17 문서 객체 모델(DOM)

17–1. 문서 객체 모델 알아보기

문서 객체 모델이란

문서 객체 모델(DOM : Document Object Model)

- **자바스크립트를 이용해 웹 문서에 접근하고 제어할 수 있도록 객체를 사용해 체계적으로 정리하는 방법**
  - 자바스크립트를 사용하면 웹 문서를 구성하는 요소들에 동적인 효과를 부여할 수 있음
    자바스크립트로 동적인 효과를 부여하려면 각 구성 요소를 구문할 수 있어야 하며 접근할 수 있는 방법이 필요함

- HTML DOM
  - HTML 언어로 작성된 웹 문서의 DOM을 말함
    웹 문서를 구성하는 모든 요소(텍스트, 이미지 등)는 객체로 정의됨
    웹 문서 내의 이미지 → document 객체(웹 문서) 내의 image 객체(이미지)

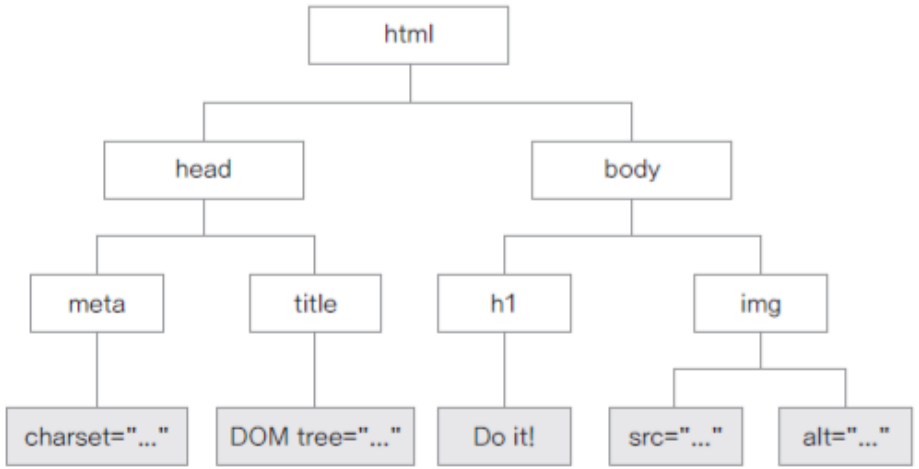
- **태그와 문서 객체의 차이**
  - 태그
    HTML에서 사용되는 <html>이나 <body>와 같은 요소를 의미
  - 문서 객체
    태그를 자바스크립트에서 이용할 수 있는 객체로 만든 경우를 의미

```
<head>
<script>
    window.onload = function () {
      let hd1 = document.getElementById('header');    // 문서 객체
    };
</script>
</head>
<body>
    <h1 id="header">HEADER</h1>    // 태그를 의미
</body>
```

문서 객체 모델이란

DOM Tree

```
chapter17 > <> DomTree1.html > ...
  1   <!DOCTYPE html>
  2   <html lang="en">
  3   <head>
  4       <meta charset="UTF-8">
  5       <title>DOM Tree 알아보기</title>
  6   </head>
  7   <body>
  8       <h1>Do it!</h1>
  9       <img src="images/doit.jpg" alt="공부하는 이미지">
 10   </body>
 11   </html>
```



- **종류에 따른 노드 분류**

종류	설명
요소 노드 (element node)	HTML 태그로 구성된 노드
텍스트 노드 (text node)	HTML 태그 내에 포함되어 있는 텍스트(문자열)
속성 노드 (attribute node)	HTML 태그에 포함된 속성과 값의 집합

- **계층에 따른 노드 분류**

종류	설명
부모 노드 (parent node)	특정 HTML 태그를 포함하고 있는 노드
자식 노드 (child node)	특정 HTML 태그에 포함되어 있는 노드
형제 노드 (sibling node)	같은 레벨에 존재하는 자식 노드

17-2. DOM 요소에 접근하고 속성 가져오기

DOM에 접근하기

getElementById() 메서드

- 요소가 가지는 id를 사용해 요소를 선택하는 메서드

```
document.getElementById(아이디이름)
```

- [예] document.getElementById('main')
  id 속성의 값이 main으로 지정된 요소를 선택
- id 값을 사용하므로 한번에 하나의 요소만 가져올 수 있음

getElementsByTagName() 메서드

- 요소의 이름을 사용해 요소를 선택하는 메서드

```
document.getElementsByTagName(태그이름)
```

- [예] document.getElementsByTagName('p')
  문서 내에 존재하는 모든 <p> 태그들을 선택

- 하나 이상의 요소를 선택 가능
  - 지정한 요소가 하나 이상 존재할 경우 순서에 따라 인덱스로 표현
  - [예] document.getElementsByTagName('p')[3]
    문서 내에 존재하는 <p> 태그들 중 네 번째 <p>태그

