



# DOM API (update)



알고리즘 및 정보보호 연구실

2005. 05. 30

발표자 : 이 승 구



# Contents

---



- ❖ What is DOM?
- ❖ DOM API (Update)
  - ❖ Node
  - ❖ Element
  - ❖ Document
  - ❖ Attr
- ❖ Reference



# What's DOM?



- ❖ DOM (Document Object Model)?
  - ❖ HTML과 XML 문서의 data를 효과적으로 다루기 위한 API
  - ❖ 문서의 논리적 구조와 문서가 접근되고 다루어지는 방법 정의
  - ❖ 객체(Object)로 모델링, 트리(tree)구조의 논리적인 형태로 표현
  
- ❖ DOM 의 장점
  - ❖ 문법으로부터 내용을 추출
  - ❖ 내부 문서 조작의 단순화
  - ❖ 전형적인 계층적 또는 관계형 DATABASE 구조와 유사

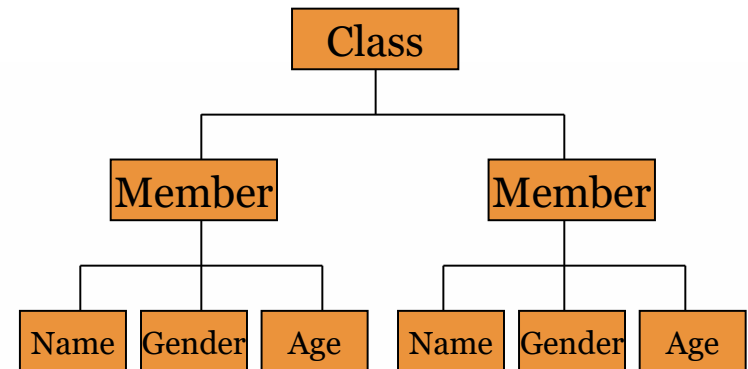


# DOM API (update)



## ❖ XML 문서에서의 DOM 해석

```
<Class>  
  <Member>  
    <Name id="1111">Paul</Name>  
    <Gender>1</Gender>  
    <Age>27</Age>  
  </Member>  
  <Member>  
    <Name id="2222">Jane</Name>  
    <Gender>2</Gender>  
    <Age>25</Age>  
  </Member>  
</Class>
```





# DOM API (update)



## ❖ Node Object

Method	Desc
InsertBefore(newChild, refChild)	자식 목록 마지막에 Node 추가
replaceChild(newChild, oldChild)	자손 리스트에 oldchild와 newchild node를 갖음 oldchild에 newchild node 추가
removeChild(oldChild)	자손 리스트에 자손 node 삭제
appendChild(newChild)	자손 리스트 마지막 인수로 주어진 node 추가
hasChildNodes()	Node가 child를 갖는지 판단
cloneNode(deep)	개체의 참조를 복사하여 컬렉션이나 개체를 반환
normalize()	문서 개체 모델을 정상화 하기 위한 개체 통합

