

7강. DOM & BOM



목 차

1

DOM을 이용한 이벤트 처리

2

addEventListener() 함수

3

DOM 요소 선택자

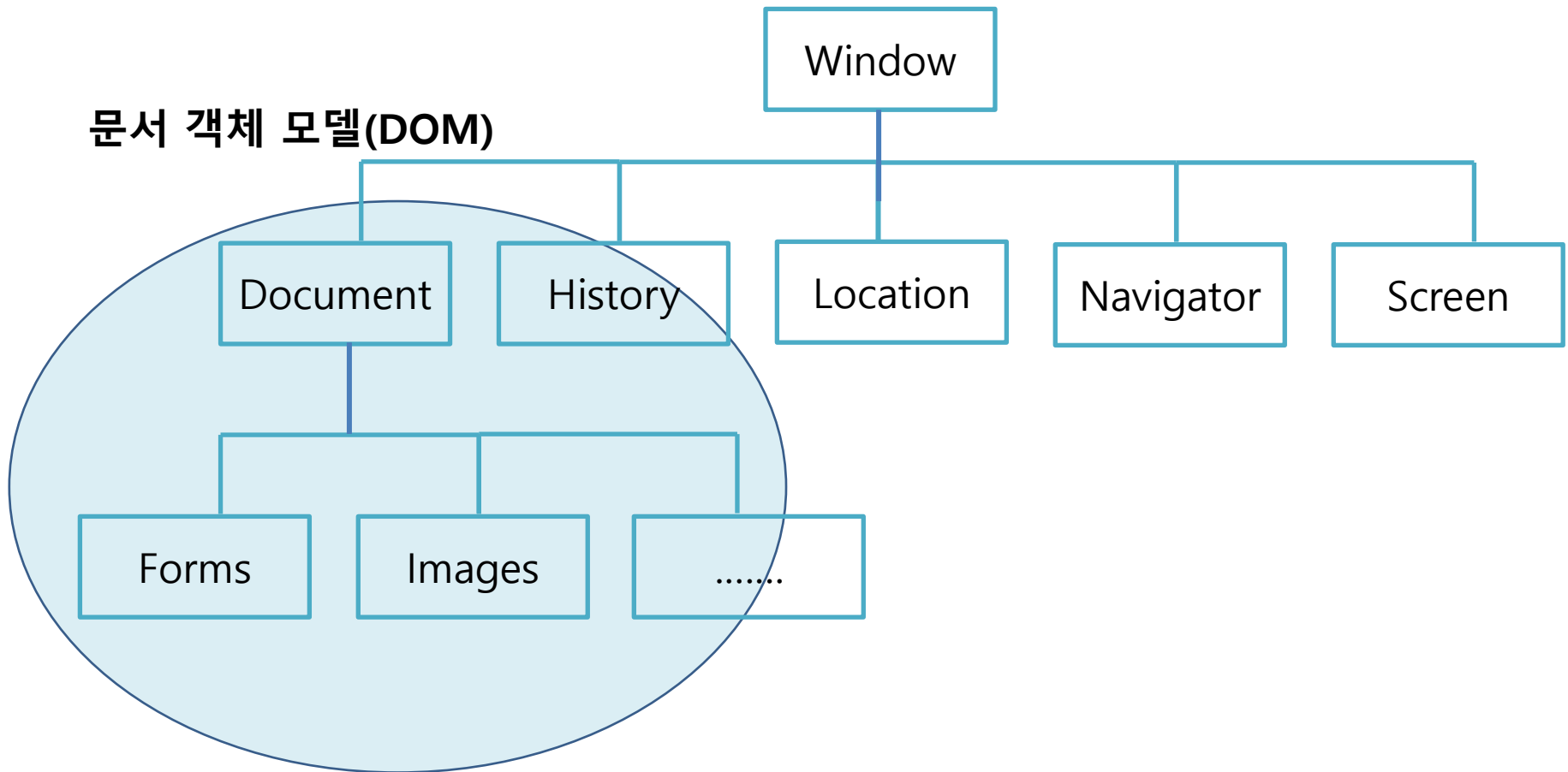
4

BOM을 이용한 이벤트 처리

객체 구조도

브라우저 객체 모델(BOM)

문서 객체 모델(DOM)



이벤트 다루기

이벤트 효과

마우스 이벤트 - **addEventListener()**에서는 **on**을 붙이지 않는다.

속성	설명
onclick	사용자가 HTML 요소를 클릭할 때 이벤트가 발생
mouseover	마우스 포인터가 요소 위로 옮겨질때 이벤트 발생
mouseout	마우스 포인터가 요소를 벗어날 때 이벤트 발생

키보드 이벤트

속성	설명
keypress	키를 눌렀을때 이벤트가 발생
keydown	키를 누르고 있는 동안 이벤트 발생
keyup	키에서 손을 뗄 때 이벤트 발생

이벤트 다루기

이벤트 효과

문서로딩 이벤트

속성	설명
load	문서 로딩이 끝나면 이벤트가 발생
abort	웹 문서가 로딩되기 전 멈췄을때 이벤트 발생
scroll	문서 화면이 스크롤 되었을때 이벤트 발생

폼 이벤트

속성	설명
change	목록이나 체크 상태 등이 변경되었을때 이벤트 발생
focus	폼 요소에 포커스(커서)가 놓였을때 이벤트 발생
submit	submit 버튼을 눌렀을때 이벤트 발생

문서 객체 모델(DOM)

문서 객체 모델(Document Object Model)

- 웹 문서의 모든 요소를 자바스크립트를 이용하여 조작할 수 있도록 객체를 사용해 문서를 해석하는 방법.

