

2024년 상반기 K-디지털 트레이닝

문서 객체 모델

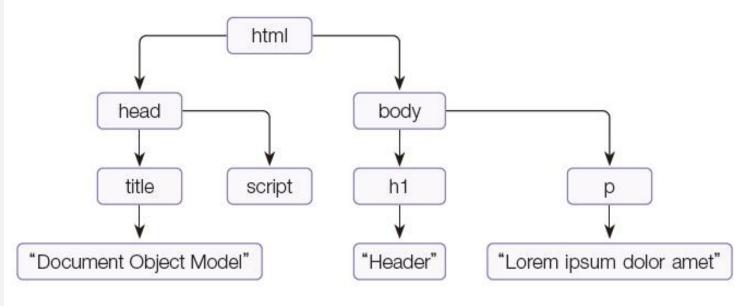
[KB] IT's Your Life



문서 객체 모델의 기본 용어와 개념

☑ 문서 객체 모델 기본 용어

- o 문서 객체(Document Object)
 - HTML 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것.
- o HTML 문서의 요소는 부모-자식 관계를 가짐 --> DOM Tree로 표현
 - 요소노드와 텍스트 노드



1 문서 객체 모델의 기본 용어와 개념

☑ 문서 객체 모델 기본 용어

ㅇ 텍스트 노드를 갖지 않는 태그

```
<br><hr><img src="rintiantta.png">
```

- o 정적 생성: 웹페이지를 처음 실행할 때 HTML 태그로 적힌 문서 객체를 생성.
- o 동적 생성: 페이지를 실행 중에 자바스크립트를 사용해 문서 객체를 생성.
- o 객체 모델: 웹 브라우저가 HTML 파일을 분석하고 출력하는 방식

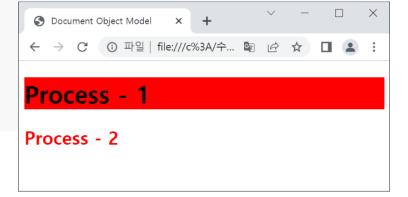
☑ dom_fault.html 실행 순서에 따른 문서 객체 사용 오류

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>
         // h1 태그의 배경 색상을 변경합니다.
         document.querySelector('h1').style.backgroundColor = 'red';
         // h2 태그의 글자 색상을 변경합니다.
         document.querySelector('h2').style.color = 'red';
    </script>
</head>
<body>
    <h1>Process - 1</h1>
                                                                     DevTools - file:///c%3A/%EC%88%98%EC%97%85/chapter_10/dom_fault.html
    <h2>Process - 2</h2>
                                                                      Elements Console Sources Network Performance Memory >>>
                                                                            </body>
                                                                                                             Default levels ▼ No Issues 🔯
                                                                      ❸ Uncaught TypeError: Cannot read properties of null (reading 'style') dom fault.html:7
</html>
                                                                          at dom fault.html:7:33
```

문서 객체 모델의 기본 용어와 개념

☑ dom_correct.html 해결방법1: script 태그 위치 옮기기

```
<!DOCTYPE html>
<html>
<head>
   <title>Document Object Model</title>
</head>
<body>
   <h1>Process - 1</h1>
   <h2>Process - 2</h2>
   <script>
       // h1 태그의 배경 색상을 변경합니다.
       document.querySelector('h1').style.backgroundColor = 'red';
       // h2 태그의 글자 색상을 변경합니다.
       document.querySelector('h2').style.color = 'red';
   </script>
</body>
</html>
```



문서 객체 모델의 기본 용어와 개념

dom_event.html

해결 방법 2: 이벤트 활용하기

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>
        window.onload = function () {
            // h1 태그의 배경 색상을 변경합니다.
            document.querySelector('h1').style.backgroundColor = 'red';
            // h2 태그의 글자 색상을 변경합니다.
            document.querySelector('h2').style.color = 'red';
        };
    </script>
</head>
<body>
    <h1>Process - 1</h1>
    \langle h2 \rangle Process - 2 \langle /h2 \rangle
</body>
</html>
```

