

HTML DOM

1. HTML DOM이란?

HTML DOM 은 HTML documents에 접근과 조작에 대한 표준을 정의한다.

2. DOM이란?

DOM 은 W3C (World Wide Web Consortium) 표준이다.

DOM은 HTML 과 XML과 같은 documents에 접근에 대한 표준을 정의한다.

“ W3C Document Object Model (DOM)은 프로그램과 스크립트가 document의 content, structure, 그리고 style에 동적으로 접근하고 업데이트하는 것을 허락하는 platform이고 language-neutral (언어 중립) interface이다.”

DOM은 세 가지의 다른 parts/levels로 분리된다.

- Core DOM - 구조화된 document에 대한 표준 모델.
- XML DOM - XML documents에 대한 표준 모델.
- HTML DOM - HTML documents에 대한 표준 모델.

DOM 은 모든 document elements의 objects 와 properties, 그리고 그것들에 접근하기 위한 methods (interface)를 정의한다.

3. XML DOM이란?

XML DOM 은 모든 XML document elements의 objects 와 properties, 그리고 그것들에 접근하기 위한 methods (interface)를 정의한다.

4. HTML DOM이란?

- HTML에 대한 표준 오브젝트 모델이다.
- HTML에 대한 표준 프로그래밍 인터페이스이다.
- Platform과 언어에 독립적이다.
- W3C 표준이다.

HTML DOM은 모든 HTML elements의 objects와 properties, 그리고 그것들에 접근하기 위한 methods(interface)를 정의한다. 다른 말로하면 HTML DOM은 HTML element들을 얻거나, 변경하거나, 추가하거나, 삭제하는 방법에 대한 표준이다.

HTML DOM Node Tree

1. HTML DOM Nodes

HTML DOM은 node-tree로서 HTML document를 보며, tree안의 모든 node들은 서로 각각 관계를 가지고 있다.

2. HTML DOM Node Tree

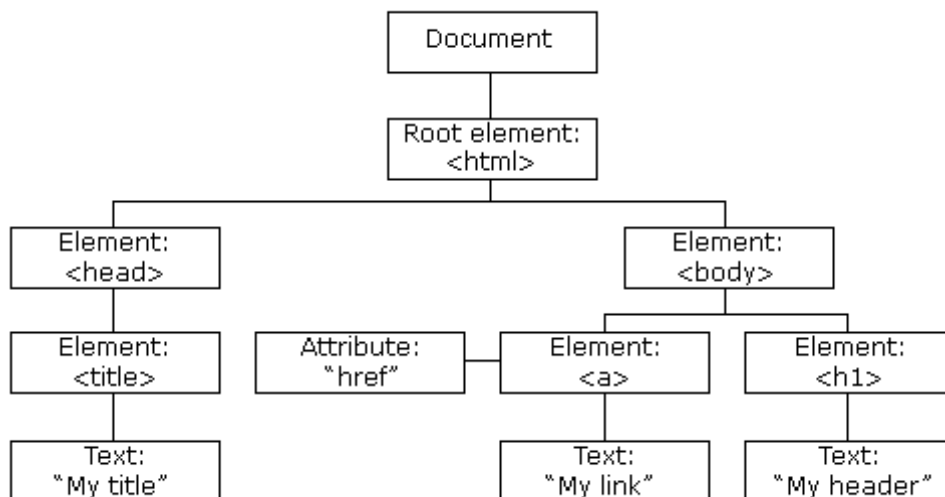
HTML DOM은 tree구조로 HTML document를 본다. 이 구조를 Node-Tree라고 한다.

모든 node들은 tree를 통해 접근할 수 있다.

그들의 content들은 변경 또는 삭제될 수 있고, 새로운 element를 생성할 수 있다.

아래의 node tree는 node들의 집합을 보여주고 그들 사이의 관계를 보여준다.

tree는 root node에서 시작하고, tree의 마지막 레벨에 text node들의 가지가 있다.