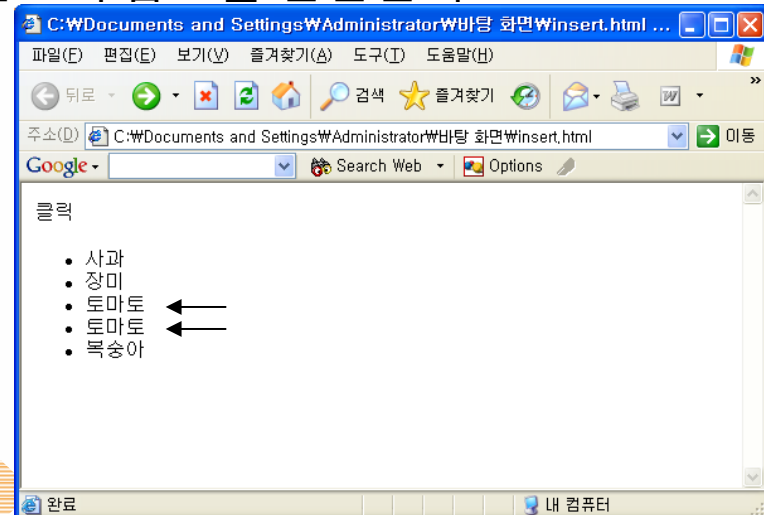


# DOM API (update)



- ❖ insertBefore(newChild, refChild)
  - ❖ newChild
    - ❖ 문서의 계통도에 삽입될 새로운 element를 지정  
createElement 메서드에 의하여 생성될 수 있는 element
  - ❖ refChild
    - ❖ 새로운 element가 위치할 자리를 지정하는 개체
  - ❖ 반환값
    - ❖ 문서에 새로이 삽입된 요소의 참조를 반환한다.

```
1 <SCRIPT>
2 function insertElement(){
3   var nod=document.createElement('LI');
4   ulObj.insertBefore(nod,oLi3);
5   nod.innerText='토마토';
6 }
7 </SCRIPT>
8
9 <SPAN onclick=insertElement()>클릭</SPAN>
10 <UL id="ulObj">
11 <LI id="oLi1">사과</LI>
12 <LI id="oLi2">장미</LI>
13 <LI id="oLi3">복숭아</LI>
14 </UL>
```



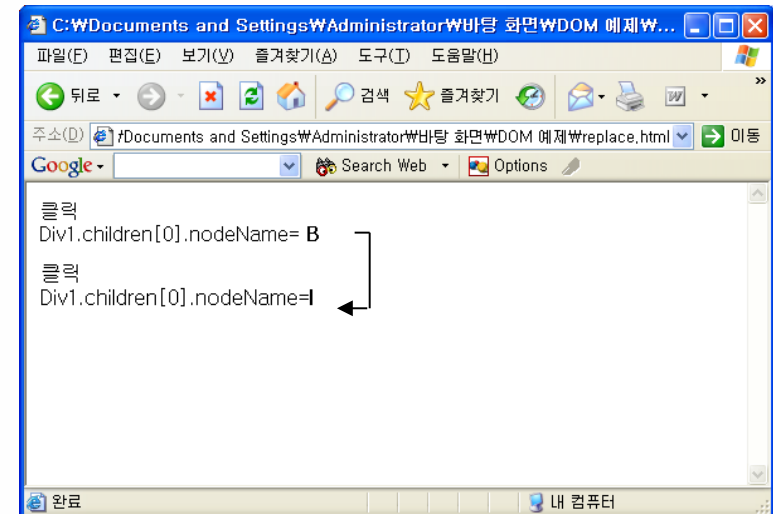


# DOM API (update)



- ❖ **replaceChild(newChild, oldChild)**
  - ❖ **newChild**
    - ❖ 문서에 삽입할 새로운 element를 지정하는 개체이다.
  - ❖ **oldChild**
    - ❖ 교체당할 기존의 element를 지정하는 개체이다.
  - ❖ **반환값**
    - ❖ 교체된 개체를 참조한다.

```
1 <HEAD>
2 <SCRIPT>
3 function replaceElement(){
4     var oChild=Div1.children(0);
5     var textObj = oChild.innerHTML;
6     if (oChild.tagName=='B'){
7         oNewChild=document.createElement('I');
8         Div1.replaceChild(oNewChild,oChild);
9         oNewChild.innerHTML=textObj
10    } else {
11        oNewChild=document.createElement('B');
12        Div1.replaceChild(oNewChild,oChild);
13        oNewChild.innerHTML=textObj
14    }
15    showA.innerHTML=Div1.children[0].nodeName;
16 }
```