DOM 요소에 접근하기

1. DOM 요소를 id 선택자로 접근하는 함수

document.getElementById("display")

객체.getElementById(선택자)

2. DOM 요소를 태그, id, class 선택자로 접근하는 함수

document.querySelector("img")

document.querySelector("#pic")

document.querySelector(".pic")

객체.querySelector(선택자 또는 태그)

3. 객체.querySelectorAll(리스트) - 여러 개의 값을 반환

DOM 요소 선택자

- `querySelector`(선택자 또는 태그) - **value** 속성

이름 :

클릭하면 이름이 변경됩니다.

클릭

querySelector1.html

```
<p>이름 : <input type="text" id="myText" value="오상식" disabled></p>
<p>클릭하면 이름이 변경됩니다.</p>
<button onclick="myFunction()">클릭</button>

<script>
  function myFunction(){
    document.querySelector("#myText").value = "최정보";
  }
</script>
```

DOM 요소 선택자

● querySelectorAll("선택자") -> 배열 생성

국가의 수도(Capital)

대한민국의 수도는 서울입니다.

세계의 수도

- 베를린
- 도쿄
- 뻬이징
- 워싱턴

```
<h2 id="heading">국가의 수도(Capital)</h2>  
<p id="desc">대한민국의 수도는 서울입니다.</p>
```

```
<h2>세계의 수도</h2>
```

```
<ul>
```

```
<li class="city">베를린</li>
```

```
<li class="city">도쿄</li>
```

```
<li class="city">뻬이징</li>
```

```
<li class="city">워싱턴</li>
```

```
</ul>
```

```
<script>
```

```
//1개의 요소 선택
```

```
document.querySelector("#heading").style.color="blue";
```

```
//모두 선택하여 cityList 배열 생성
```

```
var cityList = document.querySelectorAll(".city");
```

```
cityList[0].style.color = "green";
```

```
cityList[3].style.color = "orange";
```

```
</script>
```

querySelectorAll.html

DOM 요소 선택자

웹 요소의 스타일 수정하기

국가의 수도

대한민국의 수도는 서울입니다.

세계의 수도

시드니

도쿄

베이징

워싱턴



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following code and its output:

```
> document.querySelectorAll(".city")
< ▶ NodeList(4) [li.city, li.city, li.city, li.city]

> document.querySelectorAll(".city")[0].style.color="blue";
< "blue"

> document.querySelectorAll(".city")[1].style.color="purple";
< "purple"
```

html 태그의 onclick 속성

이벤트 처리1 - html 태그의 onclick 속성 사용하기



```
<div id="container">
  
</div>
<script>
  function changePic(){
    //document.getElementById("pic").src = "images/cup-2.png";

    var pic = document.querySelector("#pic");
    pic.src = "images/cup-2.png";
  }
</script>
```

자바스크립트 객체의 onclick 속성

이벤트 처리2 – 자바스크립트 객체의 onclick 속성 사용하기



```
<div id="container">
  
</div>
<script>
  var pic = document.getElementById("pic")
  /*pic.onclick = changePic;

  function changePic(){
    pic.src = "images/cup-2.png";
  }*/

  //실행 함수
  pic.onclick = function(){
    pic.src = "images/cup-2.png";
  }
</script>
```

addEventListener() 함수

addEventListener() 함수 사용하기

한 요소에 여러 이벤트가 발생했을때 이벤트 처리기를 연결해 주는 함수이다.

Syntax

```
element.addEventListener(event, function, useCapture)
```

false가 기본임

Parameter Values

이벤트 유형('on'을 붙이지 않음)

Parameter	Description
<i>event</i>	<p>Required. A String that specifies the name of the event.</p> <p>Note: Do not use the "on" prefix. For example, use "click" instead of "onclick".</p> <p>For a list of all HTML DOM events, look at our complete HTML DOM Event Object Reference.</p>
<i>function</i>	<p>Required. Specifies the function to run when the event occurs.</p> <p>When the event occurs, an event object is passed to the function as the first parameter. The type of the event object depends on the specified event. For example, the "click" event belongs to the MouseEvent object.</p>

addEventListener() 함수

addEventListener() 함수 사용하기 – 한 요소에 여러 이벤트 처리



```
<body>
  <div id="container">
    
  </div>
  <script src="js/event.js"></script>
</body>
```

addEventListener() 함수

event.js

```
let pic = document.querySelector('#image');

//이름 있는 함수로 정의
pic.addEventListener("mouseover", changePic); //마우스 올림
pic.addEventListener("mouseout", originPic); //마우스 벗어남

function changePic() {
    pic.src = "images/easys-2.jpg";
}

function originPic() {
    pic.src = "images/easys-1.jpg";
}

//이름 없는 함수(익명 함수 or 실행 함수)
/*pic.addEventListener("mouseover", function(){
    pic.src = "./images/easys-2.jpg";
});

pic.addEventListener("mouseout", function(){
    pic.src = "./images/easys-1.jpg";
});*/
```

DOM 요소 선택자

- 속성을 가져오거나 수정하는 함수

getAttribute(속성명), setAttribute(속성명, 값)

입력상자를 버튼으로 변경하려면 클릭하세요.