DOM에 접근하기

getElementsByclassName() 메서드

- **요소에 지정된 클래스 이름을 사용해 요소를 선택하는 메서드**

```
document.getElementsByClassName(클래스이름)
```

  - [예] document.getElementsByClassName('headers')
    Headers라는 클래스가 적용된 모든 요소들을 선택

- **하나 이상의 요소를 선택 가능**
  - 지정한 요소가 하나 이상 존재할 경우 순서에 따라 인덱스로 표현
  - [예] document.getElementsByClassName('haeder')[2]
    Header 클래스가 적용된 요소들 중 세 번째 요소를 선택

DOM에 접근하기

chapter17 > getElement.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script>
        window.onload = function() {
            const header1 = document.getElementById('header1');
            header1.style.color = 'green';

            const header3 = document.getElementsByClassName('header2');
            header3[1].style.color = 'blue'

            const headAll = document.getElementsByTagName('h1');
            headAll[3].style.color = 'red';
        }
    </script>
</head>
<body>
    <h1 id="header1">HEADER-1</h1>
    <h1 class="header2">HEADER-2</h1>
    <h1 class="header2">HEADER-3</h1>
    <h1>HEADER-4</h1>
</body>
</html>
```

HEADER-1

HEADER-2

HEADER-3

HEADER-4

DOM에 접근하기

querySelect() 메서드

- **요소가 가지는 id를 사용해 요소를 선택하는 메서드**

```
document.getElementById(아이디이름)
```

- [예] document.getElementById('main')
  id 속성의 값이 main으로 지정된 요소를 선택
- id 값을 사용하므로 한번에 하나의 요소만 가져올 수 있음

```
document.getElementsByTagName(태그이름)
```

DOM에 접근하기

querySelector() 메서드/ querySelectorAll() 메서드

- **인자에 아이디, 클래스, 태그 이름을 모두 지정해 요소를 선택할 수 있는 메서드**

```
document.querySelector(선택자)
```

- 아이디 선택자를 지정하는 경우처럼 하나의 요소를 선택할 때 사용

```
document.querySelectorAll(선택자 또는 태그)
```

- 태그나 클래스 이름을 지정하 하나 이상의 요소를 선택할 때 사용

- **인자 지정**
  - 아이디를 사용할 경우 선택자 이름 앞에 #기호를 지정
  - 클래스를 사용할 경우 선택자 이름 앞에 dot(.)기호를 지정
  - 태그이름을 사용할 경우 추가로 기호를 지정하지 않음

DOM에 접근하기

chapter17 > querySelect.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script>
        window.onload = function() {
            const header1 = document.querySelector('#header1');
            header1.style.color = 'green';

            const header2 = document.querySelectorAll('.header2');
            header2[0].style.color = 'blue'
            header2[1].style.color = 'red'

        }
    </script>
</head>
<body>
    <h1 id="header1">HEADER-1</h1>
    <h1 class="header2">HEADER-2</h1>
    <h1 class="header2">HEADER-3</h1>
</body>
</html>
```

HEADER-1

HEADER-2

HEADER-3

DOM 내용의 수정

innerHTML 속성과 textContent 속성

- innerHTML 속성

객체.innerHTML = "요소(HTML 코드)"

- 값으로 지정된 요소(html 태그)를 추가
  요소를 삽입하는 속성이므로 html 태그를 인식함

- textContent 속성

객체.textContent = "문자열"

- 객체에 인자로 지정한 문자열을 삽입
  문자열을 삽입하는 속성이므로 html 태그를 인식하지 못함

chapter17 > inner.html > ...

