CHAPTER 13

문서 객체 모델

자바스크립트 프로그래밍 입문 (2판)

한빛아카데미 Hanbit Academy, Inc.

Contents

학습목표

- 문서 객체 모델이 무엇인지 이해합니다.
- 문서 객체를 선택하는 방법을 이해합니다.
- 문서 객체를 조작하는 방법을 이해합니다.
- 문서 객체에 이벤트를 연결하는 방법을 이해합니다.

내용

- 문서 객체 모델 관련 용어
- 웹 페이지 생성 순서
- 문서 객체 선택
- 문서 객체 조작
- 이벤트

1. 문서 객체 모델

- 문서 객체 모델
  - 넓은 의미 : 웹 브라우저가 HTML 페이지를 인식하는 방법
  - 좁은 의미 : document 객체와 관련된 객체의 집합을 나타냄

코드 13-1 기본적인 웹 페이지

```
<!DOCTYPE html>
<html>
<head>
    <title>웹 페이지</title>
    <script>

    </script>
</head>
    <h1>Header 1</h1>
    <p>Lorem ipsum dolor amet</p>
</body>
</html>
```

1. 문서 객체 모델

- 문서 객체 : HTML 태그를 자바스크립트에서 사용할 수 있는 객체로 만듬
  - 문서 객체를 조작한다는 말은 태그를 조작한다는 말과 같음
  - 노드 : 각 요소
    - 요소 노드 : h1 태그와 script 태그처럼 요소를 생성하는 노드
    - 텍스트 노드 : 화면에 출력되는 문자열인 Lorem ipsum dolor amet 등

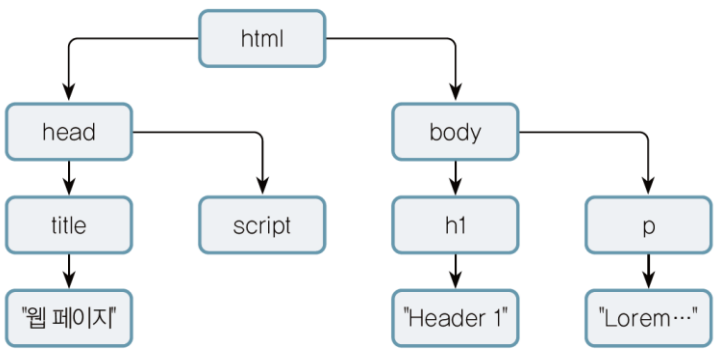


그림 13-1 노드

1. 문서 객체 모델

- 텍스트 노드가 없는 태그

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

- '정적으로 문서 객체를 생성한다'
  - 웹 페이지를 처음 실행할 때 HTML 페이지에 있는 태그를 읽으면서 생성하는 것
- '동적으로 문서 객체를 생성한다'
  - 자바스크립트를 사용해 프로그램 실행 중에 문서 객체를 생성하는 것

2. 웹 페이지 생성 순서

- 웹 브라우저는 웹 페이지를 실행 시 HTML 코드를 위에서 아래로 실행함
  - HTML 페이지 내부에서 alert ( ) 함수를 사용해 중간중간 실행 흐름을 끊음

코드 13-2 실행 순서 확인 전용 코드

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>alert('Process - 0')</script>
</head>
<body>
    <h1>Process - 1</h1>
    <script>alert('Process - 2')</script>
    <h2>Process - 2</h2>
    <script>alert('Process - 3')</script>
</body>
</html>
```

2. 웹 페이지 생성 순서

- 문서 객체가 생성되기 전에 문서 객체를 사용하는 코드
  - ❶ 자바스크립트 코드를 실행하고
  - ❷ h1 태그를 생성하고
  - ❸ h2 태그를 생성

```
<!DOCTYPE html>
<html>
<head>
  <title></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>h1 태그</h1>
  <h2>h2 태그</h2>
</body>
</html>
```

2. 웹 페이지 생성 순서

- 하지만 script 태그를 읽을 당시에는 h1 태그와 h2 태그가 생성되지 않음 → 오류 발생

```
Uncaught TypeError: Cannot read properties of null (reading 'style')
```

2. 웹 페이지 생성 순서

- ❶ h1 태그를 생성
- ❷ h2 태그를 생성
- ❸ 자바스크립트 코드를 실행하는 순서로 바꾸면 문제 해결

코드 13-3 태그 자체를 뒤로 보내서 문제 해결하기

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
  <h1>h1 태그</h1>
  <h2>h2 태그</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>
```

