

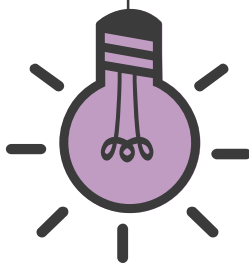
# JavaScript

J a v a S c r i p t



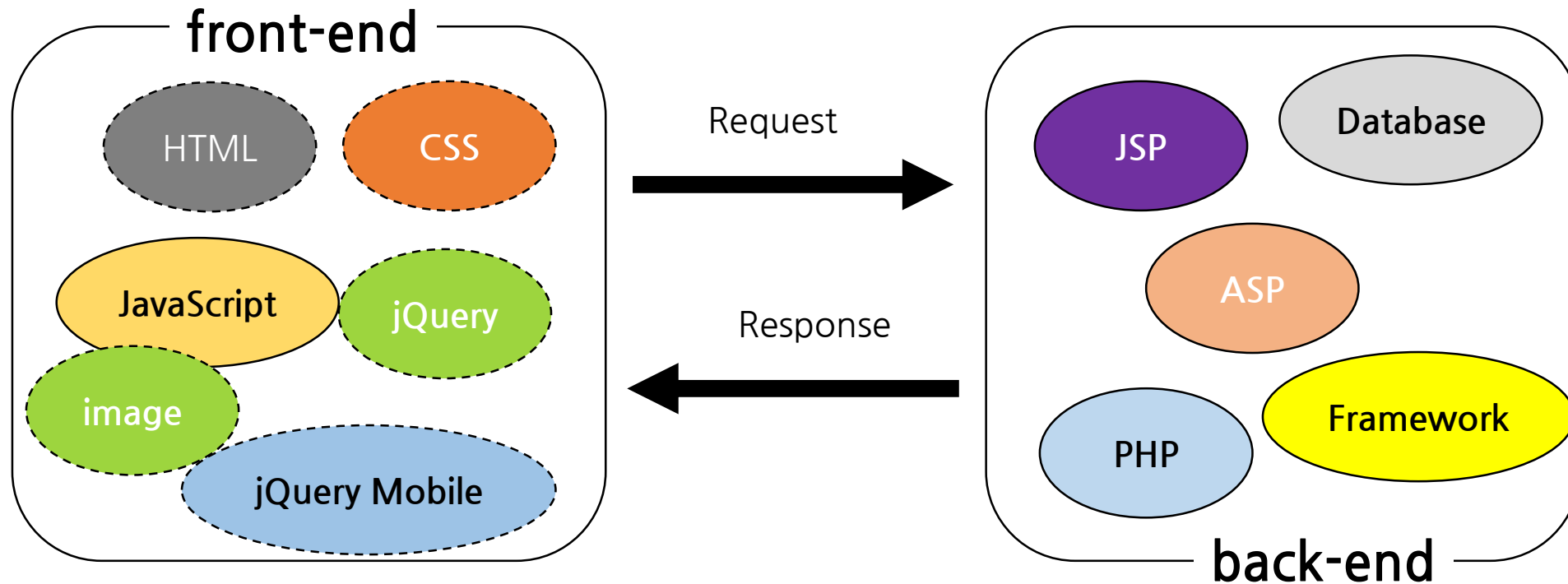
01

## 자바스크립트의 소개



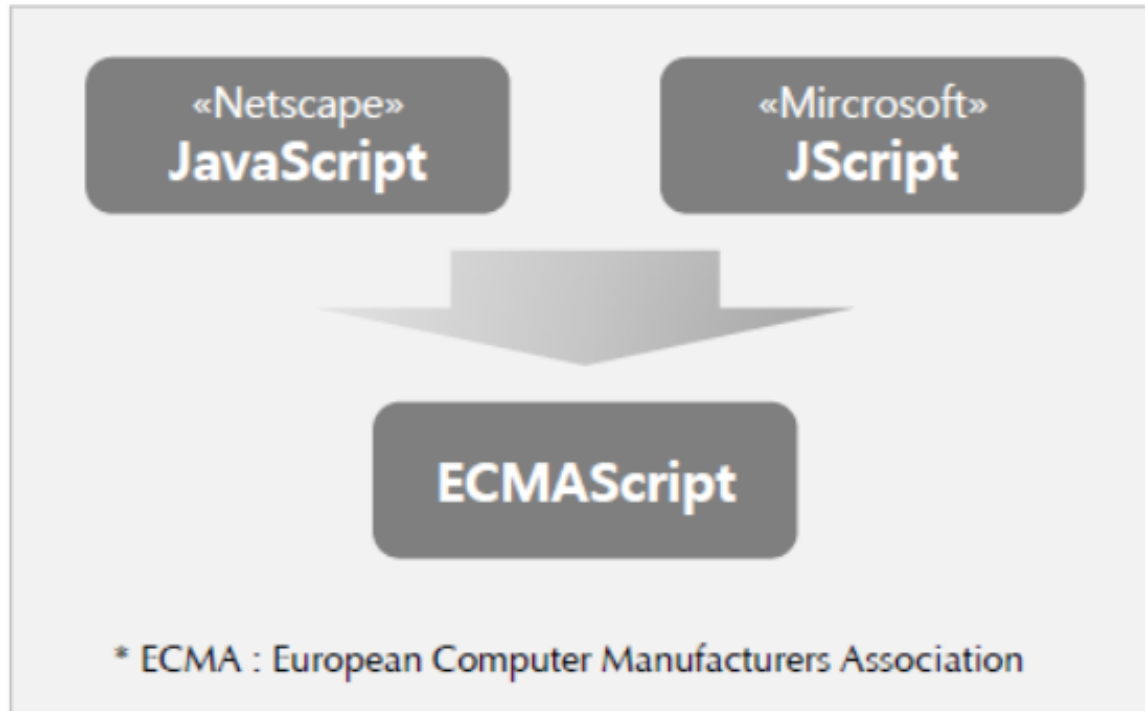
# 1. 자바스크립트 (JavaScript) 란?

웹 브라우저 내에서 사용하는 객체 기반의 스크립트 프로그래밍 언어이다.



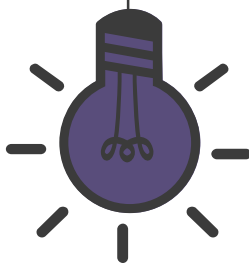
## 2. 자바스크립트의 역사

- 1996년 3월 Netscape는 'Netscape Navigator 2.0'으로 JavaScript를 지원하기 시작했다.
- Microsoft사는 웹에 호환되는 JScript를 개발해서 1996년 8월에 출시하였다.
- Netscape는 JavaScript 표준화를 위해 기술 규격을 ECMA에 제출했다.



02

## 자바스크립트의 문법



# 1. 자바스크립트 구현

자바스크립트의 선언은 일반적으로 <head>와 <head> 사이에 작성한다.

```
<script type="text/javascript">
```

실행문;

```
</script>
```

## [작성 규칙]

1. 자바스크립트는 대소문자를 구분한다.
2. 한 라인을 끝내려면 원칙적으로 세미콜론(;)으로 닫아준다.  
그러나, Enter로 라인을 구분하기 때문에 생략이 가능하다.
3. 한 번에 2개 이상의 문장이 기술되면 세미콜론을 생략할 수 없다.

# 1. 연산자

## 산술 연산자

연산자	설명	연산자	설명
+	더하기	-	빼기
*	곱하기	/	나누기
%	나머지		

## 관계 연산자

연산자	설명	연산자	설명
A > B	A가 B보다 크다.	A >= B	A가 B보다 크거나 같다.
A < B	A가 B보다 작다.	A <= B	A가 B보다 작거나 같다.
A == B	A와 B는 같다.	A != B	A와 B는 같지 않다.
A === B	A와 B는 같다. (데이터 형의 비교)	A !== B	A와 B는 같지 않다. (데이터 형의 비교)

## 대입(할당) 연산자

연산자	사용 예	의미	설명
+	A=10	A=10	A에 10을 저장한다.
+=	A+=10	A=A+10	A에 10을 더해서 다시 A에 저장한다.
-=	A-=10	A=A-10	A에서 10을 빼서 다시 A에 저장한다.
*=	A*=10	A=A*10	A에 10을 곱해서 다시 A에 저장한다.
/=	A/=10	A=A/10	A를 10으로 나누어서 다시 A에 저장한다.
%=	A%=10	A=A*10	A를 10으로 나눈 나머지를 다시 A에 저장한다.

## 논리 연산자

연산자	의미	설명
A && B	AND(그리고)	A, B 둘 다 만족해야 참
A    B	OR(또는)	A, B 둘 중 하나만 만족해도 참
!A	NOT(논리부정)	A의 반대

## 증감 연산자

연산자	사용 예	설명
++	++A	A의 값을 1증가 시킨 후에 사용
	A++	A의 값을 사용한 후 1증가
--	--A	A의 값을 1감소 시킨 후에 사용
	A--	A의 값을 사용한 후 1 감소

## 삼항 연산자

연산자	설명
(조건식)? A:B;	조건식 결과가 참이면 A를 수행하고 거짓이면 B를 수행한다.

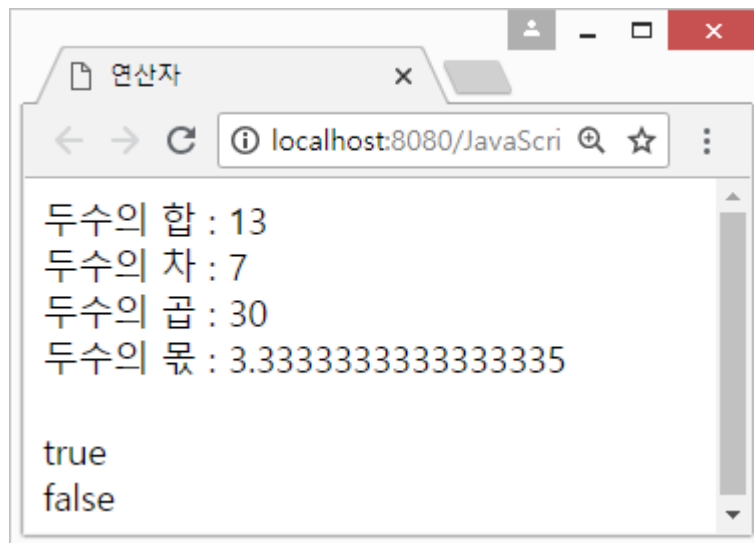


## ex01.html

```
<script>
  var a=10, b=3, c,d,e,f;
  c=a+b;
  d=a-b;
  e=a*b;
  f=a/b;
  document.write("두수의 합 : " + (a + b) + "<br>");
  document.write("두수의 차 : " + d + "<br>");
  document.write("두수의 곱 : " + e + "<br>");
  document.write("두수의 몫 : " + f + "<br><br>");

  var str="korea";
  document.write(str=="korea" || str=="KOREA");
  document.write("<br>");

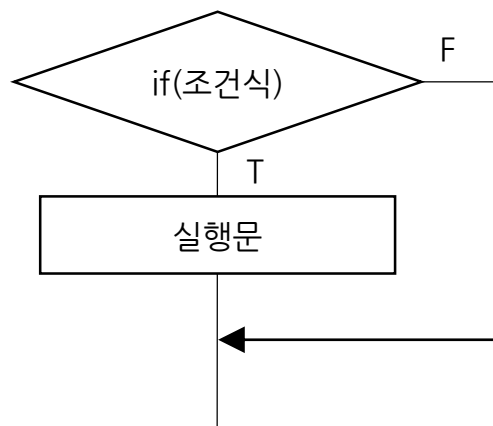
  var score=57;
  document.write(score>=70 && score<90);
  document.write("<br>");
</script>
```



