■ DOM (Document Object Model)

- 웹 브라우저가 HTML을 인식하는 방식 (넓은 의미)
- document 객체와 관련된 객체의 집합 (좁은 의미)
- HTML의 Tag를 자바스크립트에서 이용할 수 있도록 객체로 만든 것
- **Element Node** / Text Node (Text Content)

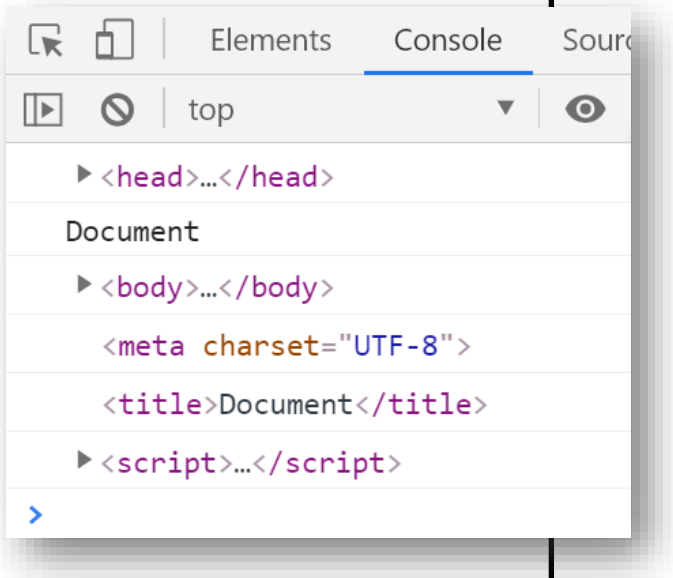
```
<body>

  <h1 title='Header'>
    Text Node
  </h1>

  <p id='content'>
    Text Node
  </p>

</body>
```

■ DOM (Document Object Model)

● 문서 객체 가져오기

- document.head / document.body / document.title

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <script>
    console.log(document.head);
    console.log(document.title);
    console.log(document.body);

    console.log(document.head.childNodes[1]);
    console.log(document.head.childNodes[3]);

    console.log(document.body.childNodes[1])
  </script>
</body>
</html>
```



■ DOM (Document Object Model)

● 자바스크립트로 HTML 내용 작성

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
  <script>
    const h1 = (text) => `<h1>${text}</h1>`;
  </script>
  <script>
    document.body.innerHTML += h1('1번째 script 태그');
  </script>
</head>
<body>
  <script>
    document.body.innerHTML += h1('2번째 script 태그');
  </script>
  <h1>1번째 h1 태그</h1>
  <script>
    document.body.innerHTML += h1('3번째 script 태그');
  </script>
  <h1>2번째 h2 태그</h1>
</body>
</html>
```

실행되지 않음
(body 생성 전)

2번째 script 태그

1번째 h1 태그

3번째 script 태그

2번째 h2 태그

■ DOM (Document Object Model)

- 웹 브라우저가 문서 객체를 모두 읽고 난 후 동작

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
  <script>
    const h1 = (text) => `<h1>${text}</h1>`;
  </script>
  <script>
    document.addEventListener('DOMContentLoaded', () => {
      document.body.innerHTML += h1('h1 태그 추가');
    });
  </script>
</head>
<body>
  <h1>h1 태그</h1>
</body>
</html>
```

h1 태그

h1 태그 추가

■ DOM (Document Object Model)

● 문서 객체 가져오기

- head / body 요소 내부에 만든 다른 요소 가져오기

```
document.querySelector('CSS 선택자')
document.querySelectorAll('CSS 선택자')
```

※ 자주 사용되는 CSS 선택자

이름	선택자 형태	설명
태그 선택자	태그	지정 태그를 가진 객체 조회
아이디 선택자	#아이디	지정 아이디를 가진 객체 조회
클래스 선택자	.클래스	지정 클래스를 가진 객체 조회
속성 선택자	[속성=값]	지정 속성 값을 가진 객체 조회
후손 선택자	선택자1 선택자2 선택자1 > 선택자2	선택자1 하위의 선택자2를 가진 객체 조회

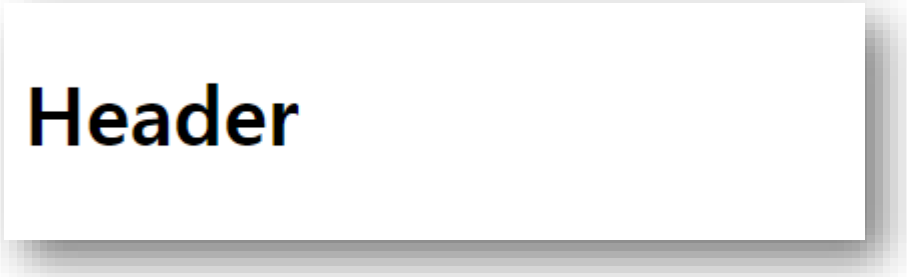
■ DOM (Document Object Model)

● 문서 객체 생성

