>
                <input type="password" id="user-pw1" name="pw1"
                       placeholder="8자리 이상" required>
            </li>
        </ul>
    </form>
</div>
```

입력값 검증 프로그램

register.html

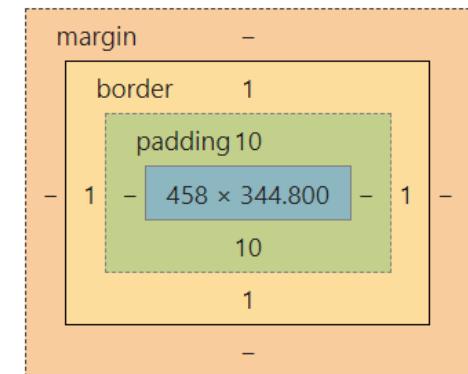
```
</li>
<li>
    <label for="user-pw2" class="field">비밀번호 확인</label>
    <input type="password" id="user-pw2" name="pw2" required>
</li>
<li>
    <label class="field">메일링 수신</label>
    <label><input type="radio" value="yes" name="mailing">예</label>
    <label><input type="radio" value="no" name="mailing">아니오</label>
</li>
<li>
    <input type="submit" value="가입하기" class="button">
    <input type="reset" value="취소하기" class="button">
</li>
</form>
</div>
<script src="js/validation.js"></script>
```

입력값 검증 프로그램

```
#container{  
    width: 460px;  
    margin: 30px auto;  
}  
  
ul li{  
    list-style: none;  
    margin: 20px;  
}  
  
h2{padding-left: 50px;}  
  
form{  
    padding: 10px 0;  
    border: 1px solid #ccc;  
}  
  
.field{  
    width: 120px;  
    float: left;  
    line-height: 2em;  
}
```

register.css

```
input[type="text"], [type="email"], [type="password"]{  
    width: 220px;  
    height: 30px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
}  
  
input[type="radio"]{  
    margin-top: 10px;  
}  
  
.button{  
    border: 1px solid #ccc;  
    border-radius: 2px;  
    font-size: 0.9em;  
    padding: 5px;  
    margin: 5px 10px 5px 0;  
}
```



입력값 검증 프로그램

```
// 입력값 검증(유효성 체크)
var id = document.getElementById("user-id");
var pw1 = document.getElementById("user-pw1");
var pw2 = document.getElementById("user-pw2");

id.onchange = checkId; //checkId 함수 호출

//이벤트 처리기 함수를 사용했을 경우
//id.addEventListener("change", checkId);

pw1.onchange = checkPw;
pw2.onchange = comparePw;

function checkId(){
    if(id.value.length < 4 || id.value.length > 15){
        alert("4~15자리의 영문과 숫자로 입력해 주세요");
        id.select(); //선택 범위 지정
    }
}
```

validation.js



입력값 검증 프로그램

```
function checkPw(){
    if(pw1.value.length < 8 ){
        alert("비밀번호는 8자리 이상이어야 합니다.");
        pw1.value = "";      //기존 문자 지우기
        pw1.focus();         //커서가 위치함
    }
}
```

validation.js

```
function comparePw(){
    if(pw1.value != pw2.value){
        alert("비밀번호가 일치하지 않습니다.");
        pw2.value = "";
        pw2.focus();
    }
}
```



정규 표현식과 유효성 검사

정규 표현식(Regular Expression)

- 특정한 규칙을 가진 문자열의 집합을 표현하고 처리하는 것을 말한다.
- 데이터의 유효성 검사에 주로 사용된다.

표현식	설명
[0-9]	숫자
[a-zA-Z]	영문 대, 소문자
[가-힣]	한글
[~!@#\$%^&*]	특수문자
\d{6}[-]\d{1-4}\d{6}	주민등록번호
\d{3}[-]\d{2}	우편번호

정규 표현식과 유효성 검사

정규 표현식(Regular Expression)

메타문자	설명
*	문자 반복
[x]	x를 찾음
\d	숫자(decimal)
\s	공백(space)
\w	알파벳+숫자(word)
{3}	반복횟수

유효성 검사(validation)

- `test()` 함수 : 매개 문자가 정규 표현식과 일치하면 `true`, 아니면 `false`

```
<script>
    //문자열 일치 여부
    //방법1 - RegExp(표현식)
    //방법2 - /표현식/

    let text = "Hello World"

    let pattern1 = new RegExp("Hello")
    let result = pattern1.test(text)
    console.log(result)

    let pattern2 = /World/g      //'g'는 전역 검색(생략 가능)
    let result2 = pattern2.test(text)
    console.log(result2)