Try it

setAttribute1.html

```
<input type="text" value="OK" class="input">

<p>입력상자를 버튼으로 변경하려면 클릭하세요.</p>

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

<script>
  //setAttribute(속성, 속성값) 사용
  function myFunction() {
    document.querySelector('.input').setAttribute("type", "button");
  }
</script>
```

DOM 요소 선택자

◆ 마우스로 선택한 사진으로 바꾸기



DOM 요소 선택자

◆ 마우스로 선택한 사진으로 바꾸기

```
<section id="content">
  <h2>커피의 세계</h2>
  <div id="pic">
    
    <div id="smallPic">
      
      
      
    </div>
  </div>
</section>

<script src="js/showBig.js"></script>
```

DOM 요소 선택자

```
#content{width: 600px; margin: 0 auto; text-align: center;}
#smallPic{margin: 20px 5px;}
.small{width: 100px; height: 100px;}
```

showBig.css

```
//큰 사진 선택
let bigPic = document.querySelector("#cup");

//작은 사진 모두 선택 - 배열
let smallPics = document.querySelectorAll(".small");

//onclick 이벤트
//smallPics[1].onclick = showBig - 인덱스 1번 이미지 확대
for(let i = 0; i < smallPics.length; i++){
    smallPics[i].onclick = showBig;    //showBig()함수 호출
}

function showBig(){
    let newPic = this.src; //this-이벤트 일어난 대상의 속성 값
    bigPic.setAttribute("src", newPic);
    //setAttribute("속성", "속성값")
}
```

showBig.js

DOM 요소 선택자

선택한 사진으로 바꾸기

- 콘솔에서 실행하기



```
> document.querySelector("#cup").getAttribute("src")
< "images/coffee-pink.jpg"
> document.querySelector("#prod-pic > img").getAttribute("src")
< "images/coffee-pink.jpg"
> document.querySelector("#prod-pic > img").setAttribute("src", "images/coffee-blue.jpg");
< undefined
```

이벤트 다루기

상세 설명 보기/닫기 – toggle 방식 구현하기

에디오피아 게덱



상품명 : 에디오피아 게덱

판매가 : 9,000원

배송비 : 3,000원

(50,000원 이상 구매 시 무료)

적립금 : 180원(2%)

로스팅 : 2019.06.17

장바구니 담기

상세 설명 보기

DOM 요소 선택자

상세 설명 보기/닫기

에디오피아 게덱



상품명 : 에디오피아 게덱

판매가 : 9,000원

배송비 : 3,000원

(50,000원 이상 구매 시 무료)

적립금 : 180원(2%)

로스팅 : 2019.06.17

장바구니 담기

[상세 설명 닫기](#)

상품 상세 정보

- 원산지 : 에디오피아
- 지 역 : 이르가체프 코체레
- 농 장 : 게덱
- 고 도 : 1,950 ~ 2,000 m
- 품 종 : 지역 토착종
- 가공법 : 워시드

Information

2차 세계대전 이후 설립된 게덱 농장은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 SCAA 인증을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티 커피를 생산합니다.

DOM 요소 선택자

product.html

```
<div id="container">
  <h1 id="heading">에디오피아 게텔</h1>
  <div id="prod-pic">
    
    <div id="small-pic">
      
      
      
    </div>
  </div>
  <div id="desc">
    <ul>
      <li>상품명 : 에디오피아 게텔</li>
      <li class="bluetext">판매가 : 9,000원</li>
      <li>배송비 : 3,000원<br>(50,000원 이상 구매 시 무료)</li>
      <li>적립금 : 180원(2%)</li>
      <li>로스팅 : 2019.06.17</li>
      <button>장바구니 담기</button>
    </ul>
    <a href="#" id="view">상세 설명 보기</a>
  </div>
</div>
<hr>
```

이벤트 다루기

```
<div id="detail">
  <h2>상품 상세 정보</h2>
  <ul>
    <li>원산지 : 에디오피아</li>
    <li>지 역 : 이르가체프 코체레</li>
    <li>농 장 : 게덱</li>
    <li>고 도 : 1,950 ~ 2,000 m</li>
    <li>품 종 : 지역 토착종</li>
    <li>가공법 : 워시드</li>
  </ul>
  <h3>Information</h3>
  <p>2차 세계대전 이후 설립된 게덱 농장은 유기농 인증 농장으로 여성의 고용 창출과 지역사회 발전에 기여하며 3대째 이어져 내려오는 오랜 역사를 가진 농장입니다. 게덱 농장은 SCAA 인증을 받은 커피 품질관리 실험실을 갖추고 있어 철저한 관리를 통해 스페셜티 커피를 생산합니다.</p>
  <h3>Flavor Note</h3>
  <p>은은하고 다채로운 꽃향, 망고, 다크 체리, 달달함이 입안 가득.</p>
</div>
<script src="js/product.js"></script>
```

