

대학 주도형 아카데미 K-디지털 트레이닝
AI 데이터분석 풀스택 웹 개발자 양성과정

DOM

(Document Object Model)

김경민

pnumin@pusan.ac.kr



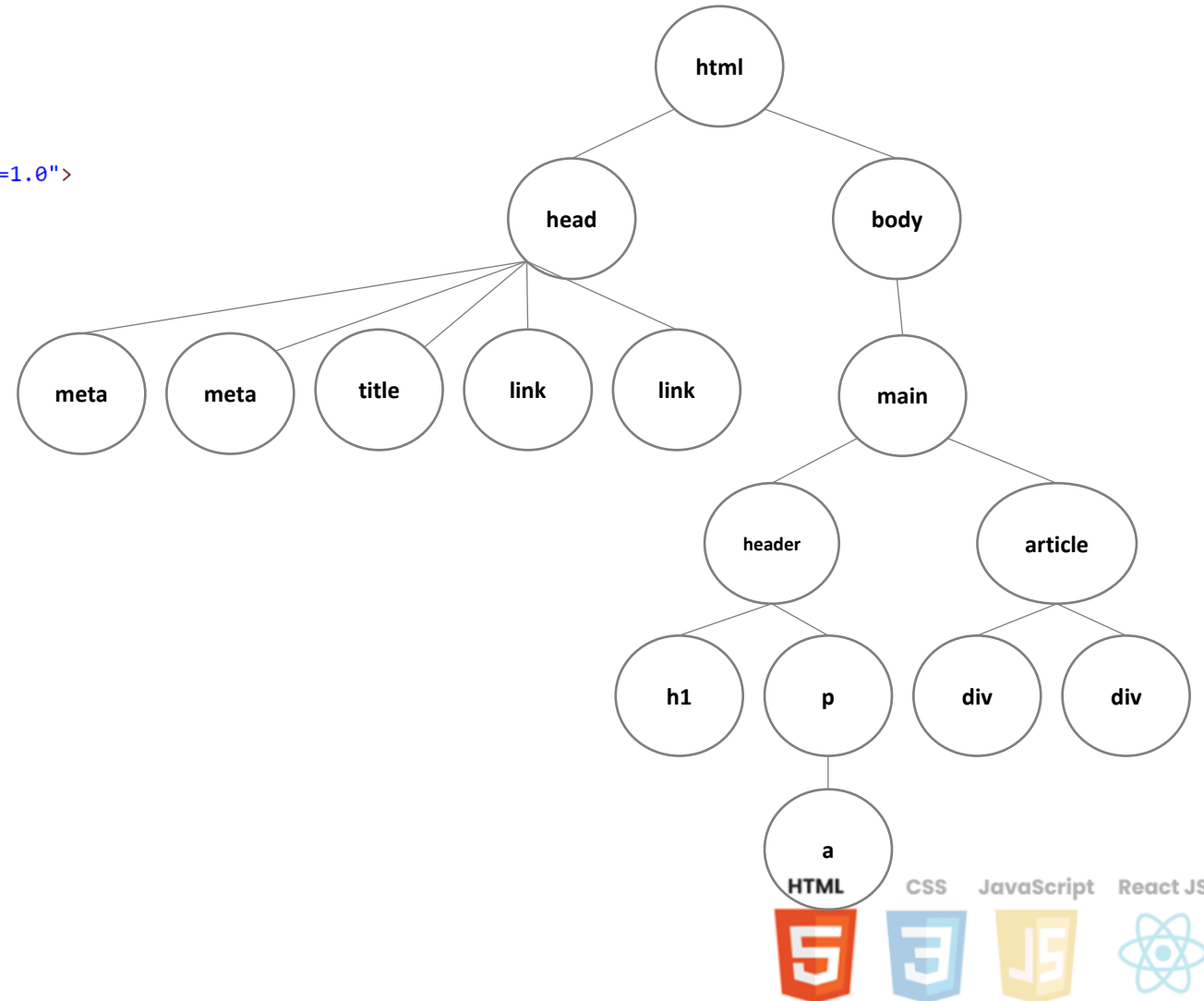
DOM(Document Object Model)