## 2. 제어문

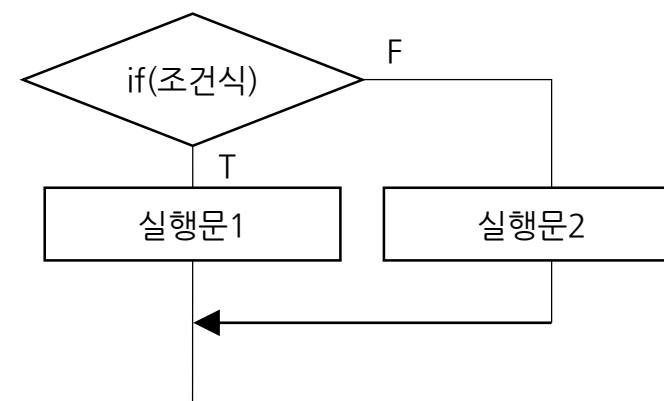
### (1) if

```
if(조건식) {  
    실행문;  
}
```



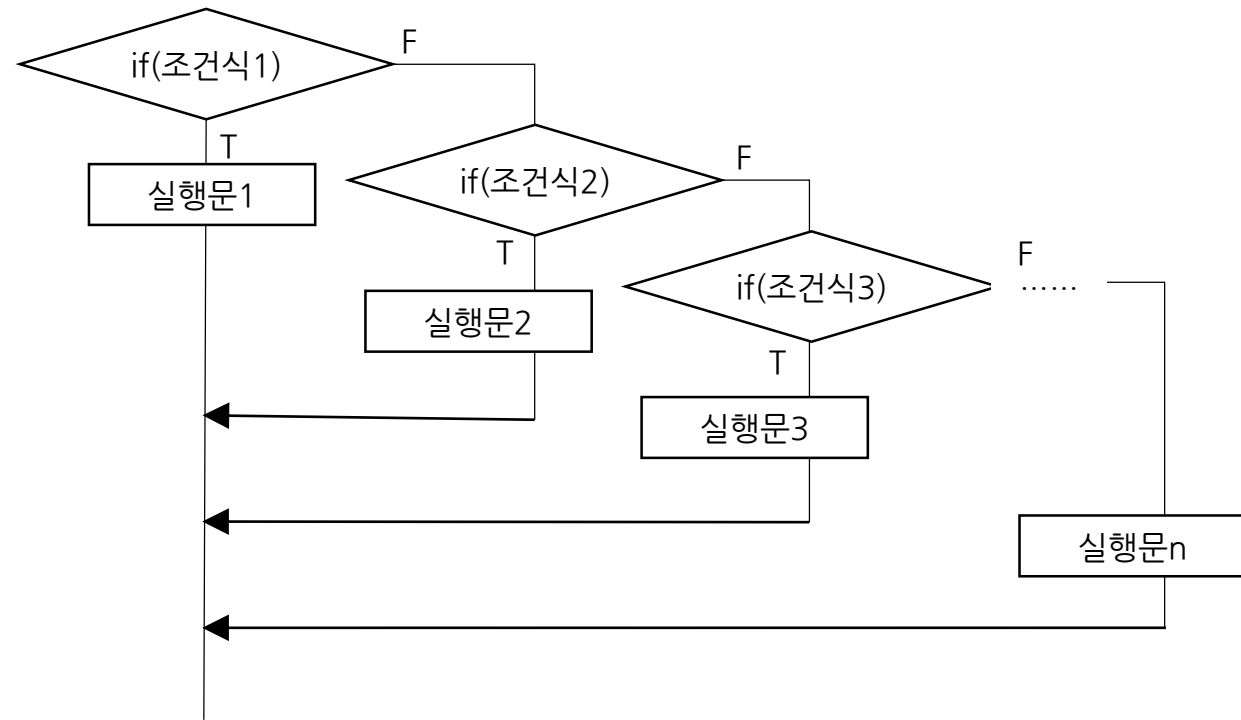
### (2) if~else

```
if(조건식){  
    실행문1;  
} else {  
    실행문2;  
}
```



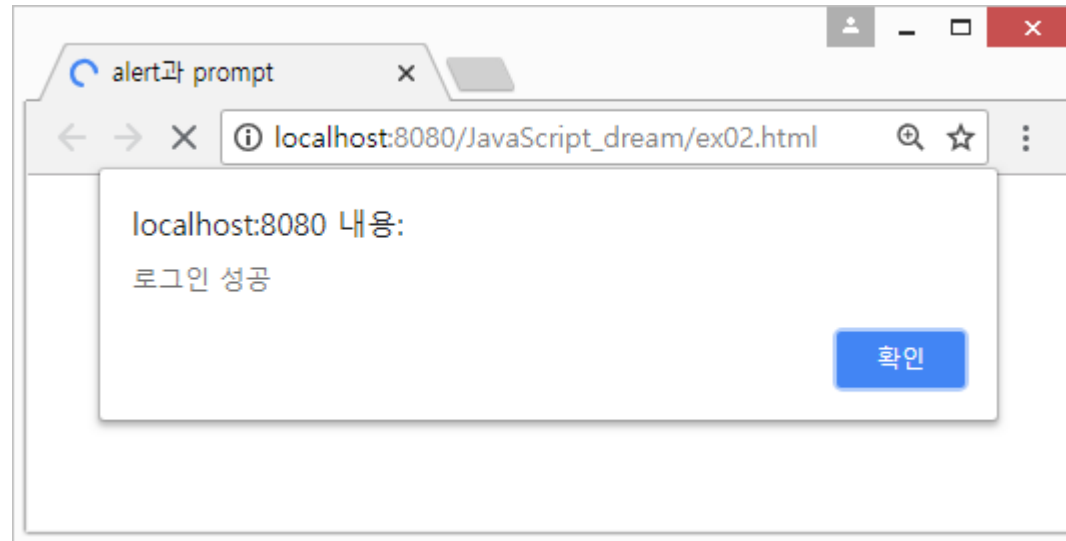
### (3) if~else if

```
if(조건식1){  
    실행문1;  
} else if(조건식2) {  
    실행문2;  
} else {  
    실행문n;  
}
```



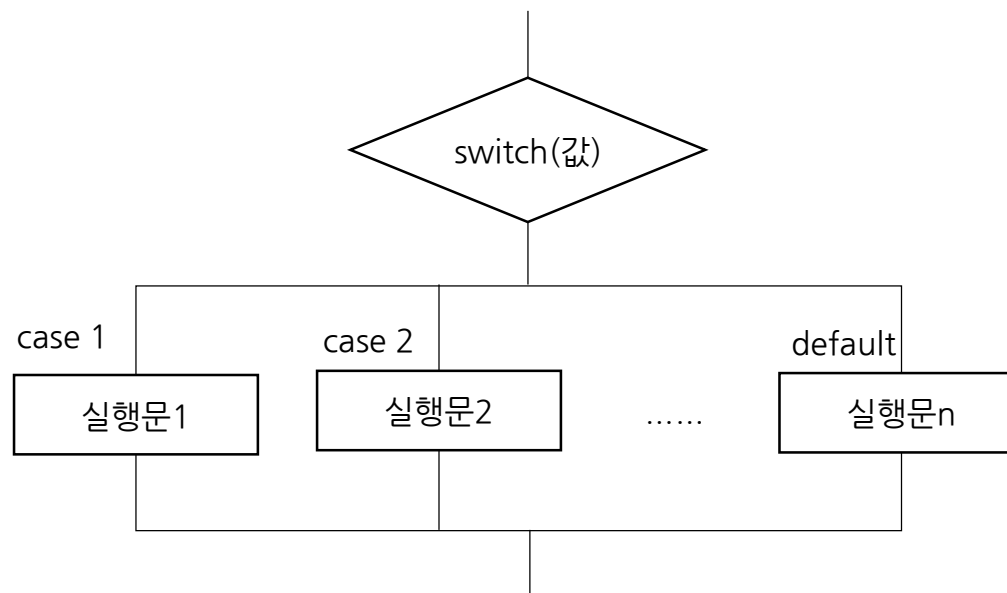
## ex02.html

```
<script>  
  var id="admin"; var pw="1234";  
  if(id=="admin" && pw=="1234")  
    alert("로그인 성공");  
  else  
    alert("ID 또는 PW가 일치하지 않습니다");  
</script>
```



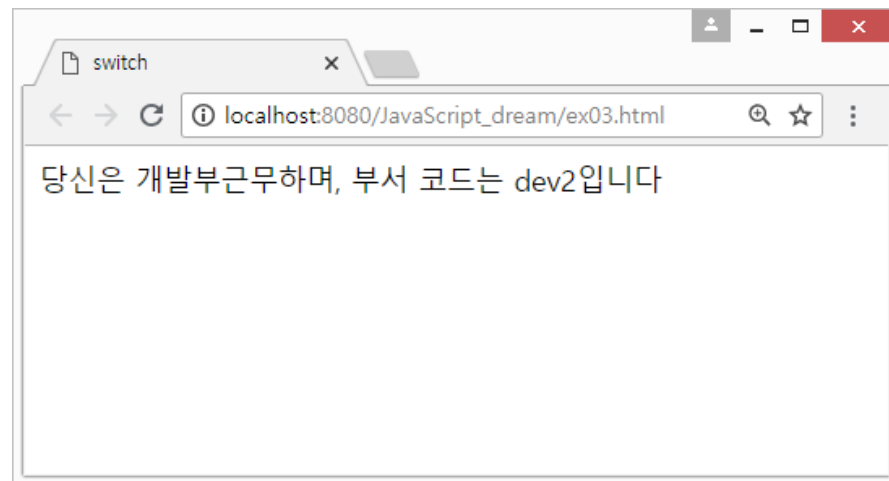
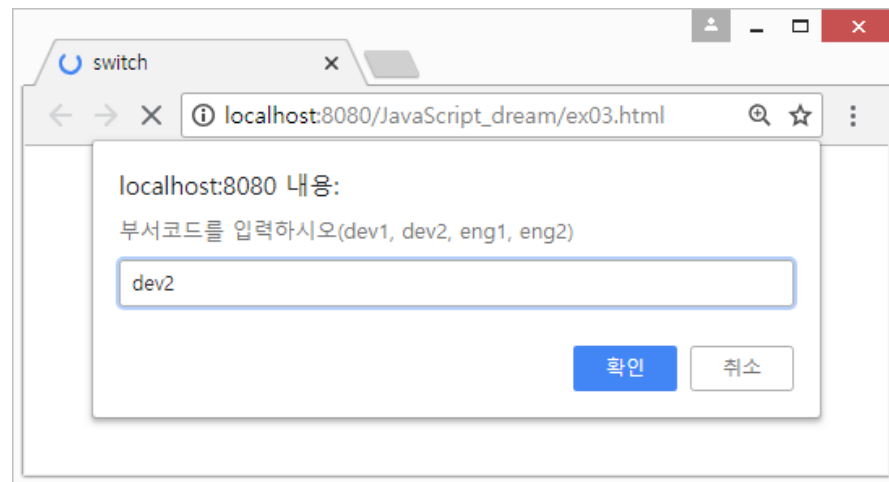
#### (4) switch

```
switch(값){  
    case n1: 실행문1; break;  
    case n2: 실행문2; break;  
    case n3: 실행문3; break;  
    :  
    [default:] 실행문n;  
}
```



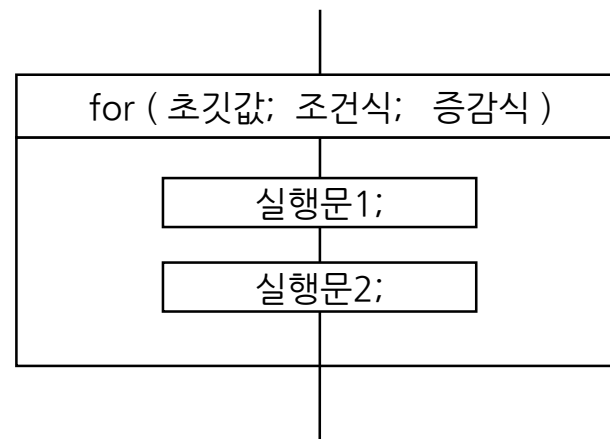
## ex03.html

```
<script>
var code=prompt("부서코드를 입력하시오(dev1, dev2, eng1, eng2)","");
var dept;
switch(code)
{
case "dev1":
case "dev2": dept="개발부"; break;
case "eng1":
case "eng2": dept="기술부"; break;
}
document.write("당신은 " + dept+ "근무하며, 부서 코드는 "
+ code +"입니다");
</script>
```



