

□ 개념 확인

(1) 빈 괄호를 채워 넣으시오

- ① 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것을 (문서 객체)라고 한다.
- ② (className) 프로퍼티는 클래스 속성값을 가져오거나 설정하며 (id) 프로퍼티는 id 속성의 값을 가져오거나 설정한다
- ③ (textContent) 프로퍼티는 마크업이 포함되면 문자열로 인식하며 (innerHTML) 프로퍼티는 마크업이 포함된 새로운 콘텐츠를 DOM에 추가할 수 있다
- ④ (style) 프로퍼티는 요소 노드의 인라인 스타일 추가, 제거, 변경한다
- ⑤ (innerHTML)프로퍼티는 마크업이 포함된 새로운 콘텐츠를 지정하면 새로운 요소를 DOM에 추가할 수 있다
- ⑥ (querySelectorAll(cssSelector)) 메소드는 지정된 css 선택자를 사용하여 요소 노드를 모두 선택하여 NodeList 객체로 반환한다
- ⑦ (getElementsByClassName(class)) 메소드는 class 어트리뷰트 값으로 요소 노드를 모두 선택하여 HTMLCollection 객체로 반환한다
- ⑧ (hasChildNodes()) 메소드는 자식 노드를 확인하여 있으면 true, 없으면 false 반환한다
- ⑨ (children)프로퍼티는 자식 노드 중 요소 노드만 모두 탐색하여 반환한다.
- ⑩ (인라인 이벤트 모델)은 HTML 태그 내부에 자바스크립트 코드를 넣어 이벤트를 연결하는 방식이다
- ⑪ 등록된 이벤트를 제거하려면 (removeEventListener())메소드를 사용 한다.
- ⑫ 인라인 이벤트 핸들러에서 이벤트 객체를 전달할 때는 (event)라는 이름으로 전달해야 한다

(2) 다음 질문에 OX를 표시하시오.

- ① querySelector() 메서드는 선택자로 요소를 여러 개 선택할 가능성이 있으므로, 배열로 문서 객체를 리턴한다. (X) →만족하는 요소노드가 여러 개 이면 첫번째 요소만 반환
- ② textContent 프로퍼티는 마크업을 포함한다 (X) →마크업 무시
- ③ nodeValue 프로퍼티는 요소 노드의 경우 문자열을 반환한다 (X) →요소 노드의 경우 null 반환
- ④ 클래스 선택자를 사용할 경우 #을 함께 사용해야 한다 (X) →ID 선택자를 사용할 경우에 #을 사용한다
- ⑤ DOM을 사용하여 웹 페이지를 조작할 때 요소 노드를 먼저 선택한 후 텍스트를 조작한다 (O)
- ⑥ HTML 요소 사이의 스페이스, 탭, 줄 바꿈 등의 공백 문자는 텍스트 노드를 생성하지 않는다 (X)
→텍스트 노드를 생성한다
- ⑦ 텍스트 노드는 자식노드를 갖지 않는다 (O)
- ⑧ 요소 노드 객체에는 HTML 어트리뷰트에 대응하는 DOM 프로퍼티가 존재한다 (O)
- ⑨ DOM에 추가된 요소는 임의로 제거할 수 없다. (X)
→제거 가능하다
- ⑩ 이벤트 리스너 내부에서 this 키워드를 사용하면 이벤트를 발생한 자기 자신을 의미한다 (O)
- ⑪ 이벤트 객체를 사용하면 이벤트와 관련한 정보를 알아낼 수 있다. (O)

⑫ a 태그를 클릭했을 때 발생하는 웹 페이지 이동 등 기본 이벤트는 제거할 수 없다. (X)

→이벤트 리스너에서 false 리턴 혹은 이벤트 객체의 preventDefault()호출시 취소가능

⑬ 등록된 이벤트는 제거할 수 없다 (X)

→이벤트 제거 가능

(3) 이벤트 핸들러 등록 방법 3 가지를 설명하시오.