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <script>
6           function inntext(){
7               var now = new Date();
8               document.getElementById("current").innerText = now;
9           }
10          function innhtml() {
11              var now = new Date();
12              document.getElementById("current").innerHTML = "<em>"+now+"</em>";
13          }
14
15      </script>
16  </head>
17  <body>
18      <button onclick="inntext()">innerText로 표시하기</button>
19      <button onclick="innhtml()">innerHTML로 표시하기</button>
20      <h1>현재 시각: </h1>
21      <div id="current"></div>
22  </body>
23  </html>
```

DOM 속성의 추출과 수정

setAttribute() 메서드와 getAttribute() 메서드

- setAttribute() 메서드
  - 객체에 대해 인자로 지정한 속성을 추가하거나 변경

```
객체.setAttribute(name, value)
```

name : 객체의 지정할 속성을 의미
value : 속성에 지정될 값을 의미

- getAttribute() 메서드
  - 인자로 지정한 속성을 추출

```
객체.getAttribute(name)
```

chapter17 > setAttrgetAttr.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script>

        window.onload = function () {

            let img = document.createElement('img');
            img.setAttribute('src', 'images/flower1.gif');
            img.setAttribute('width', 200);
            img.setAttribute('height', 200);

            document.body.appendChild(img);

            let new_width = img.getAttribute('width');
            let new_height = img.getAttribute('height');

        };

    </script>
</head>
<body>

</body>
</html>
```

17-4. DOM에서 노드 추가와 삭제

노드의 생성

- createElement( ) 메서드
  - 요소 노드 생성 (생성하고자 하는 요소를 인자로 지정)

```
document.createElement( 태그이름 )
```

  - [예] let header = document.createElement('h1')
    header 라는 이름을 가진 <h1> 요소 노드를 동적으로 생성

- createTextNode( ) 메서드
  - 텍스트 노드 생성 (생성하고자 하는 텍스트를 인자로 지정)

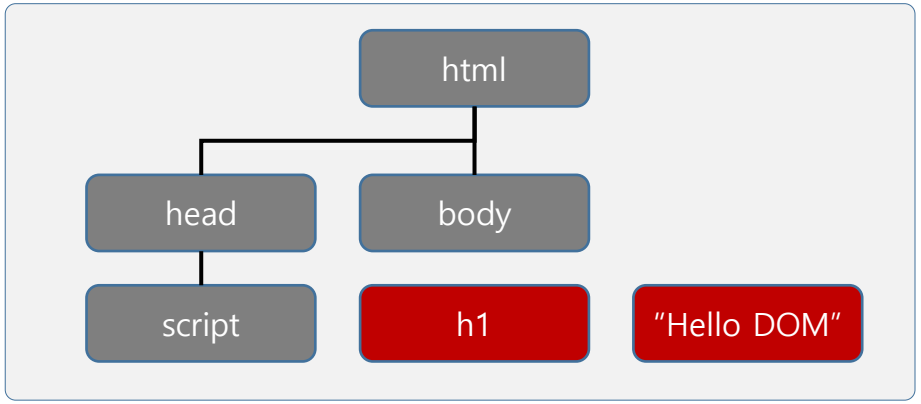
```
document.createTextNode( 텍스트 )
```

  - [예] let textNode = document.createTextNode('Hello DOM')
    textNode라는 이름을 가진 'Hello DOM' 텍스트 노드를 동적으로 생성