## (5) for

```
for(초깃값; 조건식; 증감식){  
    실행문;  
}
```



## ex04.html

```
<script>  
for(var a=1; a<=10; a++){  
    document.write(a+"&nbsp;&nbsp;&nbsp;");  
}  
  
for(var b=1; b<=6; b++){  
    document.write("<h"+b+">Hello World<"+"/h"+b+">");  
}  
</script>
```



1 2 3 4 5 6 7 8 9 10

**Hello World**

**Hello World**

**Hello World**

**Hello World**

**Hello World**

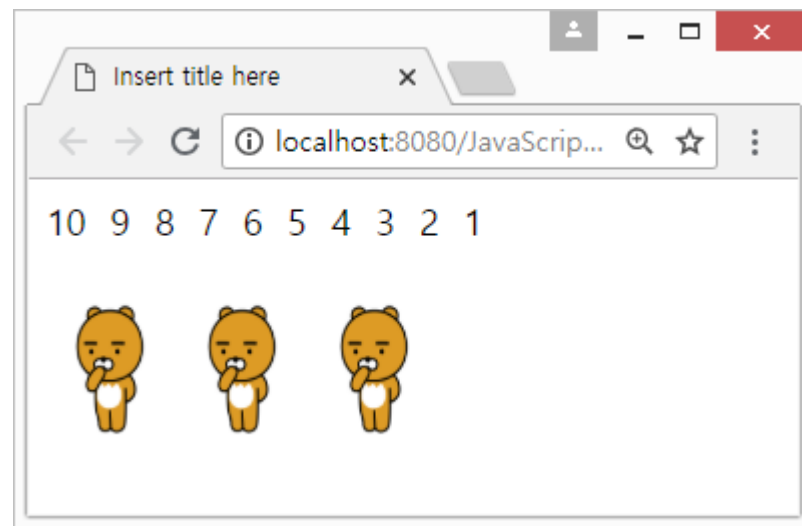
**Hello World**

## (6) while

```
while(조건식){  
    실행문;  
}
```

### ex05.html

```
<script>  
  var n=10;  
  while(n >= 1)  
  {  
    document.write(n+"&nbsp;&nbsp;&nbsp;");  
    --n;  
  }  
  document.write("<br><br>");  
  var m=0;  
  while(m < 3)  
  {  
    document.write("<img src='image/bear.gif' width='60px' height='70px'>");  
    ++m;  
  }  
</script>
```



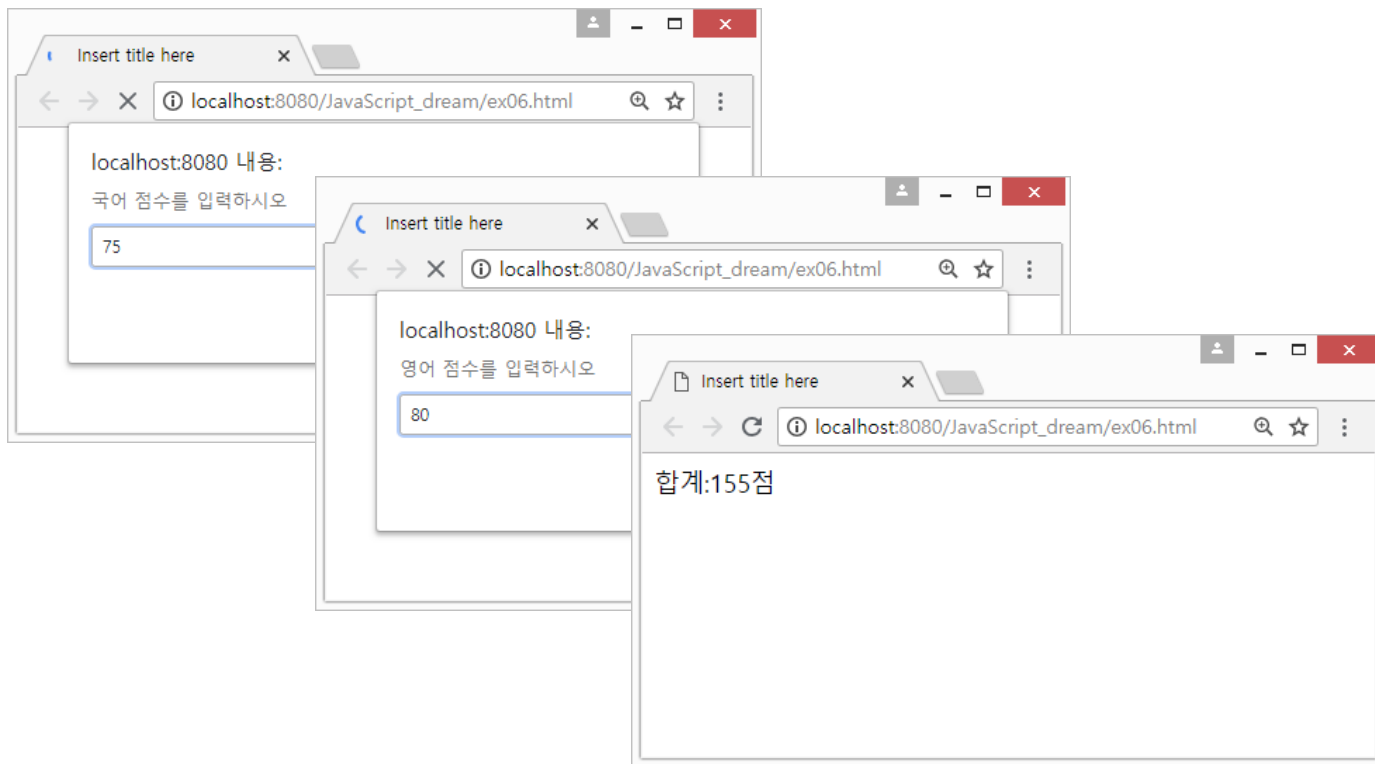


## (7) do~while

```
do{  
    실행문;  
} while(조건식);
```

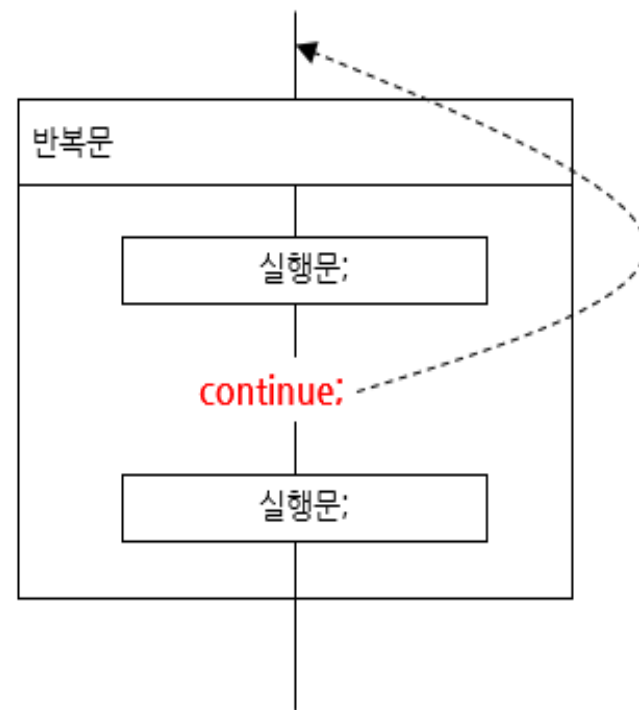
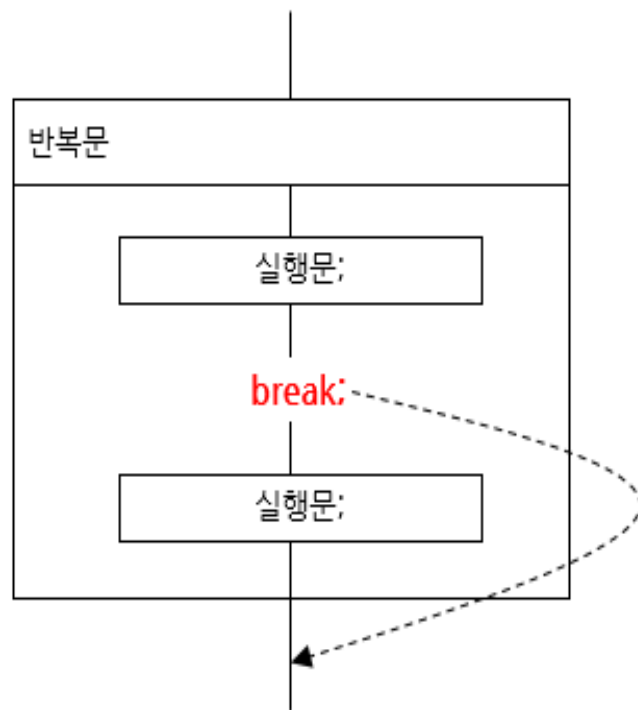
### ex06.html

```
<script>  
do{  
    kor=prompt("국어 점수를 입력하십시오","");  
    if(kor<0 || kor>100)  
        alert("0-100사이의 숫자를 입력하십시오");  
    }while(kor<0 || kor>100);  
  
do{  
    eng=prompt("영어 점수를 입력하십시오","");  
    if(eng<0 || eng>100)  
        alert("0-100사이의 숫자를 입력하십시오");  
    }while(eng<0 || eng>100);  
  
sum=parseInt(kor)+parseInt(eng);  
document.write("합계:" +sum+"점");  
</script>
```



## (8) 블록 탈출

```
반복문 {  
    break; 또는 continue;  
}
```



### 3. 함수(function)

관련 코드끼리 묶어서 분리하는 구조적 프로그램이다.

#### (1) 선언적 함수

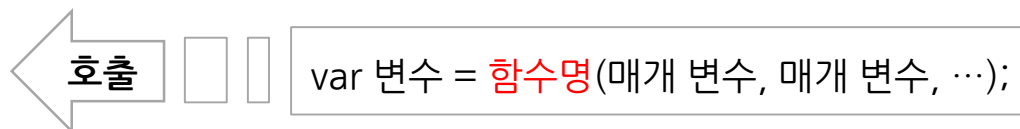
##### [형식 1]

```
function 함수명(){  
    실행문;  
}
```



##### [형식 2]

```
function 함수명(매개 변수, 매개 변수, ...){  
    return 리턴값;  
}
```



## (2) 익명 함수

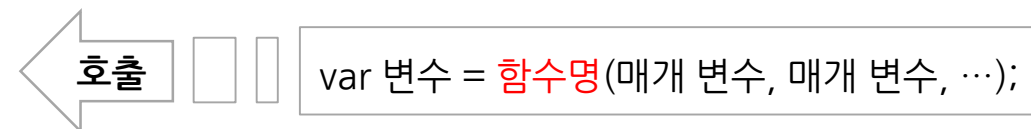
### [형식 1]

```
var 변수명 = function 함수명(){  
    실행문;  
}
```



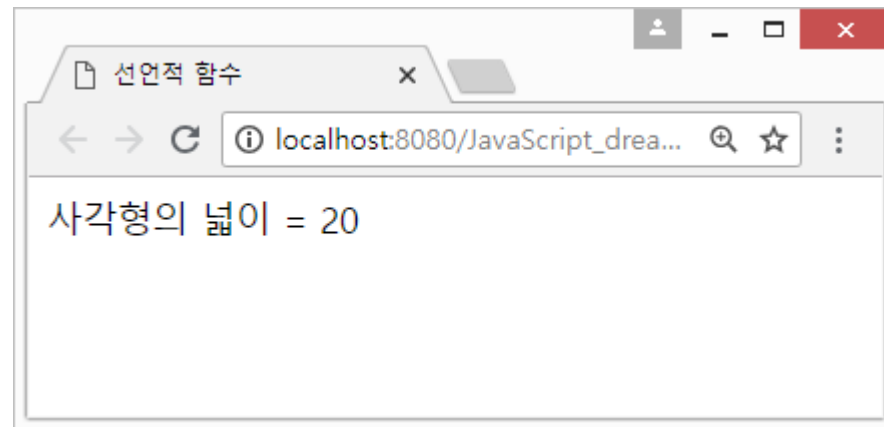
### [형식 2]

```
var 변수명 = function 함수명(매개 변수, 매개 변수, ...){  
    return 리턴값;  
}
```