2 <mark>문서 객체 선택</mark>

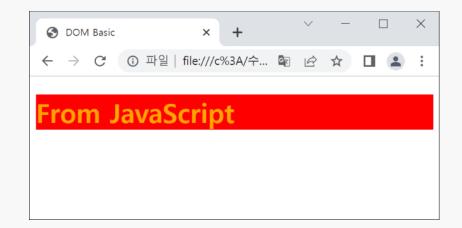
☑ 문서 객체 선택

- o HTML 태그를 자바스크립트에서 문서 객체로 변환
- o 문서 객체를 선택하는 메서드

구분	메서드	설명
1개 선택	document.getElementById(아니다)	아이디로 1개 선택
	document,querySelector(선택자)	선택자로 1개 선택
여러 개 선택	document.getElementsByName(이름)	name 속성으로 여러 개 선택
	document.getElementsByClassName(클래스)	class 속성으로 여러 개 선택
	document.querySelectorAll(선택자)	선택자로 여러 개 선택

🗹 select_id.html getElementById() 메서드를 사용해 문서 객체 1개 선택하기

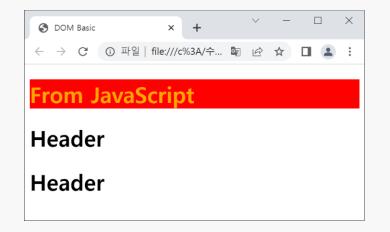
```
<!DOCTYPE html>
<html>
<head>
   <title>DOM Basic</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 문서 객체를 선택합니다.
           let header = document.getElementById('header');
           // 문서 객체를 조작합니다.
           header.style.color = 'orange';
           header.style.background = 'red';
           header.innerHTML = 'From JavaScript';
   </script>
</head>
<body>
   <h1 id="header">Header</h1>
</body>
</html>
```



select_query.html

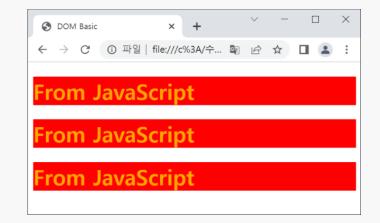
querySelector() 메서드를 사용해 문서 객체 1개 선택하기

```
<head>
   <title>DOM Basic</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 문서 객체를 선택합니다.
           let header = document.querySelector('h1');
           // 문서 객체를 조작합니다.
           header.style.color = 'orange';
           header.style.background = 'red';
           header.innerHTML = 'From JavaScript';
       };
   </script>
</head>
<body>
   <h1>Header</h1>
   <h1>Header</h1>
   <h1>Header</h1>
</body>
```



☑ select_all.html querySelectorAll() 메서드를 사용해 문서 객체 여러 개 선택하기

```
<script>
   // 이벤트를 연결합니다.
   window.onload = function () {
       // 문서 객체를 선택합니다.
       let headers = document.querySelectorAll('h1');
       for (let i = 0; i < headers.length; i++) {
           // 변수를 선언합니다.
           let header = headers[i];
           // 문서 객체를 조작합니다.
           header.style.color = 'orange';
           header.style.background = 'red';
           header.innerHTML = 'From JavaScript';
</script>
</head>
<body>
   <h1>Header</h1>
   <h1>Header</h1>
   <h1>Header</h1>
</body>
```



🥝 글자 조작

o 글자 속성

속성	설명
textContent	문서 객체 내부 글자를 순수 텍스트 형식으로 가져오도록 변경
innerHTML	문서 객체 내부 글자의 HTML 태그를 반영해 가져오도록 변경

문서 객체 조작

☑ control_text.html 내부 글자 변경

```
OM Basic
<head>
                                                                 ① 파일 | file:///c%3A/수업/chapt...
    <title>DOM Basic</title>
    <script>
                                                       <h1>Header - 0</h1><h1>Header - 1</h1><h1>Header - 2</h1>
        // 이벤트를 연결합니다.
                                                       <h1>Header - 3</h1><h1>Header - 4</h1><h1>Header - 5</h1>
                                                       <h1>Header - 6</h1><h1>Header - 7</h1><h1>Header - 8</h1>
        window.onload = function () {
                                                       <h1>Header - 9</h1>
             // 변수를 선언합니다.
             let output = '';
                                                                                            ← → C ① 파일 | file:///c%3A/수... 🔄 🖻 🗘 🔲 🚨
             for (let i = 0; i < 10; i++) {
                                                                                           Header - 0
                  output += '<h1>Header - ' + i + '</h1>/
                                                                                           Header - 1
                                                                                           Header - 2
             // 문서 객체 내부의 글자를 변경합니다/
                                                                                           Header - 3
             document.body.textContent = output;
             // innerHTML 속성을 사용합니다.
                                                                                           Header - 4
             // document.body.innerHTML = output;-
                                                                                           Header - 5
        };
                                                                                           Header - 6
    </script>
</head>
                                                                                           Header - 7
<body>
                                                                                           Header - 8
                                                                                           Header - 9
</body>
```