chapter17 > createNode.html > html

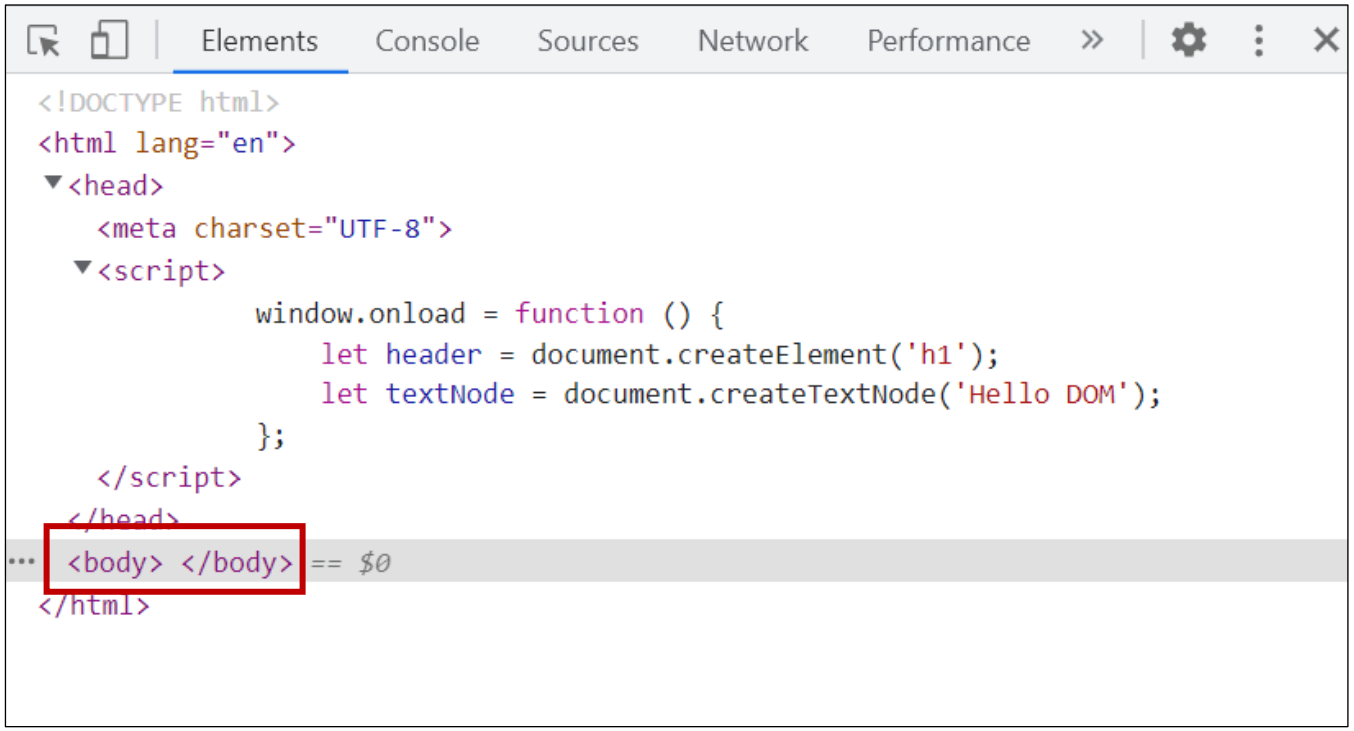
```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <script>
6
7           window.onload = function () {
8               let header = document.createElement('h1');
9               let textNode = document.createTextNode('Hello DOM');
10          };
11
12      </script>
13  </head>
14  <body>
15
16  </body>
17  </html>
```

노드의 추가



요소들이 생성은 되었지만 문서에 삽입되지 않았음

(DOM의 구조에 포함되지 않았음)



노드 삽입

- appendChild() 메서드
  - 생성한 노드를 문서에 삽입(연결)

```
부모노드.appendChild( 노드 )
```

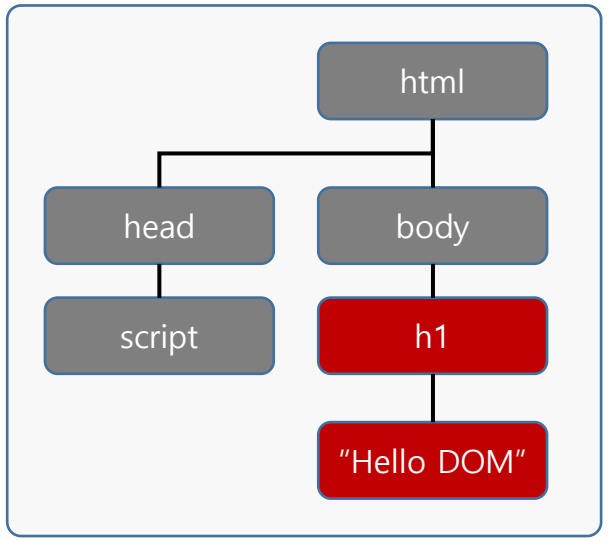
  - 부모 노드에 인자로 지정되어 있는 노드를 자식 노드로 삽입
  - [예] header1.appendChild(textNode1)
    header1이라는 문서 객체의 자식으로 textNode1을 지정해 연결함

chapter17 > appendChild.html > ...

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <script>
6
7           window.onload = function () {
8
9               let header = document.createElement('h1');
10              let textNode = document.createTextNode('Hello DOM');
11
12              header.appendChild(textNode);
13              document.body.appendChild(header);
14
15          };
16
17      </script>
18  </head>
19  <body>
20
21  </body>
22  </html>
```

Hello DOM

노드 삽입



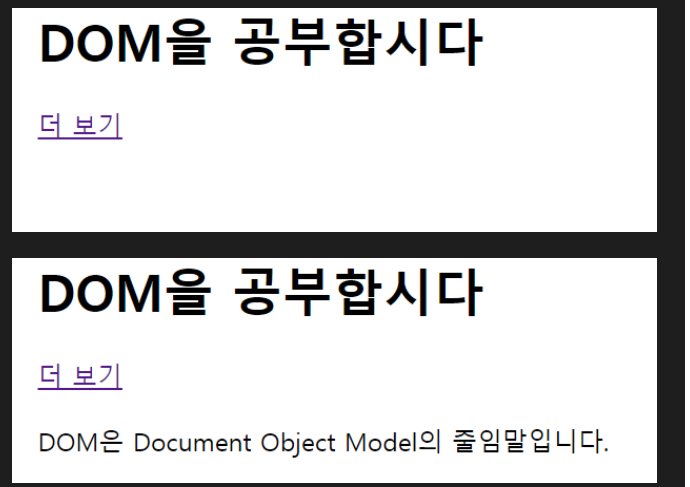
Elements Console Sources Network Performance

