



Chapter 05-4

객체-4 (문서 객체 모델)

1



문서 객체 모델

문서 객체 모델의 개요

■ 문서 객체 모델 (DOM : Document Object Model)

• 문서 객체 모델의 정의

- 각종 태그들로 구성되어 있는 HTML 문서의 구조를 의미

• 문서 객체

- HTML 문서를 구성하고 있으면서 기능(method)과 속성(Properties)을 가지고 있는 태그

- [예] 태그의 경우

- 이미지를 출력하는 기능, alt와 src 등의 속성을 가지는 객체

• 문서 객체 모델을 사용하는 이유

- 표준화된 방법으로 문서 객체에 접근할 수 있는 방법을 제공하기 위해 사용

- [예] 자바스크립트를 사용해 문서 객체에 접근할 수 있으며, 속성과 스타일을 지정할 수 있음

• 문서 객체 모델의 한계와 해결

- IE8 버전 이하와 파이어폭스, 사파리, 크롬 등의 브라우저 간 호환성이 떨어짐

- jQuery를 사용해 해결 가능

문서 객체 모델의 개요

■ 태그와 문서 객체

• 태그

- HTML에서 사용되는 <html>이나 <body>와 같은 요소를 의미

• 문서 객체

- 태그를 자바스크립트에서 이용할 수 있는 객체로 만든 경우를 의미
- 문서 객체의 집합을 문서 객체 모델(DOM)이라고 함

```
<head>
  <script type="text/javascript">
    window.onload = function () {
      var header = document.getElementById('header'); // 태그를 문서 객체로 변환
    };
  </script>
</head>
<body>
  <h1 id="header">HEADER</h1> // 태그를 의미
</body>
```

문서 객체 모델의 개요

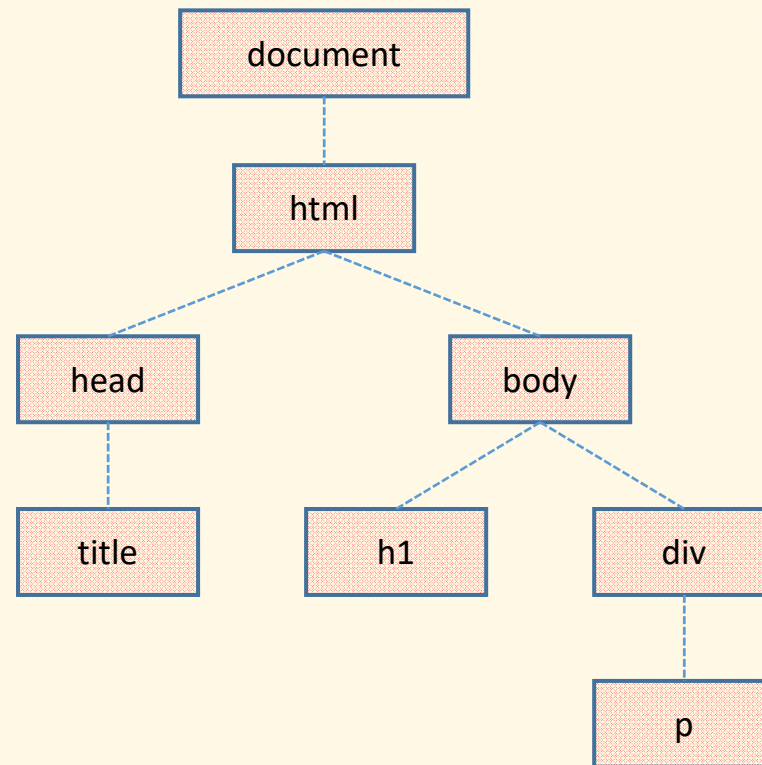
■ 문서 객체 모델의 예

```
<html>

<head>
  <title> text </title>
</head>

  <body>
    <h1> txt </h1>
    <div>
      <p> txt </p>
    </div>
  </body>

</html>
```