===== 풀이 =====

1.인라인 이벤트 핸들러 방식

- HTML 요소의 이벤트 핸들러 속성에 이벤트 핸들러를 등록하는 방법

2.이벤트 핸들러 프로퍼티 방식

-요소 노드의 이벤트 속성으로 이벤트를 연결

-자바스크립트와 HTML 이 분리

-한번에 하나의 이벤트 리스너만 가질 수 있음

3.addEventListener 메소드 방식

- addEventListener 메소드를 이용하여 대상 DOM 요소에 이벤트를 바인딩하고 해당 이벤트 가 발생했을 때 실행될 콜백 함수(이벤트 핸들러)를 지정

-하나의 이벤트에 여러 개의 이벤트 핸들러 추가 가능

(4) 다음 코드의 문제와 해결 방법을 제시하시오.

```
<!DOCTYPE html>
<html>
<head>
  <title>Document Object Model</title>
  <script>
    let element = document.getElementById('element');
    element.style.color = 'red';
    element.innerHTML = 'From JavaScript';
  </script>
</head>
<body>
  <h1 id="element">Lorem ipsum dolor amet</h1>
</body>
</html>
```

===== 풀이 =====

```
<script>
  window.onload = function () {
    let element = document.getElementById('element');
    element.style.color = 'red';
    element.innerHTML = 'From JavaScript';
  }
</script>
```

(5) 다음 코드에서 h1 태그 한 개를 선택할 수 있는 자바스크립트 코드를 2 개 이상 제시하시오.

```
<!DOCTYPE html>
<html>
<body>
  <h1 id="element" class="header">Lorem ipsum dolor amet</h1>
</body>
```

===== 풀이 =====

```
let element = document.getElementById('element');
let element = document.querySelector('.header');
```

(6) 질문에 해당하는 스크립트를 작성하시오

① p 태그 노드를 생성하여 nodeP 에 저장한다

```
const nodeP = document.createElement('p');
```

② "텍스트 추가" 텍스트 노드를 생성하여 nodeT 에 저장한다

```
const nodeT = document.createTextNode('텍스트 추가');
```

③ ①에서 생성된 노드에 ②에서 생성된 텍스트 노드를 자식 노드로 추가한다

```
nodeP.appendChild(nodeT);
```

④ ①에서 생성된 노드에 자식 노드가 있는지를 확인한다

```
if(nodeP.hasChildNodes()){
  document.write("자식노드 있음");
}
else{
  document.write("자식노드 없음");
}
```

(7) DOM 트리 노드 종류에 대하여 설명하시오

===== 풀이 =====

1.문서 노드(document node)

-트리의 최상위에 존재, DOM tree 에 접근하기 위한 시작점(entry point)

2.요소 노드(element Node)

- HTML 요소 표현 • 어트리뷰트, 텍스트 노드에 접근하려면 먼저 요소 노드를 찾아 접근

3.어트리뷰트 노드(Attribute Node)

- HTML 요소의 어트리뷰트, 해당 요소 노드를 찾아 접근하면 참조, 수정 가능

4.텍스트 노드(Text Node)

- HTML 요소의 텍스트 표현, 요소 노드의 자식이며 자식 노드를 가질 수 없다

(8) 다음 코드의 실행 결과를 제시하고 설명하시오.

```
<!DOCTYPE html>
<html>
<head>
  <title>Query Selector</title>
</head>
<body>
  <h1>Header - 1</h1>
  <h1>Header - 2</h1>
  <h1>Header - 3</h1>
  <script>
    let header = document.querySelector('h1');
    header.style.color = 'white';
    header.style.backgroundColor = 'black';
  </script>
</body>
</html>
```

```
let header = document.querySelector('h1');
```

querySelector 는 요소 노드가 여러 개일경우 첫번째 요소만 반환하기 때문에 <h1>Header - 1</h1>요소를 반환하여 header 에 저장한다.

```
header.style.color = 'white';
header.style.backgroundColor = 'black';
```

그 요소의 색깔은 하얀색으로 배경색은 검정색으로 설정한다.

□ 개념 활용 응용 프로그래밍

(1) class 어트리뷰트 값이 red인 요소에 대하여 글자색을 'green', 나머지는 'blue'로 출력하는 자바 스크립트 코드를 작성하시오. 단 li 태그명으로 모든 요소 노드를 선택하도록 한다. getElementsByTagName() 사용

```
<body>
  <h1>Cities</h1>
  <ul>
    <li class="red">hallym</li>
    <li class="blue">software</li>
    <li class="red">big data</li>
    <li class="green">IoT</li>
    <li class="red">contentIT</li>
  </ul>
</body>
```



[소스]

```
<script>
  window.onload = function () {
    let elems = document.getElementsByTagName('li');
```

```

        for(let value of elems){
            if(value.className == "red"){
                value.style.color = 'green';
            }
            else{
                value.style.color = 'blue';
            }
        }
    }
}
</script>

```

[실행결과]

