

JavaScript

J a v a S c r i p t



Index

CONTENTS



01 JavaScript의 시작



03 JavaScript와 표준 API 활용



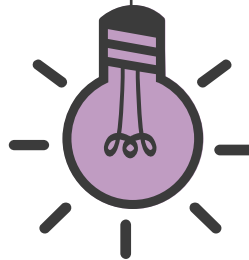
02 JavaScript 객체



04 실습3

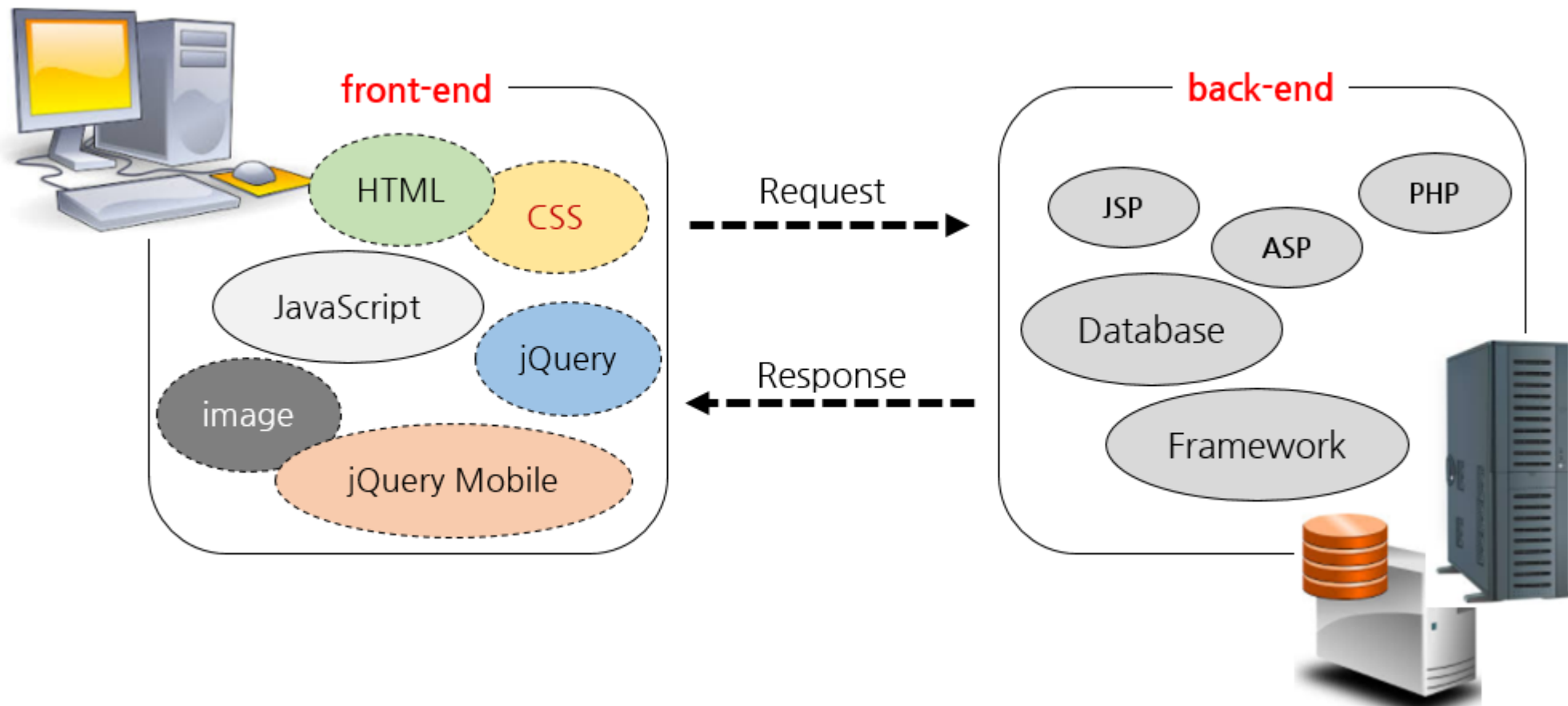
01

JavaScript의 시작



JavaScript 소개

- 정적인 웹페이지에 프로그램 언어를 사용해서 애니메이션 처리, 계산 처리, 데이터 처리 등을 가능하게 한다.



3.1.1 JavaScript의 역사와 특징

- 1995년에 넷스케이프에서 일하던 브랜든 아이크(Brendan Eich)에 의해 처음 개발되었다.
- 넷스케이프사가 사이트의 처리능력을 높이기 위해 개발한 스크립트 언어이다.
- ECMAScript 6은 모든 브라우저에서 부분적으로 지원하고 ECMAScript 7은 제대로 지원하지 않는다.

A. 코어(ECMAScript)

프로그램 구현 언어로 2008년부터 주요 브라우저는 모두 ECMA-262를 준수한다.

B. 문서 객체 모델(DOM)

DOM API를 이용하여 노드의 추가, 삭제, 수정이 가능하며 계속적으로 확장 지원하고 있다.

C. 브라우저 객체 모델(BOM)

브라우저에 접근하고 조작할 수 있도록 지원한다.

3.1.2 JavaScript의 구현 방법

- JavaScript의 코드는 일반적으로 <HEAD>와 <HEAD> 사이에서 작성한다.
- 그러나 <BODY></BODY>의 어떤 곳에도 작성할 수 있다.
- type="text/javascript"는 JavaScript의 유형이 텍스트이고 사용 언어는 JavaScript라는 뜻이다.

❖ 자바스크립트 기본 구성

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

```
    실행문;
```

```
</script>
```

❖ JavaScript를 사용하는 3가지 방법

1. HTML문서 안의 <SCRIPT>과 </SCRIPT> 안에 JavaScript 코드를 추가하는 방법이다.

<head> 태그 안에서 작성하는 방법

HTML

```
<html>
<head>
  <title>Insert title here</title>
  <script>
    document.write("Hello World");
  </script>
</head>
<body>
</body>
</html>
```

2. <BODY></BODY>안에서 작성할 수 있다.

<body> 태그 안에서 작성하는 방법

HTML

```
<html>
<head>
  <title>Insert title here</title>
</head>
<body>
  <script>
    document.write("Hello World");
  </script>
</body>
</html>
```


3. 확장자를 js로 하는 외부 파일을 불러서 사용할 수 있다.

외부 파일로 작성하는 방법

HTML

```
<html>
<head>
  <title>Insert title here</title>
  <script src="script.js"></script>
</head>
<body>
</body>
</html>
```

script.js

```
document.write("Hello World");
```

3.1.3 JavaScript의 주요 문법

- a. JavaScript는 대소문자를 구분한다.
- b. 한 라인을 끝내려면 원칙적으로 세미콜론(;)으로 닫아준다.
- c. Enter로 라인을 구분하기 때문에 생략이 가능하다.
- d. 한번에 2개 이상의 문장이 기술되면 세미콜론을 생략할 수 없다.

사용 예1)

```
var str1="very"  
var str2="good"
```

사용 예2)

```
var str1="very"; var str2="good";
```

❖ 변수 선언과 데이터 타입

- 변수(variable)란 데이터를 저장하는 공간이다.
- 변수에 저장되는 값을 데이터 값 또는 상수(constant) 값이라고 한다.

[형식]

var 변수명; 또는 var 변수명1, 변수명2.....; 또는 var 변수명=데이터 값;

[변수 선언 방법]

- 변수명은 대, 소문자를 구별하고 영문, \$, _ (밑줄 문자), 숫자를 포함할 수 있다.
- 변수명의 맨 앞에는 숫자가 올 수 없다.
- 영문, _(밑줄 문자), \$만 가능하다.

	변수 선언	설명
맞음	var num=10; var num10=100; var \$num=100; var _num=100;	영문자와 숫자의 조합은 가능하다. 변수명 맨 앞에 \$나 _(밑줄문자)로 시작할 수 있다.
틀림	var 2num=100; var num%=100;	변수명 맨 앞에는 숫자가 올 수 없다. %(특수문자)는 포함할 수 없다.