product.html

이벤트 다루기

```
#container {
  width:600px;
  margin:0 auto;
}
.small {width:60px; height:60px;}
#small-pic {margin-top:20px; padding:0;}
#prod-pic{float:left;}
#prod-pic {margin:20px 20px auto 10px; padding:0;}
#desc {
  width:300px;
  float:left;
  padding-top:20px;
  margin-bottom:50px;
}
hr {clear: both; border: 1px dashed #f5f5f5;}
#desc ul {list-style:none;}
#desc li{font-size:0.9em; line-height:1.8;}
.bluetext {color:#4343ff; font-weight:bold;}
#desc button {
  margin-top:20px;
  margin-bottom:20px;
  width:100%;
  padding:10px;
}
```

product.css

```
#desc a {
  text-decoration: none;
  font-size:0.9em;
  color:blue;
  padding-left:40px;
}
#detail {
  padding-top:10px;
  display:none;
}
#detail li {font-size:0.9em; line-height:1.4;}
h2 {
  font-size:1.5em;
  color:#bebebe;
  font-weight:normal;
}
h3 { font-size:1.1em; color:#222;}
p {
  font-size:0.9em;
  line-height:1.4;
  text-align: justify; /*양쪽 정렬*/
}
```


DOM 요소 선택자

product.js

```
//큰 사진 선택
var bigPic = document.querySelector("#cup");

//작은 사진 모두 선택 - 배열
var smallPics = document.querySelectorAll(".small");

//click 이벤트
for(var i=0; i<smallPics.length; i++){
    smallPics[i].addEventListener("click", function(){
        newPic = this.src;    //이벤트가 발생한 대상의 src 속성값
        bigPic.setAttribute("src", newPic);
    });
}
```

DOM 요소 선택자

product.js

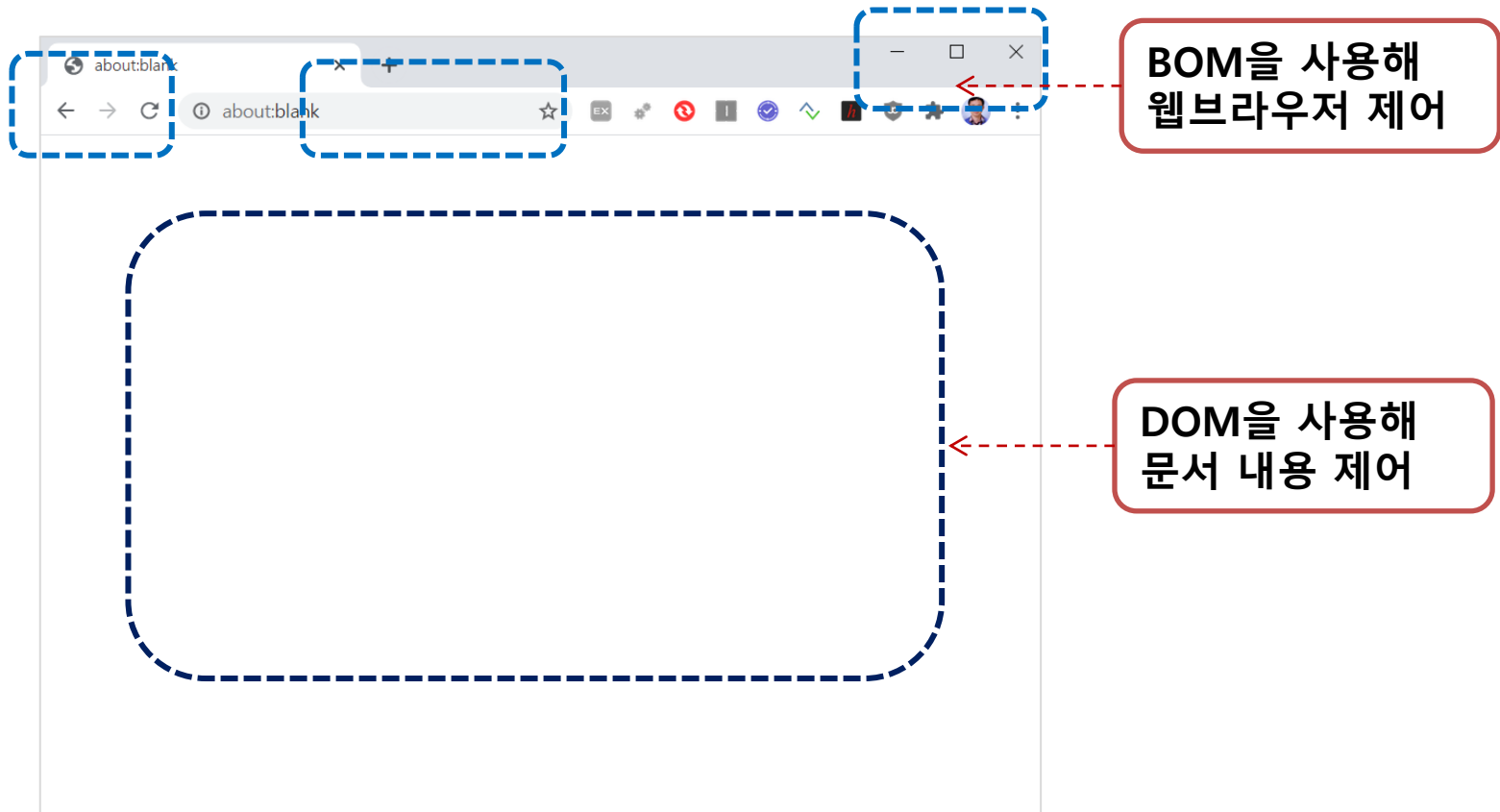
```
// 상세 설명 단기를 클릭하여 텍스트 보이기 / 감추기(toggle방식)
var isOpen = false;
var view = document.querySelector("#view");

view.addEventListener("click", function(){
    //document.querySelector("#detail").style.display = "block";
    if(isOpen==false){
        document.querySelector("#detail").style.display = "block";
        view.innerText = "상세 설명 닫기";
        isOpen = true; //isOpen에 true를 저장
    }
    else{ //isOpen==true
        document.querySelector("#detail").style.display = "none";
        view.innerText = "상세 설명 보기";
        isOpen = false;
    }
});
```