Cities

- hallym
- software
- big data
- IoT
- contentIT

(2) css 셀렉터를 사용하여 클래스 어트리뷰트 값이 hallym인 요소 노드를 모두 선택하여 다음과 같이 출력되는 자바 스크립트 코드를 작성하시오. 모든 요소는 width 값을 '100px'로 한다. css 사용 불가-
querySelectorAll()사용

```

<body>
  <h2>css 셀렉터 사용하기</h2>
  <h4 class = "hallym" >css3</h4>
  <h4 class = "hallym">html</h4>
  <h4 class = "hallym">JavaScript</h4>
  <h4 class = "hallym">TypeScript</h4>
  <h4 class = "hallym">jQuery</h4>
</body>

```

css 셀렉터 사용하기

css3

html

JavaScript

TypeScript

jQuery

[소스]

```

<script>
    window.onload = function () {
        const elems = document.querySelectorAll('.hallym');
        let count = 0;
        for(let value of elems){
            value.style.width = '100px';
            if (count % 2 == 0) {
                value.style.backgroundColor = "yellow";
            }
            else {
                value.style.backgroundColor = "gray";
            }
            count = count + 1;
        }
    }

```

```
}  
  
}  
</script>
```

[실행결과]

css 셀렉터 사용하기

css3

html

JavaScript

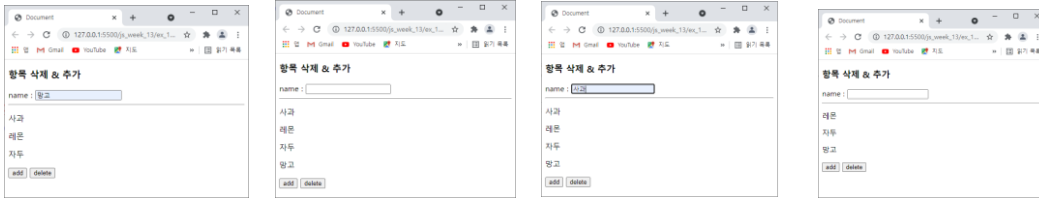
TypeScript

jQuery

(3) 글 상자로 입력된 데이터를 웹 문서에 추가하거나 삭제하는 자바 스크립트 코드를 작성하시오

- ① 'add' 버튼을 클릭하면 글 상자로 입력된 데이터를 추가
- ② 'delete' 버튼을 클릭하면 글 상자에 입력된 데이터와 일치하는 항목을 삭제.

```
<body>  
  <div id="container">  
    <h3> 항목 삭제 & 추가</h3>  
    <form>  
      name : <input type="text" id="name" required> <br>  
    </form>  
  </div>  
  <hr>  
  <div id="list">  
    <p>사과</p>  
    <p>레몬</p>  
    <p>자두</p>  
  </div>  
  <input type="button" value="add" onclick="childNodesAdd()">  
  <input type="button" value="delete" onclick="childNodesDelete()">  
</body>
```



[소스]

```
<script>
  function childNodeAdd() {
    var fruit = document.getElementById("name");

    const newElem = document.createElement('p');
    const newText = document.createTextNode(fruit.value);
    newElem.appendChild(newText);

    const container = document.querySelector('#list');
    container.appendChild(newElem);
  }
  function childNodeDelete(){
    var fruit = document.getElementById("name").value;

    const elems = document.querySelectorAll('p');
    for(value of elems){
      if(value.innerHTML == fruit){
        value.innerHTML = "";
      }
    }
  }
</script>
```

[실행결과]

항목 삭제 & 추가

name :

사과

레몬

자두

사과

사과

add

delete

항목 삭제 & 추가

name :