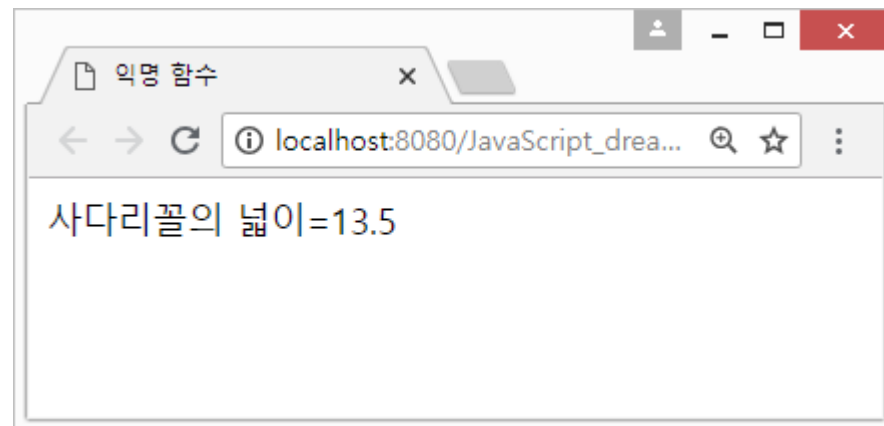
## ex07.html

```
<script>
function getSquareSize(bottom, height){
    return bottom * height;
}
document.write("사각형의 넓이 = " + getSquareSize(4,5));
</script>
```



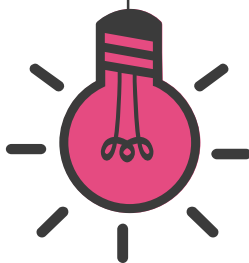
## ex08.html

```
<script>
var getTrapezoid = function(bottom, top, height){
    return (bottom + top)* height / 2.0;
}
document.write("사다리꼴의 넓이=" + getTrapezoid(5,4,3));
</script>
```



03

## 자바스크립트 객체



# 1. 표준 객체

기본 자료형을 제외하고는 배열, 함수 등 모두가 객체이다.  
객체란 데이터와 데이터를 처리하는 모든 기술의 조합이다.

## Date객체

	메서드	설명
	getFullYear( ) / getFullYear( )	연도만 설정하거나 반환한다.
	setMonth( ) / getMonth( )	월만 설정하거나 반환한다.
	getDate( ) / getDate( )	일(월 기준)을 설정하거나 반환한다.
	setDay( ) / getDay( )	일(주 기준)을 설정하거나 반환한다.
	setHour( ) / getHour( )	시간을 설정하거나 반환한다.

## String객체

	메서드	설명
속성	length	문자열의 길이를 구한다.
	big( )/small( )	글자를 한 단계 크게/작게 설정한다.
	blink( )	글자를 깜박이게 설정한다.
문자	bold( )	글자를 굵게 설정한다.
	sub( )/sup( )	아래 첨자 / 위 첨자로 설정한다.
	fontSize(크기)	글자의 크기를 설정한다.(1-7까지)
	fontcolor(색상)	글자의 색상을 설정한다.
	toLowerCase( )	글자를 소문자로 변경한다.
	toUpperCase( )	글자를 대문자로 변경한다.

## Array객체

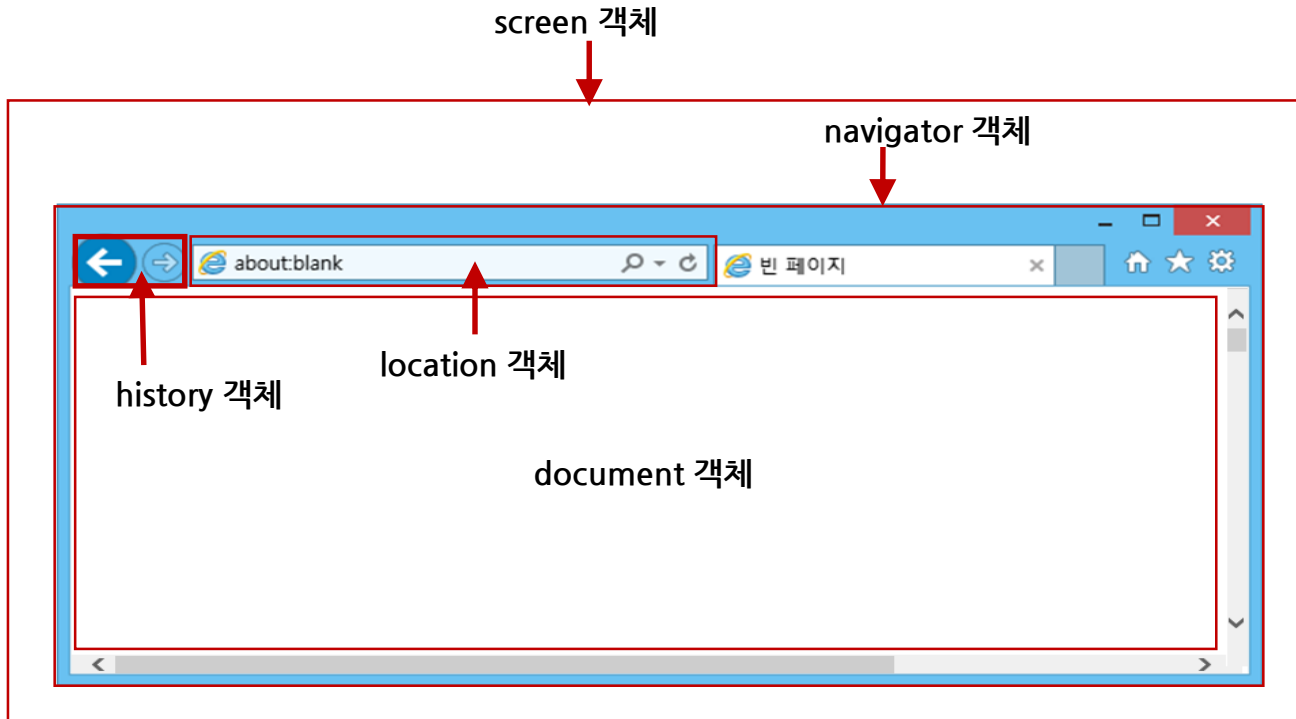
	메서드	설명
	sort( )	배열 값들을 오름차순으로 정렬한.
	reverse( )	배열 값들을 역순으로 바꾼다.
	concat(array)	두 개의 배열을 합하여 하나의 배열로 만든다.

## Math객체

	메서드	설명
	max(n1, n2, ...)	가장 큰 값을 반환한다.
	min(n1, n2, ...)	가장 작은 값을 반환한다.
	round(n)	반올림 값을 반환한다.
	ceil(n)	올림 값을 반환한다.
	floor(n)	내림 값을 반환한다.
	abs(n)	절댓값을 반환한다.
	random(n)	0~1사이의 임의의 수를 반환한다.
	pow()	숫자의 거듭제곱을 계산하여 반환한다.

## 2. 브라우저 객체모델 (BOM)

브라우저 객체 모델(Browser Object Model)은 웹 브라우저와 관련된 모든 객체들의 집합이다.



객체	설명
location 객체	브라우저 객체 모델의 종류로는 URL의 정보를 제공
navigator 객체	현재 실행 중인 브라우저의 정보를 제공
history 객체	방문 기록에 대한 정보를 제공
screen 객체	모니터의 정보를 제공
document 객체	문서에 대한 정보를 제공



## ex09.html

```
<script>
  var n=0;
  function changeEx()
  {
    f=document.getElementById("fm");
    n++;
    switch(n)
    {
      case 1: f.style.color="red";      break;
      case 2: f.style.color="green";    break;
      case 3: f.style.color="blue"; n=0; break;
    }
    window.setTimeout("changeEx()", 500) //호출함수, 호출주기
  }
</script>

<body onload="changeEx()">
  <font id="fm">안녕하세요</font>
</body>
```



안녕하세요



0.5초 간격으로 changeEx()호출

안녕하세요

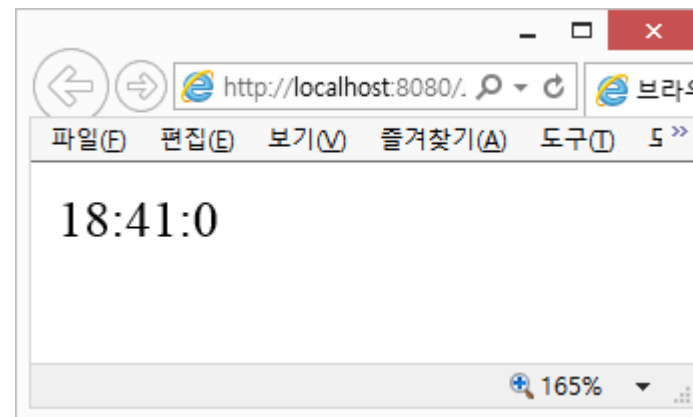


0.5초 간격으로 changeEx()호출

안녕하세요

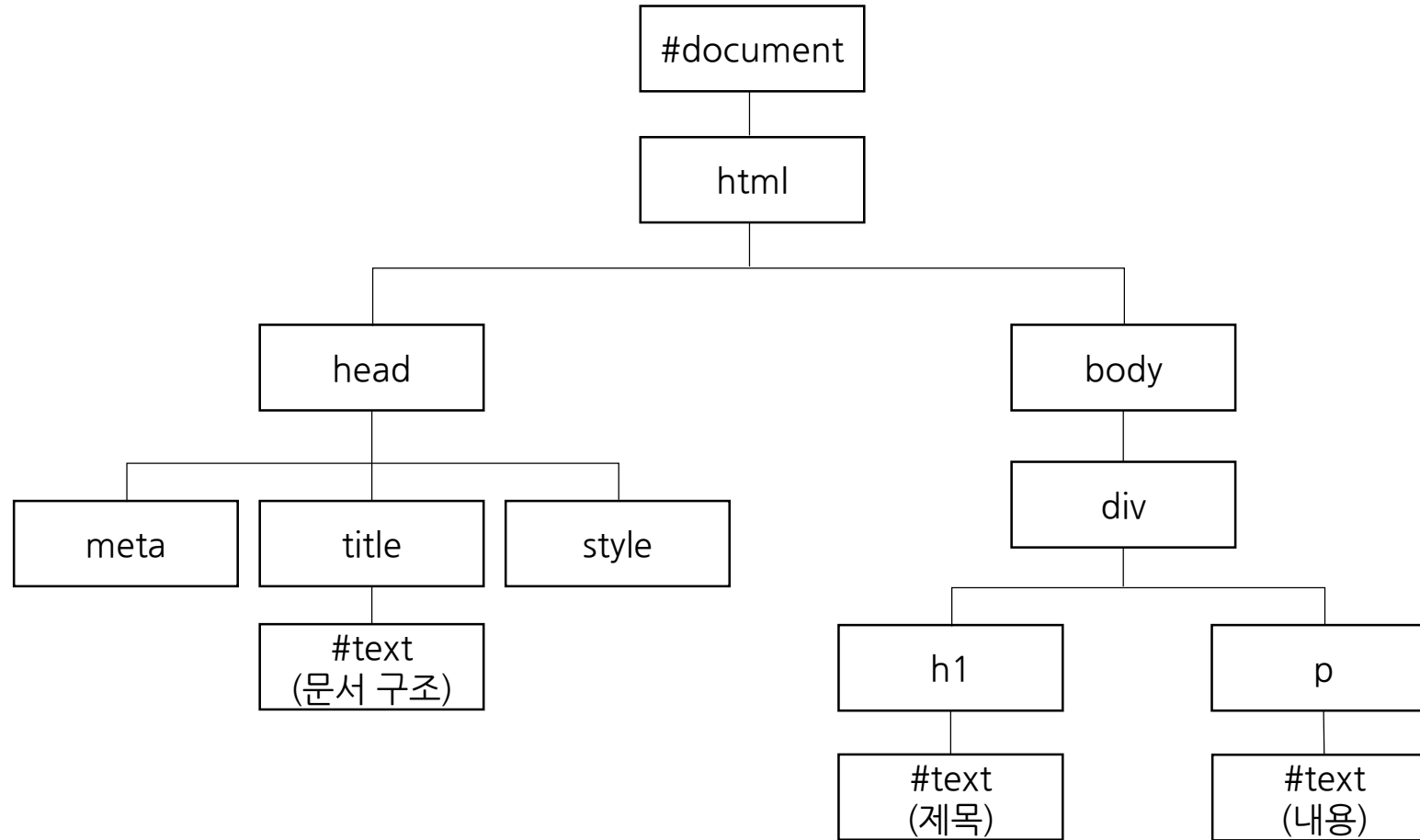
## ex10.html

```
<script>
  var t=new Date();
  document.write( t.getHours()+":"+ t.getMinutes()+":"+ t.getSeconds());
  setInterval(function(){
    location.reload();
  },1000);
</script>
```



### 3. 문서 객체모델 (DOM)

문서 객체 모델(Document Object Model)은 document와 관련된 집합이다.