[변수에 저장되는 데이터 타입]

데이터 타입	사용 방법
문자형(string)	var str="hello"; 또는 var str='hello';
숫자형(number)	var num=10; 또는 var num=50.4;
논리형(boolean)	var check=true; 또는 var check=false;
널형(null)	var str=null; 또는 var str=" ";

❖ 연산자

산술 연산자

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

[ex03_01.html] 산술 연산자

두수의 합 : 13
 두수의 차 : 7
 두수의 곱 : 30
 두수의 몫 : 3.3333333333333335

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)

[ex03_02.html] 관계 연산자

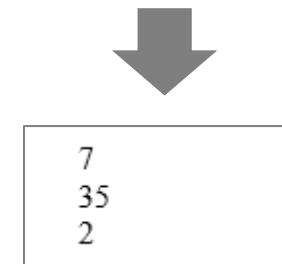
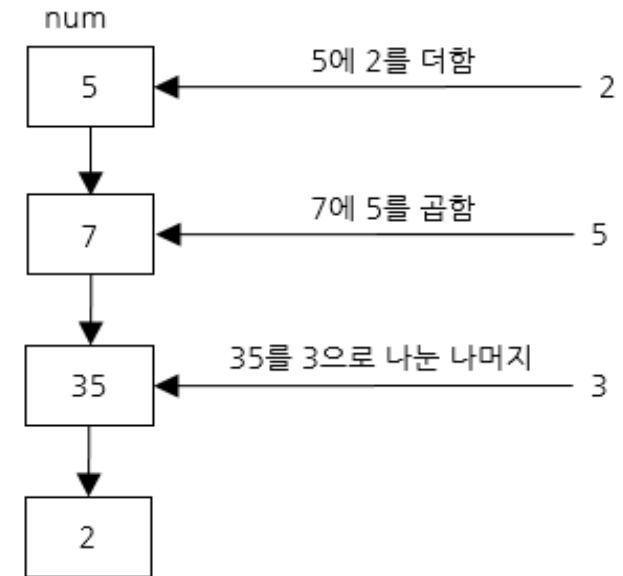
true
 true
 false

❖ 연산자

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.

[ex03_03.html] 산술 연산자



❖ 연산자

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)	A 가 true이면 false로, false이면 true 로 변경

[ex03_04.html] 논리 연산자

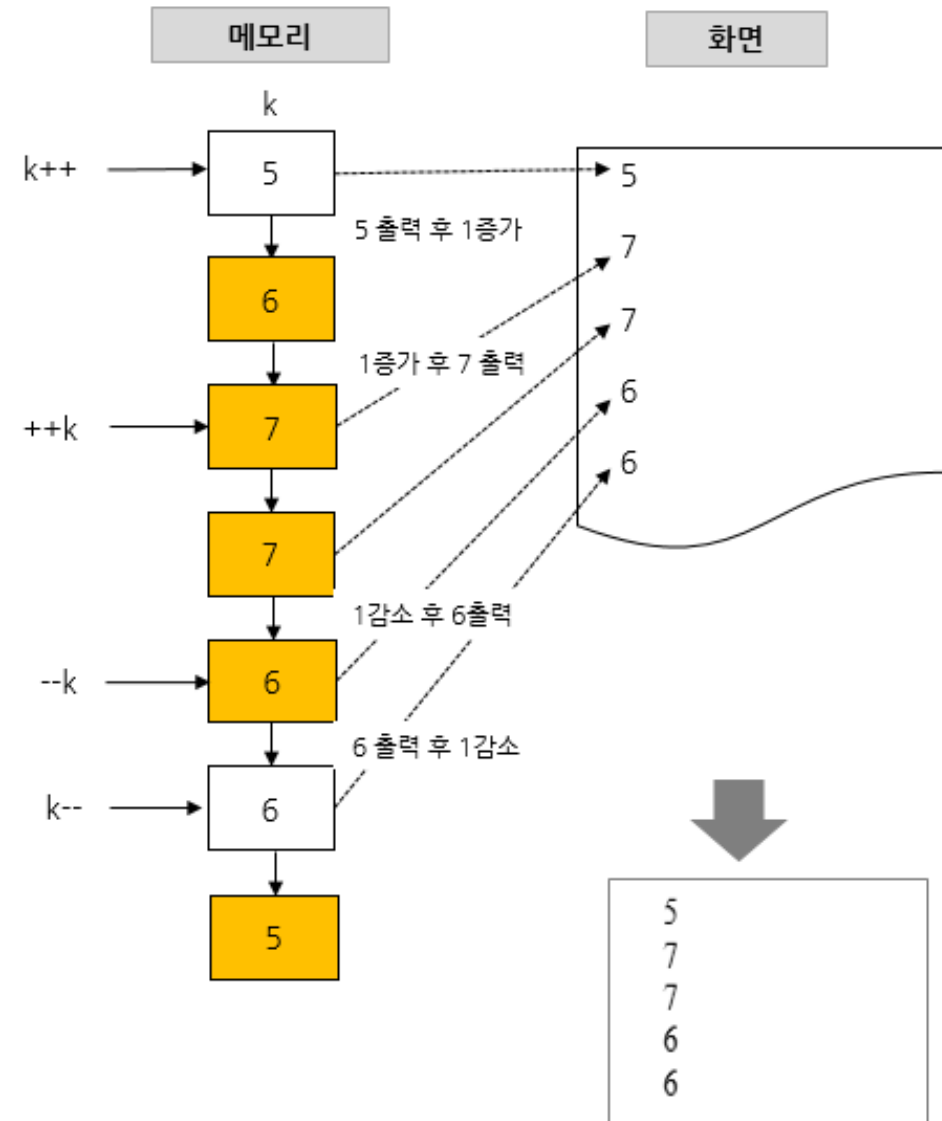
```
true  
false  
false
```


❖ 연산자

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

[ex03_05.html] 증감 연산자



❖ 연산자

문자 결합 연산자

연산자	사용 예
+	"문자" + "문자"
	"문자" + 숫자

[ex03_06.html] 문자 결합 연산자

```
korea fighting
abc123
```

삼항 연산자

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

[ex03_07.html] 삼항 연산자로 합격/불합격 판단하기

prompt("입력 데이터", "초기값")

localhost:8080 내용:

점수를 입력하십시오

확인 취소



alert("메시지")

localhost:8080 내용:

합격

확인

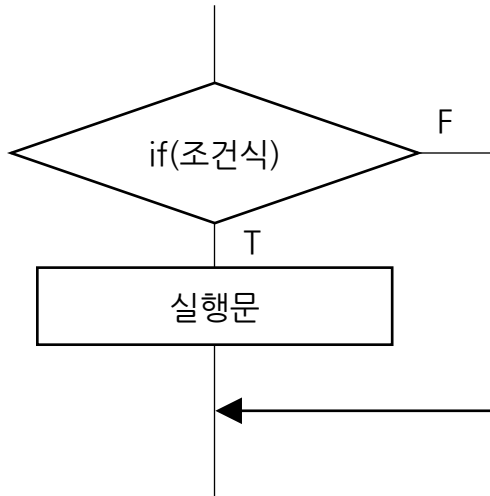
❖ 제어문

조건문

- 조건문은 조건식이 참(true)일 경우에 실행문을 수행하는 제어문이다.
- if와 if~else, if~else if 등 3가지 타입이 있다.