- 웹 문서의 구조화된 표현을 나타내는 객체 기반의 모델
 - 웹 페이지가 로드되면, 브라우저는 HTML 문서를 파싱하여 DOM을 생성
 - JavaScript가 웹 페이지의 구조와 콘텐츠를 동적으로 접근하고 조작할 수 있도록 하는 인터페이스를 제공
- 특징
 - 트리 구조로 표현되며, 문서의 각 요소와 콘텐츠는 이 구조 내에서 부모-자식 관계로 연결되어 특정 요소를 쉽게 탐색 하고 조작
 - 자바스크립트를 사용하여 DOM의 요소들을 동적으로 추가, 삭제, 수정 가능하여 실시간으로 변경되는 웹 페이지 제작



DOM(Document Object Model)

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>자바스크립트 기초</title>
  <link rel="stylesheet" href="../index.css">
  <link rel="stylesheet" href="../01/01.css">
  <script src='../02.js'></script>
</head>
<body>
  <main id='container'>
    <header>
      <h1>자바스크립트 기초</h1>
      <p><a href='../index.html'>홈</a></p>
    </header>
    <article>
      <div id='msg'></div>
      <div id='btDiv'>
      </div>
    </article>
  </main>
</body>
</html>
```



.addEventListener()

• document.addEventListener()

- DOM(Document Object Model)에서 이벤트를 감지하고 처리하기 위해 사용되는 메서드
- DOM 요소에 이벤트 리스너를 추가하고, 해당 이벤트가 발생할 때 호출되는 콜백 함수를 지정
- .addEventListener(eventType, callbackFunction, options);
 - eventType: 감지하려는 이벤트의 종류를 나타내는 문자열
예) 'click', 'mousemove', 'DOMContentLoaded', 'change', 'input' 등
 - callbackFunction: 이벤트가 발생할 때 호출되는 함수 : 이벤트 객체를 매개변수로 받아 처리
 - options: 선택적 매개변수로, 이벤트 리스너의 동작을 구성하는 객체
 - capture, once, passive 등의 속성

```
sel1.addEventListener('change', (event)=>{  
    txtId1.innerHTML = event.target.value ;  
    txt1.value = '';  
    txt2.value = '';  
});
```



JavaScript DOM 제어

• DOM이 랜더링이 된 후 제어

- `document.addEventListener("DOMContentLoaded", function(){})`
- 단 한번만 실행

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>자바스크립트 기초</title>
  <link rel="stylesheet" href="../index.css">
  <link rel="stylesheet" href="../01/01.css">
  <script src='../02.js'></script>
</head>
<body>
  <main id='container'>
    <header>
      <h1>자바스크립트 기초</h1>
      <p><a href='../index.html'>홈</a></p>
    </header>
    <article>
      <div id='msg'></div>
      <div id='btDiv'>
        </div>
    </article>
  </main>
</body>
</html>
```



```
document.addEventListener('DOMContentLoaded', () => {
  console.log('DOM 완성') ;
  const msg = document.querySelector('#msg') ;
  const btDiv = document.getElementById('btDiv') ;

  //요소 추가
  const bt1 = document.createElement('button') ;
  const bt1Txt = document.createTextNode('확인') ;

  bt1.setAttribute('id', 'bt11') ;
  bt1.appendChild(bt1Txt) ;
  btDiv.appendChild(bt1) ;

  bt1.addEventListener('click', ()=>{
    if (msg.innerHTML == '')
      msg.innerHTML = '<h1>안녕하세요!!</h1>' ;
    else
      msg.innerHTML = '' ;
  });
});
```