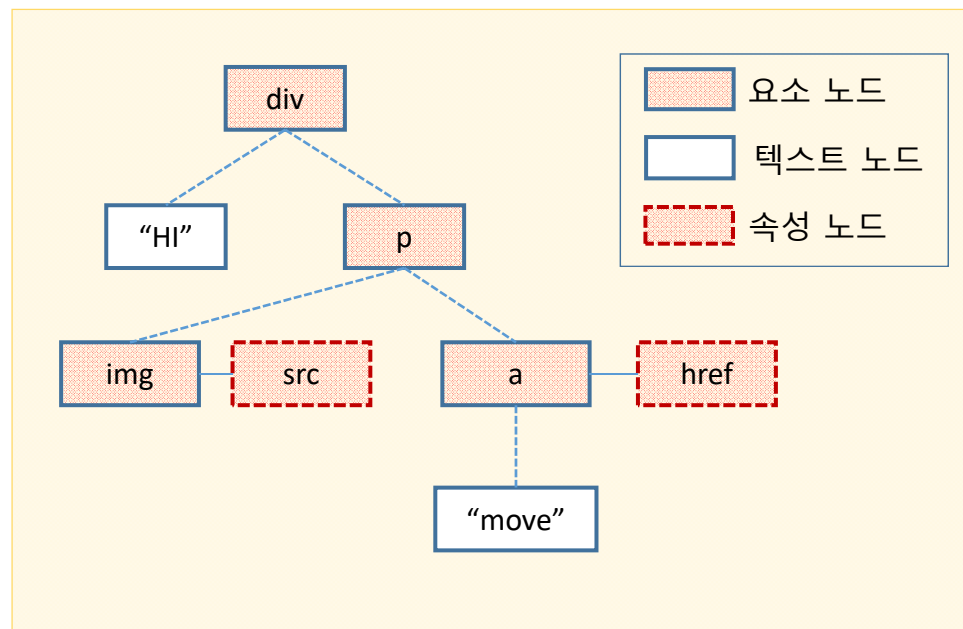


문서 객체 모델의 개요

■ DOM 노드의 분류

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

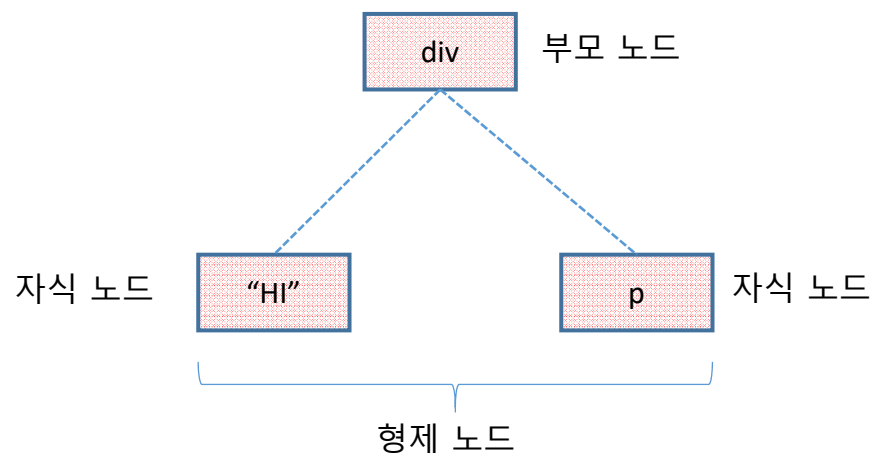
```
<div>
  HI
  <p>
    
    <a href="#"> move </a>
  </p>
</div>
```



문서 객체 모델의 개요

■ 계층에 따른 노드의 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



문서 객체의 생성과 삭제

정적 객체와 동적 객체

■ 정적 객체와 동적 객체

• 정적 객체 (Static Object)

- HTML 문서에 나타나 있는 요소들
- 내용이 변하지 않으며 실행할 때마다 항상 같은 내용이 출력되는 객체
- 정적 객체로 작성되는 문서를 정적 객체 문서라고 함

• 동적 객체 (Dynamic Object)

- 문서의 실행에 따라 새로 생성되는 요소들
- 실행할 때마다 다른 내용이 출력될 수 있는 객체
- 동적 객체로 작성되는 문서를 동적 객체 문서라고 함
- 문서 객체 모델을 사용하거나 제이쿼리 등을 사용해 생성

문서 객체의 생성

■ 문서 객체를 생성하기 위한 메서드

• 요소 노드 생성

```
createElement( tagName )
```

- 인자로 지정한 요소 노드를 생성
- [예] `var header = document.createElement('h1')`
 - header 라는 이름을 가진 <h1> 태그를 동적으로 생성

• 텍스트 노드 생성

```
createTextNode( text )
```

- 인자로 지정한 텍스트 노드를 생성
- [예] `var textNode = document.createTextNode('Hello DOM')`
 - textNode라는 이름을 가진 'Hello DOM'이라는 텍스트 노드를 동적으로 생성