BOM (Browser Object Model)

브라우저 객체 모델(Browser Object Model)

- 웹 브라우저 전체를 객체로 관리하는 것



BOM – window 객체

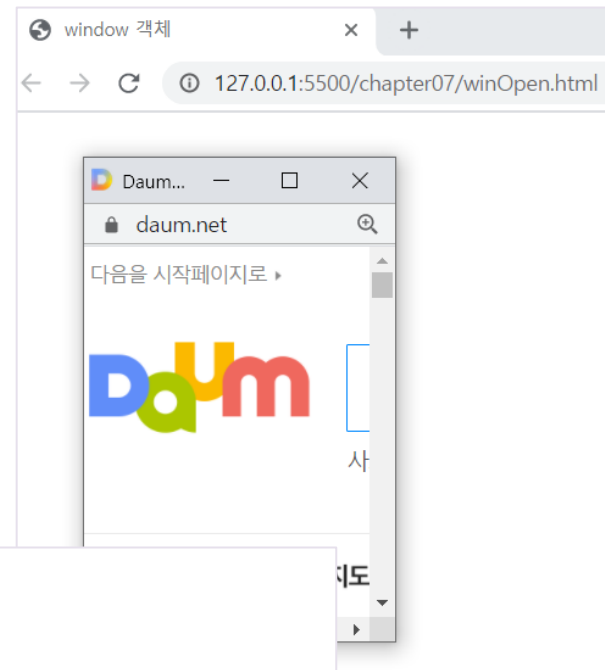
window 객체

- 웹 브라우저의 상태를 제어하는 객체, 최상위 객체
- `window.open()` , `window.alert()` 등 -> window는 생략 가능함

`window.open(경로, 창 이름, 창 옵션)` :

브라우저 창열기

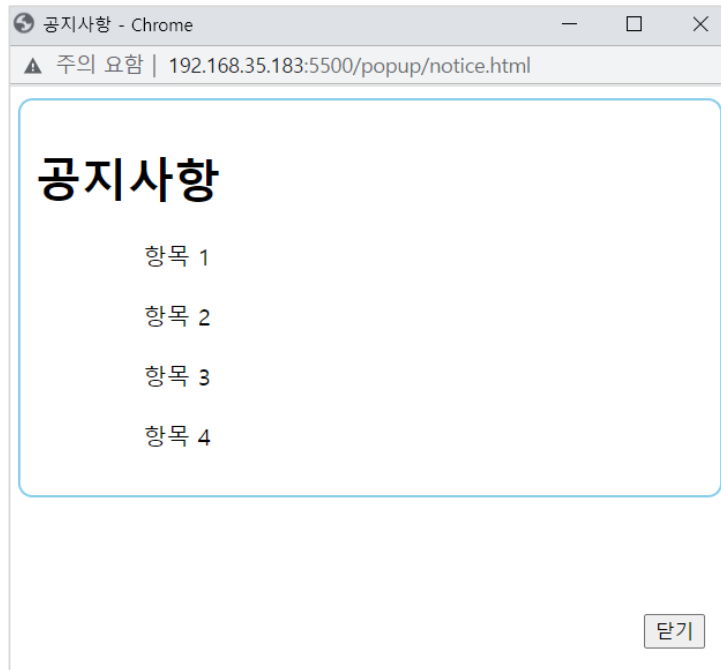
`window.close()` : 창닫기



```
<script>  
    //window.open("http://www.naver.com");  
    window.open("http://www.daum.net", "",  
                "top=100, left=50, width=200, height=250");  
</script>
```

팝업 창 띄우기

- 공지 창(popup) 만들기



popup.html

```
<p>문서를 열면 팝업 창이 표시됩니다</p>
<script>
  window.open("notice.html","", "width=500, height=400");
</script>
```

팝업 창 띄우기

■ 공지 창(popup) 만들기

```
<div id="content">
<h1>공지사항</h1>
  <ul>
    <li>항목 1</li>
    <li>항목 2</li>
    <li>항목 3</li>
    <li>항목 4</li>
  </ul>
  <button onclick="javascript:window.close();">닫기</button>
</div>
```

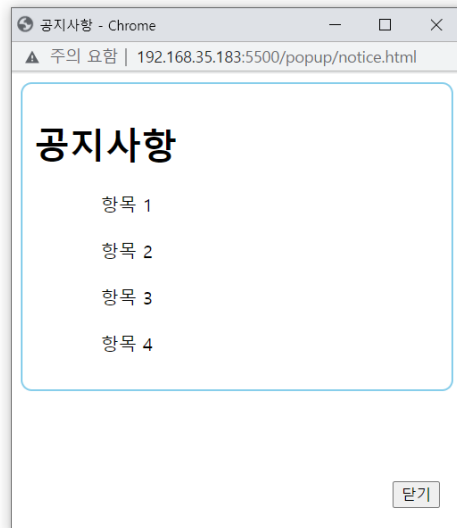
notice.html

```
#content {
  border : 2px double skyblue;
  border-radius:10px;
  padding:10px;
}
ul {list-style-type:none;}
ul li {margin : 20px}
button{position: absolute; bottom: 20px; right: 20px;}
```

