

08

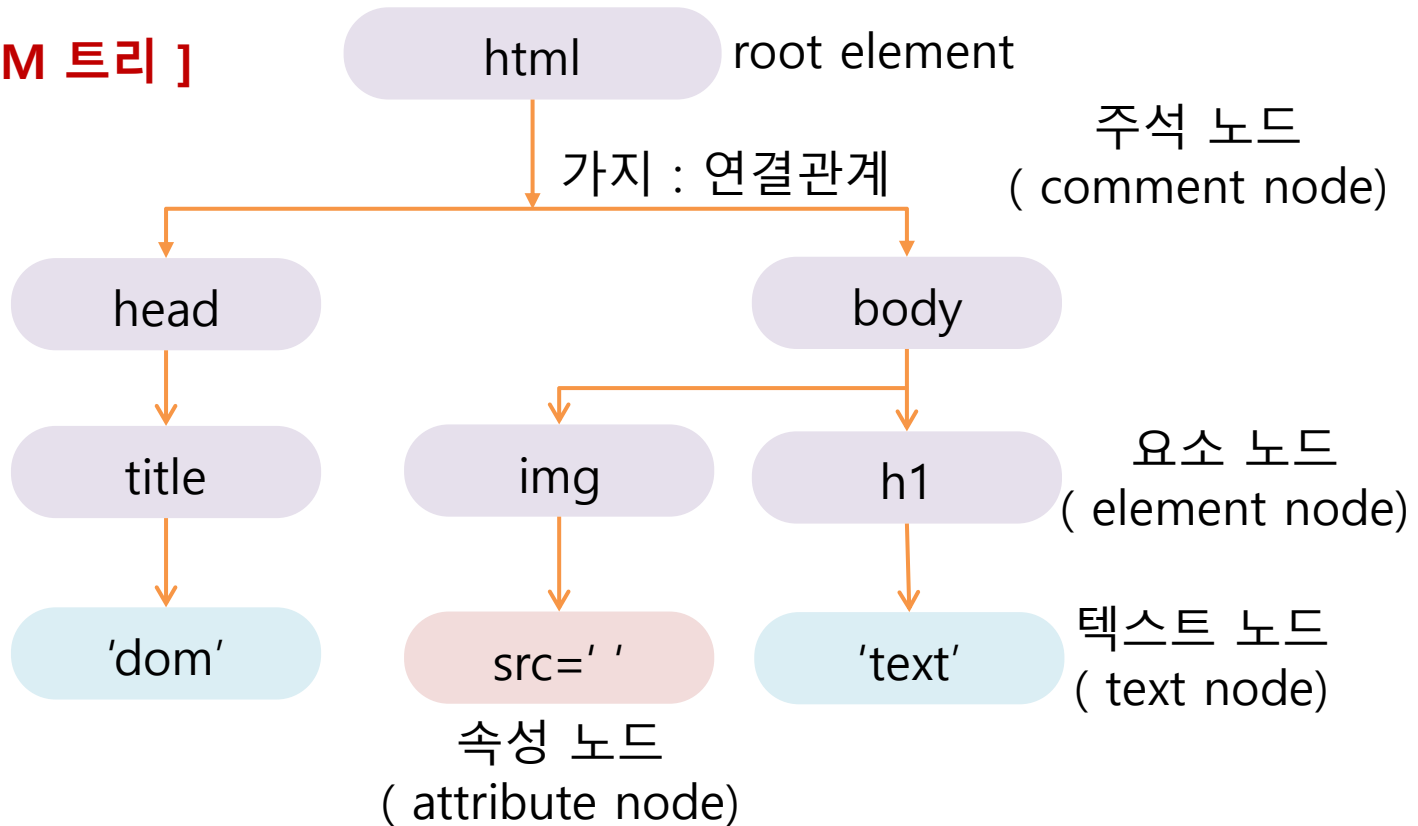
DOM과 Event

- DOM 객체
- 이벤트
- HTML 제어

❖ document 객체(DOM)

- 웹 문서와 관련된 객체의 집합
- HTML 페이지 읽으면서 문서 객체 생성(정적)
- Javascript 로 문서 객체 생성(동적)

[DOM 트리]



❖ document 객체(DOM)