    //문자열 및 글자수 일치 여부
    let id = "a123"

    let regEx = /[0-9A-Za-z]{5}/ //정규 표현식 사용
    console.log(regEx.test(id))
</script>
```

유효성 검사(validation)

- 회원가입 유효성 검사

회원가입

아이디 내용:
아이디는 5자만 가능합니다.

비밀번호

이름

유효성 검사(validation)

- 회원가입 유효성 검사

```
<h2>회원가입</h2>
<form action="" method="post" name="regForm">
    <fieldset>
        <p>
            <label for="mid">아이디</label>
            <input type="text" id="mid" name="mid">
        </p>
        <p>
            <label for="passwd">비밀번호</label>
            <input type="text" id="passwd" name="passwd">
        </p>
        <p>
            <label for="name">이름</label>
            <input type="text" id="name" name="name">
        </p>
    </fieldset>
    <div>
        <input type="button" value="가입" onclick="checkMember()">
        <input type="reset" value="취소">
    </div>
</form>
```

유효성 검사(validation)

- 회원가입 유효성 검사

```
function checkMember(){
    let form = document.regForm;
    let id = form.mid.value;
    let pwd = form.passwd.value;
    let name = form.name.value;

    //비밀번호 정규 표현식
    let pwd_pat1 = /[0-9A-Za-z]/;
    let pwd_pat2 = /[~!@#$%^&*]/;

    if(id.length != 5){
        alert("아이디는 5자까지 가능합니다.");
        form.mid.select();
        return false;
    }
}
```

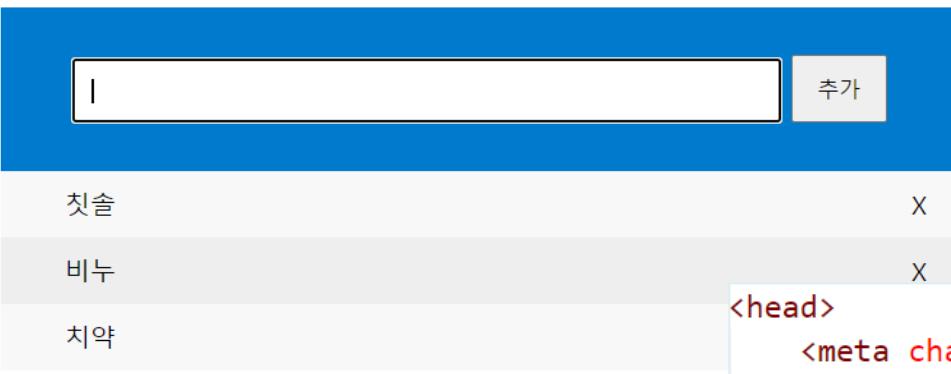
유효성 검사(validation)

- 회원가입 유효성 검사

```
else if(pwd.length != 8 || !pwd_pat1.test(pwd) || !pwd_pat2.test(pwd) ){
    alert("비밀번호는 영문자, 숫자, 특수문자 포함 8자까지 가능합니다.");
    form.mid.select();
    return false;
}
else if(name == ""){
    alert("이름은 필수 입력 항목입니다.")
    form.mid.focus()
    return false;
}
else{
    form.submit();
}
```

여행 준비물 프로그램 만들기

여행 준비물 점검 목록



The screenshot shows a user interface for a travel checklist. At the top is a search bar with a placeholder 'I' and a '추가' (Add) button. Below it is a table with three rows: '칫솔' (Toothbrush), '비누' (Soap), and '치약' (Toothpaste). Each row has an 'X' button to its right. A modal window is open, displaying the following HTML code:

```
<head>
<meta charset="UTF-8">
<title>여행 준비물 점검 목록</title>
<link rel="stylesheet" href="css/travel.css">
<link rel="stylesheet" href="css/list.css">
</head>
```

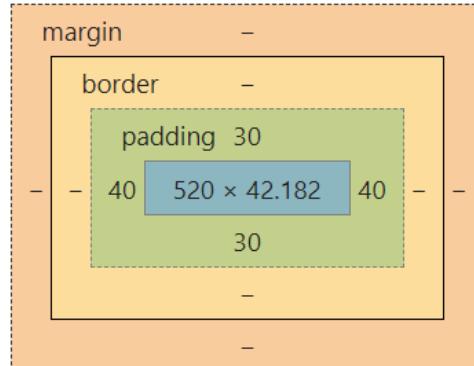
Below the modal, the main page's code is shown:

```
<div id="wrapper">
  <h2>여행 준비물 점검 목록</h2>
  <form action="addItem.jsp" method="post">
    <input type="text" name="item" id="item" autofocus>
    <button type="button" id="add" class="addBtn">추가</button>
  </form>
  <div id="itemList"></div>
</div>
<script src="js/travel.js"></script>
```

여행 준비를 프로그램 만들기

```
#wrapper{  
    width: 600px;  
    margin: 0 auto;  
}  
  
h2{text-align: center;}  
  
form{  
    background: #0094ff;  
    text-align: center;  
    padding: 30px 40px;  
}  
  
input{  
    width: 420px;  
    padding: 10px;  
}  
  
button{  
    width: 60px;  
    padding: 10px;  
}
```

item.css



```
ul{  
    margin: 0; padding: 0;  
    list-style: none;  
}  
  
ul li{  
    padding: 10px 10px 10px 40px;  
    background: #eee;  
    position: relative;  
}  
  
.close{  
    position: absolute;  
    top: 0;  
    right: 0;  
    padding: 10px 20px;  
}  
  
.close:hover{  
    background: rgb(28, 111, 194);  
    color: white;  
}
```

list.css

여행 준비를 프로그램 만들기

1. 물품(아이템) 추가

```
var itemList = [];

//버튼 클릭시 이벤트 처리
var addBtn = document.getElementById("add");
addBtn.addEventListener("click", addList);

//물품 추가
function addList(){
    var item = document.getElementById("item").value; //물품 입력

    if(item != null){
        itemList.push(item); //요소 추가
        document.getElementById("item").value = ""; //초기화
        document.getElementById("item").focus(); //커서 위치
    }
    showList();
}
```

console에서 조회하기

```
> itemList
< ▶ []
> itemList
< ▶ ["비누"]
> itemList
< ▶ (2) ["비누", "지퍼"]
```

여행 준비를 프로그램 만들기

2. 물품(아이템) 조회 테스트

여행 준비물 점검 목록



```
function showList(){
    for(var i=0; i<itemList.length; i++){
        document.getElementById("itemList").innerHTML = itemList[i];
    }
}
```

여행 준비를 프로그램 만들기

여행 준비물 점검 목록

추가

- 칫솔
- 비누
- 치약

```
//물품 조회
function showList(){
    var list = "<ul>";
    for(var i=0; i<itemList.length; i++){
        //document.getElementById("itemList").innerHTML = itemList[i];
        list += "<li>" + itemList[i] + "<span class='close' id=" + i + "X</span></li>";
    }
    list += "<ul>";
    document.getElementById("itemList").innerHTML = list;
/*
<ul>
    <li>itemList[0]</li>
    <li>itemList[1]</li>
    <li>itemList[2]</li>
</ul>
*/
}
```

여행 준비를 프로그램 만들기

3. 물품(아이템) 삭제

The screenshot shows a browser's developer tools with the 'Console' tab selected. At the top, there is a list of items with 'X' buttons for deletion:

비누	X
칫솔	X
치약	X

The console contains the following JavaScript code:

```
//아이템 삭제 - 'X' 클릭
var removeList = document.querySelectorAll(".close");
//class='close'인 요소를 모두 가져오므로 배열이 됨

for(var i=0; i<removeList.length; i++){
    removeList[i].addEventListener("click", remove);
}

function remove(){
    console.log(this); //this 값을 콘솔에 표시
}
```

A red callout box with a red arrow points from the text "X를 클릭(삭제)하면 이
벤트의 대상을 알 수 있
다. this 예약어 사용" to the line "console.log(this);".

여행 준비를 프로그램 만들기

3. 물품(아이템) 삭제

```
//아이템 삭제 - 'X' 클릭
var removeList = document.querySelectorAll(".close");
//class='close'인 요소를 모두 가져오므로 배열이 됨

for(var i=0; i<removeList.length; i++){
    removeList[i].addEventListener("click", remove);
}

function remove(){
    //console.log(this); //this 값을 콘솔에 표시
    var id = this.getAttribute("id");
    itemList.splice(id, 1); //인덱스가 id인 요소 1개 삭제
    showList(); // 삭제 후 변경된 itemList를 다시 화면에 표시
}
```