팝업 창 띄우기

■ 공지 창(popup) 만들기

왼쪽에서 100픽셀, 위에서 200픽셀 떨어진 위치에
팝업 창이 표시됩니다



window.open()에서 창이름이 없으면 브라우저가 새로고침되면 계속 팝업 창이 열리는 문제가 발생함

popup2.html

```
<p>왼쪽에서 100픽셀, 위에서 200픽셀 떨어진 위치에 <br>  
| | 팝업 창이 표시됩니다</p>  
<script>  
| window.open("notice.html","pop", "width=400, height=400, left=100, top=200");  
</script>
```

이벤트 공지

❖ 이벤트 공지



location 객체

❖ 이벤트 공지

```
<p>문서를 열면 팝업 창이 표시됩니다.</p>
```

```
<script>
```

```
  popWin = window.open("event.html", "popup", "width=700, height=600");
```

```
  popWin.creator = self;
```

```
</script>
```

main.html

```
<div id="container">
```

```
  <h1>이벤트 공지</h1>
```

```
  
```

```
  <p><a href="event-main.html">자세히 보기</a></p>
```

```
</div>
```

event.html

```
#container
```

```
{width:600px;margin:10px auto;padding:10px;text-align:center;}
```

location 객체

❖ 이벤트 공지

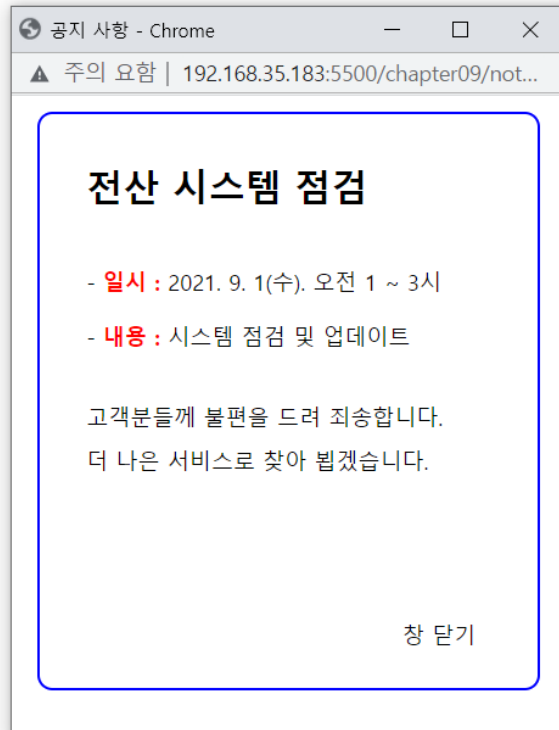
```
<style>
  #container{
    text-align: center;
  }
</style>
</head>
<body>
  <div id="container">
    
  </div>
</body>
```

event-main.html

팝업 창 띄우기

■ 공지 창(popup) 만들기

문서를 열면 공지 창이 표시됩니다.



팝업 창 띄우기

```
<h3>문서를 열면 공지 창이 표시됩니다.</h3>
```

main-popup.html

```
<script>
```

```
  window.open("notice2.html", "popup", "width=350, height=400, top=150, left=50");  
</script>
```

```
<div id="content">
```

```
  <h2>전산 시스템 점검</h2>
```

```
  <p class='detail'>
```

```
    - <b>일시 : </b>2021. 9. 1(수). 오전 1 ~ 3시<br>
```

```
    - <b>내용 : </b>시스템 점검 및 업데이트
```

```
  </p>
```

```
  <p>고객분들께 불편을 드려 죄송합니다.<br>
```

```
  <p>  더 나은 서비스로 찾아 뵙겠습니다.</p>
```

```
  <p class="close">
```

```
    <a href="javascript:window.close()">창 닫기</a>
```

```
  </p>
```

```
</div>
```

notice2.html

팝업 창 띄우기

```
/* 전체 선택 - 박스 크기 : border-box */
*{margin: 0; padding: 0; box-sizing: border-box;}
#content{
    width: 330px;
    height: 380px;
    border: 2px solid ■#00f;
    border-radius: 10px;
    margin: 10px auto;
    position: relative; /* 부모 기준 */
}
h2{padding: 30px;}
p{line-height: 2em; padding-left: 20px; font-size: 1.1em;}
.detail{margin-bottom: 20px;}
.detail b{color: ■red;}
.close{position: absolute; bottom: 20px; right: 40px;} /*창닫기*/
.close a{text-decoration: none; color: ■#00f;}
.close a:hover{text-decoration: underline; }
```

notice.css

location 객체

location 객체

- 브라우저의 주소표시줄과 관련된 객체
- 현재 브라우저 창에서 열어야 할 사이트나 문서를 지정할 수 있다.