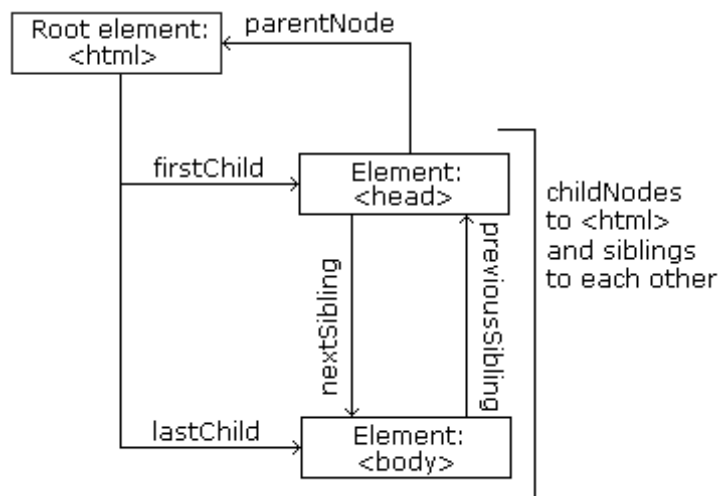


3. Node Parents, Children, and Siblings

node tree안의 node들은 서로 각각 계층적인 관계를 가지고 있다.

parent, child, 그리고 sibling(형제, 자매)라는 용어는 관계를 설명하기 위해 사용되어 진다. Parent node들은 children을 가지고 있다. 동일한 레벨의 children은 siblings라고 불린다.

- node tree에서, top node는 root라고 불린다.
- root를 제외한 모든 node는 하나의 parent node를 가지고 있다.
- node는 다수의 children을 가질 수 있다.
- leaf는 children이 없는 node이다.
- sibling들은 동일한 parent의 node들이다.



4. HTML 구조 Sample

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
</body>
</html>
```

위의 HTML로부터

- <html>node는 parent node를 가지고 있지 않다. root node이다.
- <head>와 <body> node의 parent node는 <html> node 이다.
- "Hello world!" text node의 parent node는 <p> node이다.

그리고

- <html>node는 <head>와 <body> 2개의 child node를 가지고 있다.
- <head>node는 <title> node 1개의 child node를 가지고 있다.
- <title>node 또한 text node "DOM Tutorial" 하나의 child node를 가지고 있다.
- <h1>과 <p>node는 siblings이다. 그리고 둘 다 <body>의 child node이다.

First Child / Last Child

위의 HTML로부터

- <head>element는 <html> element의 첫 번째 child이고,
 <body>element는 <html> element의 마지막 child 이다.
- <h1>element는 <body> element의 첫 번째 child이고,
 <p>element는 <body> element의 마지막 child 이다.

HTML DOM Properties and Methods

Property들과 Method들은 HTML DOM의 Programming Interface를 정의한다.

1. Programming Interface

DOM에서 HTML documents는 node오브젝트들의 집합으로 구성된다.
node들은 JavaScript 또는 다른 프로그래밍 언어들을 사용하여 접근할 수 있다.

DOM의 Programming Interface는 표준 Property들과 Method들을 사용하여 정의된다.

Properties는 그것이 무엇인지를 나타낸다. (예) node의 이름)

Methods는 그것이 하는 일을 나타낸다. (예) node를 제거)

2. HTML DOM Methods

a.innerHTML	a의 text 값	a.parentNode	a의 parent node
a.nodeName	a의 이름	a.childNodes	a의 child nodes
a.nodeValue	a의 값	a.attributes	a의 attributes nodes

getElementById(id)	명시된 id를 사용하여 element를 얻는다.
getElementsByName(name)	명시된 tag name들을 사용하여 모든 element들을 얻는다.
a.appendChild(node)	a에 child node를 삽입한다.
a.removeChild(node)	a로부터 child node를 삭제한다.
a.createAttribute()	a에 Attribute를 생성한다.
a.createElement()	a에 element를 생성한다.
a.createTextNode()	a에 TextNode를 생성한다.

3. The innerHTML Property

element의 content를 얻거나 변경하는 가장 쉬운 방법은 innerHTML property를 사용하는 것이다. innerHTML은 W3C DOM specification의 부분이 아니다. 하지만, 모든 주요 브라우저에서 지원된다.

innerHTML property는 HTML element들의 content들을 (<html> 과 <body>를 포함) 반환 또는 대체하는데 유용하고, 또한 동적으로 변경된 페이지의 소스를 보는데 사용되어 질 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

  <p id="intro">Hello World!</p>

  <script>
    var txt=document.getElementById("intro").innerHTML;
    document.write(txt);
  </script>

</body>
</html>
```

※ 위 예제에서 getElementById는 modthod인 반면, innerHTML은 property이다.