객체 노드 생성

	메서드	설명
1	createElement()	Element node 생성
2-1	createTextNode()	Text node 생성
	appendChild()	Text node를 element node 자식 노드로 추가
2-2	createAttribute()	속성 노드 생성
	setAttributeNode()	속성 노드를 요소 노드에 연결
3	appendChild()	생성된 요소 노드를 부모 노드에 추가

❖ document 객체(DOM)

```
//<p class="accent">주문 완료</P>
```

```
// 1. element node : <p>
```

```
let newP = document.createElement('p');
```

```
// 2. text node : 주문 완료
```

```
let newText = document.createTextNode('주문 완료');
```

```
// 3. 자식노드 추가
```

```
newP.appendChild(newText);
```

```
// 4. 부모노드에 추가
```

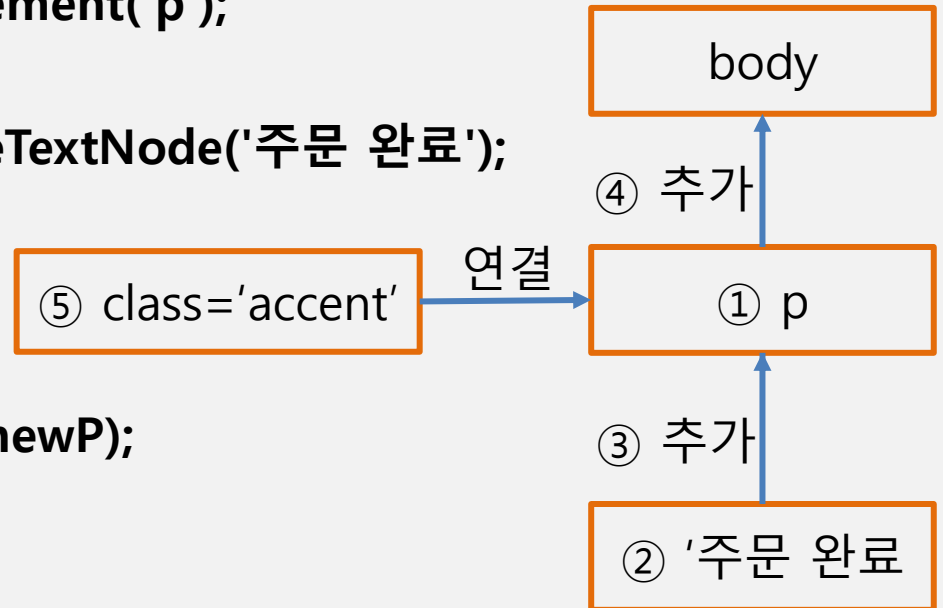
```
document.body.appendChild(newP);
```

```
// 5. 속성 노드 생성, 연결
```

```
let newAttr = document.createAttribute('class');
```

```
newAttr.value = 'accent';
```

```
newP.setAttributeNode(newAttr); newP.setAttribute('class', 'accent');
```



❖ document 객체(DOM)

문서 객체 가져오기

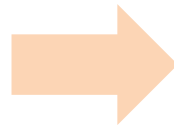
메서드	설명
getElementById(id)	Id가 일치하는 요소
getElementsByClassName(class)	class가 일치하는 요소들
getElementsByTagName(tagName)	tagName이 일치하는 요소들
getElementsByName(name)	name이 일치하는 요소들
querySelector('선택자')	선택자로 첫번째 요소
querySelectorAll('선택자')	선택자와 일치하는 모든 요소
<ul style="list-style-type: none"> - 직접 선택자 : id, class, formName, elementName - 간접 선택자 : parentNode, childNode, firstChild, children, nextSibling, previousSibling, 	

❖ document 객체(DOM)

문서 객체 속성 변경

```
<h1 id="heading"> 문서 객체 속성 변경 </h1>
```

문서 객체 속성 변경



이미지 변경



```
img src="coffee-pink.jpg" id="cup" width="200" height="200">
```

❖ document 객체(DOM)

문서 객체 속성 변경

```
<h1 id="heading">문서 객체 속성 변경</h1>
```

```

```

```
<script>
```

```
document.querySelector('#heading').innerHTML = '이미지 변경';
```

```
let bigPic = document.querySelector("#cup");
```

```
bigPic.setAttribute('src', 'coffee-gray.jpg');
```

```
</script>
```

❖ document 객체(DOM)

