

JavaScript

4. DOM(Document Object Model)

JavaScript DOM (DOCUMENT OBJECT MODEL)

DOM 이란

- HTML 문서에 접근하기 위한 객체 모델.
 - 객체: HTML element들
 - 속성: HTML element들의 구성 요소들
 - 메소드: HTML element에 접근하기 위한 방법
 - 이벤트: HTML element에 대한 이벤트
- HTML DOM은 HTML 요소(element)를 얻고, 변경하고, 추가하고, 삭제 할 수 있는 방법을 제공한다.

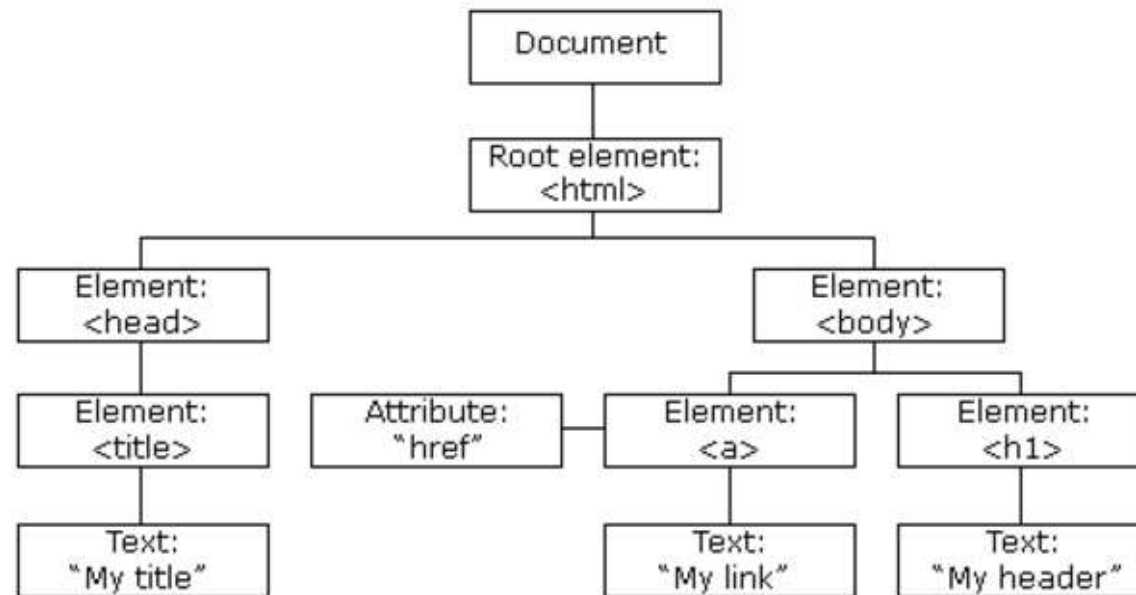
DOM 이란

- 페이지가 loading될 때 그 페이지의 DOM 이 생성된다.
- DOM의 활용
 - 요소, 속성, CSS 스타일 변경
 - 요소, 속성 삭제
 - 요소, 속성 추가
 - 이벤트에 대응하는 반응
 - 새로운 이벤트 생성

DOM tree and node

- HTML 내의 모든 것은 node로 표현된다.