## 선택자

메서드	설명
<code>getElementById( ' id명 ' )</code>	태그의 id 속성이 <code>getElementById( ' id 명 ' )</code> 과 일치하는 문서 객체를 가져오는 선택자이다.
<code>getElementsByName( ' name명 ' )</code>	태그의 name 속성이 <code>getElementsByName( ' name 명 ' )</code> 과 일치하는 문서 객체를 가져오는 선택자이다.
<code>getElementsByTagName( ' tag명 ' )</code>	' tag 명 ' 과 일치하는 문서 객체를 가져오는 선택자이다.

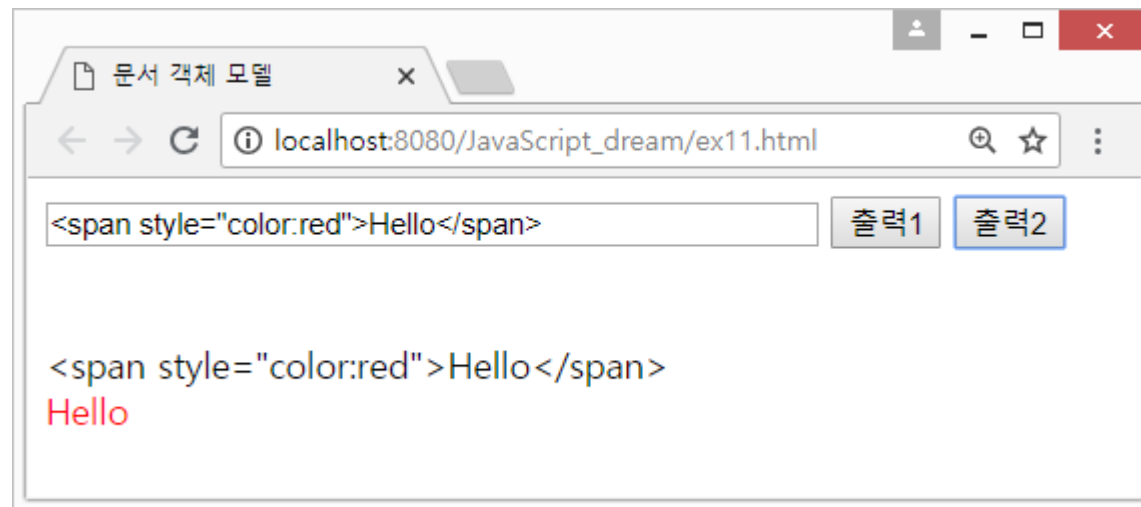
## DOM 객체 메서드

메서드	설명
<code>createElement(tagName)</code>	요소 노드를 생성한다.
<code>createTextNode(text)</code>	텍스트 노드를 생성한다.
<code>removeChild(child)</code>	자식 노드를 삭제한다.
<code>appendChild(node)</code>	자식 노드를 추가한다.
<code>setAttribute(name, value)</code>	객체의 속성을 설정한다.
<code>getAttribute(name)</code>	객체의 속성을 가져온다.

## ex11.html

```
<script>
function process1(){
    var ob=document.getElementById("txt").value;
    document.getElementById("view1").innerText=ob;
}
function process2(){
    var ob=document.getElementById("txt").value;
    document.getElementById("view2").innerHTML=ob;
}
</script>

<body>
    <input type="text" id="txt" size="50" name="text">
    <input type="button" value="출력1" onclick="process1()">
    <input type="button" value="출력2" onclick="process2()">
    <br><br><br>
    <div id="view1"></div>
    <div id="view2"></div>
</body>
```

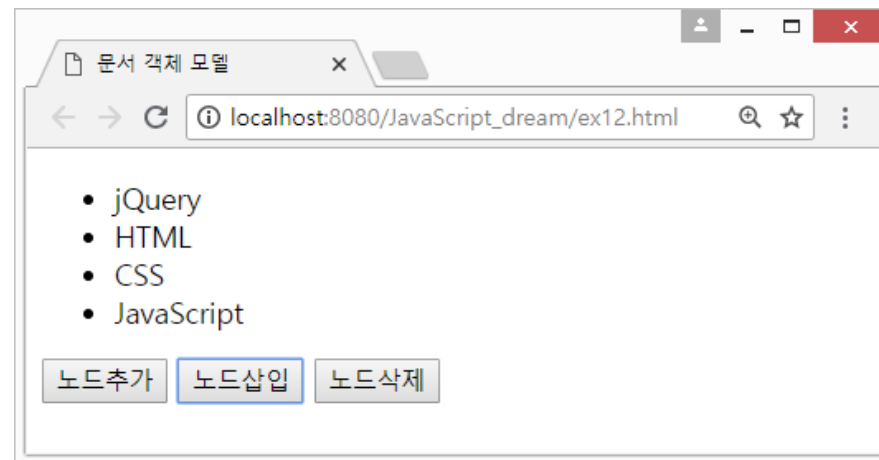


## ex12.html

```
<script>
function appendNode(){
    var target=document.getElementById("target");
    var li=document.createElement("li");
    var text=document.createTextNode("JavaScript");
    li.appendChild(text);
    target.appendChild(li);
}
function insertNode(){
    var target=document.getElementById("target");
    var li=document.createElement("li");
    var text=document.createTextNode("jQuery");
    li.appendChild(text);
    target.insertBefore(li, target.firstChild);
}
function removeNode(){
    var target=document.getElementById("target");
    target.parentNode.removeChild(target);
}
</script>

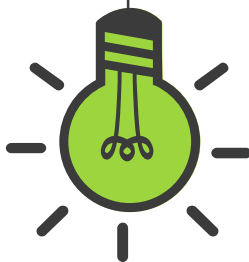
<body>
<ul id="target">
    <li>HTML</li>
    <li>CSS</li>
</ul>

<input type="button" onclick="appendNode()" value="노드추가">
<input type="button" onclick="insertNode()" value="노드삽입">
<input type="button" onclick="removeNode()" value="노드삭제">
</body>
```



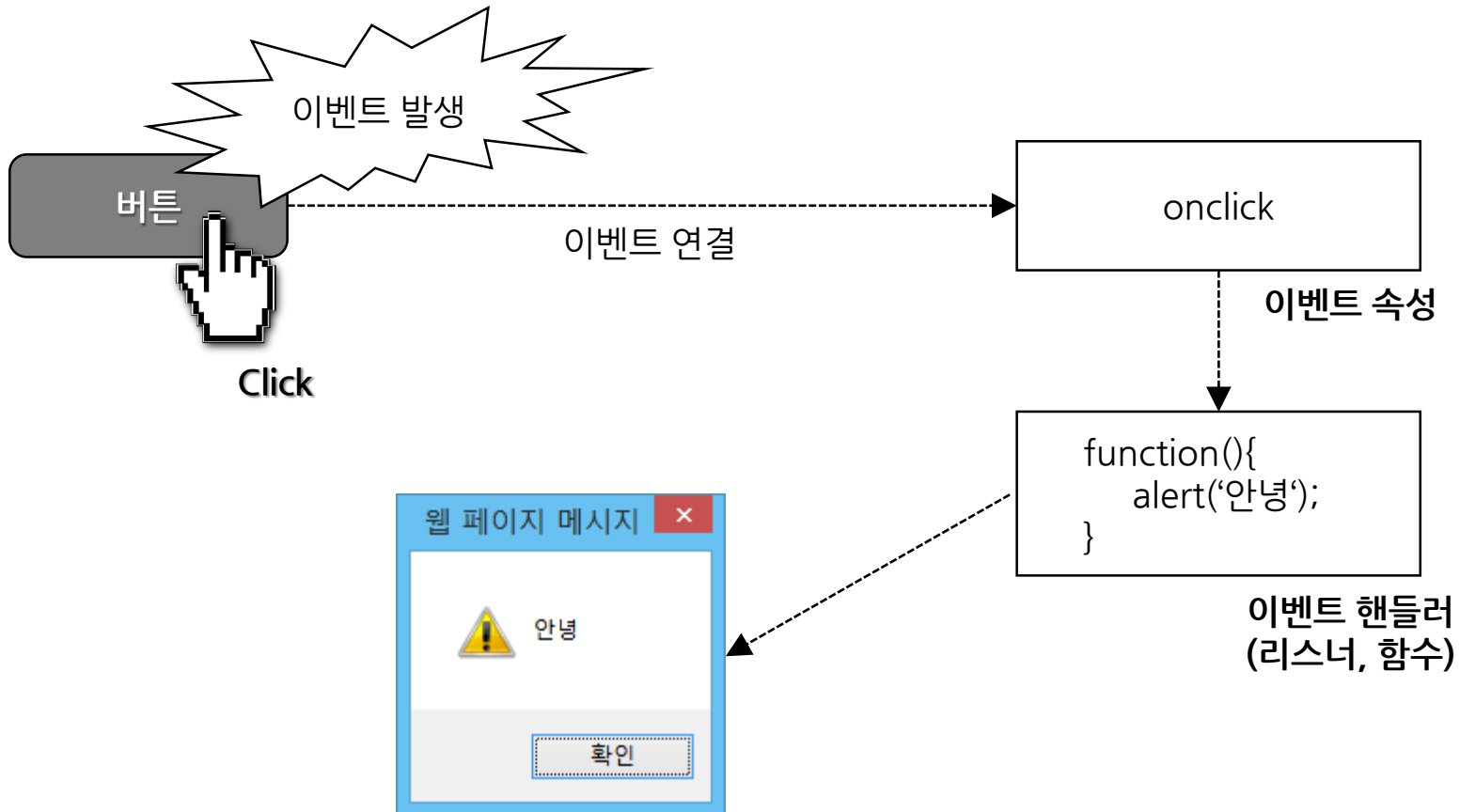
04

## 이벤트 모델



# 1. 이벤트 모델이란?

이벤트란 키보드 입력, 마우스 클릭과 같이 사용자가 특정 행위를 했을 때 행위에 대한 결과 동작을 보여주는 것이다.





## (1) 이벤트 유형



## 마우스 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
click	onclick()	마우스가 요소를 클릭할 때 발생한다.
mouseup	onmouseup()	마우스를 요소 위에 놓을 때 발생한다.
mousedown	onmousedown()	마우스로 요소를 누를 때 발생합니다.
mousemove	onmousemove()	마우스가 요소 위에서 움직일 때 발생한다.
mouseenter	onmouseenter()	포인터가 요소 위로 이동할 때 발생한다.
mouseleave	onmouseleave ()	포인터가 요소 밖으로 이동할 때 발생한다.
mouseover	onmouseover()	포인터를 요소 위로 이동하거나 자식 요소 중 하나 위로 이동할 때 발생한다.
mouseout	onmouseout()	포인터를 요소 밖으로 이동하거나 자식 요소 중 하나에서 벗어날 때 발생한다.