4. childNodes and nodeValue

element의 content를 얻기 위해 childNodes 와 NodeValue property를 사용할 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

  <p id="intro">Hello World!</p>

  <script>
    var txt=document.getElementById("intro").childNodes[0].nodeValue;
    document.write(txt);
  </script>

</body>
</html>
```

※ 위 예제에서 getElementById는 method인 반면, childNodes 와 nodeValue는 property이다.

HTML DOM Access

※ DOM을 사용하여, HTML document안의 모든 node에 접근할 수 있다.

세 가지 방법을 사용하여 node에 접근할 수 있다.

- 1) getElementById() method를 사용함으로써
- 2) getElementsByTagName() method를 사용함으로써
- 3) node 관계를 사용하여 node tree를 탐색함으로써

1. The getElementById() Method

getElementById() method는 명시된 ID를 사용하는 element를 반환한다.

```
node.getElementById("id");
```

```
document.getElementById("intro");  
id="intro"를 사용하는 element를 얻는다.
```

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
  <meta charset="UTF-8">  
  <title>HTML DOM</title>  
</head>  
<body>  
  
  <p id="intro">Hello World!</p>  
  <p>This example demonstrates the <b>getElementById</b> method!</p>  
  
  <script>  
    var txt=document.getElementById("intro");  
    document.write("<p>이번 예제의 결과 값: " + txt.innerHTML + "</p>");  
  </script>  
  
</body>  
</html>
```

2. The `getElementsByName()` Method

`getElementsByName()` method는 명시된 tag name을 사용하는 모든 element들을 반환한다.

```
node.getElementsByTagName("tagname");
```

```
document.getElementsByTagName("p");  
document안의 모든 <p> element들의 nodeList를 반환한다.
```

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
  <meta charset="UTF-8">  
  <title>HTML DOM</title>  
</head>  
<body>  
  
  <p id="intro">Hello World!</p>  
  <p>The DOM is very useful!</p>  
  <p>This example demonstrates the <b>getElementsByTagName</b> method.</p>  
  
  <script>  
    var txt=document.getElementsByTagName("p");  
    document.write("Text of first paragraph: " + txt[0].innerHTML);  
  </script>  
  
</body>  
</html>
```

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
  <meta charset="UTF-8">  
  <title>HTML DOM</title>  
</head>  
<body>  
  
  <p>Hello World!</p>  
  
  <div id="main">  
    <p>The DOM is very useful.</p>
```

```

    <p>This example demonstrates the <b>getElementsByTagName</b>
method</p>
</div>

<script>
    var txt=document.getElementById("main").getElementsByTagName("p");
    document.write("First paragraph inside the div: " + txt[0].innerHTML);
</script>

</body>
</html>

```

위 예제는 id="main"을 사용하는 element의 자손인 모든<p> element들의 nodeList를 반환한다.

3. DOM Node List

getElementByTagName() method는 node-list를 반환한다. node-list는 node의 배열이다.

```
var txt=document.getElementsByTagName("p");
```

위의 코드는 node-list안의 모든 <p> node들을 선택한다.
node들은 index 번호로 접근될 수 있다. 두 번째 <p>에 접근하기 위해 다음과 같이 작성한다.

```
y=txt[1];
```

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>HTML DOM</title>
</head>
<body>

    <p>Hello World!</p>
    <p>The DOM is very useful!</p>

<script>
var txt=document.getElementsByTagName("p");
document.write("The innerHTML of the second paragraph is: " + txt[1].innerHTML);
</script>

```

```
</body>
</html>
```

※ index는 0부터 시작한다.

4. DOM Node List Length

length property는 node-list안의 node들의 수를 정의한다.
length property를 사용함으로써 node-list를 loop할 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

  <p>Hello World!</p>
  <p>The DOM is very useful!</p>
  <p>This example demonstrates the <b>length</b> property.</p>

  <script>
var txt=document.getElementsByTagName("p");

for (i=0;i<txt.length;i++){
  document.write(txt[i].innerHTML);
  document.write("<br />");
}
</script>

</body>
</html>
```

5. Navigating Node Relationships

parentNode, firstChild, lastChild 세 개의 property는 document structure를 따르고 document안을 돌아다니는 것을 허락한다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

<p>Hello World!</p>

<div>
  <p>The DOM is very useful!</p>
  <p>This example demonstrates node relationships.</p>
</div>

</body>
</html>
```

위 HTML 코드에서,
첫 번째 p element는 body element의 첫 번째 child node(firstChild)이고,
div element는 body element의 마지막 child node(lastChild)이다.
첫 번째 p element와 div element의 parent node(parentNode)는 body element이고,
div element안의 p element들의 parent node는 div element이다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

<p id="intro">Hello World!</p>

<script>
var txt=document.getElementById("intro");
```

```
document.write(txt.firstChild.nodeValue);  
</script>
```

```
</body>  
</html>
```

firstChild property는 element의 text에 접근하기 위해 사용되어질 수 있다.