문서 객체의 생성

5-DOM-create.htm

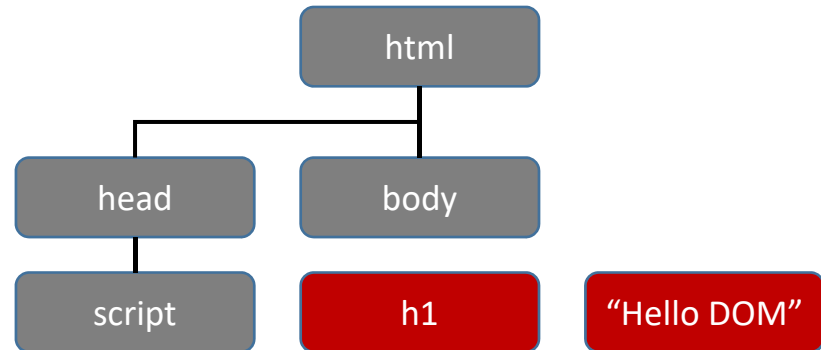
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {
      var header = document.createElement('h1');
      var textNode = document.createTextNode('Hello DOM');

    };

  </script>
</head>
<body> </body>
</html>
```

문서 객체의 생성



요소들이 생성은 되었지만 문서에 삽입되지 않았음
(DOM의 구조에 포함되지 않았음)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script type="text/javascript">
      window.onload = function () {
        var header = document.createElement('h1');
        var textNode = document.createTextNode('Hello DOM');
      };
    </script>
  </head>
  <body>
  </body>
</html>
```

문서 객체의 삽입(연결)

■ 문서 객체의 삽입(연결)

- 생성한 문서 객체를 문서에 삽입(연결)해 출력되게 하는 방법

```
객체.appendChild( node )
```

- 객체에 인자로 지정되어 있는 노드를 연결
 - 객체의 자식 노드로 삽입
- [예] `header1.appendChild(textNode1)`
 - `header1`이라는 문서 객체의 자식으로 `textNode1`을 지정해 연결함

문서 객체의 삽입(연결)

5-DOM-append.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {
      var header1 = document.createElement('h1');
      var textNode1 = document.createTextNode('Hello DOM1');

      var header2 = document.createElement('h1');
      var textNode2 = document.createTextNode('Hello DOM2');

      header1.appendChild(textNode1);
      document.body.appendChild(header1);

      header2.appendChild(textNode2);
      document.body.appendChild(header2);

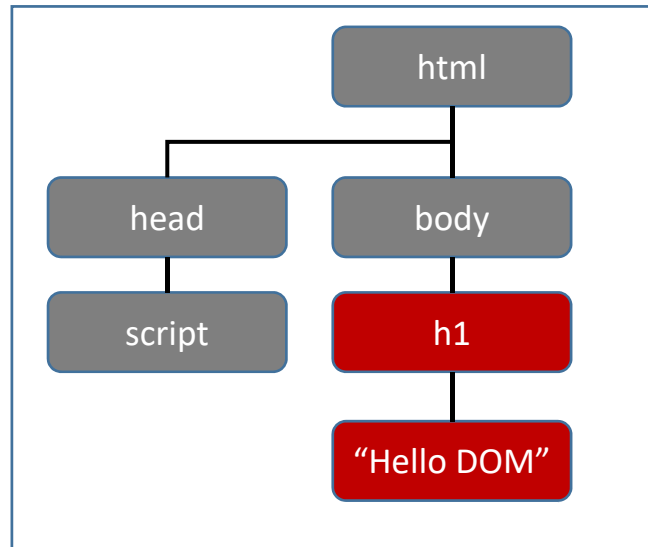
    };

  </script>
</head>
<body> </body>
</html>
```

Hello DOM1

Hello DOM2

문서 객체의 삽입(연결)