DOM Read

- DOM 내의 특정 요소를 조회
- `document.getElementById(id)`
 - 주어진 문자열과 일치하는 id 속성을 가진 요소를 찾고, 이를 나타내는 Element 객체를 반환
- `document.querySelector(selectors)`
 - 제공한 선택자 또는 선택자 문치와 일치하는 문서 내 첫 번째 Element를 반환
- `document.querySelectorAll(selectors)`
 - 지정된 셀렉터 그룹에 일치하는 다큐먼트의 엘리먼트 리스트를 나타내는 정적(살아 있지 않은) NodeList 를 반환
 - 요소.forEach(함수)
- `document.getElementsByTagName(name)`
 - 주어진 태그명을 가진 요소를 찾고, 이를 나타내는 Element 객체를 반환

```
document.addEventListener('DOMContentLoaded', () => {  
  console.log('DOM 완성');  
  const msg = document.querySelector('#msg') ;  
  const btDiv = document.getElementById('btDiv') ;  
  
  //요소 추가  
  const bt1 = document.createElement('button') ;  
  const bt1Txt = document.createTextNode('확인');  
  
  bt1.setAttribute('id', 'bt11') ;  
  bt1.appendChild(bt1Txt) ;  
  btDiv.appendChild(bt1) ;  
  
  bt1.addEventListener('click', ()=>{  
    if (msg.innerHTML == '')  
      msg.innerHTML = '<h1>안녕하세요!!</h1>' ;  
    else  
      msg.innerHTML = '' ;  
  });  
});
```

HTML



CSS



JavaScript



React JS



DOM Create

- **createElement()**
 - document 내에 새로운 요소 생성
 - 생성된 요소는 변수에 저장
- **append()**
 - 새로 생성한 요소된 요소를 특정 노드에 연결
 - 여러 개의 노드 객체나 요소의 텍스트를 매개변수로 사용
- **appendChild()**
 - 선택한 요소 안에 자식요소를 추가
 - 단 하나의 노드 객체만 사용

```
document.addEventListener('DOMContentLoaded', () => {  
  console.log('DOM 완성') ;  
  const msg = document.querySelector('#msg') ;  
  const btDiv = document.getElementById('btDiv') ;  
  
  //요소 추가  
  const bt1 = document.createElement('button') ;  
  const bt1Txt = document.createTextNode('확인') ;  
  
  bt1.setAttribute('id', 'bt11') ;  
  bt1.appendChild(bt1Txt) ;  
  btDiv.appendChild(bt1) ;  
  
  bt1.addEventListener('click', ()=>{  
    if (msg.innerHTML == '')  
      msg.innerHTML = '<h1>안녕하세요!!</h1>' ;  
    else  
      msg.innerHTML = '' ;  
  });  
});
```

DOM Update

- 요소 값 변경
- innerHTML
 - 요소 내의 HTML 소스를 가져옴
- innerText
 - style이 적용된 웹 브라우저에서 보이는 텍스트를 가져옴
 - 만약 style.display = 'none'이면 값을 가져오지 못함
- textContent
 - style에 상관없이 요소의 텍스트를 가져옴

```
<article>
  <header><h1 id="jsH1">자바스크립트</h1></header>
  <div class="grid">
    <button class="bt1" id="btid1">1</button>
    <button class="bt2" id="btid2">2</button>
    <button class="bt1" id="btid3">3</button>
    <button class="bt2" id="btid4">4</button>
  </div>
</article>

<script>
document.addEventListener("DOMContentLoaded", () => {
  const bts = document.querySelectorAll("button");

  for(let i of bts) {
    i.innerHTML = "버튼" + i.textContent ;
  }
})
</script>
```

