



DoiT Web Project Team

7. JavaScript: Interaction

User Interface

- 사용자와 맞닿는 요소.
- 자바스크립트로 UI를 조작할 수 있다.

🤖 그래서 어떻게 할 수 있나요?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page</title>
  </head>
  <body>
    <div id="button">Button</div>
    <script src="index.js"></script>
  </body>
</html>
```

```
var el = document.getElementById('button');

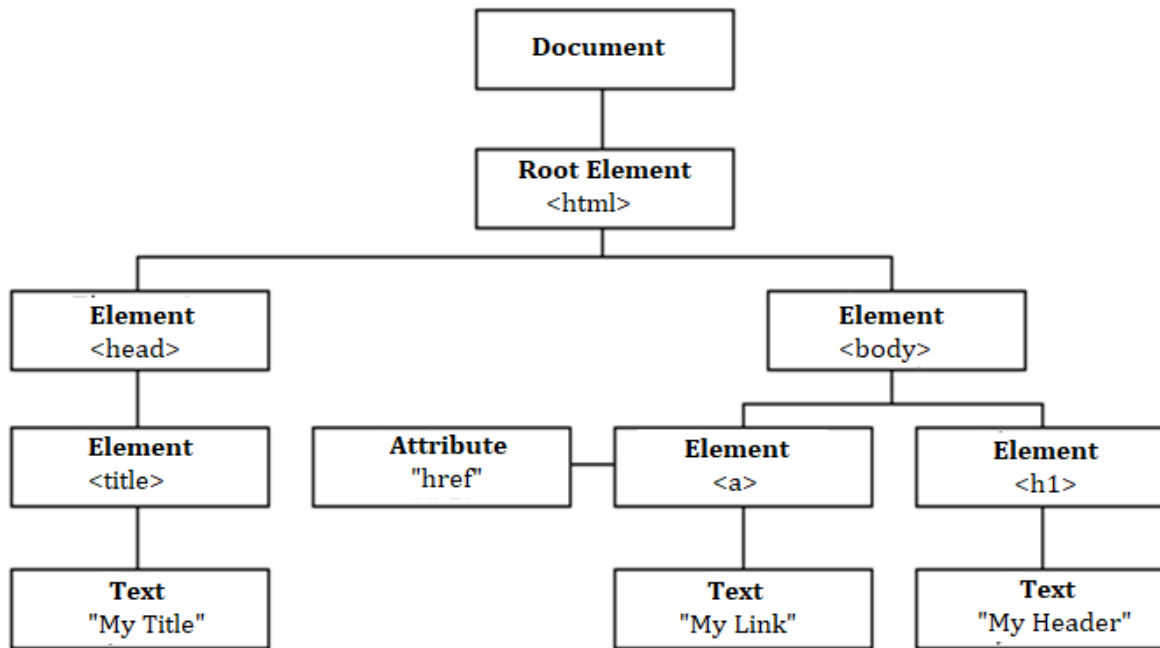
el.addEventListener('click', function() {
  alert('clicked');
});
```

요소에 접근하기

```
var el = document.getElementById('button');
```

- DOM에서 id가 `button` 인 요소를 가져온다.

DOM (Document Object Model)



- 브라우저가 만드는 문서 모델.
- `document` 객체를 통해 요소에 접근할 수 있다.

📁 특정 요소에 접근하기

```
// <div id="button">Button</div>  
document.getElementById('button');
```

- `button` 이라는 id를 가진 첫번째 요소.

```
// <div id="button">Button</div>  
document.querySelector('#button');
```

```
// <div class="box">Box</div>  
document.querySelector('.box');
```

- `box` 라는 class를 가진 첫번째 요소.
- CSS selector를 사용한다.

📁 여러 요소에 접근하기

```
// <div class="box">Box</div>  
// <div class="box">Box</div>  
document.getElementsByClassName('box');
```

- `box` 라는 class를 가진 모든 요소.

```
// <div>Section A</div>  
// <div>Section B</div>  
document.getElementsByTagName('div');
```

- 모든 `div` 요소.

```
// <div class="box">Box</div>  
// <div class="box">Box</div>  
document.querySelectorAll('.box');
```

- `box` 라는 class를 가진 모든 요소.

📁 여러 요소에 접근하기

NodeList 객체

```
document.getElementsByClassName('box');  
document.getElementsByTagName('div');  
document.querySelectorAll('.box');
```

- 여러 요소를 가져오는 메소드들은 **NodeList** 객체를 반환한다.

📁 여러 요소에 접근하기

NodeList 객체

```
// <div class="box">Box</div>
// <div class="box">Box</div>

var boxes = document.getElementsByClassName('box');

for (var i = 0; i < boxes.length; i += 1) {
  boxes[i].className = 'section';
  boxes[i].textContent = 'Section #' + i;
}

// <div class="section">Section #0</div>
// <div class="section">Section #1</div>
```

- 여러 요소를 가져오는 메소드들은 NodeList 객체를 반환한다.
- NodeList 객체는 iterable하다.

DOM 조작하기

```
<body>
  <ul id="list">
  </ul>
</body>
```

```
var el = document.querySelector('#list');

for (var i = 0; i < 10; i += 1) {
  el.innerHTML += '<li>Item #' + i + '</li>';
}
```

- 반복 작업을 자동화하거나,
- 동적으로 콘텐츠를 구성할 수 있다.

이벤트

```
var el = document.querySelector('button');  
  
el.addEventListener('click', function() {  
  alert('clicked');  
});
```

- 요소 객체에 **이벤트 리스너**를 등록할 수 있다.
- 이벤트가 발생하면 콜백 함수의 내용을 실행한다.