그림 13-2 문서 객체 조작

2. 웹 페이지 생성 순서

- script 태그를 아래에 삽입하면 HTML 페이지의 규모가 클 때 유지 보수가 어려움 → 이벤트 기능 사용

코드 13-4 문서 객체 조작 순서 – 이벤트 활용

```
<!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>
    <h2>Process - 2</h2>
</body>
</html>
```

3. 문서 객체 선택

- 문서 객체 선택
  - HTML 태그를 자바스크립트에서 문서 객체로 변환
  - 문서 객체를 선택하면 자바스크립트로 실행 중에 내부 글자를 변경하거나 스타일을 변경할 수 있음

1개의 문서 객체 선택

표 13-1 1개의 문서 객체를 선택하는 메소드

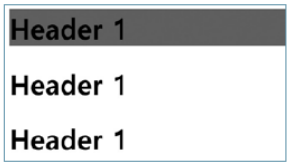
메소드	설명
document.querySelector(선택자)	선택자를 사용해 문서 객체를 선택합니다.

3. 문서 객체 선택

- [예제 13-1] querySelector() 메소드
  - 매개 변수로 전달한 CSS 선택자로 선택되는 첫 번째 태'만 선택

코드 13-6 querySelector() 메소드 QuerySelector.html

```
<!DOCTYPE html>
<html>
<head>
    <title>웹 페이지</title>
    <script>
        // 화면 구성이 완료되면
        window.onload = function () {
            // h1 태그의 색상을 흰색으로 변경합니다.
            const header = document.querySelector('h1');
            header.style.backgroundColor = 'red';
        };
    </script>
</head>
<body>
    <h1>Header 1</h1>
    <h1>Header 1</h1>
    <h1>Header 1</h1>
</body>
</html>
```



3. 문서 객체 선택

여러 개의 문서 객체 선택

표 13-2 여러 개의 문서 객체를 선택하는 메소드

메소드	설명
document.querySelectorAll(선택자)	선택자로 여러 개의 문서 객체를 선택합니다.

- [예제 13-2] document.querySelectorAll( ) 메소드

코드 13-7 document.querySelectorAll() 메소드 QuerySelectorAll.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            const headers = document.querySelectorAll('h1');
```

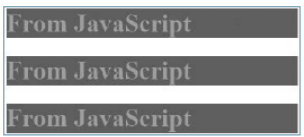
3. 문서 객체 선택

코드 13-7 document.querySelectorAll() 메소드 — QuerySelectorAll.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            const headers = document.querySelectorAll('h1');

            for (let i = 0; i < headers.length; i++) {
                // 변수를 선언합니다.
                const 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>
</html>
```



4. 문서 객체 조작

문자 조작

표 13-3 글자를 조작하는 속성

속성	설명
innerHTML	문서 객체 내부의 글자를 나타냅니다.

- [예제 13-3] innerHTML 속성
  - 문서 객체의 innerHTML 속성을 변경해서 내부 문자를 조작

코드 13-8 innerHTML 속성 innerHTML.html

```
<!DOCTYPE html>
<html>
<head>
    <title>웹 페이지</title>
    <script>
        // 화면 구성이 완료되면
        window.onload = function () {
            // h1 태그 내부의 텍스트를 변경합니다.
            const header = document.getElementById('header');
```

4. 문서 객체 조작

```
            const originalText = header.innerHTML;
            header.innerHTML = '자바스크립트로 변경했어요!<br />';
            header.innerHTML += '원래는 ' + originalText + '였답니다!';
        };
    </script>
</head>
<body>
    <h1 id="header">Header 1</h1>
</body>
</html>
```

4. 문서 객체 조작

- [예제 13-4] innerHTML 속성과 이스케이프 문자
  - 이스케이프 문자를 사용해 태그 내부에 HTML 태그 형식의 글자를 입력

코드 13-9 innerHTML 속성과 이스케이프 문자 Escape.html

```
<!DOCTYPE html>
<html>
<head>
    <title>웹 페이지</title>
    <script>
        // 화면 구성이 완료되면
        window.onload = function () {
            // h1 태그 내부의 문자를 변경합니다.
            const header = document.getElementById('header');
            header.innerHTML = '<i>i 태그입니다</i><br />';
            header.innerHTML += '&lt;i&gt;i 태그입니다&lt;/i&gt;';
        };
    </script>
</head>
<body>
    <h1 id="header">Header 1</h1>
</body>
</html>
```