6. DOM Root Nodes

parentNode, firstChild, lastChild 세 개의 property는 document structure를 따르고 document안을 돌아다니는 것을 허락한다.

tags에 접근하는 것을 허락하는 특별한 2가지의 property가 있다.

- document.documentElement - document의 root node를 반환한다.
- document.body - <body> tag에 direct access를 제공한다.

```
<!DOCTYPE html>  
<html lang="ko">  
<head>  
  <meta charset="UTF-8">  
  <title>HTML DOM</title>  
</head>  
<body>  
  
  <p>Hello World!</p>  
  <div>  
    <p>The DOM is very useful!</p>  
    <p>This example demonstrates the <b>document.body</b> property.</p>  
  </div>  
  
  <script>  
    alert(document.body.innerHTML);  
  </script>  
  
</body>  
</html>
```

HTML DOM Node Information

※ nodeName, nodeValue, 그리고 nodeType property들은 node들에 대한 정보를 포함한다.

1. Node Properties

HTML DOM에서, 각각의 node들은 object이다.

object들은 JavaScript에 의해 접근과 변경될 수 있는 method들과 property들을 가지고 있다.

세 가지 중요한 node property들이 있다.

- nodeName
- nodeValue
- nodeType

2. The nodeName Property

nodeName property는 node의 이름을 명시한다.

- nodeName은 읽기 전용이다.
- element node의 nodeName은 tag name과 동일하다.
- attribute node의 nodeName은 attribute name이다.
- text node의 nodeName은 항상 #text이다.
- document node의 nodeName은 항상 #document이다.

※ nodeName은 항상 HTML element의 대문자 tag name을 포함한다.

3. The nodeValue Property

nodeValue property는 node의 값을 명시한다.

- element에 대한 nodeValue는 정의되지 않는다.
- text node에 대한 nodeValue는 text 자신이다.
- attribute node에 대한 nodeValue는 attribute value이다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
```

```
<body>
```

```
<p id="intro">Hello World!</p>
```

```
<script>
```

```
var txt=document.getElementById("intro");
```

```
document.write(txt.firstChild.nodeValue);
```

```
</script>
```

```
</body>
```

```
</html>
```

4. The nodeType Property

nodeType property는 node의 type을 반환한다. nodeType은 읽기 전용이다.

※ 가장 중요한 node type들이다.

Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

HTML DOM - Change HTML Elements

※ HTML element들은 JavaScript, HTML DOM과 events를 사용하여 변경될 수 있다.

1. Change an HTML Element

HTML DOM과 JavaScript는 HTML element들의 inner content와 attributes를 변경할 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

<script type="text/javascript">
  document.body.style.backgroundColor="yellow";
</script>

<p>스크립트로 배경컬러 바꾸기</p>

</body>
</html>
```

2. Change the Text of an HTML Element - innerHTML

element의 content를 얻거나 변경하는 가장 쉬운 방법은 innerHTML property를 사용하는 것이다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

<p id="p1">Hello World!</p>

<script type="text/javascript">
<!--
  document.getElementById("p1").innerHTML="New text!";
//-->
</script>

<p>스크립트로 element의 text 변경하기</p>

</body>
</html>
```

3. Change an HTML Element Using Events

event handler는 event가 발생할 때 코드를 실행하는 것을 허락한다. event들은 사용자가 element를 클릭할 때, 페이지가 로드될 때, 폼이 submit될 때 등 브라우저에 의해 생성되어진다.

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<body>

<input type="button" onclick="document.body.style.backgroundColor='lightblue';"
value="Change background color">

<p>Click 이벤트가 발생하면 &lt;body> element의 background color 변경하기</p>

</body>
</html>
```

4. Change the Text of an Element - with a Function

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<head>
<script type="text/javascript">
<!--
function ChangeText(){
  document.getElementById("p1").innerHTML="New text!";
}
//-->
</script>
</head>

<body>

<p id="p1">Hello world!</p>
<input type="button" onclick="ChangeText()" value="Change text" />

</body>
</html>
```

5. Using the Style Object

각각의 HTML element의 style object는 각각의 style을 표현한다.

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<head>
<script type="text/javascript">
<!--
function ChangeBackground(){
    document.body.style.backgroundColor="#ffffcc";
}
//-->
</script>
</head>

<body>

<p id="p1">함수를 사용하여 버튼이 클릭될 때 &lt;body&gt; element의 배경 style
변경하기</p>
<input type="button" onclick="ChangeBackground()" value="Change background
color" />

</body>
</html>
```