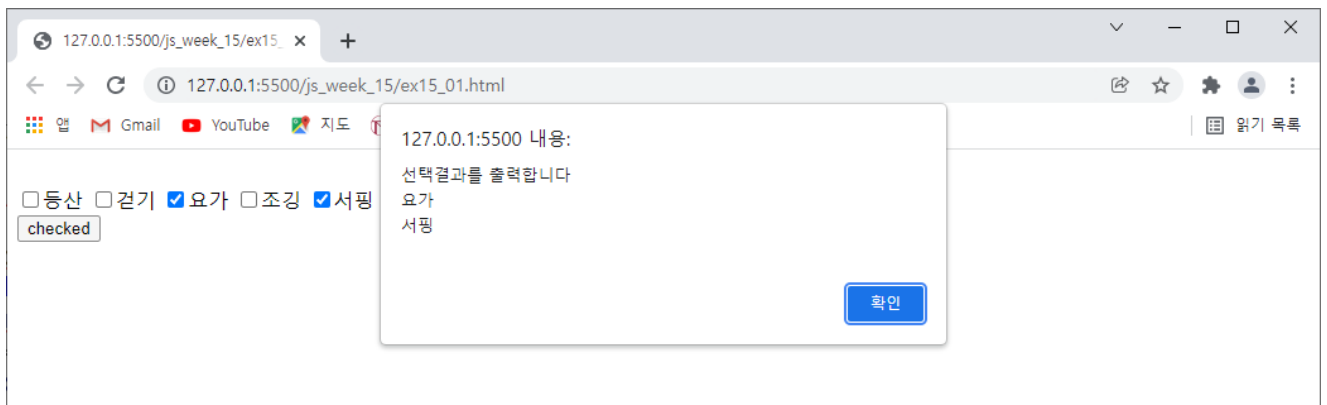
레몬

자두

add

delete

- (4) 버튼을 클릭하면 체크박스에서 체크된 항목을 알림창으로 출력하는 자바 스크립트 코드를 작성하시오. 이벤트 핸들러 등록 방식은 `addEventListener()` 사용



```
<body>
<form>
  <input type="checkbox" name="hobby" value="등산">등산
  <input type="checkbox" name="hobby" value="걷기">걷기
  <input type="checkbox" name="hobby" value="요가">요가
  <input type="checkbox" name="hobby" value="조깅">조깅
  <input type="checkbox" name="hobby" value="서핑">서핑<br>
  <input type="button" value="checked">
</form>
</body>
```

[소스]

```
<script>
  window.onload = function () {
    let em = document.querySelector('input[type=button]');
    em.addEventListener('click', clicked);
  }
  function clicked() {
    let str = "";
    let elems = document.getElementsByName('hobby');
    str += "선택결과를 출력합니다\n";
    for (value of elems) {
      if (value.checked) {
```

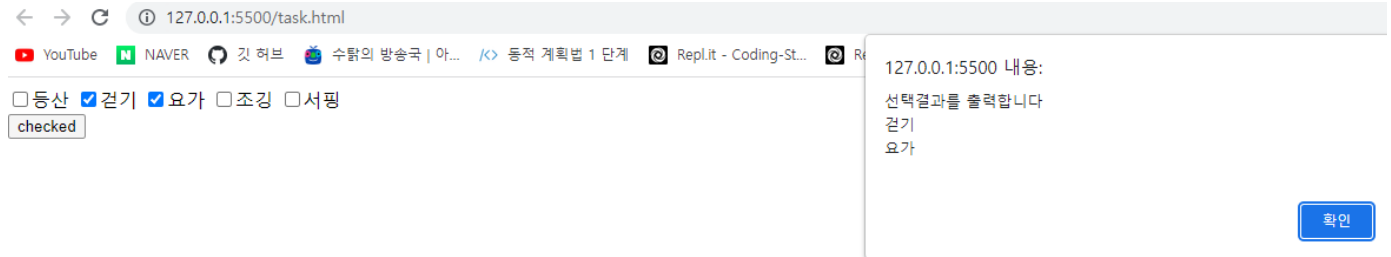


```

        str += value.value + "\n";
    }
}
alert(str);
}
</script>

```

[실행결과]

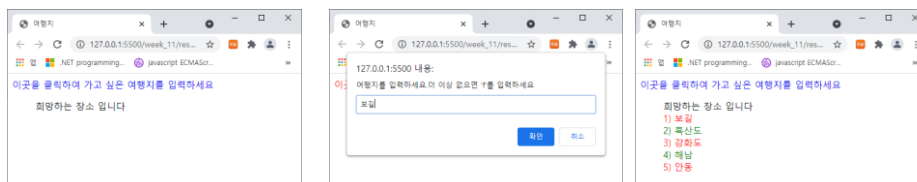


- (5) 텍스트를 클릭하면 입력창에서 데이터를 입력한다. f를 입력하면 전체 데이터를 제시된 결과처럼 출력하는 자바스크립트 코드를 작성하시오. innerHTML 속성을 사용하여 여행지를 추가할 것

```

<head>
  <meta charset="UTF-8">
  <title>여행지</title>
</head>
<body>
  <a href="javascript:insert()">이곳을 클릭하여 가고 싶은 여행지를 입력하세요</a>
  <ul>희망하는 장소입니다</ul>
</body>

```



[소스]

```

<script>
  function insert() {
    let count = 1;
    while (true) {
      let place = prompt("여행지를 입력하세요. 더 이상 없으면 'f'를 입력하세요");
      if (place == "f") {
        break;
      }
      const newElem = document.createElement('li');
      const newText = document.createTextNode("");
      newElem.appendChild(newText);
      newElem.innerHTML = count + ")" + place;

      const container = document.querySelector('ul');
      container.appendChild(newElem);
    }
  }