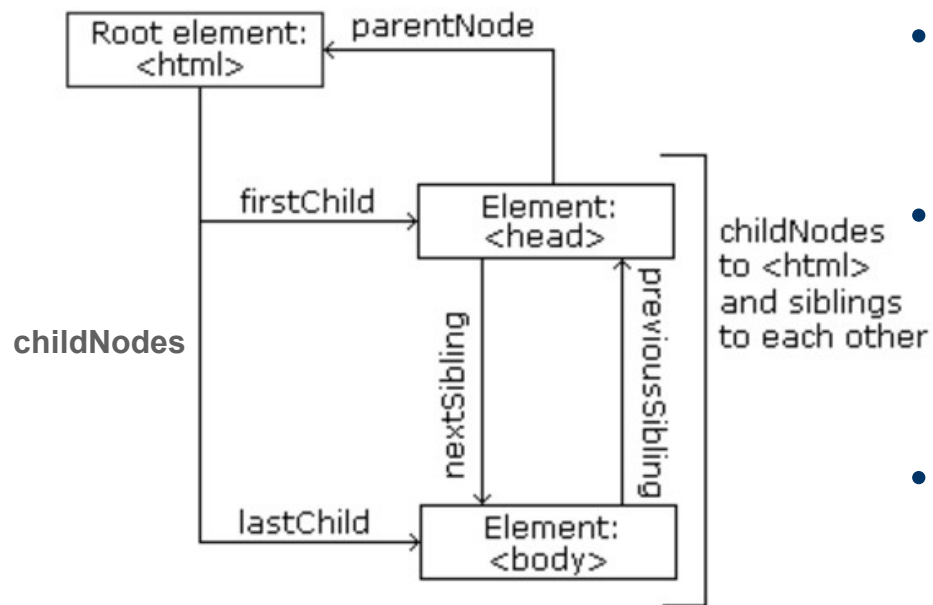
The HTML DOM Tree of Objects



DOME TREE AND NODE

DOM tree and node

- DOM tree의 노드들은 계층적 구조를 가진다.



- 노드 트리의 최상위 노드는 루트 노드(root node)이다.
- 루트 노드를 제외한 모든 노드는 단 하나의 부모 노드(parent node)를 가진다.
- 노드는 여러 개의 자식 노드(child node)를 가진다.
- 같은 부모 노드를 가진 노드들을 형제 노드(sibling node)라고 한다.

Node 간의 관계

- <html> 은 root node 이다.
- <head>와 <body>는 <html>의 자식노드이고, <html>은 <head>와 <body>의 부모 노드이다.
- <head>는 <html>의 첫 번째 자식 노드이다.
- <body>는 <html>의 두 번째이자 마지막 자식 노드이다.

```
<html>
  <head>
    <title>Multi media</title>
  </head>
  <body>
    <h1>Web contents</h1>
    <p>Lesson Week-11</p>
  </body>
</html>
```


Node 간의 관계

- <head>는 하나의 자식 노드(<title>)를 가진다.
 - <title>은 하나의 자식 노드를 가지며, 이는 text node이다.
- <body>는 두 개의 자식 노드(<h1>, <p>)를 가진다.
 - <h1>은 하나의 자식 노드를 가진다.
 - <p> 하나의 자식 노드를 가진다.
 - <h1>과 <p>는 형제 노드이다.

Node 간의 접근

- 노드 속성을 이용하여 노드간의 이동이 가능하다.
 - parentNode
 - childNodes[]
 - firstChild
 - lastChild
 - nextSibling
 - previousSibling

자식 노드와 노드 값

```
<html>

  <head>
    <title>Multi media</title>
  </head>

  <body>
    <h1 id="subject">Web contents</h1>
    <p> Lesson Week-11 </p>
    <p id="demo"></p>
  </body>

  <script>
    myText = document.getElementById("subject").firstChild.nodeValue;
    document.getElementById("demo").innerHTML = myText;
  </script>

</html>
```

Node 속성

- nodeName 속성
 - 수정할 수 없음. (read-only)
 - HTML element 노드의 nodeName 값은 tag 이름이다.
 - HTML element 속성의 nodeName 값은 attribute 이름이다.
 - 모든 text 노드의 이름은 항상 '#text'이다.
- nodeValue 속성
 - HTML element 노드의 nodeValue는 없다.
 - 모든 text 노드의 nodeValue 값은 문자열이다.
 - attribute 노드의 nodeValue는 attribute가 가진 값이다.

Node 속성

- nodeType 속성

Element type	Node type
<i>Element</i>	1
<i>Attribute</i>	2
<i>Text</i>	3
Comment	8
Document	9

HTML DOM DOCUMENT OBJECT

HTML DOM Document 객체

- Document 객체
 - 웹 페이지의 최상위 객체이다.
 - 웹 페이지의 다른 객체에 접근하기 위해서는 항상 document 객체로 부터 시작해야 한다.