## 키보드 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
keydown	onkeydown()	키보드를 처음 누를 때 발생한다.
keyup	onkeyup()	키보드에서 손을 땔 때 발생한다.
keypress	onkeypress()	키보드에 문자가 입력될 때 발생한다.

## 폼 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
change	onchange( )	내용을 변경할 때 발생한다.
submit	onsubmit( )	입력 내용을 전송할 때 발생한다.
reset	onreset( )	입력 내용을 초기화할 때 발생한다.

## UI 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
load	onload( )	웹페이지의 로드가 완료되었을 때 발생한다
resize	onresize()	브라우저의 창 크기를 변경할 때 발생한다.

## 포커스 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
focus	onfocus( )	포커스를 얻었을 때 발생한다.
blur	onblur( )	포커스를 잃었을 때 발생한다.

## 드래그 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
drag	ondrag( )	요소가 드래그 되고 있을 때 발생한다.
dragend	ondragend( )	요소의 드래그가 끝나면 발생한다.
drangenter	ondrangenter( )	요소에서 드롭 될 때 발생한다.
dragleave	ondragleave( )	요소가 놓을 위치를 떠날 때 발생한다.
dragover	ondragover( )	요소가 놓을 위치에 있을 때 발생한다.
dragstart	ondragstart( )	요소를 드래그하기 시작할 때 발생한다.
drop	ondrop( )	요소가 놓을 위치에 놓이면 발생한다.
draggable		요소를 드래그 할 수 있는지 여부를 지정한다. (true 또는 false)


## 터치 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
touchcancel	ontouchcancel( )	터치가 중단되면 발생한다.
touchmove	ontouchmove( )	스크린에 드래그하면 발생한다.
touchstart	ontouchstart( )	스크린에 터치하면 발생한다.

## (2) 이벤트 모델 종류

DOM Level 0		DOM Level 2
모델 종류	<ul style="list-style-type: none"><li>• 인라인 이벤트 모델</li><li>• 고전 이벤트 모델</li></ul>	<ul style="list-style-type: none"><li>• 인터넷 익스플로러 이벤트 모델</li><li>• 표준 이벤트 모델</li></ul>
이벤트 연결	하나의 이벤트에 하나의 리스너만 연결 가능하다.    하나의 이벤트에 여러 개의 리스너를 연결할 수 있다.	

ex13.html (인라인 이벤트 모델)

[illegible]

A screenshot of a web browser window. The address bar shows the URL 'localhost:8080/JavaScript\_dream/ex18.html'. The page content displays a simple calculator interface with the following text and inputs:

- A값 :
- B값 :
- 결과 :
-

## ex14.html (인라인 이벤트 모델)

```
<script>
function process()
{
    var x=parseInt(document.getElementById('x').value);
    var y=parseInt(document.getElementById('y').value);
    var op=document.getElementById('ope').value;
    var solValue;

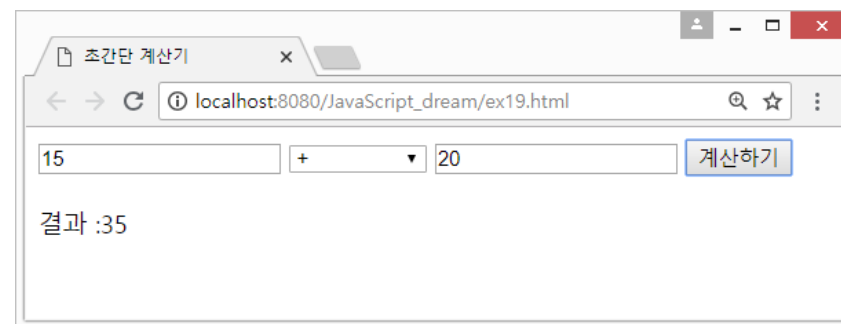
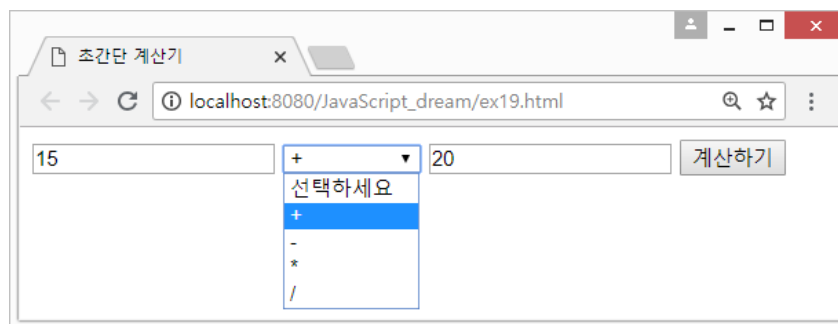
    switch(op)
    {
        case "+":solValue=x+y;    break;
        case "-":solValue=x-y;    break;
        case "*":solValue=x*y;    break;
        case "/":solValue=x/y;    break;
        default:alert("연산자 오류");
    }
    document.getElementById('sol').innerText="결과 : " + solValue;
}
</script>
```

```
<body>
    <input type="text" name="x" id="x" />

    <select id="ope">
        <option >선택하세요</option>
        <option value="+>+</option>
        <option value="->-</option>
        <option value="*>*</option>
        <option value="/>/</option>
    </select>

    <input type="text" name="y" id="y" />
    <input type="button" value="계산하기" onclick="process()"><br><br>

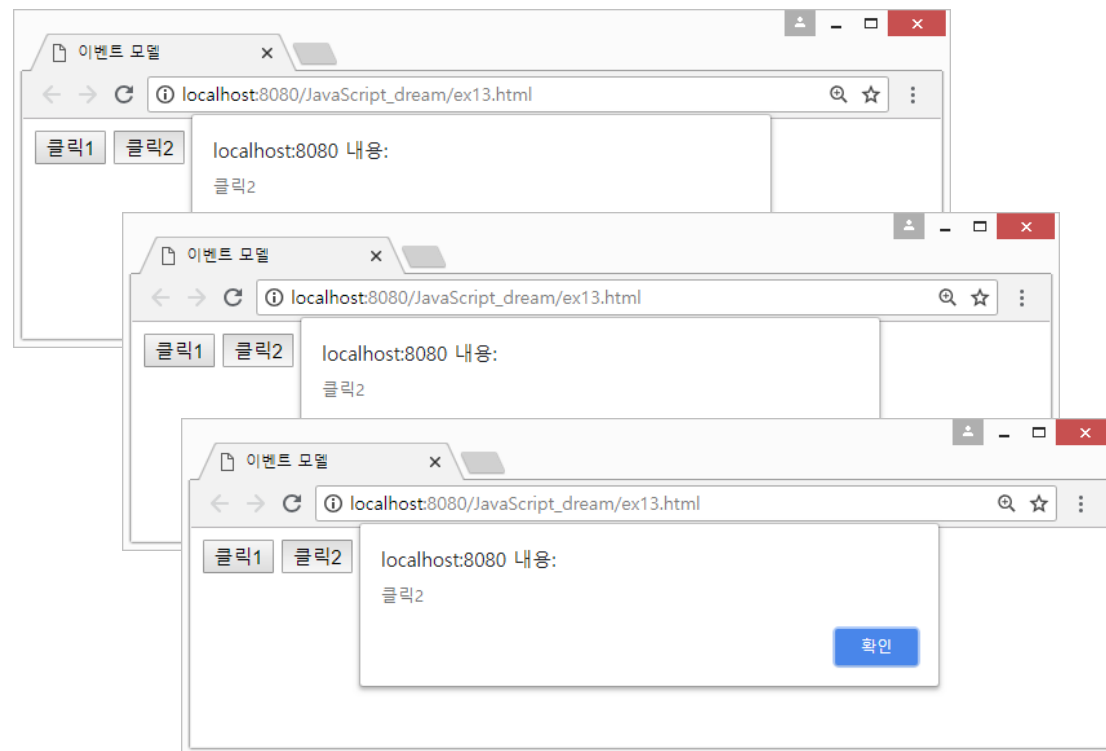
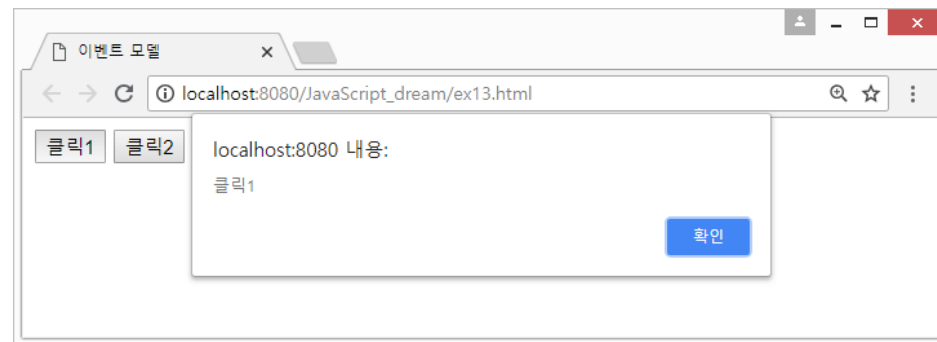
    <span id="sol"></span>
</body>
```



## ex15.html (고전 이벤트 모델)

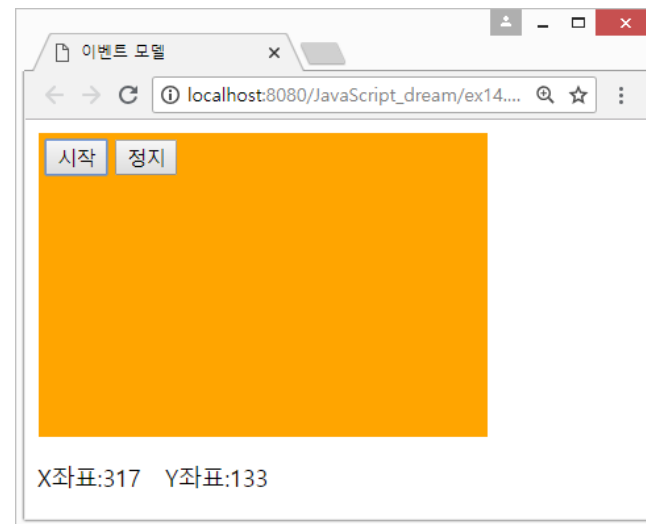
```
<script>
  window.onload = function() {
    var btn = document.getElementById('btn1');
    btn.onclick = function() {
      alert('클릭1');
      btn.onclick = null;
    }
  };
</script>

<body>
<input type="button" id="btn1" value="클릭1">
<input type="button" id="btn2" value="클릭2" onclick="alert('클릭2')">
</body>
```



ex16.html (표준 이벤트 모델)

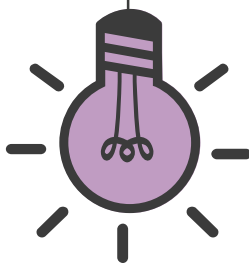
```
<body>  
  <div id="box">  
    <button onclick="addEventHandler()" id="startbtn">시작</button>  
    <button onclick="removeHandler()" id="stopbtn">정지</button>  
  </div>  
  <p id="demo"></p>  
  
  <script>  
    function myFunction() {  
      document.getElementById("demo").innerHTML = " X좌표:"  
        + event.x + "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;" + "Y좌표:" + event.y;  
    }  
    function addEventHandler() {  
      document.getElementById("box").addEventListener("mousemove", myFunction);  
    }  
    function removeHandler() {  
      document.getElementById("box").removeEventListener("mousemove", myFunction);  
    }  
  </script>  
</body>
```





05

## JSON과 정규표현식



# 1. JSON

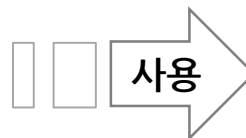
JSON은 저중량(lightweight) 데이터를 교환하는 형식으로 컴퓨터 시스템이 파싱하고 생성하기 쉽다.