4. 문서 객체 조작

스타일 조작

표 13-4 스타일 시트의 스타일 속성과 자바스크립트의 스타일 속성 차이

스타일 시트의 스타일 속성	자바스크립트의 스타일 속성
background-color	backgroundColor
border-radius	borderRadius
border-bottom	borderBottom

- [예제 13-5] style 속성을 사용한 스타일 조작

코드 13-11 style 속성을 사용한 스타일 조작 Style.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <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;
```

4. 문서 객체 조작

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

</body>
</html>
```

256개의 div 태그가 위에서 아래로 그레이디언트를 표현

4. 문서 객체 조작

속성 조작

표 13-5 문서 객체의 속성 조작 메소드

메소드	설명
setAttribute(속성_이름, 속성_값)	속성을 지정합니다.
getAttribute(속성_이름)	속성을 추출합니다.

- 웹 표준에서 지정한 속성 접근 방법

```
image.src = 'rint.png'
alert(image.src)
```

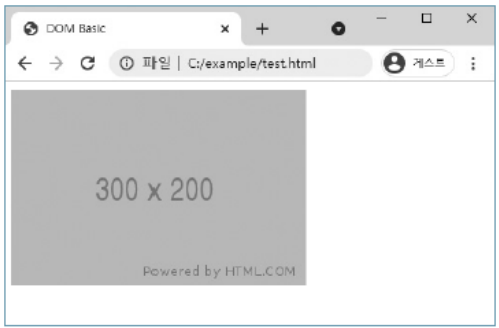
4. 문서 객체 조작

- [예제 13-6] 웹 표준에서 정의한 속성 조작
  - 속성 조작 방법을 사용해 img 태그의 src 속성, width 속성, height 속성을 변경

코드 13-12 웹 표준에서 정의한 속성 조작 StandardAttribute.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 변수를 선언합니다.
            const image = document.getElementById('image');

            // 속성을 변경합니다.
            image.src = 'https://via.placeholder.com/300x200';
            image.width = 300;
            image.height = 200;
        };
    </script>
</head>
<body>
    <img id="image" />
</body>
</html>
```



4. 문서 객체 조작

- 웹 표준에서 지원하지 않는 속성을 지정할 때는 setAttribute ( ) 메소드와
- getAttribute ( ) 메소드를 사용
  - [그림 13-3]에서 data-role 속성은 웹 표준에서 지원하지 않음

```
<body>
    <div data-role="page">
        <div data-role="header"></div>
        <div data-role="content">

        </div>
        <div data-role="footer"></div>
    </div>
</body>
```

그림 13-3 jQuery Mobile 코드

4. 문서 객체 조작

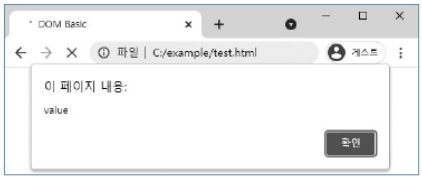
- [예제 13-7] 웹 표준에서 정의하지 않은 속성 조작
  - body 태그에 data-custom 속성을 지정하고 추출해서 출력

코드 13-13 문서 객체 조작 – 속성 조작 CustomAttribute.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 속성을 지정합니다.
            document.body.setAttribute('data-custom', 'value');

            // 속성을 추출합니다.
            const dataCustom = document.body.getAttribute('data-custom');
            alert(dataCustom);
        };
    </script>
</head>
<body>

</body>
</html>
```




```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
    <body data-custom="value"></body>
  </html>
```

5. 이벤트

- 키보드로 키를 입력하거나 마우스 클릭 등 어떤 현상이 프로그램에 영향을 미치는 것
  - 마우스 이벤트
  - 키보드 이벤트
  - HTML 프레임 이벤트
  - HTML 입력 양식 이벤트
  - 사용자 인터페이스 이벤트
  - 구조 변화 이벤트
  - 터치 이벤트

5. 이벤트

이벤트 관련 용어 정리

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

- 이벤트 속성 : onload
- 이벤트 이름, 이벤트 타입 : load
- 이벤트 리스너, 이벤트 핸들러 : 이벤트 속성에 넣는 함수
- 이벤트 모델 : 문서 객체에 이벤트를 연결하는 방법