```
<!DOCTYPE html>
<html lang="en">
▼<head>
    <meta charset="UTF-8">
▼<script> == $0
            window.onload = function () {

                let header = document.createElement('h1');
                let textNode = document.createTextNode('Hello DOM');

                header.appendChild(textNode);
                document.body.appendChild(header);

            };
    </script>
  </head>
▼<body>
    <h1>Hello DOM</h1>
  </body>
</html>
```

노드 삽입

chapter17 > createNode1.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <style>
        #container{
          width:500px; margin:10px auto; padding:20px;
        }
        #info { margin-top:20px; }
    </style>
</head>
<body>
    <div id="container">
        <h1>DOM을 공부합시다</h1>
        <a href="#" onclick="addP(); this.onclick='';">더 보기</a>
        <div id="info"></div>
    </div>
    <script>
        function addP() {
            var newP = document.createElement("p");
            var txtNode = document.createTextNode("DOM은 Document Object Model의 줄임말입니다.");
            newP.appendChild(txtNode);
            document.getElementById("info").appendChild(newP);
        }
    </script>
</body>
</html>
```



노드 삽입

속성 노드 삽입

- createAttribute() 메서드
  - 요소에 속성 노드를 생성하기 위한 메서드

```
document. createAttribute( 속성 )
```

  - 속성의 값은 value 속성을 사용해 지정

```
let srcNode = document.createAttribute('src');
srcNode.value = 'images/dom.jpg'
```

노드 삽입

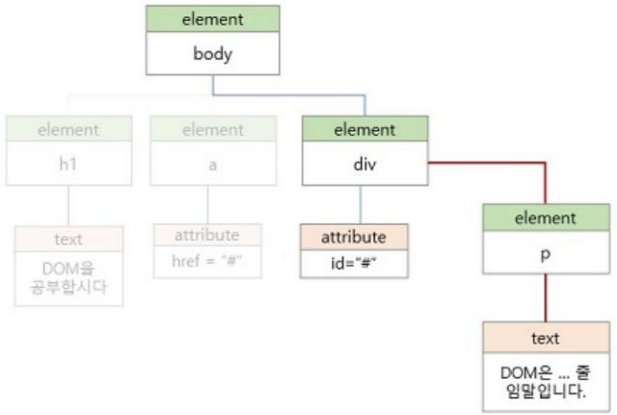
chapter17 > createAttr.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <style>
        #container{
          width:500px; margin:10px auto; padding:20px;
        }
        #info { margin-top:20px; }
    </style>
</head>
<body>

    <div id="container">
        <h1>DOM을 공부합시다</h1>
        <a href="#" onclick="addContents(); this.onclick='';">더 보기</a>
        <div id="info"></div>
      </div>

    <script>
        function addContents() {
          var newP = document.createElement("p");
          var txtNode = document.createTextNode("DOM:Document Object Model");
          newP.appendChild(txtNode);

          var newImg = document.createElement("img");
          var srcNode = document.createAttribute("src");
          var altNode = document.createAttribute("alt");

          srcNode.value = "images/dom.jpg";
          altNode.value = "돔 트리 예제 이미지";
          newImg.setAttributeNode(srcNode);
          newImg.setAttributeNode(altNode);

          document.getElementById("info").appendChild(newP);
          document.getElementById("info").appendChild(newImg);
        }
    </script>
</body>
</html>
```

DOM을 공부합시다

더 보기

DOM:Document Object Model



노드 삭제

노드 삭제

- removeChild() 메서드
  - 문서 객체의 자식 노드를 제거 함

```
부모노드.removeChild( 노드 )
```

chapter17 > removeChild.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <script>

        window.onload = function () {
            var willRemove = document.getElementById('will-remove');
            document.body.removeChild(willRemove);
        };

    </script>
</head>
<body>

    <h1 id="will-remove">Header-1</h1>
    <h1>Header-2</h1>

</body>
</html>
```

Header-2

수고하셨습니다