문서 객체 스타일 변경

```
let header = document.getElementById('heading')  
header.style.border = '2px solid red';  
header.style.backgroundColor = 'pink';
```

문서 객체 제거

```
let cup = document.getElementById('cup');  
cup.parentNode.removeChild(cup);
```


❖ document 객체(DOM)

폼 요소 접근

- id, class
 - name
 - form name과 form 요소에 name 속성이 있어야 함
- `document.shop.shippingName.value`
`document.forms["ship"].elements["shippingName"].value`
- 폼 배열 (forms)
 - id, class, name 속성이 모두 없을 때 사용
 - 폼 태그를 가져와 배열 형태로 반환
- `document.forms[0].elements[0].value`

❖ document 객체(DOM)

선택 항목 및 옵션항목 접근 : select

- form name = "testForm"
- select name = "major"

`document.testForm.major.options[2].innerText` - 화면표시 내용

`document.testForm.major.options[2].value` -- 서버 전달 값

- 사용자가 선택한 옵션
 - option 은 배열 형태로 저장

```
let selectMenu=document.testForm.major;
```

```
selectMenu.options[selectMenu.selectedIndex].innerText;
```

❖ document 객체(DOM)

라디오 버튼과 체크상자 접근

- form name = "testForm"
- radio name = "subject"
 - radioButtonList(n) 형태로 저장
- checkbox name = "mailing1"
 - 체크 상자는 name으로 접근(모두 다른 이름 사용)

`document.testForm.subject[0].checked`

`document.testForm.mailing1.checked`

`document.testForm.mailing2.checked`

❖ 이벤트(event)

- 웹 브라우저나 사용자가 행하는 어떤 동작에 의해 발생
- 웹 문서 안에서 이루어지는 동작에서만 발생
- 키보드, 마우스, 웹 문서 불러올 때, 폼에 내용을 입력할 때 발생

```
Window.onload = function () {  
    let header = document.getElementById('header');  
    function whenClick(){ alert('CLICK'); }  
}
```

	이벤트 이름	
header.	<u>onClick</u>	= <u>whenClick</u> ;
	이벤트 속성	이벤트 리스너
		이벤트 핸들러
};		

- **이벤트 모델** : 문서 객체에 이벤트를 연결하는 방법

❖ 이벤트(Event)

마우스 이벤트 발생

속성	설명
click	요소를 마우스로 클릭했을 때
dblclick	요소를 두 번 눌렀을 때
mousemove	요소 위에서 마우스 포인터를 움직일 때
mousedown	요소 위에 마우스를 누르는 동안
mouseup	요소를 누르고 있다가 버튼에서 손을 뗄 때
mouseover	요소 위에 마우스가 올려질 때
mouseout	마우스 포인터가 요소를 벗어날 때

❖ 이벤트(Event)

키보드 이벤트 발생

속성	설명
keypress	키를 눌렀을 때
keydown	키를 누르고 있는 동안
keyup	키에서 손을 뗄 때
키워드 접근성	onmouseover : onfocus → tab onmouseout : onblur onclick → enter

❖ 이벤트(Event)

문서 로딩 이벤트 발생

속성	설명
abort	웹 문서 로딩 중에 멈췄을 때
error	문서가 정확하게 로딩되지 않았을 때
load	문서 로딩이 끝났을 때
resize	문서의 화면 크기가 바뀔 때
scroll	문서가 스크롤 되었을 때
unload	문서를 벗어날 때

❖ 이벤트(Event)

폼 이벤트 발생

속성	설명
blur	폼 요소에 포커스를 잃었을 때
change	목록이나 체크상태 등이 변경되었을 때
focus	폼 요소에 포커스가 놓였을 때
reset	폼이 다시 시작되었을 때
submit	submit 버튼을 눌렀을 때

❖ 이벤트(Event)

이벤트 처리

- 이벤트가 발생하면 바로 이벤트 처리 함수 연결
- 이벤트 처리기는 이벤트 이름 앞에 '**on**' 을 붙여 사용
- 이벤트 처리기 = 실행명령이나 함수 연결
- 태그에 직접 입력하는 인라인 이벤트 모델

```
<h1 id="heading" onclick="alert('click ')">이벤트 처리</h1>
```

- DOM을 이용한 이벤트 연결

```
document.getElementById('heading').onclick=function(){  
    alert('click');  
}
```

```
document.getElementById('heading').onclick= null --- 제거
```