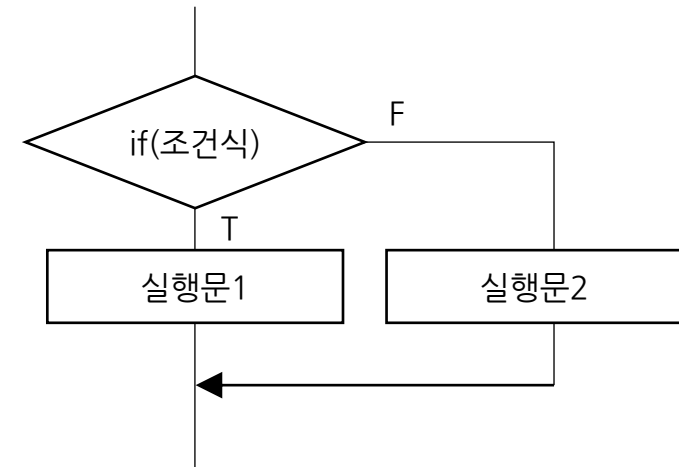
(1) if

```
if(조건식) {  
    실행문;  
}
```



(2) if~else

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



[ex03_08.html] if문을 이용한 7의 배수 구하기

35은(는) 7의 배수 입니다

[ex03_09.html] if~else문을 이용한 합격/불합격 판단하기

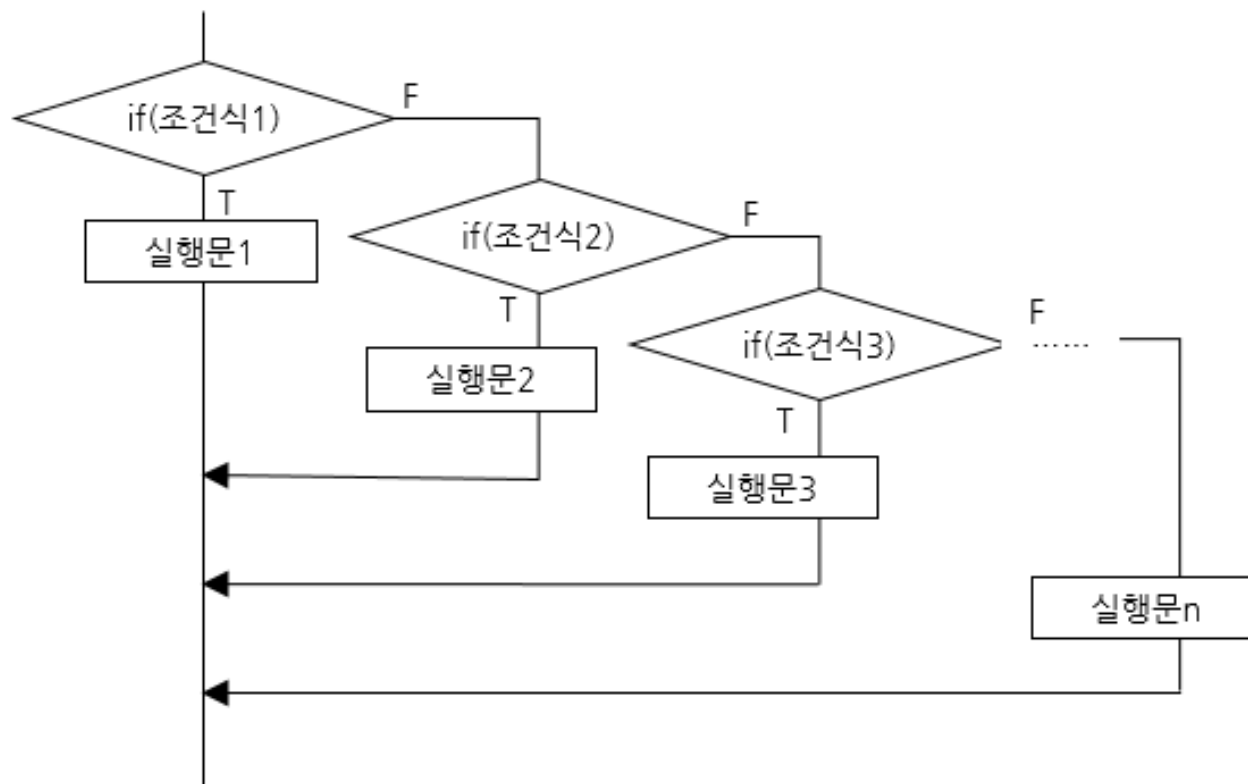
localhost:8080 내용:

점수를 입력하십시오

합격
당신의 점수는 75점 입니다

(3) if~else if

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



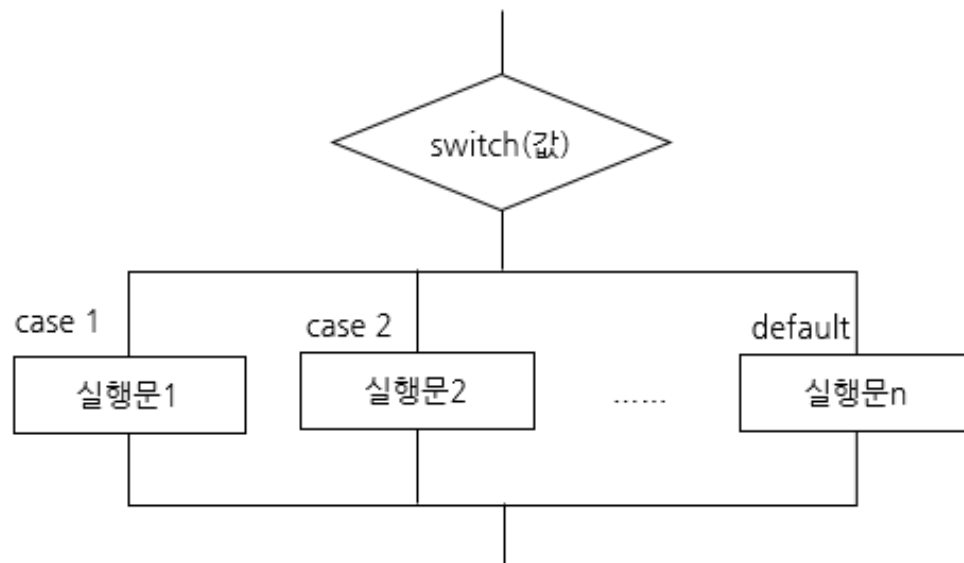
[ex03_10.html] if~else if을 이용해서 학점 구하기

나의 점수는 95점이고 학점은 A 입니다

선택문

- 선택문은 저장된 데이터의 값이 일치하는 경우에 해당 실행문을 수행하는 제어문이다.

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



[ex03_11.html] switch 문을 이용한 나이 등급 구하기

localhost:8080 내용:
나이를 입력하시오



localhost:8080 내용:
청년입니다

[ex03_12.html] switch를 이용한 부서 찾기

localhost:8080 내용:
부서코드를 입력하시오(dev1,dev2,eng1,eng2)



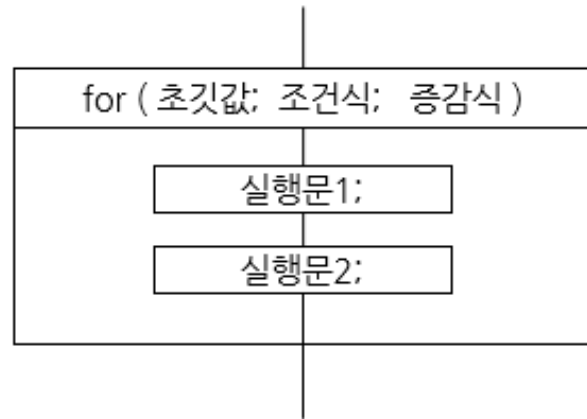
당신은 기술부에 근무하며, 부서 코드는 eng1입니다

반복문

- 반복문은 조건식이 참(true)인 경우에 실행문을 반복적으로 수행하며 for, while이 있다.
- for문은 정해진 구간만큼 반복할 때 사용한다.