```
document.createElement('태그명')
```

- h1 태그 생성 후 "Header" 텍스트 추가하여 body에 적용

```
<script>
  const obj = document.createElement('h1')
  obj.textContent = 'Header'

  // const text = document.createTextNode('Header');
  // obj.appendChild(text);

  document.body.appendChild(obj);
</script>
```

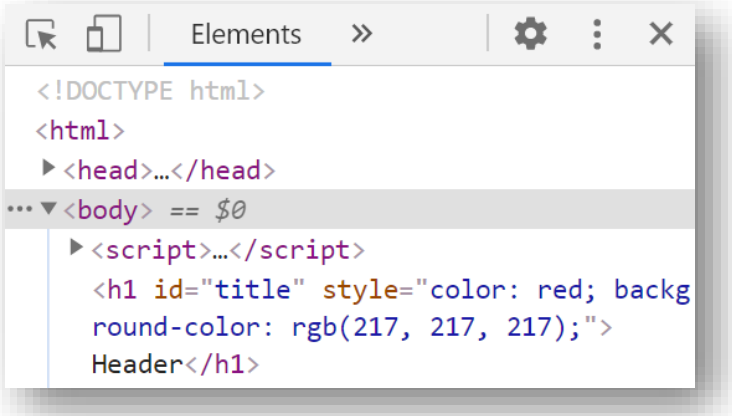


■ DOM (Document Object Model)

● 문서 객체 속성 지정

- h1 태그 생성 후 "Header" 텍스트 추가하여 body에 적용

```
<script>
  const obj = document.createElement('h1')
  obj.textContent = 'Header'
  document.body.appendChild(obj);

  obj.style.color = 'red';
  obj.style.backgroundColor = '#d9d9d9';
  obj.setAttribute('id', 'title');
</script>
```



■ DOM (Document Object Model)

● 문서 객체 제거

```
부모객체.removeChild(자식객체)
문서객체.parentNode.removeChild(문서객체)
```

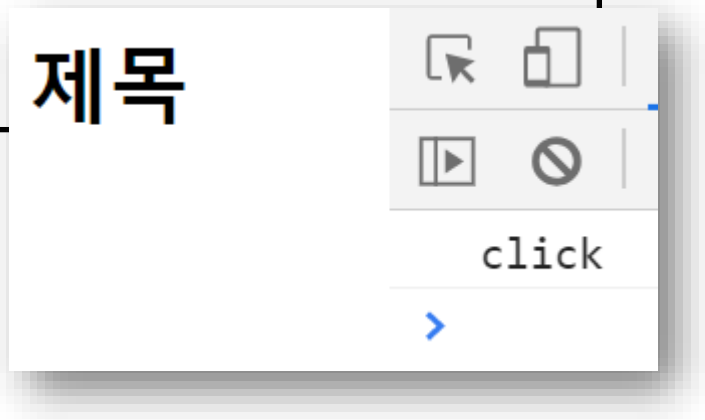
```
<script>
  const obj = document.querySelector('h1');
  document.body.removeChild(obj);

  // obj.parentNode.removeChild(obj);
</script>
```

■ DOM (Document Object Model)

● 이벤트 사용

```
문서객체.addEventListener('이벤트명', 콜백함수)
문서객체.removeEventListener('이벤트명', 콜백함수)
```

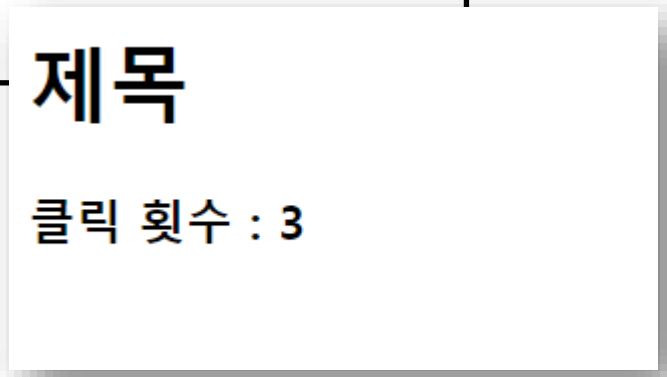
```
<body>
  <h1>제목</h1>
</body>