❖ 이벤트(Event)

표준 이벤트 처리

- 이벤트 연결과 제거

addEventListener(eventName, handler, useCapture)

removeEventListener(eventName, handler)

```
let header = document.getElementById('heading')
let handler = function(){
  this.style.color = 'red'; --- this : 이벤트 발생 객체
  alert('click');
}
header.addEventListener('click', handler); --- 연결
header.removeEventListener('click', handler); --- 제거
```

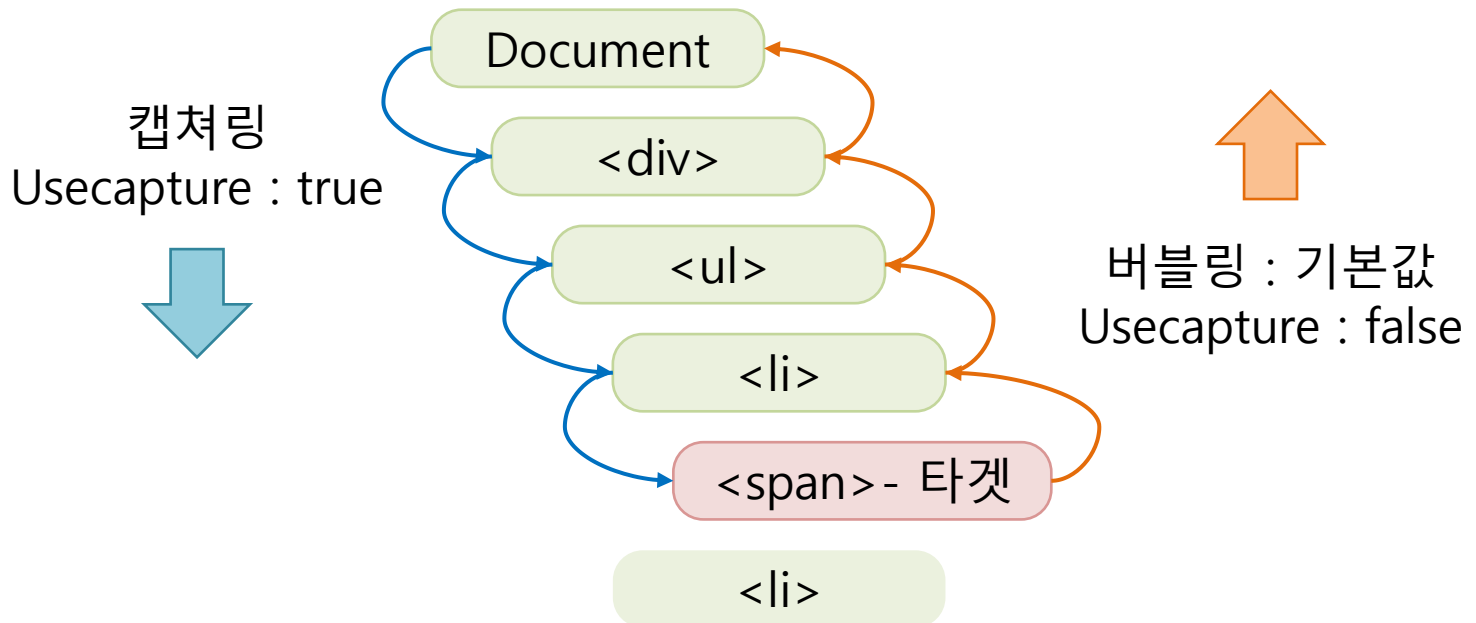
❖ 이벤트(Event)

이벤트 전달

- 이벤트 전달옵션

`addEventListener(eventName, handler, useCapture)`

`eventPropagation(true)` : 캡처링, `(false)` : 버블링



❖ 이벤트(Event)

이벤트 전달을 원하지 않는 경우

- 타겟 요소의 이벤트 리스너의 콜백 함수 마지막 위치에 **event.stopPropagation();** 메서드 호출

```
<h1 id="heading">header
  <p id="paragraph">paragraph</p>
</h1>
document.getElementById('heading').onclick=function(){
  alert('header');
};
document.getElementById('paragraph').onclick=function(){
  alert('paragraph');
  event.stopPropagation();
};
```