for

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



[ex03_13.html] for문을 이용한 1-10까지 출력하기와 h1 태그 응용하기



[ex03_14.html] 1-100까지 전체 합, 홀수 합, 짝수 합 구하기

1-100까지의 전체 합: 5050
1-100까지의 홀수 합: 2500
1-100까지의 짝수 합: 2550

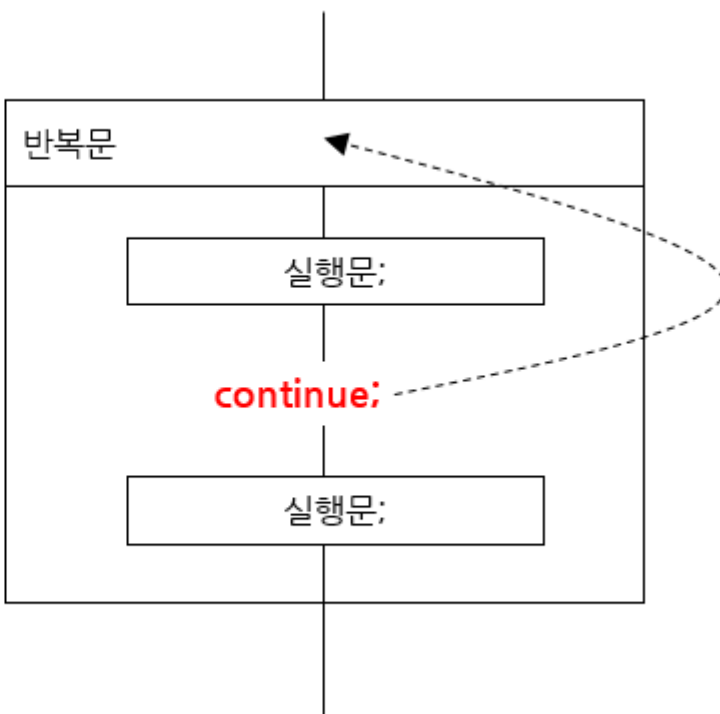
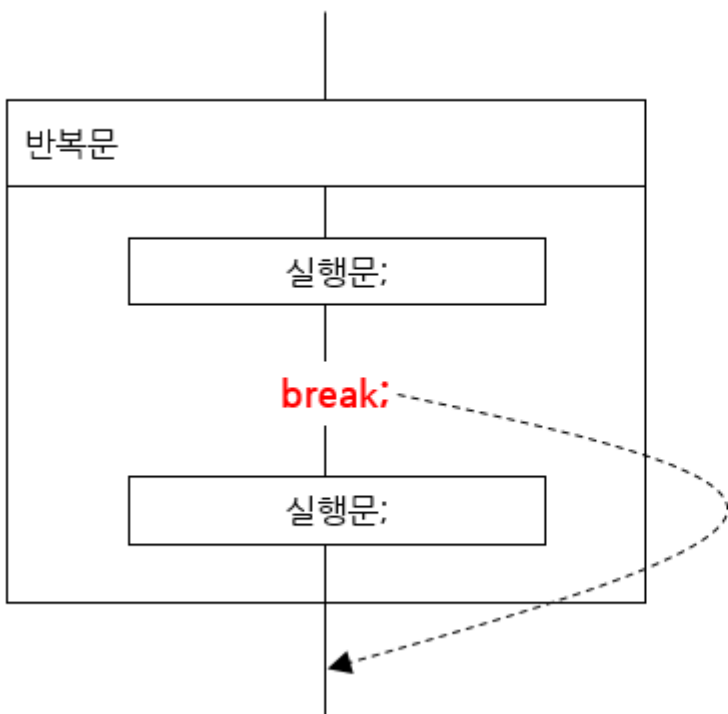
블록 탈출

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

[ex03_15.html] break와 continue

1 2 3 4

1 2 3 4 6 7 8 9 10

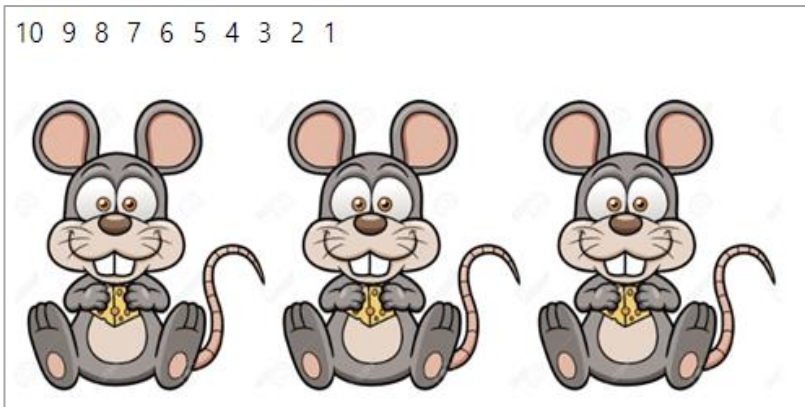


while

```
while(조건식){  
    실행문;  
}
```

- while문은 조건식이 참일 경우에 반복한다.

[ex03_16.html] while문으로 10-1까지 출력하기와 같은 이미지 3번 출력하기

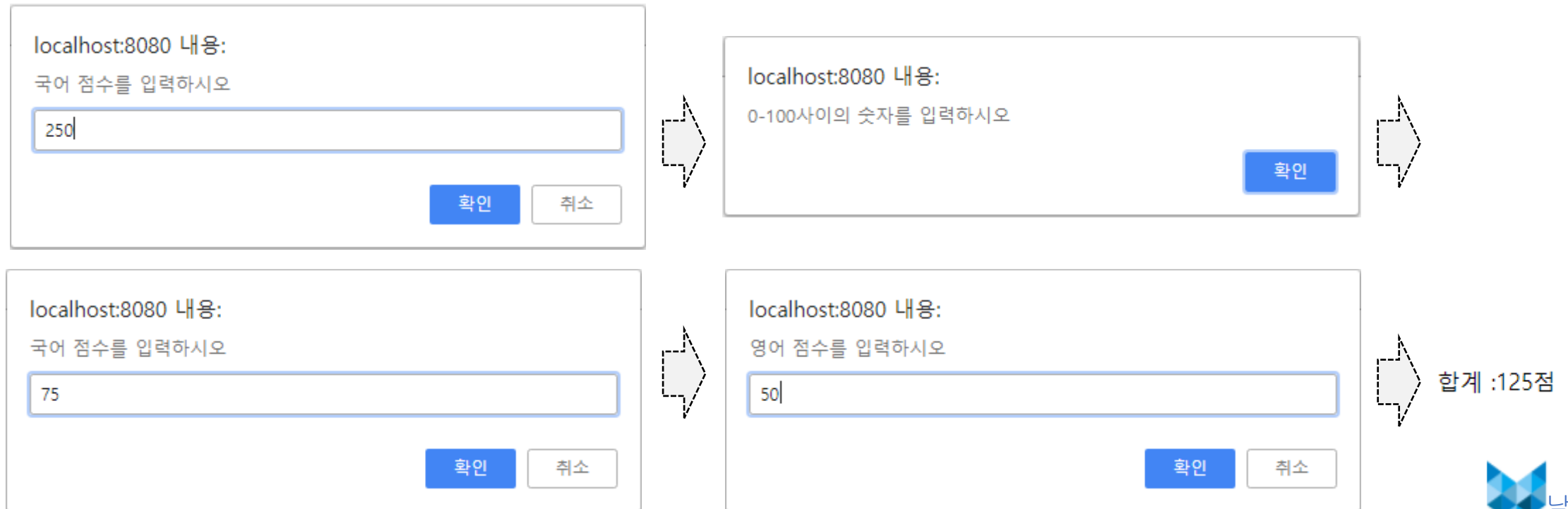


do~while

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

- do~while문은 먼저 do 안으로 들어가서 실행문을 수행한 후 조건을 비교한다.

[ex03_17.html] do~while를 이용한 과목별 점수를 입력받아 총점 구하기



3.1.4 함수(Function)

- 함수는 코드의 양이 많은 큰 프로그램을 관련 코드끼리 묶어서 분리하는 구조적 프로그램이다.
- 함수는 주기적으로 사용하는 코드를 분리해 두었다가 필요할 때 호출하여 사용하면 매우 편리하다.

❖ 선언적 함수 • 선언적 함수는 function이라는 키워드를 사용해서 함수를 선언한다.

함수의 정의와 호출

[형식 1]

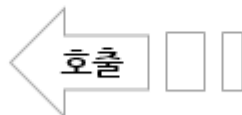
```
function 함수명(){  
    실행문;  
}
```



매개 변수와 리턴값

[형식 2]

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



```
var 변수 = 함수명(매개 변수, 매개 변수, ...);
```

return 사용 예

맞는 예	틀린 예
return "good";	return a, b, c
return num;	
return 3+5;	

[ex03_18.html] 선언적 함수를 이용한 처리 방식

```
**도형의 넓이**  
원의 넓이 = 314  
사각형의 넓이 = 20
```

❖ 익명 함수

- function 형태는 함수이지만 이름이 없기 때문에 익명 함수라고 부른다.

함수의 정의와 호출

[형식 1]

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



매개 변수와 리턴값

[형식 2]

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

호출

```
var 변수 = 함수명(매개 변수, 매개 변수, ...);
```

[ex03_19.html] 익명 함수를 이용한 처리 방식

사다리꼴의 넓이=13.5

❖ 내장 함수

- 내장 함수는 JavaScript가 자체적으로 가지고 있는 함수이다.
- 네트워크 통신용 함수와 숫자와 관련된 함수가 있다.