# DOM API (update)



## ❖ removeChild(oldChild)

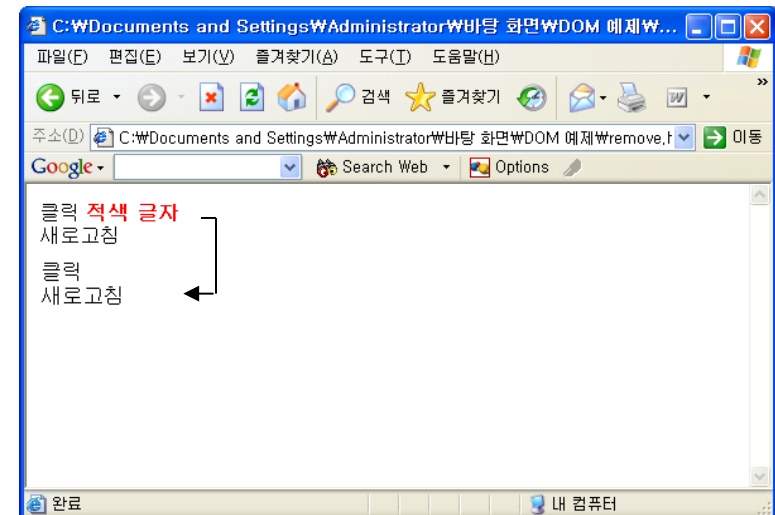
### ❖ oldChild

❖ 문서로부터 제거할 element를 지정하는 개체이다.

### ❖ 반환값

❖ 제거된 개체의 참조이다.

```
1 <HEAD>
2 <SCRIPT>
3 function removeElement(){
4   try {
5     var oChild=Div1.children(0);
6     Div1.removeChild(oChild);
7   } catch(x) {
8     alert('적색 글자의 개체가 이미 제거 되었습니다.')
9   }
10 }
11 </SCRIPT>
12 <BODY>
13 <DIV id=Div1 onclick="removeElement()">클릭
14 <B style=color:red>적색 글자</B></DIV>
15 <DIV onclick="document.location.reload();">새로고침</DIV>
16 </BODY>
```







# DOM API (update)



## ❖ appendChild(newChild)

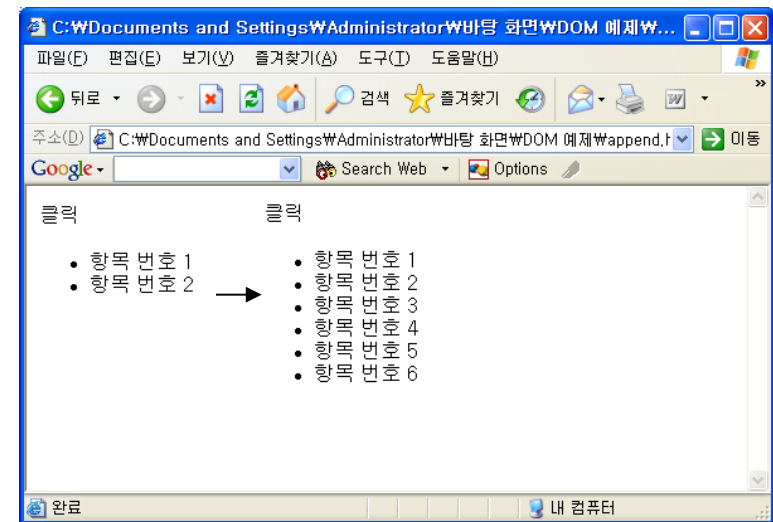
### ❖ newChild

❖ 첨부되는 element를 지정하는 개체이다.

### ❖ 반환값

❖ 첨부된 자식 element를 참조하는 개체를 반환한다.

```
1 <SCRIPT>
2 function appChild(){
3   if (oList.all.length<6){
4     var oNewNode=document.createElement('LI');
5     oList.appendChild(oNewNode);
6     oNewNode.innerText='항목 번호 '+oList.all.length;
7   }
8 }
9 </SCRIPT>
10
11 <SPAN onclick="appChild()">클릭</SPAN>
12 <UL id=oList>
13 <LI>항목 번호 1
14 <LI>항목 번호 2
15 </UL>
```