자바스크립트

버튼1

버튼2

버튼3

버튼4

JavaScript React JS



DOM 속성 다루기

• 속성 가져오기

- `element.getAttribute("속성")`

• 속성 수정하기

- `element.setAttribute("속성", 값)`

```
const show = (event) => {  
  event.preventDefault() ;  
  
  //랜덤수 생성  
  let n = Math.floor(Math.random() * 6) + 1 ;  
  document.querySelector(".h2Class > img").setAttribute("src", `./images/${n}.png`);  
}
```

```
document.addEventListener("DOMContentLoaded", ()=>{  
  //DOM요소 가져오기  
  const txt1 = document.querySelector("#txt1") ;  
  const txt2 = document.querySelector("#txt2") ;  
  
  const bts = document.querySelectorAll(".bt") ;  
  //버튼배열에 클릭이벤트 작성  
  for(let bt of bts) {  
    bt.addEventListener('click', (event)=>{  
      event.preventDefault();  
      let gubun = bt.getAttribute('id').slice(-1) ;  
      if (gubun === '1') palindrome(txt1, txt2) ;  
      else numSum(txt1, txt2) ;  
    });  
  }  
});
```

DOM Delete

- `remove()`
 - 요소 제거

```
<article>
  <header><h1 id="jsH1">자바스크립트</h1></header>
  <div class="grid">
    <button class="bt1" id="btid1">1</button>
    <button class="bt2" id="btid2">2</button>
    <button class="bt1" id="btid3">3</button>
    <button class="bt2" id="btid4">4</button>
  </div>
</article>
```



```
<script>
document.addEventListener("DOMContentLoaded", () => {
  document.querySelector("#btid4").remove();
})
</script>
```

자바스크립트

1

2

3



DOM CSS 제어

- style 속성
 - CSS 속성 값을 수정
- getComputedStyle()
 - CSS속성값을 가지고 옴
- setProperty(속성, 값)
 - style 값 설정하기
- getPropertyValue(속성)
 - style 값 가져오기
- item(인덱스)
 - style값 가져오기, 인자로 index를 사용
- removeProperty(속성)
 - style 값 삭제하기

```
const showGameOk = () => {  
  const bt1 = document.getElementById("bt1") ;  
  const bt2 = document.getElementById("bt2") ;  
  const inNum = document.getElementById("inNum") ;  
  const dice = document.getElementById("dice") ;  
  
  //주사위 이미지 숨기기  
  dice.style.display = 'block' ;  
  
  //주사위 버튼 보이기  
  bt1.style.display = 'block' ;  
  
  //확인 버튼 숨기기  
  bt2.style.display = 'none';  
  
  //번호 선택 보이기  
  inNum.style.display = 'none';  
}
```

해결문제

- 주사위 버튼을 누르면 임의의 주사위 번호가 나오도록 작성

자바스크립트

w3schools HTML

w3schools JS

주사위게임



주사위를 던져주세요.



해결문제1

- 랜덤으로 생성된 주사위 번호를 맞추는 게임을 작성

자바스크립트

w3schools HTML

w3schools JS

주사위게임

Computer



User



1 선택

2 선택

3 선택

4 선택

5 선택

6 선택

틀림

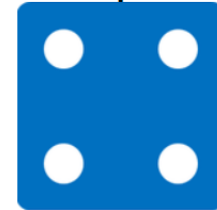
자바스크립트

w3schools HTML

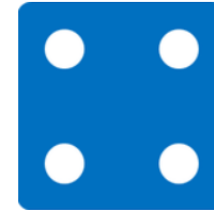
w3schools JS

주사위게임

Computer



User



1 선택

2 선택

3 선택

4 선택

5 선택

6 선택

맞춤



해결문제1 분석

```
/*  
  1. DOM에서 제어할 노드를 가져 오기  
  2. 6개 버튼 클릭을 확인  
  3. 버튼 클릭이 되면  
      - 해당하는 버튼의 숫자를 추출 => 사용자 선택 숫자(user) => 이미지 변경  
      - 랜덤수를 생성 => 컴퓨터 숫자(com) => 이미지 변경  
      - 사용자 선택수(user)와 컴퓨터 선택수(com)을 비교  
      - 결과 출력  
*/
```

```
> document.addEventListener('DOMContentLoaded', ()=>{ ...  
});
```