내장 함수	설명
<code>encodeURIComponent()</code>	파라미터를 전달하는 URI 전체를 인코딩 할 때 사용하며 특수문자(:, ;, /, =, ?, & 등)를 제외한 문자만 인코딩합니다. 주로 인터넷 주소를 인코딩할 때 사용한다 .
<code>decodeURI()</code>	<code>encodeURIComponent()</code> 로 인코딩된 데이터를 다시 되돌린다.

[ex03_20.html] 인코딩과 디코딩

연습1

localhost:8080 내용:
%EA%B0%80%EB%82%98%EB%8B%A4
가나다

확인

연습2

localhost:8080 내용:
1015

확인

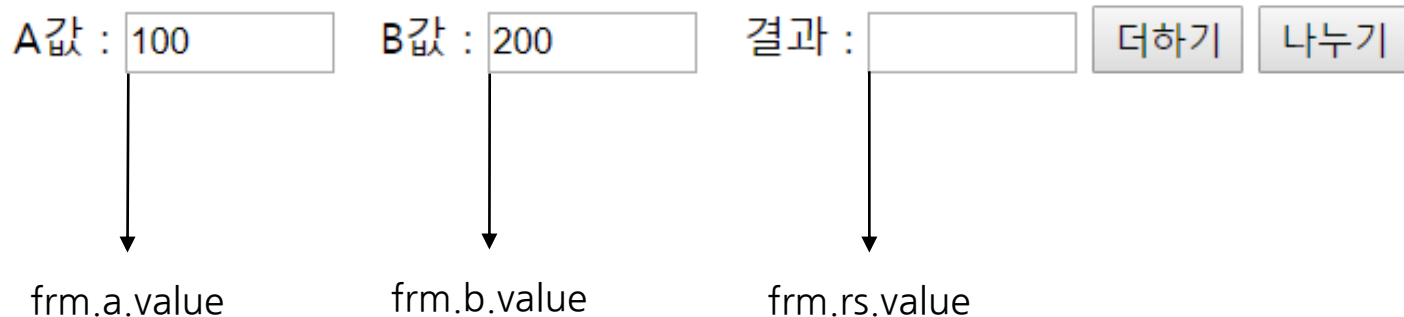
연습3

localhost:8080 내용:
7
2

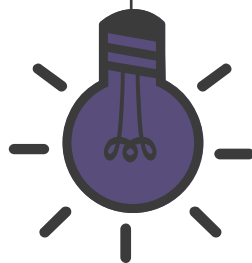
확인

내장 함수	설명
isNaN()	NaN은 "Not a Number"의 약자이다. 숫자가 아닌 문자가 포함되면 true를 반환한다.
isFinite()	값이 유한수인지를 판단을 한다.
parseInt()	문자를 정수형으로 변환한다.
parseFloat()	문자를 실수형으로 변환한다.
eval()	문자로 된 수식을 JavaScript의 수식으로 인식하여 실행시키고 결과를 변환한다.
Number()	문자를 숫자형으로 변환한다.
String()	숫자를 문자형으로 변환한다.

[ex03_21.html] 기타 내장 함수

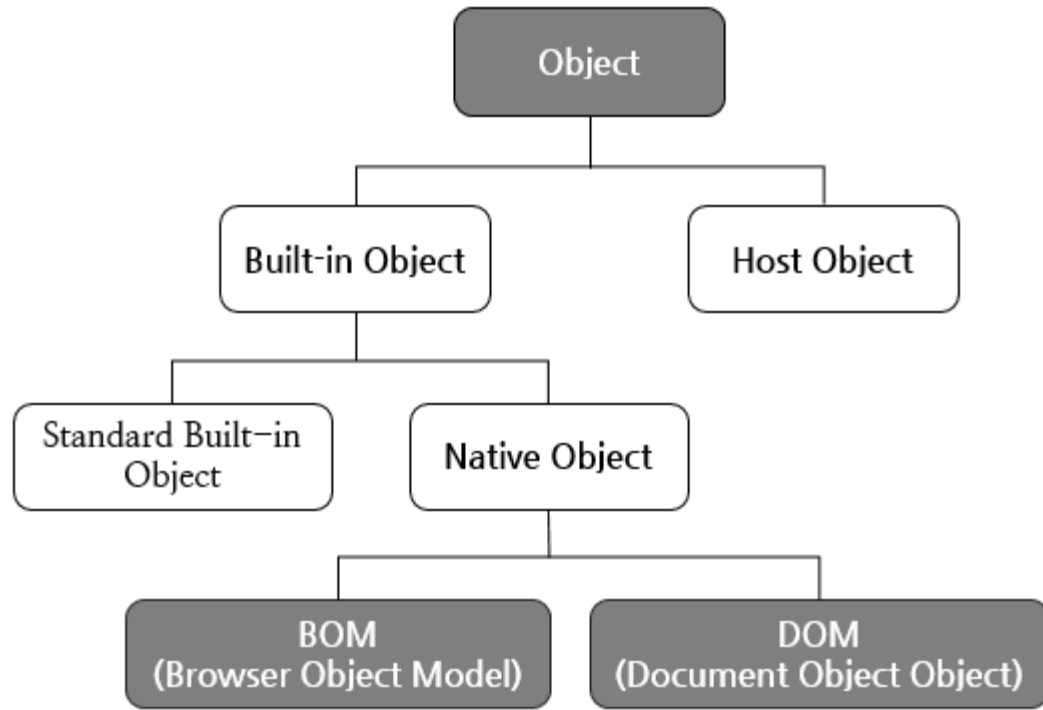


02 JavaScript 객체



객체 (Object)

- 객체란, 데이터와 데이터를 처리하는 기술의 조합을 의미한다.
- JavaScript는 객체 기반의 스크립트 언어이다.
- 기본 자료형을 제외하고는 배열, 함수 등 모두가 객체이다.



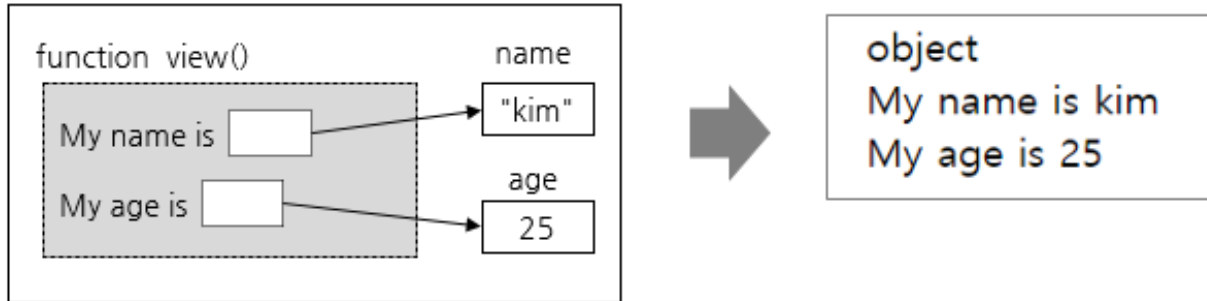
3.2.1 객체 생성

- 객체(Object)는 데이터를 의미하는 프로퍼티(또는 속성)와 데이터 처리 기술(즉, 동작인 메서드(Method))의 조합이다.

❖ 리터럴을 이용한 객체 생성 • 객체 리터럴 방식은 선언과 동시에 인스턴스를 생성한다.