◎ 스타일 조작

o 자바스크립트로 CSS 속성 값을 추가·제거·변경

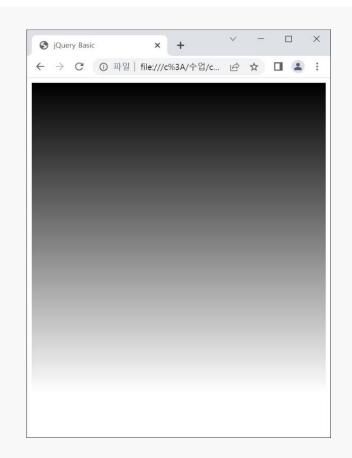
```
1 let header = document.getElementById('header');
  header.style.backgroundColor-color = 'red';
2 let header = document.getElementById('header');
  header.style.backgroundColor = 'red';
```

스타일시트의 스타일 속성	자바스크립트의 스타일 식별자	
background-image	backgroundlmage	
background-color	backgroundColor	
box-sizing	boxSizing	
list-style	listStyle	

문서 객체 조작

☑ control_style.html 스타일 조작

```
<script>
       // 이벤트를 연결합니다.
       window.onload = function () {
          // 문서 객체를 추가합니다.
          let output = '';
          for (let i = 0; i < 256; i++) {
              output += '<div></div>';
          document.body.innerHTML = output;
          // 문서 객체를 선택합니다.
          let divs = document.querySelectorAll('div');
          for (let i = 0; i < divs.length; i++) {
              // 변수를 선언합니다.
              let div = divs[i];
              // 스타일을 적용합니다.
              div.style.height = '2px';
              div.style.background = 'rgb(' + i + ', ' + i + ', ' + i + ')';
       };
   </script>
</head>
<body></body>
```



🤨 속성 조작

ㅇ 문서 객체의 속성 조작 메서드

메서드	설명
setAttribute(속성 이름, 속성 값)	속성 지정
getAttribute(속성 이름)	속성 추출

- o 웹 브라우저 제조사가 구현하지 않았다면 접근하지 못할 수 있음.
- o 'data-'로 시작하는 사용자 지정 속성은 반드시 속성 조작 메서드로 접근.

control_attribute.html

img 태그 속성 조작하기

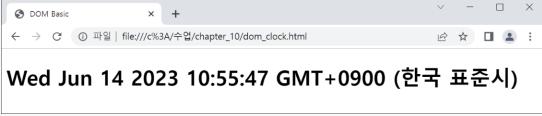
```
<!DOCTYPE html>
<html>
<head>
   <title>DOM Basic</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 변수를 선언합니다.
           let image = document.getElementById('image');
           // 속성을 변경합니다.
           image.src = 'http://via.placeholder.com/300x200';
           image.width = 300;
           image.height = 200;
                                                                               300 x 200
   </script>
</head>
<body>
   <img id="image">
</body>
</html>
```

☑ control_body.html body 태그 속성 조작하기

```
<!DOCTYPE html>
<html>
                                                          이 페이지 내용:
<head>
                                                          value
   <title>DOM Basic</title>
   <script>
                                                                                                    확인
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 속성을 지정합니다.
           document.body.setAttribute('data-custom', 'value');
           // 속성을 추출합니다.
           let dataCustom = document.body.getAttribute('data-custom');
           alert(dataCustom);
       };
                                                         <!DOCTYPE html>
   </script>
                                                         <html>
</head>
                                                         head> -- </head>
<body>
                                                          <body data-custom="value"> </body> == $0
</body>
                                                         </html>
</html>
```