[형식 1] 객체인 경우

```
{"이름1":"값1", "이름2":"값2"}
```

[형식 2] 배열인 경우

```
[{"이름1":"값1", "이름2":"값2"}, {"이름1":"값1", "이름2":"값2"}]
```



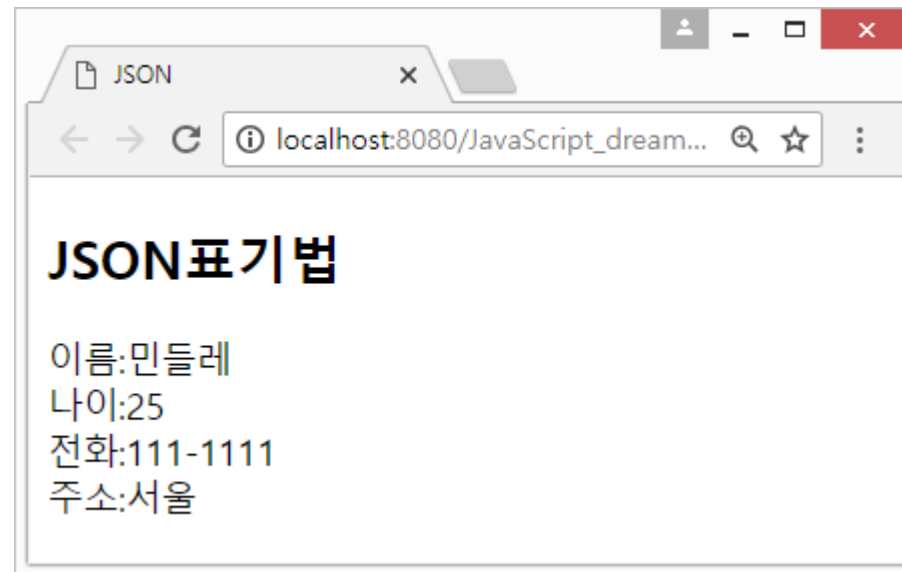
객체명.이름 또는 객체명['이름']

## ex17.html

```
<script>
function init(){
    var emp={"name":"민들레",
            "age":"25",
            "tel":"111-1111",
            "address":"서울"};

    var msg="";
    msg += "이름:" + emp.name + "<br>"
        + "나이:" + emp.age + "<br>"
        + "전화:" + emp['tel'] + "<br>"
        + "주소:" + emp['address'] + "<br>"
    document.getElementById("result").innerHTML=msg;
}
</script>

<body onload="init()">
    <h2>JSON표기법</h2>
    <div id="result"></div>
</body>
```



## ex18.html

```
<script>
  var emp = {
    "id" : "abcd",
    "pw" : "1234",
    "name" : "홍길동",
    "point" : "1250"
  };
  function login(){
    var a = document.getElementById("id").value;
    var b = document.getElementById("pw").value;

    if(emp.id == a && emp.pw == b)
      document.getElementById("confirm").innerHTML
        ="<span style='color:green'>" + emp.name
        + "님 환영합니다.<br>현재 포인트는 " + emp.point + "점 입니다.</span>";
    else
      document.getElementById("confirm").innerHTML
        ="<span style='color:red'>아이디 또는 비밀번호가 틀립니다.</span>";
  }
</script>

<body>
  <label for="input">[로그인 화면]</label><br><br>
  <input type="text" id="id"><span>아이디를 입력하십시오</span><br>
  <input type="text" id="pw"><span>비밀번호를 입력하십시오</span><br>
  <button id="log" onclick="login()">Login</button><br><br>
  <span id="confirm"></span>
</body>
```



JSON로그인 x

localhost:8080/JavaScript\_dream...

[로그인 화면]

abcd 아이디를 입력하십시오

1234 비밀번호를 입력하십시오

Login

홍길동님 환영합니다.  
현재 포인트는 1250점 입니다.

JSON로그인 x

localhost:8080/JavaScript\_dream...

[로그인 화면]

abcd 아이디를 입력하십시오

1111 비밀번호를 입력하십시오

Login

아이디 또는 비밀번호가 틀립니다.

## 2. 정규 표현식

정규 표현식은 조합된 문자열이 특정 규칙에 맞게 작성되었는지 찾아내기 위한 검색 패턴이다.

### [형식]

```
var 객체명 = new RegExp('정규표현식','[Flag]');  
var 객체명 = /정규표현식/[Flag];
```

표현식	설명
^	문자열의 시작한다.
\$	문자열의 종료한다.
.	임의의 한 문자이다. (단, \w 는 제외)
*	앞 문자가 없을 수도 무한정 많을 수도 있다.
+	앞 문자가 하나 이상입니다
?	앞 문자가 없거나 하나 있습니다.
{ }	횟수 또는 범위를 나타냅니다.

사용 예) `^[0-9]*$`

^ 으로 우선 패턴의 시작을 알린다.

[0-9] 괄호 사이에 두 숫자를 넣어 범위를 지정해줄 수 있다.

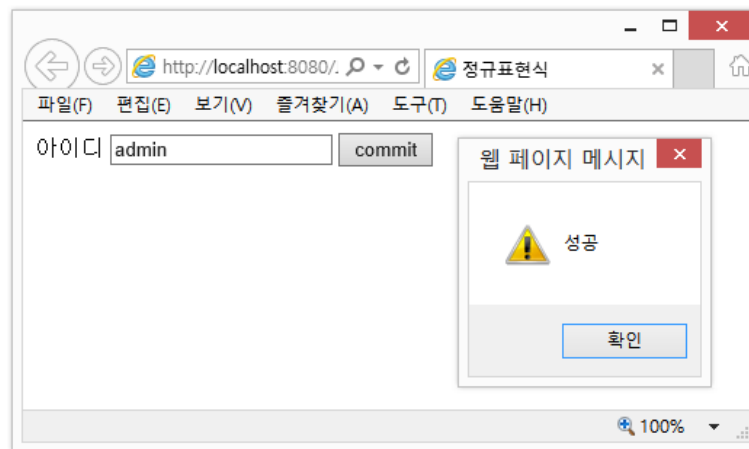
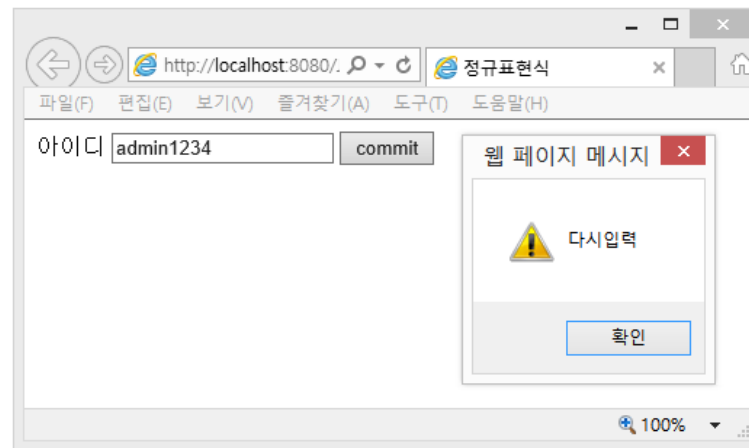
\* 를 넣으면 글자 수를 상관하지 않고 검사한다.

\$ 로 패턴의 종료를 알린다.

## ex19.html

### [아이디 입력 조건]

1. 입력되는 전체 길이는 5-8자이어야 한다.
2. 아이디는 숫자와 영문자로만 사용 가능하다.

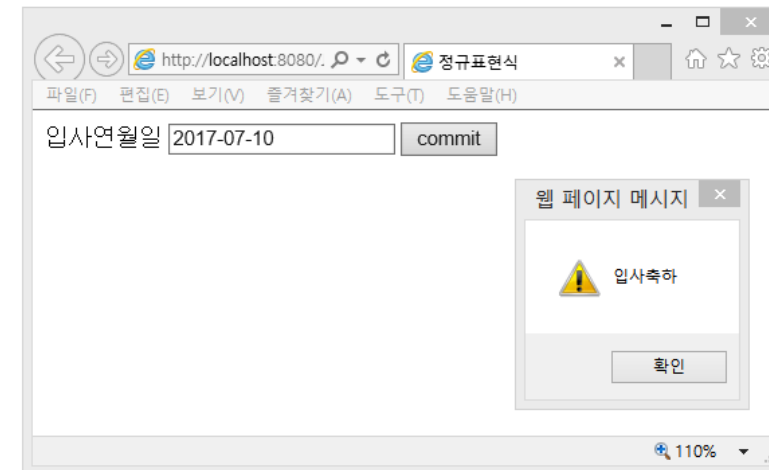
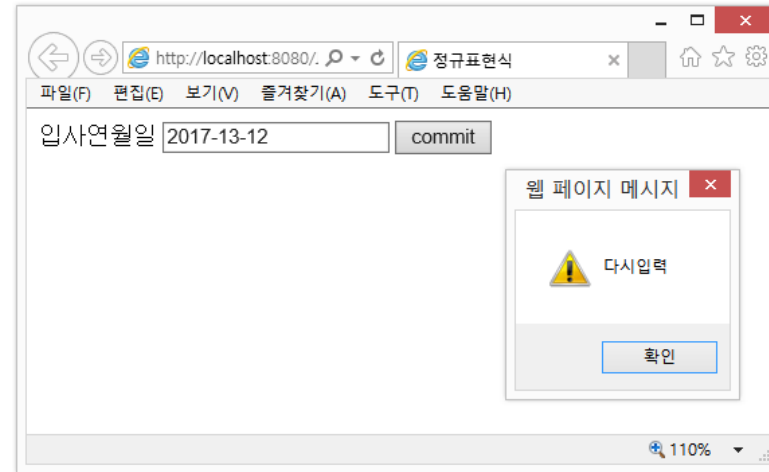


## ex20.html

[날짜 입력 조건]

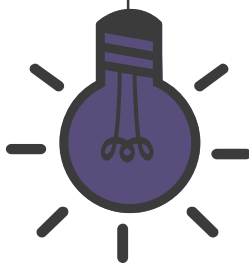
연도 4자리, 월이나 일은 1자리이거나 2자리로 합니다.