[ex03_22.html] 자기 자신을 가리키는 키워드 this 사용하기

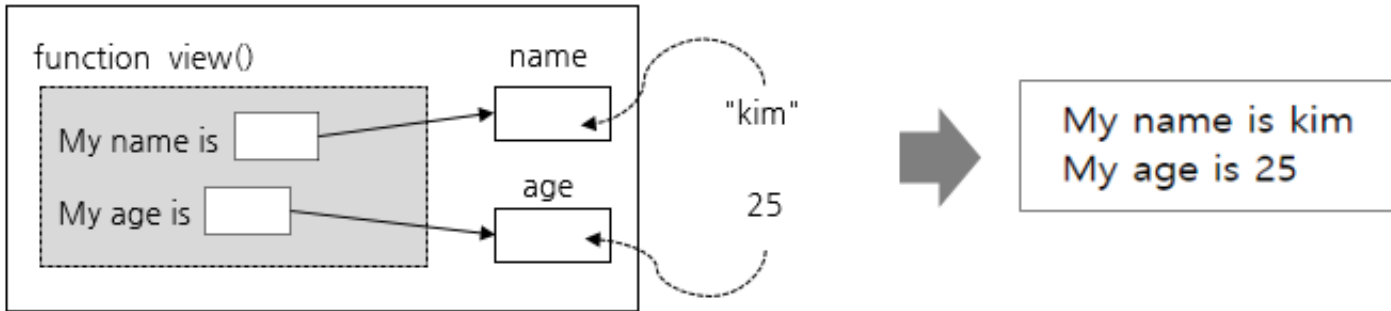
User 객체



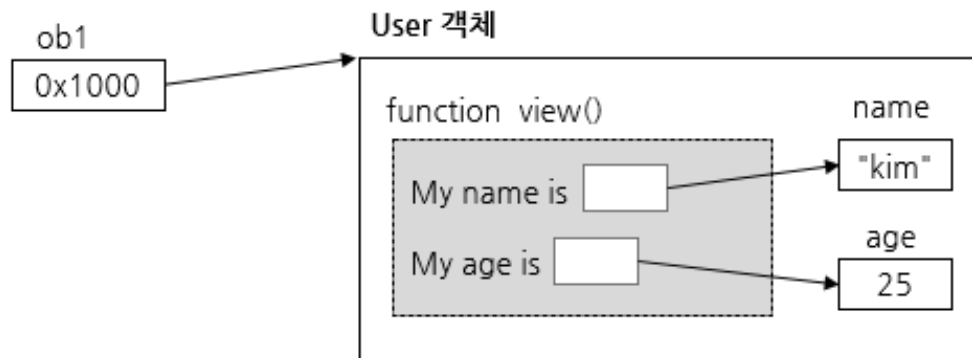
❖ **생성자 함수를 이용한 객체 생성** • Object로 빈 객체를 생성한 후 프로퍼티와 메서드를 추가한다.

[ex03_23.html] 생성자 함수를 이용한 객체 생성

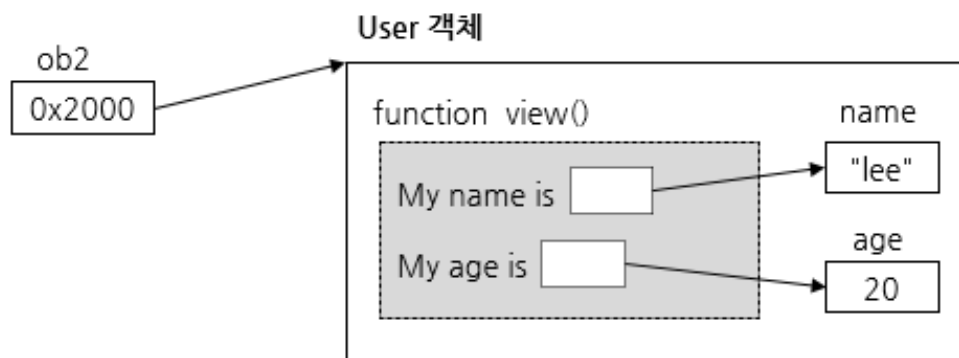
User 객체



[ex03_24.html] 생성자를 이용한 객체 생성



나의 이름은 kim이고 나이는 25세 입니다
나의 이름은 lee이고 나이는 20세 입니다



3.2.2 표준 객체

- 표준 객체(Standard Built-in Object)란 JavaScript에서 자주 사용하는 Number, String, Array, Date, Math 등의 내장 객체를 말한다.

❖ **Number객체** • Number 객체는 숫자를 다루는 객체이다.

[형식]

var 변수명 = new Number(숫자); 또는 var 변수명=숫자;

[ex03_25.html] 숫자를 다루는 Number 객체

a와 b를 더하면: 1515
15의 2진수 표현: 1111
a와 b의 타입: number object
a와 b의 값 비교: true
a와 b의 타입 비교: false

❖ String객체

- String 객체는 문자열을 다루는 객체이다.

[형식]

```
var 변수명 = new String("문자열"); 또는 var 변수명="문자열";
```

String 속성과 문자 관련 메서드

	메서드	설명
속성	length	문자열의 길이를 구한다.
	big()/small()	문자를 한 단계 크게 / 작게 설정한다.
	blink()	문자를 깜박이게 설정한다.
	bold()	문자를 굵게 설정한다.
	sub()/sup()	아래 첨자 / 위 첨자로 설정한다.
	fontSize(크기)	문자의 크기를 설정한다. (범위: 1-7)
	fontcolor(색상)	문자의 색상을 설정한다.
	toLowerCase()	문자를 소문자로 변경한다.
	toUpperCase()	문자를 대문자로 변경한다.
	anchor('#위치표시문자')	과 같은 효과이다.
문자	link('링크할 위치')	와 같은 효과이다.
	italics()/strike()	이탤릭체 / 취소선으로 설정한다.

String 문자열 관련 메서드

메서드	설명
<code>charAt(n)</code>	인덱스 값에 해당 n의 위치를 반환한다.
<code>indexOf("문자열")</code>	처음부터 시작해서 최초로 만나는 "문자열"의 위치를 반환한다.
<code>lastIndexOf("문자열")</code>	끝에서부터 시작해서 최초로 만나는 "문자열"의 위치를 반환한다.
<code>substring(n1, n2)</code>	n1에서 (n2-1) 사이의 문자열을 반환한다.
<code>slice(s, e)</code>	s부터 (e-1)의 문자열을 분리한다.
<code>substr(s, 길이)</code>	s부터 길이만큼 문자열을 추출한다.
<code>concat("문자열")</code>	두 개의 문자열을 연결한다.
<code>split("문자", 개수)</code>	문자를 기준으로 개수만큼 분리한다.
<code>replace(s1, s2)</code>	문자열 중 s1을 s2로 치환한다.

[ex03_26.html] 문자열을 다루는String 객체

```
javascript test
15
javascript test
javascript test
남가람 박스

j
av
script test
javascript sample
```

❖ Array객체

- 배열은 하나의 객체에 여러 개의 데이터 값을 저장할 때 사용한다.

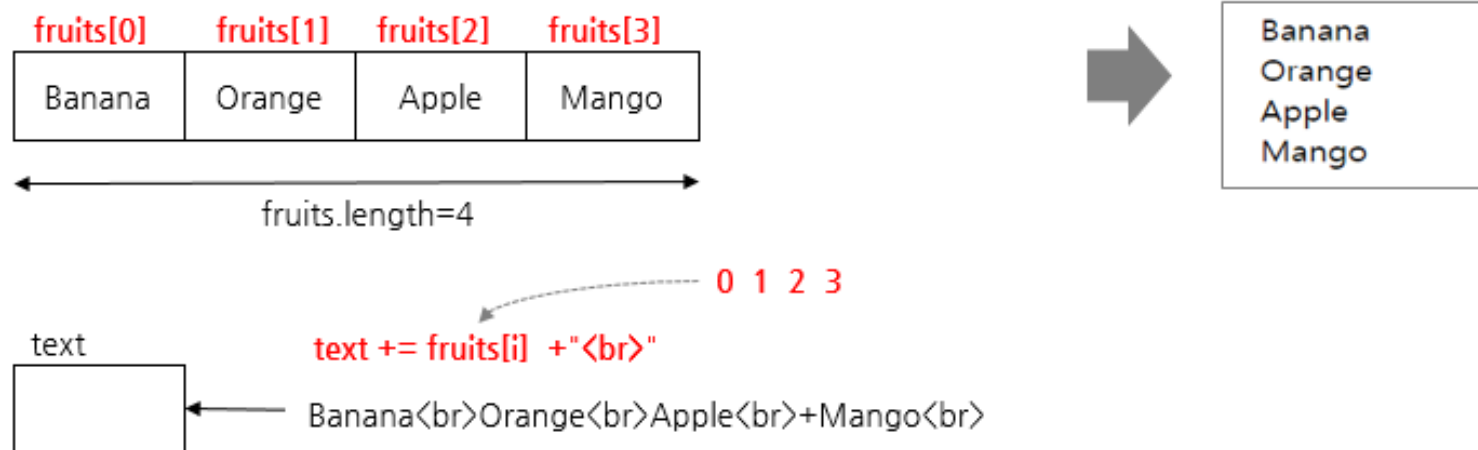
[형식]

```
var 변수명 = [값1, 값2, 값3.....값n];
```

Array 관련 메서드

메서드	설명
sort()	배열 값들을 오름차순으로 정렬한다.
reverse()	배열 값들을 역순으로 바꾼다.
concat(array)	두 개의 배열을 합하여 하나의 배열로 만든다.
join([str])	배열에 들어 있는 값을 모두 붙여서 하나의 문자열로 만든다.
push	배열의 마지막에 새로운 원소를 추가한다.
pop()	배열의 마지막 원소를 추출한다. (가장 최근에 push()한 요소)

[ex03_27.html] Array 객체에 데이터 저장하고 출력하기



- 배열 객체를 선언하는 방식은 new 키워드를 사용해서 객체 선언과 동시에 값을 인자로 추가한다.