DOM method

- method는 html 문서에서 행할 수 있는 행위를 말한다.
 - getElementById 또는 getElementsByName 등은 HTML element 객체를 얻기 위한 메소드이다.
- properties는 html 요소에 있는 속성 값이나 스타일 값이며 설정하거나 변경이 가능하다.
 - innerHTML은 element의 contents를 제어할 때 사용하는 속성 값이다.

HTML DOM Document 객체

- DOM 속성

Method	Description
node.childElementCount	Node가 가진 자식 element의 개수.
node.childNodes	Node의 자식 노드 리스트.
node.children	Node의 자식 element의 리스트.
node.innerHTML	HTML element와 contents
node.innerText	HTML element의 contents
node.Value, nodeName, nodeType	Node의 속성을 나타냄.
node.parentNode	부모 node를 나타냄
node.parentElement	부모 node의 element를 나타냄. (tag 명)
node.nextSibling	다음 이웃 node를 나타냄
node.nextElementSibling	다음 이웃 element node를 나타냄

HTML DOM Document 객체

- DOM 속성

Method	Description
previousSibling	이전 이웃 node를 나타냄
previousElementSibling	이전 이웃 element node를 나타냄
firstChild	첫번째 자식 node를 나타냄
lastChild	마지막 자식 node를 나타냄

HTML DOM Document 객체

- HTML Element 객체 찾기

Method	Description
<code>document.getElementById(id)</code>	해당 Id를 가진 Element 찾는다. 하나의 id는 하나의 element만 가질 수 있으므로 element 객체를 return한다. 없으면, null이 return 된다.
<code>document.getElementsByTagName(tag)</code>	해당 tag 를 가진 모든 Element 들 찾는다. 여러 개의 element가 동일한 Name을 가질 수 있으므로 list를 return 한다.
<code>document.getElementsByClassName(class)</code>	해당 Class name으로 Element 들 찾는다. 여러 개의 element가 동일한 Name을 가질 수 있으므로 list를 return 한다.

HTML DOM Document 객체

- HTML Element 객체 찾기
 - HTML objects collection으로 element를 찾을 수 있다.
 - document.anchors
 - document.body
 - document.documentElement
 - document.embeds
 - document.forms
 - document.head
 - document.images
 - document.links
 - document.scripts
 - document.title

HTML DOM Document 객체

- HTML Element 생성 및 삭제

Method	Description
<code>document.createElement(node name)</code>	인자로 준 tag이름의 HTML element를 생성한다. 생성된 element의 객체를 return 한다.
<code>document.createTextNode(string contents);</code>	Text node를 생성한다.
<code>node.removeChild(element object)</code>	인자로 준 HTML element를 제거한다.
<code>node.appendChild(node)</code>	HTML element를 추가
<code>node.insertBefore(new node, child node)</code>	특정 child node 앞에 HTML element 추가
<code>node.replaceChild(new node, old node)</code>	HTML element를 교체

- Event 핸들러 등록

Method	Description
<code>node.onEventname = function() {code}</code>	이벤트 핸들러에 함수 연결
<code>node.addEventListener("event name", function)</code>	

HTML DOM Document 객체

- HTML Element 변경하기
 - element 찾기 메소드를 이용해 얻은 element 객체의 속성과 style을 변경한다.

Method	Description
<i>element.innerHTML = new contents</i>	HTML의 contents 변경
<i>element.attribute = new value</i>	HTML의 속성 값 변경
<i>element.setAttribute(attribute, new value)</i>	HTML의 속성 값 변경
<i>element.style.property = new style</i>	HTML의 style 변경하기

HTML DOM Document 객체

```
<html>
  <body>
    <button onclick="myFunction()">Try it</button>

    <script>
      function myFunction() {
        var btn = document.createElement("BUTTON");
        var t = document.createTextNode("CLICK ME");
        btn.appendChild(t);
        document.body.appendChild(btn);

        var prgh = document.createElement("P");
        var txt = document.createTextNode("test");
        prgh.appendChild(txt);
        document.body.appendChild(prgh);

        var inp = document.createElement("INPUT");
        inp.setAttribute("type", "text");
        inp.setAttribute("value", "name");
        inp.style.color = "red";
        document.body.appendChild(inp);
      }
    </script>
  </body>
</html>
```