해결문제1 분석

```
<article>
  <div>주사위게임</div>
  <div id='btDiv4'>
    <section>
      <p>
        <span>Computer</span>
        <img id="com" src='../03/img/1.png' />
      </p>
      <p>
        <span>User</span>
        <img id="user" src='../03/img/1.png' />
      </p>
    </section>
    <section>
      <button class='bt4'>1선택</button>
      <button class='bt4'>2선택</button>
      <button class='bt4'>3선택</button>
      <button class='bt4'>4선택</button>
      <button class='bt4'>5선택</button>
      <button class='bt4'>6선택</button>
    </section>
  </div>
  <div id='msg'>결과</div>
</article>
```

//1. DOM에서 제어할 노드를 가져 오기

```
const comImg = document.getElementById('com') ;
const userImg = document.querySelector('#user');
const msg = document.querySelector('#msg');
```

```
const bts = document.querySelectorAll('.bt4') ;
```

해결문제1 분석

```
<article>
  <div>주사위게임</div>
  <div id='btDiv4'>
    <section>
      <p>
        <span>Computer</span>
        <img id="com" src='../03/img/1.png' />
      </p>
      <p>
        <span>User</span>
        <img id="user" src='../03/img/1.png' />
      </p>
    </section>
    <section>
      <button class='bt4'>1선택</button>
      <button class='bt4'>2선택</button>
      <button class='bt4'>3선택</button>
      <button class='bt4'>4선택</button>
      <button class='bt4'>5선택</button>
      <button class='bt4'>6선택</button>
    </section>
  </div>
  <div id='msg'>결과</div>
</article>
```

```
//2. 6개 버튼 클릭을 확인
for (let bt of bts) {
  bt.addEventListener('click', ()=>{
    // 3. 버튼 클릭이 되면
    // - 해당하는 버튼의 숫자를 추출 => 사용자 선택 숫자(user) => 이미지 변경
    // - 랜덤수를 생성 => 컴퓨터 숫자(com) => 이미지 변경
    // - 사용자 선택수(user)와 컴퓨터 선택수(com)을 비교
    // - 결과 출력

    const user = parseInt(bt.textContent.slice(0,1));
    // const user = bt.textContent.slice(0,1);
    const com = Math.floor(Math.random() * 6) + 1 ; //1~6

    comImg.setAttribute('src', `../03/img/${com}.png`) ;
    comImg.setAttribute('alt', `computer num ${com}`) ;

    userImg.setAttribute('src', `../03/img/${user}.png`) ;
    userImg.setAttribute('alt', `user num ${user}`) ;

    if (com === user) msg.innerHTML = '맞음' ;
    else msg.innerHTML = '틀림' ;

  });
}
```


해결문제2

자바스크립트

w3schools HTML

w3schools JS

단위환산

섭씨온도(°C) ▼

화씨온도(°F) ▼

°C

°F



해결문제2 분석

• <form> 태그

- HTML 문서에서 사용자 입력을 수집하고 서버로 전송하기 위한 양식을 정의하는 데 사용
- 웹 애플리케이션에서 사용자와 상호작용하는 중요한 요소로, 다양한 입력 필드를 포함

• 폼 요소

- <input>
 - 가장 많이 사용되는 폼 요소로 텍스트, 비밀번호, 체크박스, 라디오 버튼, 파일 등 다양한 유형의 입력
- <select>
 - 드롭다운 목록을 생성하여 사용자가 선택
- <button>
 - 폼을 제출하는 버튼이나 다른 동작을 실행하는 버튼