<script>
  const h1 = document.querySelector('h1');
  h1.addEventListener('click', function() {
    console.log('click');
  });
</script>
```



■ 연습문제

● 제목을 클릭하면 클릭 횟수 증가 후 출력하기

```
<body>
  <h1>제목</h1>
  <h3>클릭 횟수 : 0</h3>
</body>

<script>
  let count = 0;









</script>
```

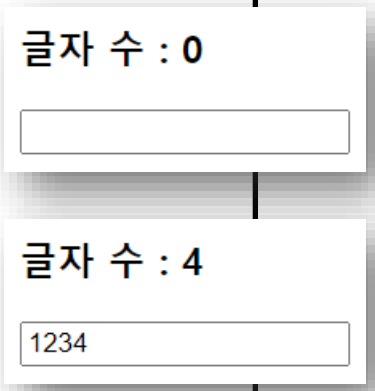


■ DOM (Document Object Model)

● 키보드 이벤트

이벤트	설명
keydown	키가 눌릴 때 실행
keypress	키가 입력되었을 때 실행
keyup	눌렀던 키를 떨어질 때 실행

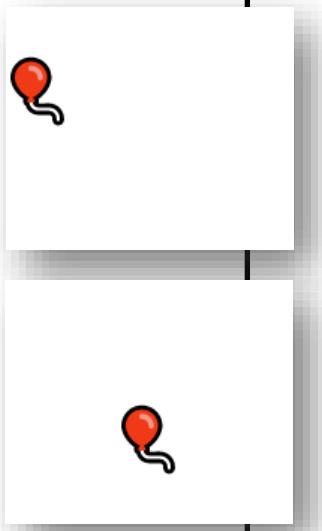
```
<body>
  <h3>글자 수 : 0</h3>
  <input type="text" name="title" id="title">
</body>

<script>
  const h3 = document.querySelector('h3');
  const input = document.querySelector('input');
  input.addEventListener('keyup', function(event) {
    const length = input.value.length;
    h3.textContent = `글자 수 : ${length}`;
  });
</script>
```



■ DOM (Document Object Model)

● 키보드로 풍선 이모지(🎈) 이동하기

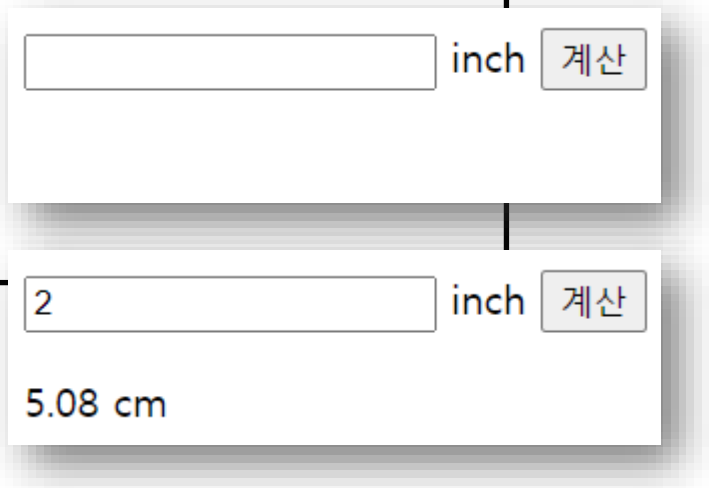
```
<body>
  <h1>🎈</h1>
</body>

<script>
    const star = document.querySelector('h1');
    star.style.position = 'absolute';
    let [x, y] = [0, 0];
    const block = 20;
    const print = () => {
      star.style.left = `${x * block}px`
      star.style.top = `${y * block}px`
    };
    print();

    const [left, up, right, down] = [37, 38, 39, 40];
    document.body.addEventListener('keydown', (event) => {
      switch (event.keyCode) {
        case left : x -= 1; break;
        case up   : y -= 1; break;
        case right: x += 1; break;
        case down : y += 1; break;
      }
      print();
    });
</script>
```



■ 연습문제

- inch를 cm로 변환하는 기능 구현하기

```
<body>
  <input type="text"> inch
  <button>계산</button>
  <p></p>
</body>

<script>
  // 변환 공식 : inch * 2.54 => cm




</script>
```



■ 연습문제

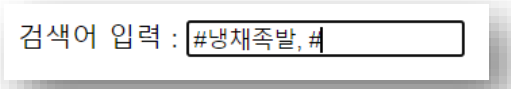
● 검색용 해시태그 입력을 위한 사용자 편의 기능 추가하기