NODELIST

Node list

- node list와 node의 갯수
 - `getElementsByTagName()` 메소드를 이용하면 node list를 반환 받을 수 있다.
 - 반환 받은 node list에는 node의 개수를 나타내는 `length`라는 속성 (property)이 있다.

```
var nl = document.getElementsByTagName("p");  
documnt.write (nl.length);
```

- node list는 array가 아니다.
 - array처럼 index로 참조하고, loop를 돌릴 수 있지만 array 연산은 불가능하다.

HTML DOM EVENT

HTML Event

- 이벤트가 발생할 때 등록된 동작을 취한다.

onEventname = Javascript function or codes

- 발생 가능한 이벤트 들
 - 사용자가 마우스를 클릭할 때
 - onclick
 - onmousedown
 - onmouseup
 - 마우스가 html element 위에 있을 때
 - onmouseover
 - onmouseout

HTML Event

- input field가 변했을 때
 - onchange:
- HTML form이 제출될 때
 - onsubmit
- 사용자가 keyboard를 입력할 때
 - onkeypress

EventListener

- HTML Element에서 하나의 이벤트에 여러 개의 이벤트 핸들러를 등록할 수 있다.

```
<html>
  <body>
    <button id="test">Try it</button>

    <p id="demo"></p>

    <script>
      var obj = document.getElementById("test");

      function displayDate() {
        document.getElementById("demo").innerHTML = Date();
      }
      function fontcolor(t) {
        t.style.color = "lightblue";
      }

      obj.addEventListener("click", displayDate);
      obj.addEventListener("click", function() {fontcolor(document.getElementById("demo"));});
    </script>
  </body>
</html>
```

FORM

JavaScript Form

- JavaScript를 이용하면 서버로 제출하기 전에 클라이언트에서 입력 값들의 오류를 검출할 수 있다.
 - 비어 있는 항목이 없는지 확인.
 - 필드에 입력 값이 유효한 값인지 확인.
 - 필드에 입력 값이 정확한 값인지 확인.
- 클라이언트 단에서의 오류 검출은 서버의 부담을 줄여 줄 수 있다.

JavaScript Form

- Form으로의 접근
 - 기존의 getElementById, getElementsByName, getElementsByTagName 등의 메소드를 이용해서 접근할 수 있다.
 - 다른 방법으로는 DOM의 form object를 사용할 수 있다.
 - document.forms.<form의name또는id> 또는 document.<form의name또는id>을 사용하면 특정 이름을 가진 form 문에 접근할 수 있다.
 - 또는 document.forms["form의name또는id"]로 접근할 수 있다.

JavaScript Form

```
<form id="myForm" name="myForm" method="post">  
  Name:  
  <input type="text" name="fname">  
  <input type="submit" value="Submit">  
</form>
```

```
var obj = document.forms.myForm;
```

```
var obj = document.forms["myForm"];
```

```
var document.forms.myForm.fname;
```

```
var document.forms["myForm"]("fname");
```

-
- Form이 제출 될 때 양식 검토
 - submit 이벤트 핸들러를 이용한다.
 - onSubmit 에는 동작이나 function 만을 등록하는 것이 아니라 이에 대한 반환 값을 return 하도록 해주어야 한다.

`onSubmit="return chkval(this) "`

- return이 없다면 지정된 함수만을 수행하고 submit 과정이 그대로 진행 된다.
- 지정된 함수의 동작에 따라 submit 과정의 진행 여부를 결정하고 싶다면 return 을 받도록 해주어야 한다.
- 반환 값이 true 이면 submit이 진행되고, 반환 값이 false이면 submit이 진행되지 않는다.

```

</font>
<script>
    function check_value(myform)
    {
    }
</script>

```

END