• 참고

- https://www.w3schools.com/html/html_forms.asp

```
<form name='myform'>
  <section>
    <h1>자바스크립트 단위환산</h1>
    <div>
      <div class='row'>
        <div>
          <select name="sel1" id="sel1">
            <option value="°C" selected>섭씨온도(°C)</option>
            <option value="°F">화씨온도(°F)</option>
          </select>
        </div>
        <div>
          <select name="sel2" id="sel2">
            <option value="°C">섭씨온도(°C)</option>
            <option value="°F" selected>화씨온도(°F)</option>
          </select>
        </div>
      </div>
      <div class='row'>
        <div>
          <input type="number" id="txt1">
          <label for="txt1">°C</label>
        </div>
        <div>
          <input type="number" id="txt2" readonly>
          <label for="txt2">°F</label>
        </div>
      </div>
    </div>
  </section>
</form>
```



해결문제2 분석 - input 요소 이벤트 리스너

- **input 이벤트**

- 사용자가 input 요소의 값을 변경할 때마다 발생하는 이벤트로 텍스트를 입력할 때마다, 또는 값을 변경하는 즉시 이벤트가 발생

- **change 이벤트**

- 사용자가 input 요소의 값을 변경하고 포커스를 잃을 때 발생하는 이벤트로 input과 달리 값이 바뀌었지만 아직 포커스를 잃지 않은 상태에서는 발생하지 않음

- **focus 이벤트**

- input 요소가 포커스를 얻을 때 발생하는 이벤트로 input 필드를 클릭하거나 탭을 통해 접근할 때 발생

- **keydown 이벤트**

- 사용자가 키보드를 눌렀을 때 발생하는 이벤트로 어떤 키가 눌렸는지 감지할 수 있음

- **keyup 이벤트**

- 사용자가 키보드를 눌렀다가 떼를 때 발생하는 이벤트로 keydown 이벤트와 함께 사용되며, 키를 떼 후의 상태를 감지

```
txt1.addEventListener("input", ()=>{  
  console.log(txt1.value)  
  if (sel1.value === "°C") {  
    //섭씨온도를 화씨온도로  
    txt2.value = (parseFloat(txt1.value) * (9 / 5)) + 32 ;  
  }  
  else {  
    //화씨온도를 섭씨온도로 변경  
    txt2.value = (parseFloat(txt1.value) - 32) * (5/9) ;  
  }  
});
```