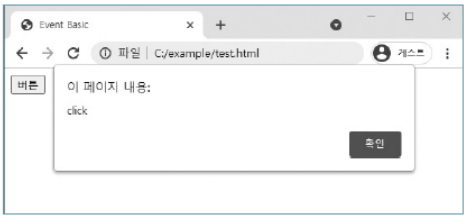
5. 이벤트

인라인 이벤트 모델

- HTML 태그 내부에서 이벤트를 연결하는 방법
- [예제 13-8] 인라인 이벤트 모델
  - button 태그 내부에서 onclick 속성을 사용해 자바스크립트 코드를 입력

코드 13-14 인라인 이벤트 모델

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Basic</title>
</head>
<body>
    <button onclick="alert('click')">버튼</button>
</body>
</html>
```



5. 이벤트

- HTML 태그에서 'on' 문자열로 시작하는 속성은 이벤트와 관련됨

표 13-6 이벤트 속성

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

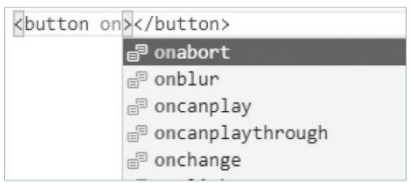


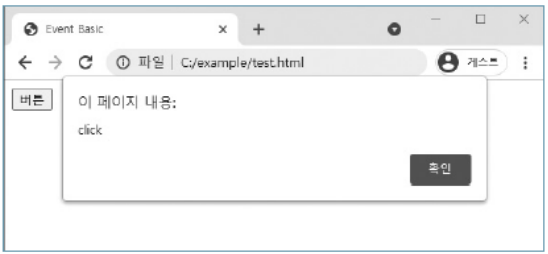
그림 13-4 이벤트 속성

5. 이벤트

- [예제 13-10] script 태그를 활용한 인라인 이벤트 모델
  - 인라인 이벤트 모델에서 script 태그 내부에 있는 함수를 호출

코드 13-15 script 태그를 활용한 인라인 이벤트 모델

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



5. 이벤트

고전 이벤트 모델

```
const image = document.getElementById('image');
image.width = 100;
image.height = 100;
```

- [예제 13-11] 고전 이벤트 모델
  - [코드 13-16]을 고전 이벤트 모델로 변경
  - 문서 객체의 이벤트 속성에 함수를 지정

코드 13-16 고전 이벤트 모델

```
<!DOCTYPE html>
<html>
<head>
    <title>Traditional Event</title>
    <script>
```

5. 이벤트

```
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            const button = document.getElementById('button');

            // 이벤트를 연결합니다.
            button.onclick = function () {
                alert('click');
            };
        };
    </script>
</head>
<body>
    <button id="button">버튼</button>
</body>
</html>
```

5. 이벤트

이벤트 객체

- 인터넷 익스플로러 : window 객체의 event 속성이 이벤트 객체

코드 13-18 표준 이벤트 객체

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

</body>
</html>
```

이벤트 객체

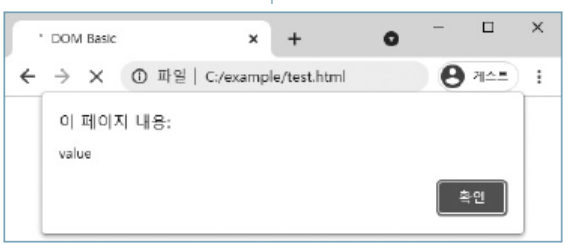


그림 13-6 표준 이벤트 객체

5. 이벤트

기본 이벤트 제거

- 기본 이벤트 : 예) a 태그를 클릭하면 href 속성에 입력한 위치로 이동
- 기본 이벤트를 막아야 할 경우
  - 예) 다음 상황에서 <확인>버튼을 누르면, 우선 사용자가 정확하게 이름과 주민등록번호를 입력했는지 확인하고 이동해야 함

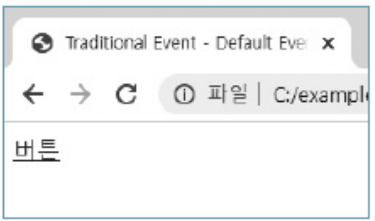
그림 13-7 기본 이벤트 제거를 사용한 유효성 검사

5. 이벤트

- [예제 13-11] 기본 이벤트 제거
  - a 태그의 click 이벤트 리스너에서 false를 리턴

코드 13-19 기본 이벤트 제거

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




THANK
YOU!