```
검색어 입력 : <input type="text" name="search">

<script>
  const input = document.querySelector('[name=search]');
  input.addEventListener('focus', (event) => {
    let search = event.currentTarget.value;



  });



</script>
```

최초 실행

input 선택 (포커스)

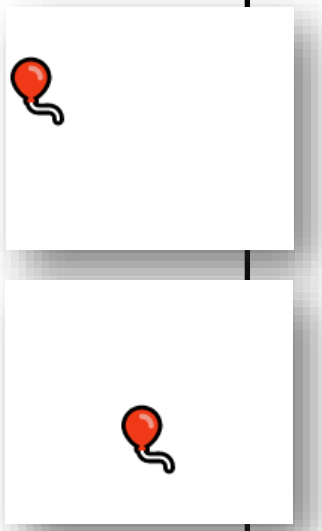
space 키 입력



■ DOM (Document Object Model)

● 드롭다운(select) 목록 활용하기

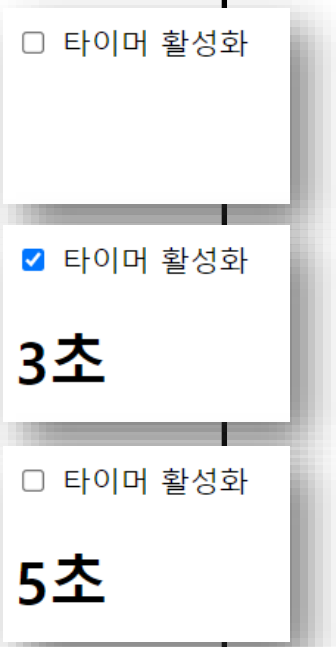
```
<body>
  <select>
    <option>떡볶이</option>
    <option>순대</option>
    <option>오뎅</option>
    <option>튀김</option>
  </select>
  <p>선택: 떡볶이</p>
</body>

<script>
  const select = document.querySelector('select');
  const p = document.querySelector('p');

  select.addEventListener('change', (event) => {
    const options = event.currentTarget.options;
    const index = event.currentTarget.options.selectedIndex;

    p.textContent = `선택: ${options[index].textContent}`
  });
</script>
```



■ DOM (Document Object Model)

● 체크박스(input[type="checkbox"]) 활용하기

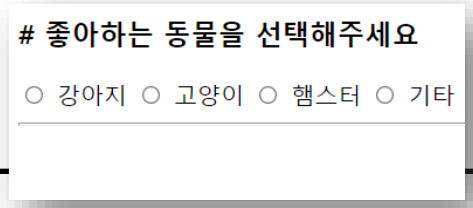
```
<body>
  <input type="checkbox">
  <span>타이머 활성화</span>
  <h1></h1>
</body>

<script>
    let [timer, timerId] = [0, 0];
    const h1 = document.querySelector('h1');
    const checkbox = document.querySelector('input');
    checkbox.addEventListener('change', (event) => {
      if (event.currentTarget.checked) {
        timerId = setInterval(() => {
          timer += 1;
          h1.textContent = `${timer}초`;
        }, 1000);
      } else {
        clearInterval(timerId);
      }
    });
</script>
```

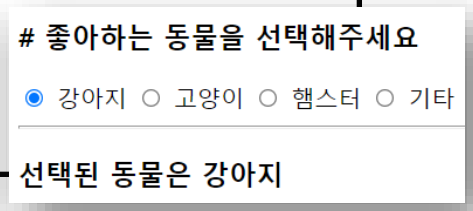


■ DOM (Document Object Model)

● 라디오버튼(input[type="radio"]) 활용하기

```
<body>
  <h3># 좋아하는 동물을 선택해주세요</h3>
  <input type="radio" name="pet" value="강아지"><span>강아지</span>
  <input type="radio" name="pet" value="고양이"><span>고양이</span>
  <input type="radio" name="pet" value="햄스터"><span>햄스터</span>
  <input type="radio" name="pet" value="기타"><span>기타</span>
  <hr>
  <h3 id="output"></h3>
</body>