# DOM API (update)

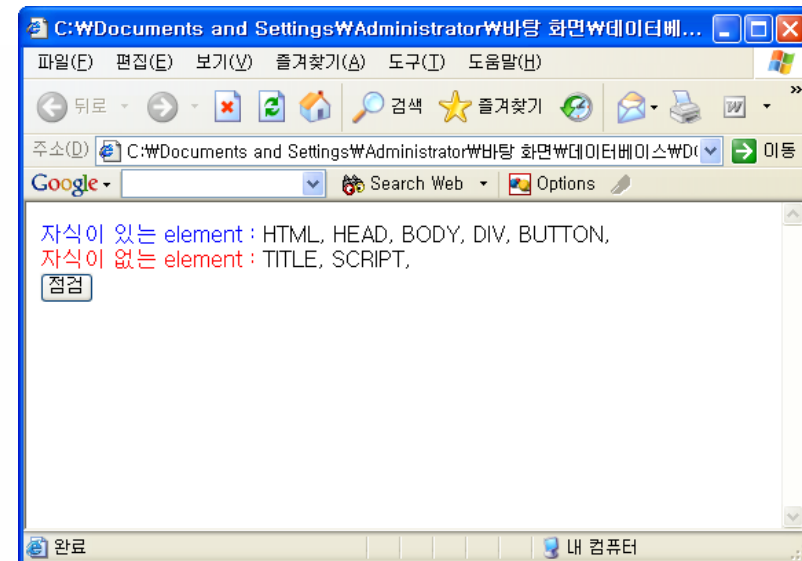


## ❖ hasChildNodes()

### ❖ 반환값

- ❖ 자식개체를 가지고 있는가를 나타내는 부울값이다.

```
1 <DIV id=showA title="결과">결과 표시창</DIV>
2 <BUTTON onClick="doShow()">점검</BUTTON>
3 <SCRIPT>
4 function doShow(){
5     coll=document.all;
6     hasChild=noChild="";
7     for (i=0;i< coll.length;i++){
8         if(coll[i].hasChildNodes()) hasChild+=coll[i].nodeName+' ';
9         else noChild+=coll[i].nodeName+' ';
10    }
11    str='<FONT color=blue>자식이 있는 element :</FONT> '+hasChild+'<BR>';
12    str+='<FONT color=red>자식이 없는 element :</FONT> '+noChild+'<BR>';
13    showA.innerHTML=str;
14 }
15 </SCRIPT>
```





# DOM API (update)



## ❖ cloneNode(deep)

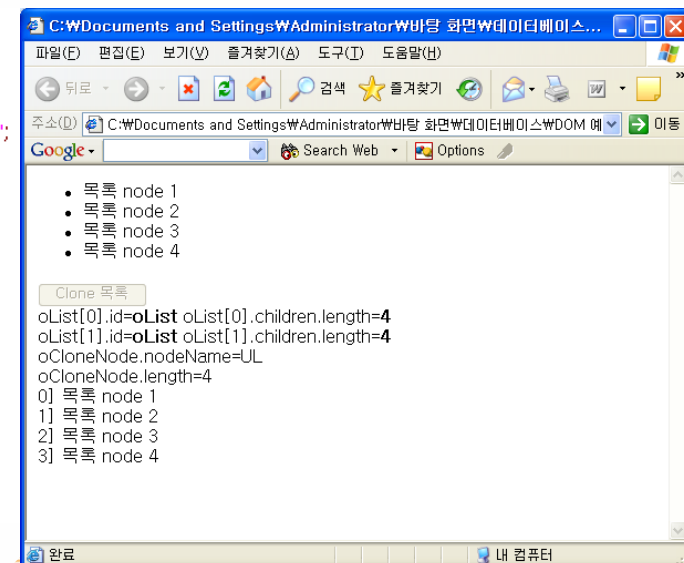
### ❖ Boolean

- ❖ 디폴트이며, 복사된 개체는 childNode를 포함하지 않는다.
- ❖ 복사된 개체는 childNodes를 갖는다.