예시: 2018-05-24



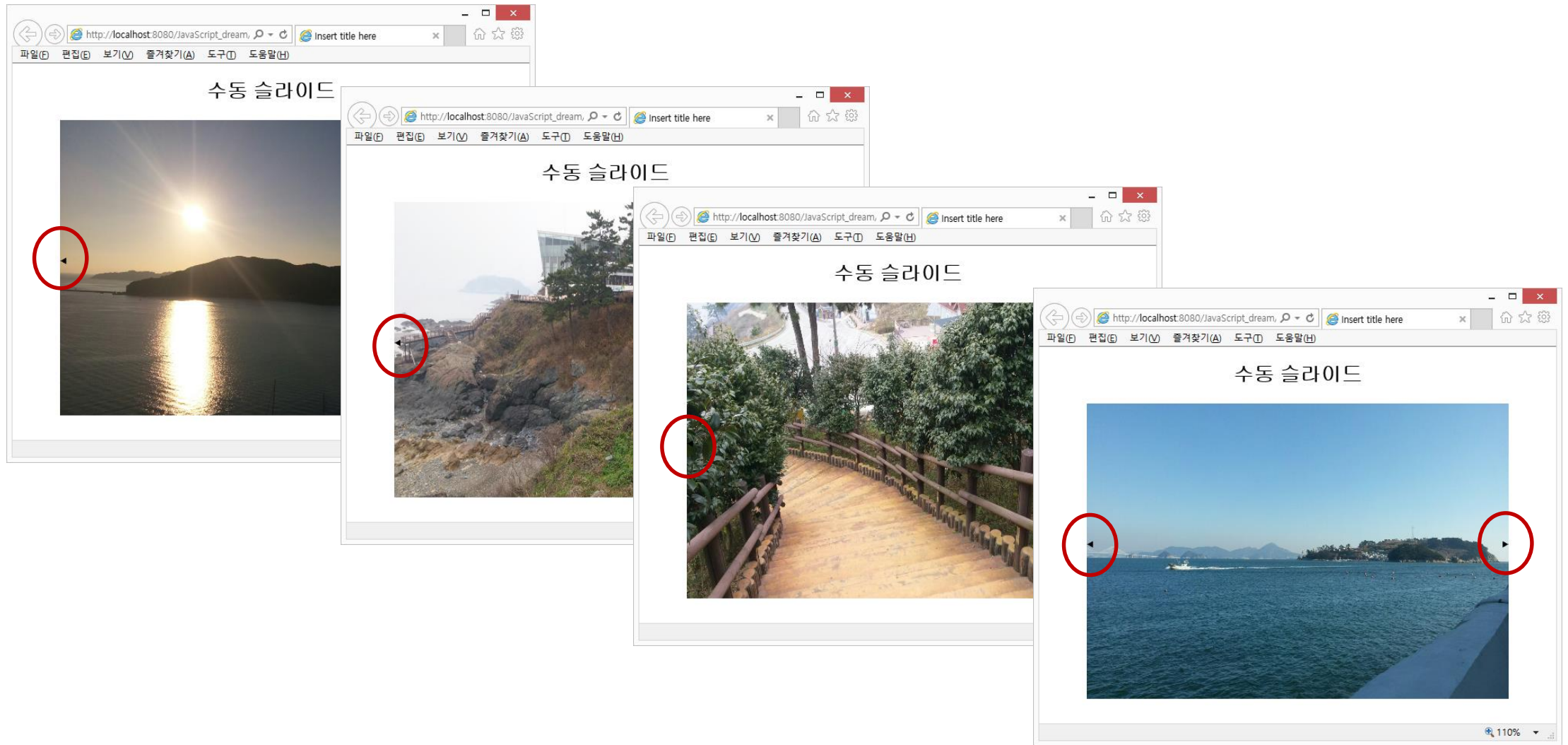
06

## 자바스크립트 종합 실습





## ex21.html (수동 슬라이드)



## 수동 슬라이드 코딩 분석

```
<body onload="showDivs(1)">
<div align="center" class="large">
  <h2 align="center">수동 슬라이드</h2>
  <div class="small">
    
    
    
    

    <a class="leftbtn" onclick="plusDivs(-1)">◀</a>
    <a class="rightbtn" onclick="plusDivs(1)">▶</a>

  </div>
</div>
```

## 수동 슬라이드 코딩 분석

```
<script>
var slideIndex=1;
function plusDivs(n) {
  showDivs(slideIndex += n);
}
function showDivs(n) {
  var i;
  var x = document.getElementsByClassName("mySlides");
  if (n > x.length) {slideIndex = 1}
  if (n < 1) {slideIndex = x.length} ;
  for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
  x[slideIndex-1].style.display = "block";
}
```



showDivs(slideIndex += n)

1

## 수동 슬라이드 코딩 분석

```
<script>
```

```
var slideIndex=1;
```

```
function plusDivs(n) {
```

```
    showDivs(slideIndex += n);
```

```
}
```

```
function showDivs(n) {
```

```
    var i;
```

```
    var x = document.getElementsByClassName("mySlides");
```

```
    if (n > x.length) {slideIndex = 1}
```

```
    if (n < 1) {slideIndex = x.length} ;
```

```
    for (i = 0; i < x.length; i++) {
```

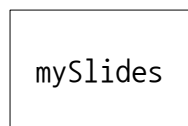
```
        x[i].style.display = "none";
```

```
    }
```

```
    x[slideIndex-1].style.display = "block";
```

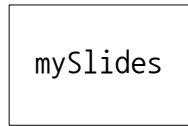
```
}
```

그림1



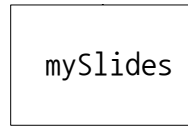
x[0]

그림2



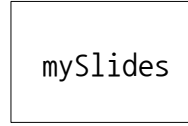
x[1]

그림3



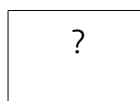
x[2]

그림4



x[3]

n



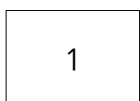
>

x.length

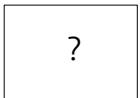


➔

slideIndex



n



<

1

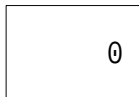
➔

slideIndex



x.length는 그림의 갯수 ➔ 4

i



➔

1

➔

2

➔

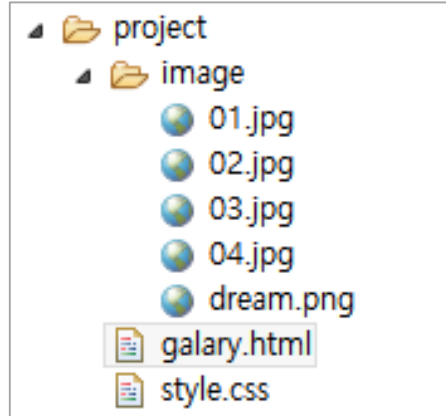
3

x[i].style.display= " none " ➔ 4개의 이미지를 안보이게 함

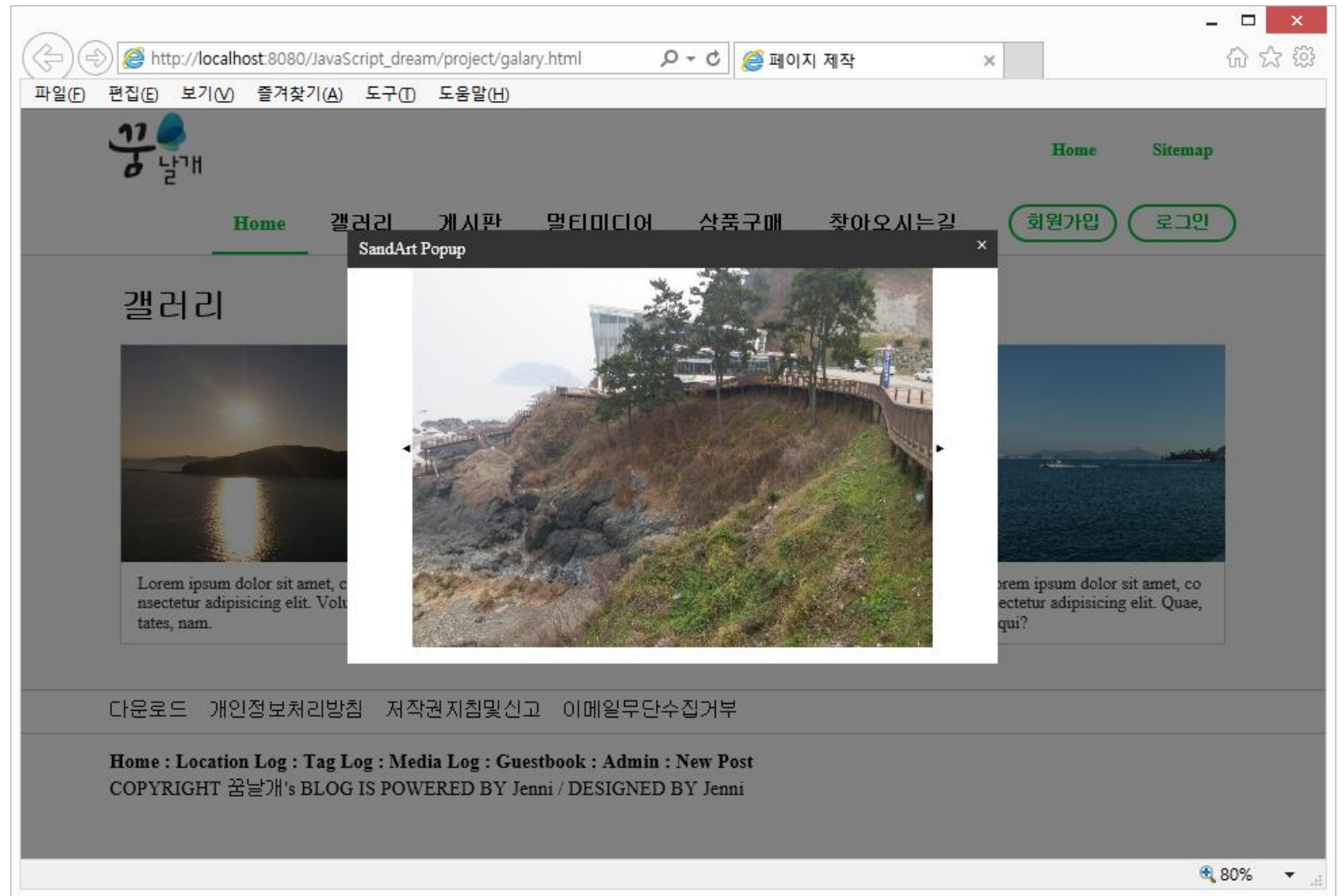
x[slideIndex-1].style.display= " block "

➔ 이미지를 보이게 함

## 프로젝트 구성



## [실행결과]



---

## ❖ 웹 표준 검증 사이트

---

- W3C Mark Up 검증 서비스: <http://validator.kldp.org/>
- W3C CSS Validation 서비스: <http://jigsaw.w3.org/css-validator/>

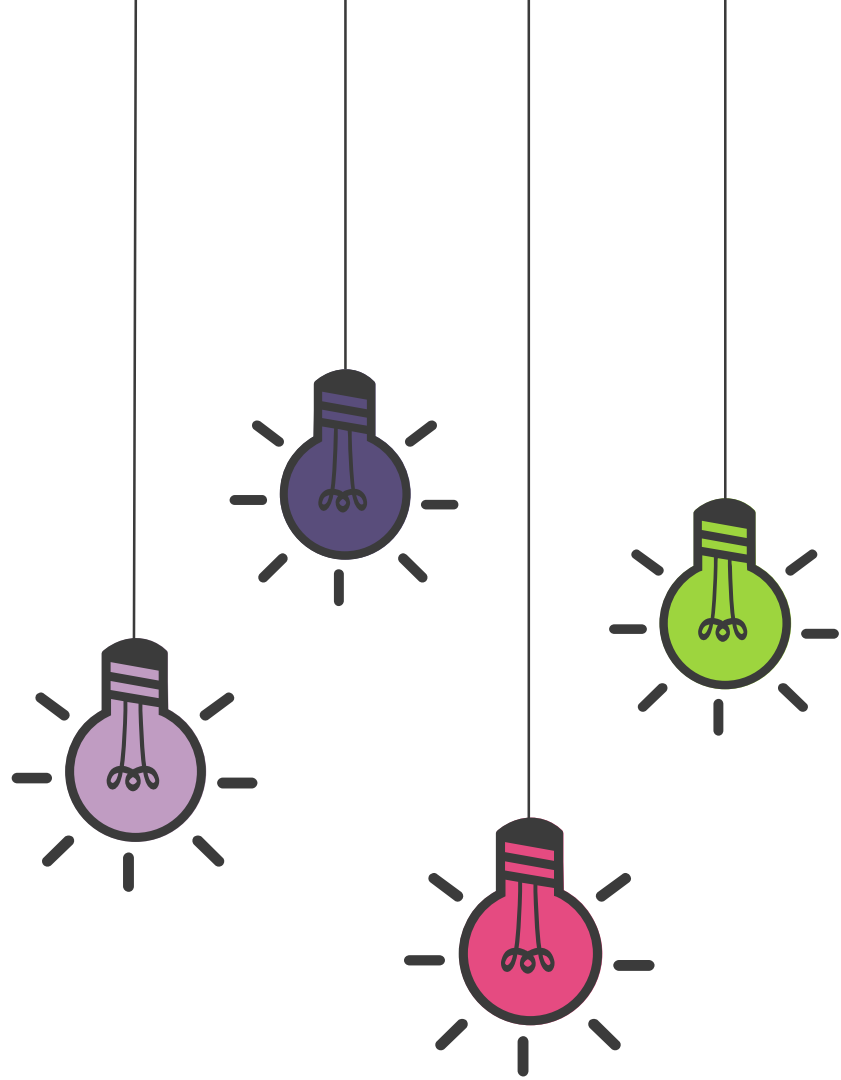


---

## ❖ 참고 사이트

---

- <http://www.w3schools.com>



# 감사합니다

THANK YOU FOR WATCHING