[형식]

```
var 변수명 = new Array (값1, 값2, 값3.....값n);
```

[ex03_28.html] Array 객체의 오름차순/내림차순 정렬 메서드

```
Apple,Banana,Mango,Orange  
Orange,Mango,Banana,Apple
```

❖ Date객체

- Date 객체는 날짜와 시간에 대한 정보를 제공한다.

[형식]

```
var 변수명 = new Date(); 또는 var 변수=new Date(년, 월, 일, 시, 분, 초);
```

날짜와 시간 관련 메서드

메서드	설명
setFullYear() / getFullYear()	연도만 설정하거나 반환한다.
setMonth() / getMonth()	월만 설정하거나 반환한다.
setDate() / getDate()	일(월 기준)을 설정하거나 반환한다.
setDay() / getDay()	일(주 기준)을 설정하거나 반환한다.
setHour() / getHour()	시간을 설정하거나 반환한다.
setMinutes() / getMinutes()	분을 설정하거나 반환한다.
setSeconds() / getSeconds()	초를 설정하거나 반환한다.
setMilliseconds() / getMilliseconds()	밀리 초를 설정하거나 반환한다.
toString()	날짜를 문자형식으로 반환한다.

날짜와 시간의 범위

날짜	범위	시간	범위
연도	1900년(00) ~ 1999년(99)	시	0시(0) ~ 23시(23)
월	1월(0) ~ 12월(11)	분	0분(0) ~ 59분(59)
일	1일(1) ~ 31일(31)	초	0초(0) ~ 59초(59)

[ex03_29.html] Date와 Array 객체를 이용한 날짜/시간 구하기

오늘은 2018년11월17일 토요일 입니다.
현재시간은 0:4:41입니다.

❖ Math객체

- Math 객체는 수학에서 자주 사용하는 메서드를 미리 정의해 놓은 객체이다.

수학 관련 메서드

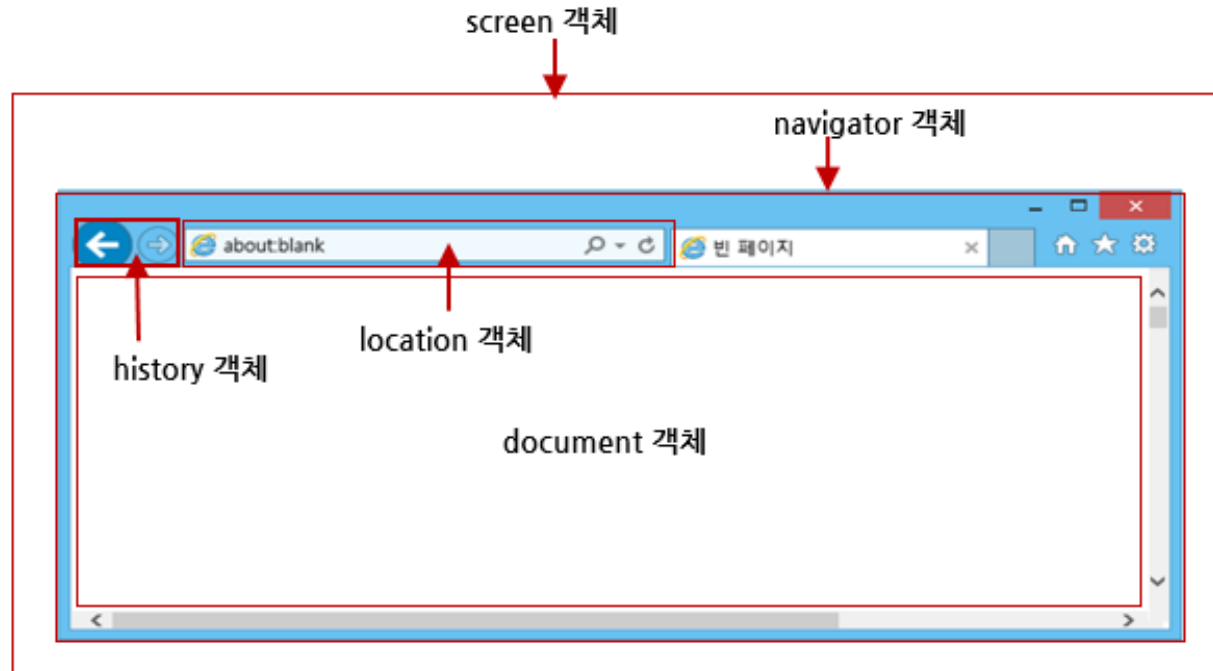
메서드	설명
max(n1, n2, ...)	가장 큰 값을 반환한다.
min(n1, n2, ...)	가장 작은 값을 반환한다.
round(n)	반올림 값을 반환한다.
ceil(n)	올림 값을 반환한다.
floor(n)	내림 값을 반환한다.
abs(n)	절댓값을 반환한다.
random(n)	0~1사이의 임의의 수를 반환한다.
pow()	숫자의 거듭제곱을 계산하여 반환한다.

[ex03_30.html] Math 객체의 수학 메서드

```
64
0.6716154743955758
4
5
5
4.4
8
```

3.2.3 브라우저 객체 모델 (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

❖ window객체

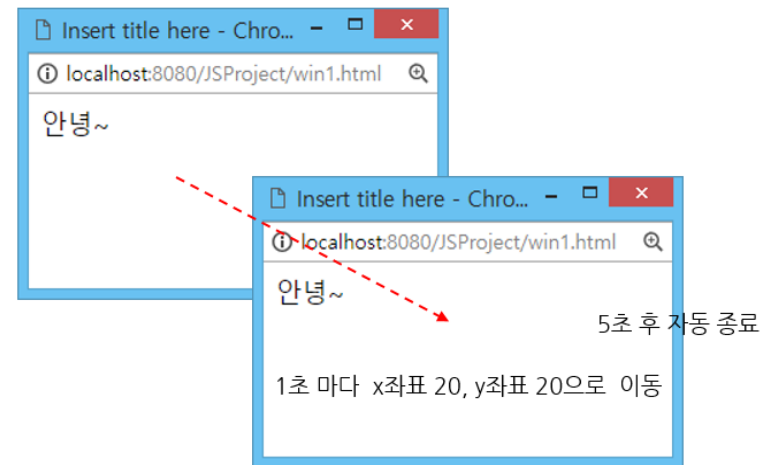
- window 객체는 브라우저 객체 모델의 최상위 객체이다.

window 객체 메서드

메서드	설명
open()	새로운 윈도우 객체를 생성한다.
alert()	경고창을 띄운다.
prompt()	입력창을 띄운다.
confirm()	확인/취소 창을 띄운다.
setInterval()	일정시간 마다 반복 실행한다.
setTimeout()	시간을 지연시킨 후 실행문을 수행한다.