location.href = "url 경로"

location 객체

```
location.href :  
http://127.0.0.1:5500/chapter09/locationObject.html  
  
location.host : 127.0.0.1:5500  
  
location.protocol : http:
```

[고용노동부 홈페이지로 이동하기](#)

location 객체

location 객체

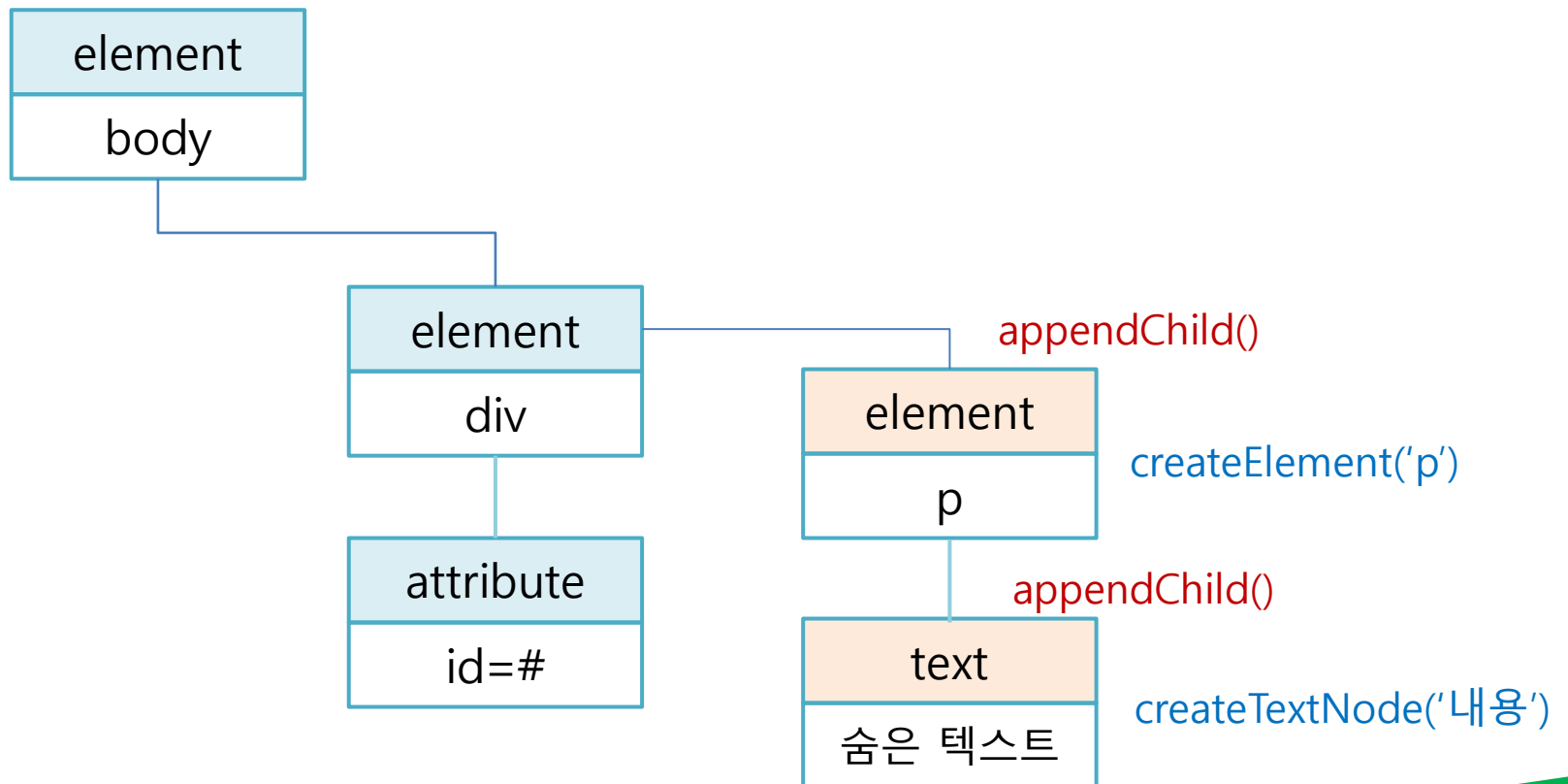
```
<section id="content">
  <h2>location 객체 </h2>
  <div class="display">
    <script>
      //url 경로
      document.write("<p><b>location.href : </b>" + location.href + "</p>");
      //호스트 컴퓨터의 주소
      document.write("<p><b>location.host : </b>" + location.host + "</p>");
      //url의 프로토콜
      document.write("<p><b>location.protocol : </b>" + location.protocol + "</p>");
    </script>
  </div>
  <button onclick="location.href = 'https://www.hrd.go.kr'">
    고용노동부 홈페이지로 이동하기
  </button>
</section>
```

```
#content{width: 600px; margin: 10px auto;}
.display {margin-top: 10px; padding: 10px; border: 1px solid ■ #222; }
button {margin-top: 20px;}
```

DOM에서 노드 추가,삭제하기

DOM 트리에 새로운 노드 추가

웹 문서에서 처음 화면에는 내용이 보이지 않다가 클릭이나 이벤트가 발생하면 내용이 나타나는 경우 사용한다.



DOM에서 노드 추가,삭제하기

- Coffee
- Tea

Click Button!

Try it



- Coffee
- Tea
- Water

Click Button!