☑ dom_clock.html 문서 객체 속성 조작

```
<html>
<head>
   <title>Clock</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 문서 객체를 선택합니다.
           let clock = document.getElementById('clock');
           // 1초마다 함수를 실행합니다.
           setInterval(function () {
               let now = new Date();
               clock.innerHTML = now.toString();
           }, 1000);
                                                    OM Basic
                                                               × +
   </script>
</head>
<body>
   <h1 id="clock"></h1>
</body>
</html>
```



☑ 이벤트

o 키보드 누르기, 마우스 클릭 등 어떤 현상이 프로그램에 영향을 미침.

```
window.onload = function () { };
```

onblur onfocus onfocusin onfocusout
onload onresize onscroll onunload
onclick ondbclick onmousedown
onmouseup onmousemove onmouseover
onmouseout onmouseenter
onmouseleave onchange onselect
onsubmit onkeydown onkeypress
onkeyup onerror



☑ 이벤트 연결

o 이벤트 모델: 문서 객체에 이벤트를 연결하는 방식.

구분	종류
DOM 레벨 0	• 인라인 이벤트 모델 • 고전 이벤트 모델
DOM 레벨 2	• 마이크로소프트 인터넷 익스플로러 이벤트 모델 • 표준 이벤트 모델

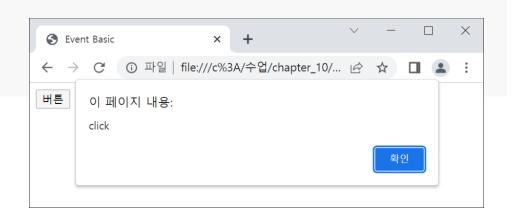
💟 이벤트 연결

- o 인라인 이벤트 모델
 - HTML 태그 내부에 자바스크립트 코드를 넣어 이벤트를 연결하는 방식.

<태그 on이벤트="자바스크립트코드">...</태그>

주로 함수 호출문을 지정

🗹 event_inline.html 인라인 이벤트 모델 사용하기



🗹 event_inlineWithScript.html 💎 script 태그에 인라인 이벤트 모델 사용하기

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline Event</title>
    <script>
        function buttonClick() {
            alert('click');
    </script>
</head>
<body>
    <button onclick="buttonClick()">버튼</button>
</body>
</html>
```

☑ event_tradition.html 고전 이벤트 모델

```
<!DOCTYPE html>
<html>
<head>
   <title>Traditional Event</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
           // 문서 객체를 선택합니다.
           let button = document.getElementById('button');
           // 이벤트를 연결합니다.
           button.onclick = function () {
               alert('click');
           };
   </script>
</head>
<body>
   <button id="button">버튼</button>
</body>
</html>
```

◎ 이벤트 사용

- ㅇ 모든 이벤트 핸들러에는 이벤트 정보를 담고 있는 이벤트 객체가 매개변수로 잔달됨
- o 이벤트 객체를 사용하면 이벤트와 관련한 정보를 알아낼 수 있음.

☑ event_object.html 이벤트 정보

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Object</title>
    <script>
        window.onload = function (event) {
            alert(event);
            console.log(event);
    </script>
</head>
<body>
</body>
</html>
```

◎ 이벤트 사용

- o 기본 이벤트:
 - 특정 태그가 가진 기본적인 이벤트.
 - <a>의 기본 이벤트
 - href에 지정한 페이지로 이동
 - 이벤트 핸들러에서 false를 리턴하면 기본 이벤트 핸들러 호출되지 않음

event_defaulthtml.html

기본 이벤트 제거

```
<!DOCTYPE html>
<html>
<head>
   <title>Traditional Event - Default Event</title>
   <script>
       // 이벤트를 연결합니다.
       window.onload = function () {
          // 문서 객체를 선택합니다.
           let button = document.getElementById('button');
           // 이벤트를 연결합니다.
           button.onclick = function () {
              // 기본 이벤트를 제거합니다.
              return false;
           };
       };
   </script>
</head>
<body>
   <a id="button" href="http://hanbit.co.kr">버튼</a>
</body>
</html>
```