```

```

        count += 1;
    }

    const container = document.querySelector('ul');
    container.style.listStyle = 'none';

    let elems = document.querySelectorAll('li');

    for(let i = 0 ; i < elems.length ; i++){
        if(i%2 == 0){
            elems[i].style.color = "red";
        }
        else{
            elems[i].style.color = "green";
        }
    }
}
</script>

```

[실행결과]

[이곳을 클릭하여 가고 싶은 여행지를 입력하세요](#)

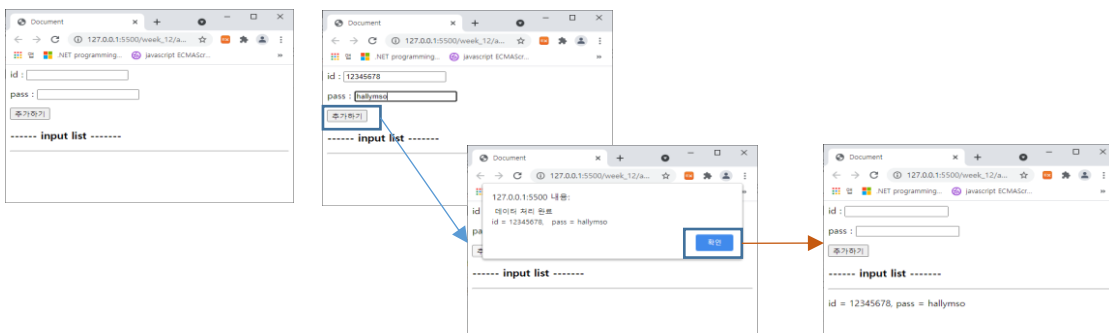
희망하는 장소 입니다

- 1)강화도
- 2)제주도
- 3)부산
- 4)서울
- 5)홍콩
- 6)유럽
- 7)체코

(6) 제시된 결과처럼 텍스트박스로 입력된 데이터를 웹 문서에 추가는 자바 스크립트 코드를 작성하시오 – 이벤트 핸들러는 addEventListener 메소드를 사용할 것

- ① id에 입력된 글자가 8이상이면 자동으로 포커스가 pass로 이동
- ② pass에서 백스페이스이고 입력된 글자가 없으면 id로 포커스 이동
- ③ 추가하기 버튼을 클릭하면 경고창으로 데이터 출력 후 웹 브라우저로 출력
- ④ 텍스트 박스에 입력된 값 추가하기 버튼을 클릭하면 삭제

힌트) keydown 이벤트 사용, keyCode 프로퍼티 값이 8이면 백스페이스



```

<body>
  <form>
    id : <input type="text" maxlength="8">
    <p></p>
    pass : <input type="text" maxlength="8">
    <p></p>
    <input type="button" value="추가하기" id="add">
  </form>
  <h3>----- input list -----</h3>
  <hr>
  <div id='container'></div>
</body>

```

[소스]

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <script>
    let del = false;
    window.onload = function () {
      let em = document.querySelector('input[type=button]');
      em.addEventListener('click', clicked);
    }
    function clicked() {
      let pw = document.querySelector('#password');
      let id = document.querySelector('#id');
      let str = "id = " + id.value + ", pass = " + pw.value;
      alert("데이터 처리 완료\n" + "id = " + str);

      let elems = document.querySelectorAll("p")
      for (value of elems) {
        del = false;
        if (value.innerHTML == str) {
          value.remove();
          del = true;
          break;
        }
      }
      if (!del) {
        const newElem = document.createElement('p');
        const newText = document.createTextNode(str);
        newElem.appendChild(newText);

        const container = document.querySelector('#container');
        container.appendChild(newElem);
      }
    }
  </script>

```

```

        del = false;
    }

}

document.addEventListener('keyup', function (event) {
    let pw = document.querySelector('#password');
    let id = document.querySelector('#id');
    if (id.value.length >= 7) {
        pw.focus();
    }

})

document.addEventListener('keydown', function (event) {
    let pw = document.querySelector('#password');
    let id = document.querySelector('#id');
    if (event.keyCode == 8) {
        if (pw.value.length == 0) {
            id.focus();
        }
    }
})

</script>
</head>

<body>
    <form>
        id : <input type="text" maxlength="8" id="id">
        <p></p>
        pass : <input type="text" maxlength="8" id="password">
        <p></p>
        <input type="button" value="추가하기" id="add">
    </form>
    <h3>----- input list -----</h3>
    <hr>
    <div id='container'></div>
</body>

</html>

```

[실행결과]

id :

pass :

----- input list -----

id = 1, pass = 12

id = 1, pass = 12345

Re

127.0.0.1:5500 내용:
데이터 처리 완료
id = id = 1, pass = 12

id :

pass :

----- input list -----

id = 1, pass = 12345