```
Elements Console Sources Network Performance Memory
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    ... <script type="text/javascript"> == $0
      window.onload = function () {
        var header1 = document.createElement('h1');
        var textNode1 = document.createTextNode('Hello DOM1');

        var header2 = document.createElement('h1');
        var textNode2 = document.createTextNode('Hello DOM2');

        header1.appendChild(textNode1);
        document.body.appendChild(header1);

        header2.appendChild(textNode2);
        document.body.appendChild(header2);
      };
    </script>
  </head>
  <body>
    <h1>Hello DOM1</h1>
    <h1>Hello DOM2</h1>
  </body>
</html>
```

문서 객체의 속성 지정과 추출

■ innerHTML 속성과 innerText 속성

• innerHTML 속성

객체.innerHTML = "요소(HTML 코드)"

- 값으로 지정된 요소를 추가
 - 지정된 요소는 자식 요소로 추가됨
 - HTML 태그와 속성을 인식함

• innerText 속성

객체.innerText = "문자열"

- 객체에 인자로 지정한 문자열을 삽입
 - HTML 태그와 속성을 인식하지 못함

문서 객체의 속성 지정과 추출

5-DOM-innerHTML-innerText

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {

      document.body.innerHTML = '<h2>Document Object Model</h2>';

    };

  </script>
</head>
<body> </body>
</html>
```

Document Object Model

innerHTML 일 경우

<h2>Document Object Model</h2>

문서 객체의 속성 지정과 추출

■ `setAttribute()` 메서드

객체.`setAttribute(name, value)`

- 객체에 대해 인자로 지정한 속성을 추가하거나 변경
 - `name` : 객체의 지정할 속성을 의미
 - `value` : 속성에 지정될 값을 의미

■ `getAttribute()` 메서드

객체.`getAttribute(name)`

- 인자로 지정한 속성을 제거

문서 객체의 속성 지정과 추출

5-DOM-set-getAttr.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {

      var img = document.createElement('img');
      img.setAttribute('src', 'images/flow1.gif');
      img.setAttribute('width', 200);
      img.setAttribute('height', 200);

      document.body.appendChild(img);

      var new_with = img.getAttribute('width');
      var new_height = img.getAttribute('height');
      alert(new_with + ", " + new_height);

    };

  </script>
</head>
<body> </body>
</html>
```

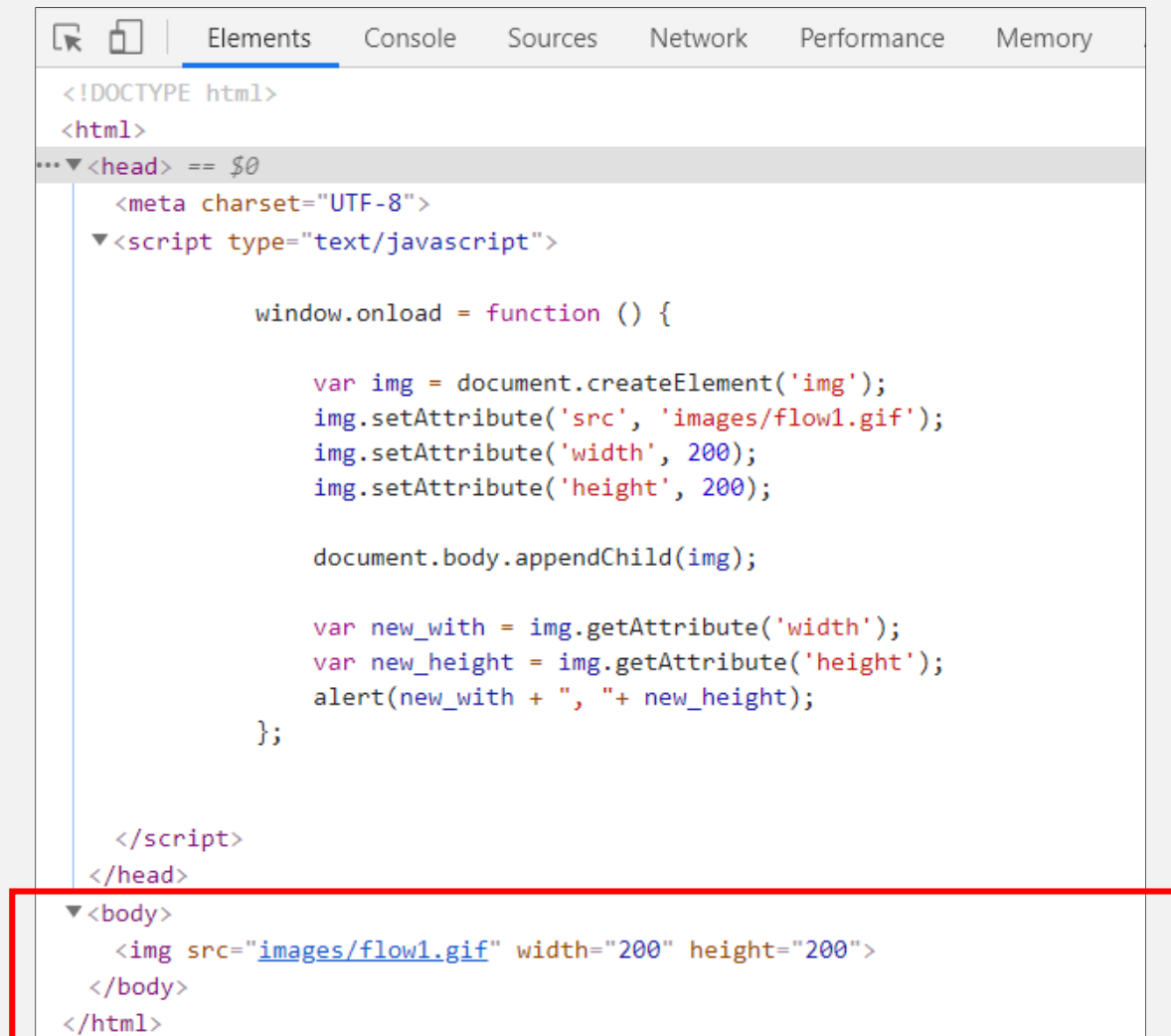


localhost:8080 내용:

200, 200

확인

문서 객체의 속성 지정과 추출



The screenshot shows a web browser's developer tools interface. The 'Elements' tab is selected, displaying the DOM tree. The tree structure is as follows:

- <!DOCTYPE html>
- <html>
- <head> == \$0
 - <meta charset="UTF-8">
 - <script type="text/javascript">