$$^{\circ}\text{F} = (^{\circ}\text{C} \times \frac{9}{5}) + 32$$

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times \frac{5}{9}$$

해결문제3

자바스크립트

w3schools HTML

w3schools JS

업다운게임

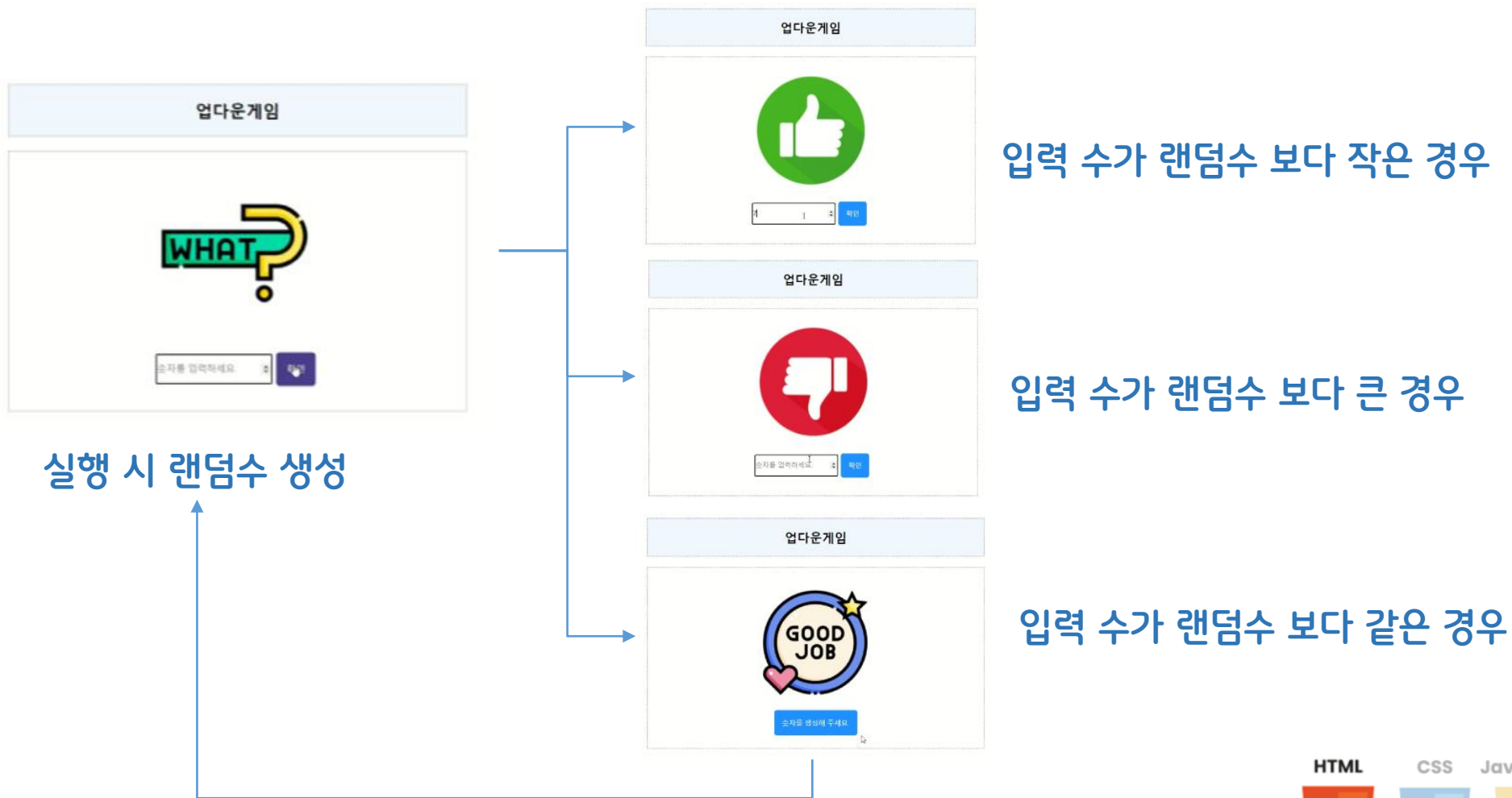


숫자를 입력하세요.

확인



해결문제3 분석



해결문제3 분석

- 랜덤 수를 맞출 때까지 랜덤수를 다시 생성하지 않도록 flag 변수 사용
- `e.preventDefault()`;
 - 자바스크립트에서 이벤트(event)의 기본 동작을 막기 위해 사용되는 메서드
 - HTML 요소의 기본 동작을 취소할 수 있는 강력한 도구로, 웹 페이지의 동작을 보다 유연하게 제어
- `parseInt()` 함수
 - 자바스크립트에서 문자열을 정수로 변환하는데 사용되는 함수

```
//랜덤수 생성 변수
let n ;
let flag = false ;

//버튼의 클릭이벤트 달기
bt.addEventListener("click", (e)=>{
  e.preventDefault();
  //input 박스 내용가져오기
  if (numInput.value == '') { ...
  }

  if (numInput.value < 1 || numInput.value > 100) { ...
  }

  //랜덤수 생성
  // if (flag === false) {
  if (!flag) { ...
  }

  //숫자비교
  msg.innerHTML = "" ;
  if (n === parseInt(numInput.value)) { ...
  }
  else if ( n > numInput.value) {
    upDownImg.setAttribute("src", "./img/up.png") ;
  }
  else {
    upDownImg.setAttribute("src", "./img/down.png") ;
  }
});
```