### ❖ 리턴값

- ❖ 새로 생성된 node의 참조하는 개체를 반환한다.

```
1 <SCRIPT>
2 function fnClone(){
3   var oCloneNode=oList.cloneNode(true);
4   showB.insertBefore(oCloneNode);
5   str='oList[0].id=<B>'+oList[0].id+'</B> oList[0].children.length=<B>'+oList[0].children.length+'</B><BR>';
6   str+='oList[1].id=<B>'+oList[1].id+'</B> oList[1].children.length=<B>'+oList[1].children.length+'</B><BR>';
7   str+='oCloneNode.nodeName='+oCloneNode.nodeName+'<BR>';
8   str+='oCloneNode.length='+oCloneNode.children.length+'<BR>';
9   for (i=0;i<oCloneNode.children.length;i++){
10     str+=' '+i+' '+oCloneNode.children[i].innerText+'<BR>';
11   }
12   showB.innerHTML=str;
13   cloneButton.disabled=true;
14 }
15 </SCRIPT>
16 <UL id=oList>
17 <LI>목록 node 1
18 <LI>목록 node 2
19 <LI>목록 node 3
20 <LI>목록 node 4
21 </UL>
22 <INPUT id="cloneButton" type="button" value="Clone 목록" onclick="fnClone()">
23 <DIV id=showB></DIV>
```





# DOM API (update)



## ❖ Element Object

Method	Desc
getAttribute(name)	지정된 attribute의 값을 반환
setAttribute(name, value)	지정된 attribute에 값을 설정
removeAttribute(name)	개체로부터 지정한 attribute 속성을 제거
getAttributeNode(name)	name을 속성으로 참조하는 attribute 개체를 반환
setAttributeNode(newAttr)	attribute 개체의 일 부분으로의 노드를 설정
removeAttributeNode(oldAttr)	개체로부터 지정된 attribute 개체를 제거



# DOM API (update)



## ❖ getAttribute(name)

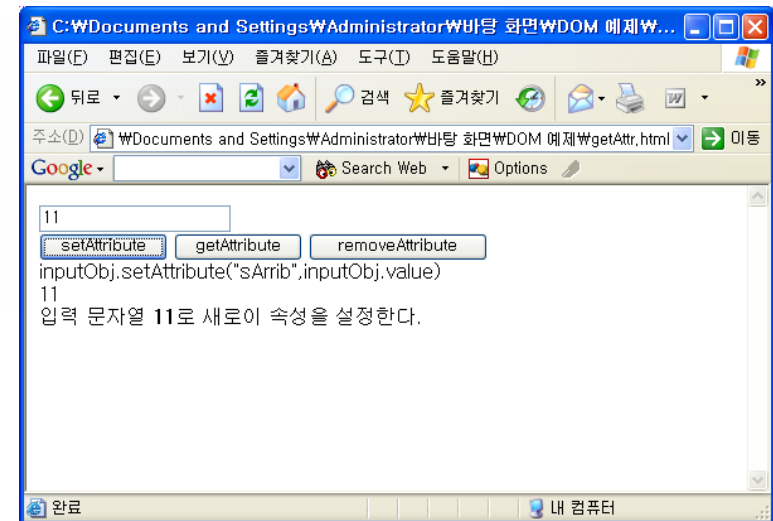
### ❖ name

❖ attribute의 이름을 나타내는 문자열이다.

### ❖ 반환값

❖ attribute에 의하여 정의된 값을 반환한다.

```
1 <STYLE>.userData {behavior:url(#default#userdata);}</STYLE>
2 <SCRIPT>
3 function doSet(){
4   if (inputObj.value){
5     showA.innerHTML='attribute를 입력하십시오.</FONT>';
6     return;
7   }
8   inputObj.setAttribute('sArrib',inputObj.value);
9   inputObj.save('oDataStore');
10  showA.innerHTML=inputObj.setAttribute("sArrib",inputObj.value)<BR>'+simpleGet();
11  showB.innerHTML='입력 문자열 <B>'+inputObj.value+'</B>로 새로이 속성을 설정한다.';
12 }
```





# DOM API (update)



## ❖ `setAttribute(name, value)`

### ❖ `name`

❖ attribute의 이름을 지정하는 문자열이다.