Try it

```
<ul id="myList">
  <li>Coffee</li>
  <li>Tea</li>
</ul>
<p>Click Button!</p>
<button onclick="addNode(); this.onclick='' ">Try it</button>
<script>
  function addNode(){
    var node = document.createElement('li');           //createElement(태그)
    var textNode = document.createTextNode('Water');   //createTextNode(텍스트내용)
    node.appendChild(textNode);                         //textNode를 자식노드로 추가
    document.getElementById('myList').appendChild(node); //node를 자식노드로 추가
  }
</script>
```

DOM에서 노드 추가.삭제하기

DOM 노드 추가하기

[더 보기](#)

--->

DOM 노드 추가하기

[더 보기](#)

안녕하세요~ 숨겨진 텍스트입니다.

```
<div id="container">
  <h1>DOM 노드 추가하기</h1>
  <a href="#" onclick="addP(); this.onclick='';">더 보기</a>
  <div id="info"></div>
</div>
<!-- this.onclick=''; addP() 함수가 1번만 실행됨 -->
<script>
  function addP(){
    var newP = document.createElement("p"); //p태그 요소 노드 만들기
    var text = document.createTextNode("안녕하세요~ 숨겨진 텍스트입니다."); //text 노드 만들기
    newP.appendChild(text); //p와 text 자식노드 연결하기
    document.getElementById("info").appendChild(newP); //p를 부모인 div에 연결하기
  }
</script>
```

DOM에서 노드 추가,삭제하기

에디오피아 게델 커피

[더 보기](#)

은은하고 다채로운 꽃향, 망고, 체리 달달함이 입안 가득...



```
Elements Console Sources Network Performance Memory
▼ <body>
  ▼ <div id="container">
    <h1>에디오피아 게델 커피</h1>
    <a href="#" onclick="addContents(); this.onclick=''>더 보기</a>
  .. ▼ <div id="info"> == $0
    <p>은은하고 다채로운 꽃향</p>
    
  </div>
</div>
▶ <script>...</script>
```

DOM에서 노드 추가,삭제하기

```
<div id="container">
  <h1>에디오피아 게덱 커피</h1>
  <a href="#" onclick="addContents(); this.onclick='';">더 보기</a>
  <div id="info"></div>
</div>
<script>
  function addContents() {
    var newP = document.createElement("p");
    var text = document.createTextNode
    | | | | | ("은은하고 다채로운 꽃향, 망고, 체리 달달함이 입안 가득...");
    newP.appendChild(text);

    var newImg = document.createElement("img");
    var src = document.createAttribute("src");
    var alt = document.createAttribute("alt");
    src.value = "images/coffee-blue.jpg";
    alt.value = "커피 이미지";
    newImg.setAttributeNode(src);
    newImg.setAttributeNode(alt);

    document.getElementById("info").appendChild(newP);
    document.getElementById("info").appendChild(newImg);
  }
</script>
```

DOM에서 노드 추가.삭제하기

❖ 구구단 출력

구구단 4단

단 입력 :

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36

DOM에서 노드 추가,삭제하기

❖ 구구단 출력

```
<h1>구구단 <span id="x">x</span>단</h1>
<p>단 입력 :
<input type="text" id="myNum">
<button onclick="printGugu(); this.onclick='';">OK</button><br>
<div id="demo"></div>
<script>
    function printGugu() {
        let strN = document.getElementById("myNum").value;
        let n = parseInt(strN);
        document.getElementById("x").innerHTML = n;
        for (let i = 1; i <= 9; i++) {
            let newDiv = document.createElement("div"); //div 태그 생성
            let textNode = document.createTextNode(n + " x " + i + " = " + (n * i)); //text
            newDiv.appendChild(textNode);
            document.getElementById("demo").appendChild(newDiv);
        }
    }
</script>
```

window 스크린

PICK
YOUR
FAVORITE



window 스크롤

1. 배경 이미지 – background url() fixed
2. 메인 이미지 – position: absolute
3. 텍스트 – 애니메이션 효과 : 스크롤 제어

index.html

```
<section>
  <h1>PICK YOUR FAVORITE</h1>
  
</section>
<script src="main.js"></script>
```


window 스크롤

// 제목 글자 애니메이션

```
let mainText = document.querySelector("h1");

window.addEventListener('scroll', function(){
  let value = window.scrollY;
  console.log("scrollY", value);

  if(value > 300){
    mainText.style.animation = 'disappear 1s ease-out forwards';
  }
  else{
    mainText.style.animation = 'slide 1s ease-out forwards';
  }
});
```

main.js

window △≡≡≡

```
body{margin: 0; height: 2000px;}
section{
  height: 800px;
  background: url("https://www.starbucks.co.kr/common/img/main/fav_prod_bg_new.jpg") fixed;
}
section h1{
  width: 300px;
  font-size: 6em;
  color: □white;
  position: absolute;
  top: 80px;
  left: 400px;
  animation: slide 1s ease-out;
}
section .photo{position: absolute; top: 100px; right: 200px;}
```

main.js

```
@keyframes slide{
  from{
    left: -100px;
    opacity: 0;
  }
  to{
    left: 400px;
    opacity: 1;
  }
}
```

```
@keyframes disappear{
  from{
    left: 400px;
    opacity: 1;
  }
  to{
    left: -100px;
    opacity: 0;
  }
}
```