6. Change the font and color of an Element

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<head>
<script type="text/javascript">
<!--
function ChangeStyle(){
    document.getElementById("p1").style.color="blue";
    document.getElementById("p1").style.fontFamily="Verdana";
    document.getElementById("p1").style.fontSize="30px";
}
//-->
</script>
</head>

<body>

<p id="p1">Hello world!</p>
<input type="button" onclick="ChangeStyle()" value="Change style" />

</body>
</html>
```

7. Creating New Html Elements

HTML DOM에 새로운 요소를 추가하려면 먼저 요소 (요소 노드)를 만든 다음 기존 요소에 추가해야 한다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>
<div id="d1">
  <p id="p1">This is a paragraph.</p>
  <p id="p2">This is another paragraph.</p>
</div>

<script>
  var para=document.createElement("p");
  var node=document.createTextNode("This is new.");
  para.appendChild(node);

  var element=document.getElementById("d1");
  element.appendChild(para);
</script>

</body>
</html>
```

HTML DOM - Events

1. Events

※ event들은 JavaScript에 의해 탐지될 수 있는 action들이다.

웹 페이지의 모든 element는 JavaScript 함수들을 trigger할 수 있는 event들을 가지고 있다. 예를 들어, 사용자가 버튼을 클릭할 때 함수가 실행될 것을 나타내기 위해 button element의 onClick event를 사용할 수 있다. HTML elements안에 event들을 정의한다.

event들은 일반적으로 함수와 결합되어 사용되어지고, 함수는 event가 발생하기 전 실행되지 않는다.

- 마우스 클릭
- 웹 페이지 또는 image 로딩
- 웹 페이지의 hot spot에 mouse over
- HTML form 안의 input box를 선택
- HTML form을 submitting
- keystroke

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML DOM</title>
</head>
<body>

<h1 onclick="this.innerHTML='Oops!'">Click on this text!</h1>

</body>
</html>
```

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

```

function changetext(id){
  id.innerHTML="Oops!";
}
//-->
</script>
</head>

<body>

<h1 onclick="changetext(this)">Click on this text!</h1>

</body>
</html>

```

2. HTML Event Attributes

```

<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<head>
<script type="text/javascript">
<!--
function displayDate(){
  document.getElementById("demo").innerHTML=Date();
}
//-->
</script>
</head>

<body>

<p>버튼을 클릭하면 displayDate() 함수를 실행한다.</p>

<button onclick="displayDate()">Try it</button>
<p id="demo"></p>

</body>
</html>

```


3. Assign Events Using the HTML DOM

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">

<body>

<p>버튼을 클릭하면 displayDate() 함수를 실행한다.</p>

<button id="myBtn">Try it</button>
<p id="demo"></p>

<script type="text/javascript">
<!--
document.getElementById("myBtn").onclick=function(){
    displayDate()
};

function displayDate(){
    document.getElementById("demo").innerHTML=Date();
}
//-->
</script>
</body>
</html>
```

4. The onload and onunload Events

onload 와 onUnload event들은 사용자가 페이지에 들어오거나 나갈 때 trigger되어 진다.

onload event는 방문자의 브라우저 type과 version을 검사하기 위해 자주 사용되고, 정보를 바탕으로 웹 페이지의 적당한 버전을 load한다.

onload 와 onUnload event들은 둘 다 사용자가 페이지를 들어오거나 나갈 때 설정되어야 하는 cookie들을 다루기 위해 자주 사용된다. 예를 들어, 사용자의 첫 번째 방문 때 사용자의 이름을 물어보는 팝업을 가질 수 있다. 이름은 cookie에 저장되어 진다. 방문자가 다음에 재 방문할 때, "Welcome Daisy!" 와 같은 환영메시지를 볼 수 있다.

5. onFocus, onBlur and onChange

onFocus, onBlur 그리고 onChange event들은 form fields의 validation과 함께 자주 사용되어 진다.

```
<!DOCTYPE html>
<html lang="ko">
<meta charset="UTF-8">
<head>
<script type="text/javascript">
<!--
function myFunction(){
var txt=document.getElementById("fname");
txt.value=txt.value.toUpperCase();
}
//-->
</script>
</head>

<body>

Enter your name: <input type="text" id="fname" onchange="myFunction()">

<p>텍스트 필드에 입력한 내용을 대문자로 변경하기</p>

</body>
</html>
```