```
window.onload = function () {  
  
    var img = document.createElement('img');  
    img.setAttribute('src', 'images/flow1.gif');  
    img.setAttribute('width', 200);  
    img.setAttribute('height', 200);  
  
    document.body.appendChild(img);  
  
    var new_with = img.getAttribute('width');  
    var new_height = img.getAttribute('height');  
    alert(new_with + ", " + new_height);  
};
```

</script>
- </head>
- <body>
 -
- </body>
- </html>

The `` element in the `<body>` is highlighted with a red rectangle, indicating it was dynamically created by the script shown in the `<script>` block above.

문서 객체의 선택

■ getElementById() 메서드

`getElementById(id)`

- 요소가 가지는 id를 사용해 요소를 선택하는 메서드
 - [예] `document.getElementById('main')`
 - HTML 문서 내에서 id 속성의 값이 main으로 지정된 부분을 의미
- 요소에 지정된 id 값을 사용하므로 한번에 하나의 요소만 가져올 수 있음

문서 객체의 선택

5-DOM-getElementById.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {

      var header1 = document.getElementById('header-1');
      var header2 = document.getElementById('header-2');

      header1.innerHTML = 'with getElementById()';
      header2.innerHTML = 'with getElementById()';

    };
  </script>
</head>
<body>

  <h1 id="header-1">Header</h1>    // <h1> 요소의 기존 문자열은 새로운 문자열로 대체됨
  <h1 id="header-2">Header</h1>    // <h1> 요소의 기존 문자열은 새로운 문자열로 대체됨

</body>
</html>
```

with getElementById()

with getElementById()

문서 객체의 선택

■ `getElementsByTagName()` 메서드

```
getElementsByTagName( tagName )
```

- 요소의 이름을 사용해 요소를 선택하는 메서드
 - [예] `document.getElementsByTagName('p')`
 - HTML 문서 내에 존재하는 <p> 태그를 의미
- 하나 이상의 요소를 선택 가능
 - 문서 내에 지정한 요소가 하나 이상 존재할 경우 나타나는 순서에 따라 인덱스로 표현
 - [예] `document.getElementsByTagName('p')[3]`
 - HTML 문서 내에 존재하는 <p> 태그들 중 네 번째 <p> 태그

문서 객체의 선택

5-DOM-getElementsbyTagName.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    window.onload = function () {

      var headers = document.getElementsByTagName('h1');
      headers[0].innerHTML = 'with getElementsByTagName()';
      headers[1].innerHTML = 'with getElementsByTagName()';

    };

  </script>
</head>
<body>

  <h1>Header</h1>
  <h1>Header</h1>

</body>
</html>
```

with getElementsByTagName()
with getElementsByTagName()

문서 객체의 제거

■ removeChild() 메서드

```
객체.removeChild( child )
```

- 문서 객체의 자식 노드를 제거 함

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    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