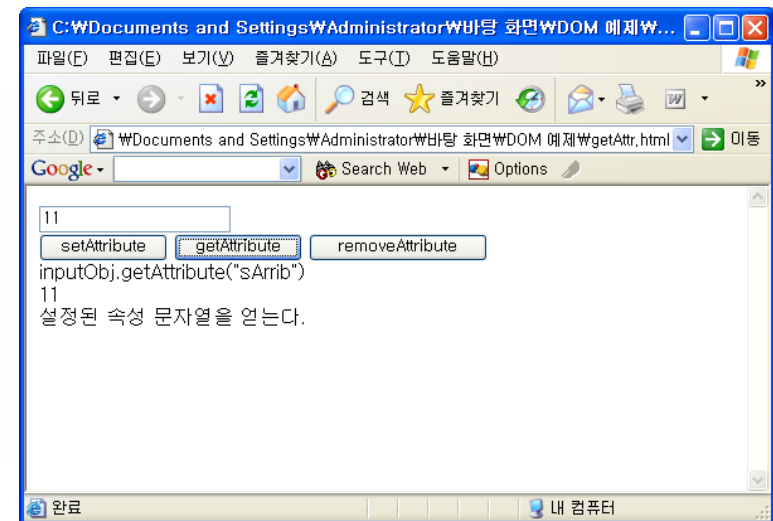
### ❖ `value`

❖ attribute의 할당값을 지정

### ❖ 반환값

❖ 없음

```
13 function doGet(){
14   inputObj.load('oDataStore');
15   inputObj.value=inputObj.getAttribute('sArrib');
16   inputObj.getAttribute("sArrib")
17   showA.innerHTML='inputObj.getAttribute("sArrib")<BR>'+inputObj.getAttribute('sArrib');
18   showB.innerText='설정된 속성 문자열을 얻는다.';
19 }
```





# DOM API (update)



## ❖ removeAttribute(name)

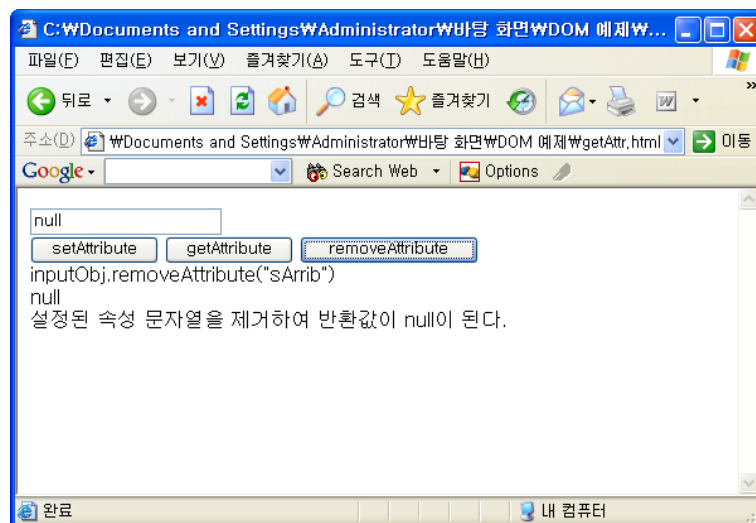
### ❖ name

❖ 제거할 attribute의 이름을 나타내는 문자열이다.

### ❖ 반환값

❖ 성공적으로 Attribute가 제거되었는지 나타내는 boolean값이다

```
25 function doRemove(){
26   inputObj.removeAttribute('sArrib');
27   inputObj.save('oDataStore');
28   showA.innerHTML=inputObj.removeAttribute("sArrib")<BR>'+simpleGet();;
29   showB.innerText='설정된 속성 문자열을 제거하여 반환값이 null이 된다.';
30 }
```





# DOM API (update)



## ❖ Document Object

Method	Desc
<code>createElement(tagName)</code>	지정된 태그의 element를 생성한다.
<code>createDocumentFragment()</code>	새로운 문서를 생성한다.
<code>createTextNode(data)</code>	지정된 값으로 텍스트 문자열 개체를 생성한다.
<code>createAttribute(name)</code>	attribute 개체를 지정된 이름으로 생성한다.
<code>getElementByID(elementID)</code>	ID attribute 의 값에 맞는 처음 개체를 반환한다.
<code>getElementsByTagName(tagname)</code>	TagName attribute의 값에 맞는 처음 개체를 반환한다.