<script>
    const output = document.querySelector('#output');
    const radios = document.querySelectorAll('[name=pet]');

    radios.forEach((radio) => {
      radio.addEventListener('change', (event) => {
        const current = event.currentTarget;
        if (current.checked) {
          output.textContent = `선택된 동물은 ${current.value}`;
        }
      });
    });
</script>
```





■ DOM (Document Object Model)

● 기본 이벤트 막기 - preventDefault()

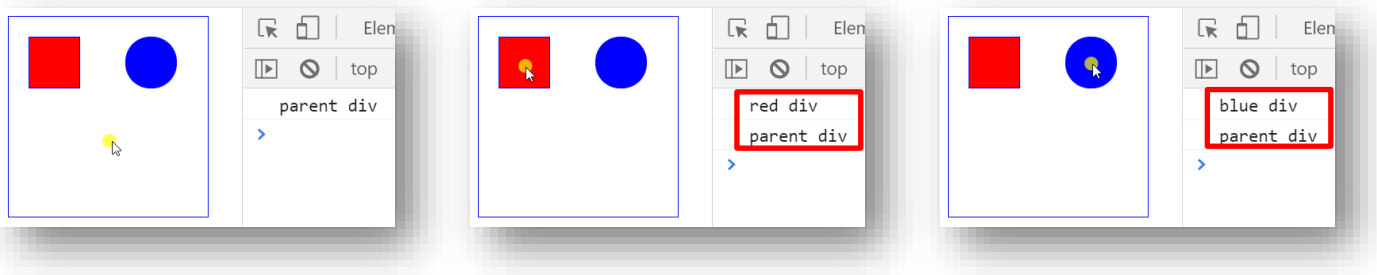
```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    const imgs = document.querySelectorAll('img')
    imgs.forEach((img) => {
      img.addEventListener('contextmenu', (event) => {
        event.preventDefault()
      })
    })
  })
</script>
<body>
  <img src="http://placekitten.com/300/300" alt="">
</body>
```

마우스 우클릭 이벤트

■ DOM (Document Object Model)

● 이벤트 전달 막기

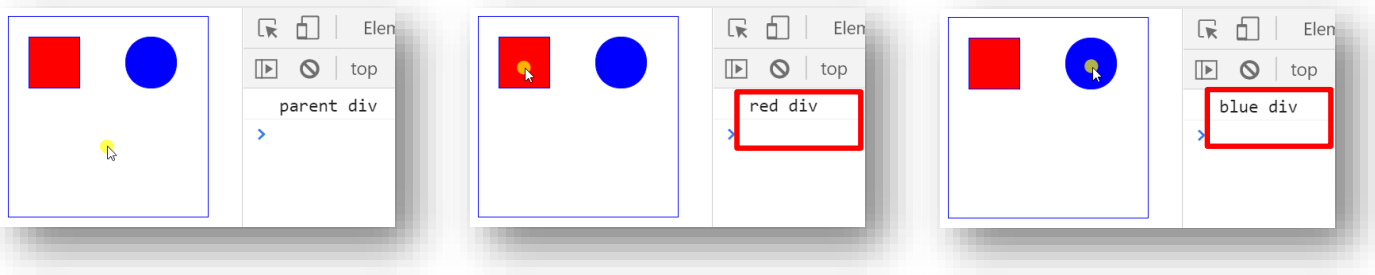
```
<style>
  div {
    border: 1px solid blue; display: inline-block;
  }
  div > div {
    width: 50px; height: 50px; margin: 20px;
  }
</style>
<div style='width: 200px; height: 200px;'>
  <div style='background-color: red;'></div>
  <div style='background-color: blue; border-radius: 25px;'></div>
</div>
<script>
  document.querySelector('div > div:first-child').addEventListener('click', (event) => {
    console.log('red div');
  });
  document.querySelector('div > div:last-child').addEventListener('click', (event) => {
    console.log('blue div');
  });
  document.querySelector('body > div').addEventListener('click', (event) => {
    console.log('parent div');
  });
</script>
```



■ DOM (Document Object Model)

● 이벤트 전달 막기 - stopPropagation()

```
<style>
  div {
    border: 1px solid blue; display: inline-block;
  }
  div > div {
    width: 50px; height: 50px; margin: 20px;
  }
</style>
<div style='width: 200px; height: 200px;'>
  <div style='background-color: red;'></div>
  <div style='background-color: blue; border-radius: 25px;'></div>
</div>
<script>
  document.querySelector('div > div:first-child').addEventListener('click', (event) => {
    console.log('red div'); event.stopPropagation();
  });
  document.querySelector('div > div:last-child').addEventListener('click', (event) => {
    console.log('blue div'); event.stopPropagation();
  });
  document.querySelector('body > div').addEventListener('click', (event) => {
    console.log('parent div'); event.stopPropagation();
  });
</script>
```



■ 연습문제

- "이동" 링크를 클릭하면 아래와 같은 경고창을 보여주고 웹사이트 이동 막기

```
<a href='http://ggoreb.com'>이동</a>

<script>




</script>
```