# DOM API (update)



## ❖ createElement(tagName)

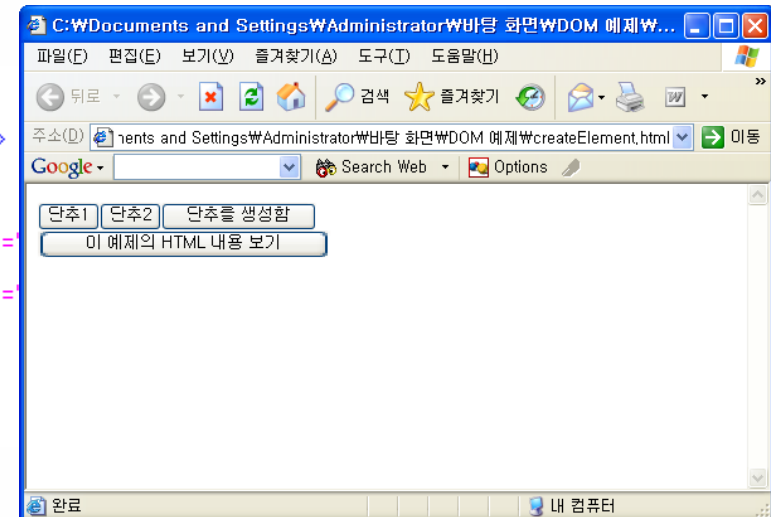
### ❖ tagName

- ❖ element 이름을 지정하는 문자열이다.

### ❖ 반환값

- ❖ 새로 생성된 element를 반환한다.

```
1 <DIV id=example>
2 <FORM id="fname">
3 <INPUT type="button" id="but1" onclick="makeButton()" value="단추를 생성함"><BR>
4 </FORM>
5 <SCRIPT>
6 function makeButton(){
7   newElem=document.createElement('<INPUT type="button" name="button1" value=
8   fname.insertBefore(newElem,fname,but1);
9   newElem=document.createElement('<INPUT type="button" name="button2" value=
10  fname.insertBefore(newElem,fname,but1);
11  // document.body.insertBefore(newElem); 문서의 제일 뒤에 삽입하려면...
12 }
13 </SCRIPT>
14 </DIV>
```



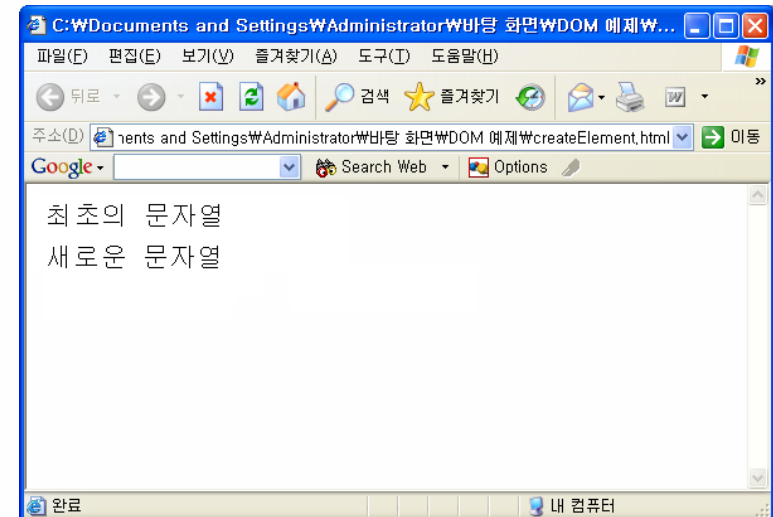


# DOM API (update)



- ❖ createTextNode(data)
  - ❖ data
    - ❖ 텍스트 노드의 노드 값을 지정한다.
  - ❖ 리턴값
    - ❖ 생성된 TextNode 개체이다.

```
1 <SCRIPT>
2 function newNode(){
3   var newObj=document.createTextNode('새로운 문자열');
4   var replaceObj=spanObj.childNodes(0);
5   replaceObj.replaceNode(newObj);
6 }
7 </SCRIPT>
8 <SPAN id="spanObj" onclick="newNode()">최초의 문자열</SPAN>
```



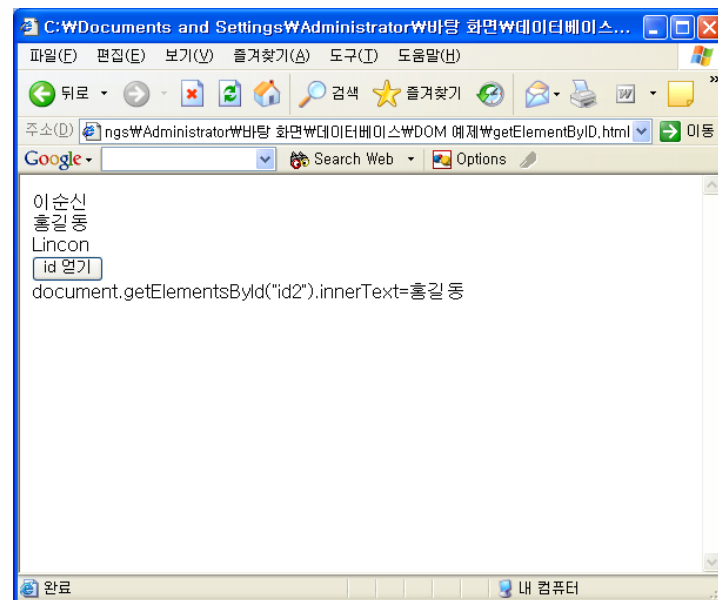


# DOM API (update)



- ❖ getElementById(elementID)
  - ❖ elementID
    - ❖ id 값을 지정하는 문자열이다.
  - ❖ 반환값
    - ❖ 지정된 element ID를 갖는 처음의 개체이다.

```
1 <SCRIPT>
2 function doShow(){
3     showA.innerHTML='document.getElementById("id2").innerHTML='
4     +document.getElementById("id2").innerHTML;
5 }
6 </SCRIPT>
7 <DIV id="id1">이순신</DIV>
8 <DIV id="id2">홍길동</DIV>
9 <DIV id="id3">Lincon</DIV>
▶ 10 <INPUT type="button" value="id 열기" onclick="doShow()">
11 <DIV id="showA">결과 표시창</DIV>
```





# DOM API (update)



- ❖ `getElementByTagName(tagname)`
  - ❖ `tagname`
    - ❖ element의 이름을 나타내는 문자열이다.
  - ❖ 반환값
    - ❖ 지정된 element 이름들을 갖는 개체들의 컬렉션이다.
  
- ❖ `createDocumentFragment()`
  - ❖ 반환값
    - ❖ 새로 생성된 문서를 반환한다.



# DOM API (update)



- ❖ `createAttribute(name)`
  - ❖ Name
    - ❖ attribute 개체의 이름을 설정하는 문자열이다.
  - ❖ 반환값
    - ❖ 새로운 attribute 개체의 참조를 반환한다.



# DOM API (update)



## ❖ Attr Object

Property	Desc
getName	Attribute 이름 반환
getSpecified	속성이 명시 되었는지 판단 Boolean
getValue	Attribute 값을 반환



# Reference



- ❖ (JUMP TO)XML : Web Programing
  - ❖ 허준회 등...저, 정보게이트, 2001
- ❖ <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>
- ❖ <http://koxo.com/lang/js/method/jsList.html>