메서드	설명
moveBy(x, y)	윈도우를 현재 위치에서 상대적 위치로 이동한다
moveTo(x, y)	윈도우를 현재 위치와 상관없이 절대적 위치로 이동한다.
resizeBy(x, y)	윈도우의 화면을 현재 크기에서 (x, y) 만큼 증가시킨다. (상대적)
resizeTo(x, y)	윈도우의 화면을 (x, y) 크기로 한다. (절대적)
scrollBy(x, y)	윈도우 스크롤의 위치를 상대 위치로 이동한다.
scrollTo(x, y)	윈도우 스크롤의 위치를 절대 위치로 이동한다.
focus()	윈도우에 초점을 맞춘다.
blur()	윈도우에 초점을 제거한다.
close()	윈도우를 닫는다.

[ex03_31.html / win1.html] window 객체를 이용하여 창 이동하기



❖ location객체

- location 객체는 URL과 관련된 인터넷의 주소와 관련된 속성이다.

location 객체 속성

속성	사용 방법	설명
href	localhost:80/index.html	문서의 url 주소를 반환한다.
host	localhost:80	호스트 이름과 포트 번호를 반환한다.
hostname	localhost	호스트 이름을 반환한다.
port	80	포트 번호를 반환한다.
pathname	/docs/index.html	파일 경로를 반환한다.
hash	http://localhost:80/#test	앵커의 이름이다.
search	?param=100	요청 값(쿼리)을 반환한다.
protocol	http://	프로토콜을 반환합니다.

location 객체 메서드

메서드	설명
reload()	화면을 "새로 고침" 한다.
replace(link)	현재 페이지를 새로운 페이지로 바꾼다.

[ex03_32.html] location.reload()로 1초마다 현재 시간 출력하기

18:37:23

[ex03_33.html] location 객체를 이용해서 지정 파일에 데이터 전송하기 / [win2.html]

사용자명 : 전화번호 :



주소: http://localhost:8080/JSProject/win2.html?kim&010-123-4567
 프로토콜: http:
 호스트: localhost:8080
 파일 경로명: /JSProject/win2.html
 호스트: ?kim&010-123-4567

❖ **navigator 객체** • navigator 객체는 웹 브라우저에 대한 정보를 제공하는 객체이다.

navigator 객체 속성

메서드	설명
appCodeName	브라우저의 코드명을 반환한다.
appName	브라우저의 이름을 반환한다.
language	브라우저의 사용 언어를 반환한다.
product	브라우저의 사용 엔진이름을 반환한다.
platform	사용중인 운영체제의 시스템 환경 정보를 반환한다.

[ex03_34.html] 사용 중인 브라우저의 정보 확인

브라우저 명: Netscape
코드명: Mozilla
운영체제: Win32
사용 언어: ko-KR
사용 엔진: Gecko

❖ history객체 • 인터넷 방문 기록에 대한 정보를 제공하는 객체이다.

history 객체의 메서드

메서드	설명
go(숫자)	숫자만큼 다음 또는 이전 페이지로 이동한다.
back()	이전 페이지로 이동한다.
forward()	다음 페이지로 이동한다.
length	방문 기록에 저장된 목록의 개수를 반환한다.

[ex03_35.html] 방문 페이지를 history 객체에 저장하기
[test1.html] / [test2.html]



❖ screen객체

- screen 객체는 현재 화면의 해상도나 색상, 화면의 크기 정보 등의 속성을 제공하는 객체이다.

screen 객체 속성

속성	설명
width	전체 화면의 너비이다.
height	전체 화면의 높이이다.
availWidth	전체화면에서 사용 가능한 화면의 너비이다. (작업 표시줄 제외)
availHeight	전체화면에서 사용 가능한 화면의 높이이다. (작업 표시줄 제외)
colorDepth	이미지를 표시하기 위한 색상 팔레트 비트의 깊이 (픽셀 당 비트 수) 이다.
pixelDepth	화면의 색상 해상도(픽셀당 비트 수)이다.

[ex03_36.html] screen 객체로 화면의 속성

화면의 실제 넓이:1600
화면의 실제 높이:860
화면의 넓이:1600
화면의 높이:900
사용 가능한 색상수:24
한 픽셀당 비트 수:24

3.2.4 문서 객체 모델 (DOM)

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

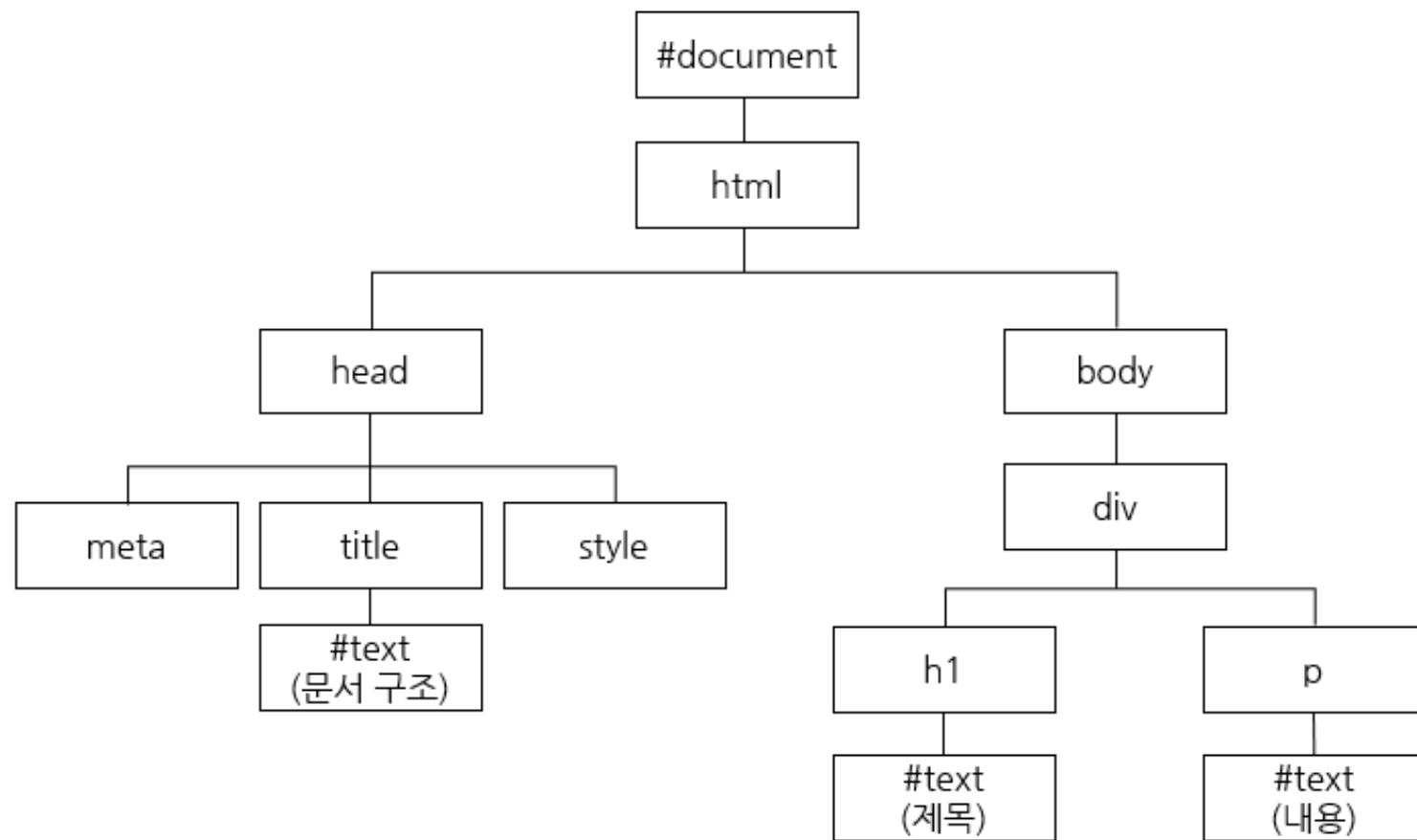
HTML 문서의 DOM 트리 구조

```
<html>
  <head>
    <title>문서 구조</title>
  </head>
  <body>
    <h1>제목</h1>
  </body>
</html>
```

XML 문서의 DOM 트리 구조

```
<books>
  <book>
    <title>알고리즘</title>
    <author>이상진</author>
  </book>
  <book>
    <title>스타트 스프링 부트</title>
    <author>구멍가게 코딩단</author>
  </book>
  <book>
    <title>웹 개발 마스터 북</title>
    <author>김동섭</author>
  </book>
</books>
```

HTML의 DOM 구조



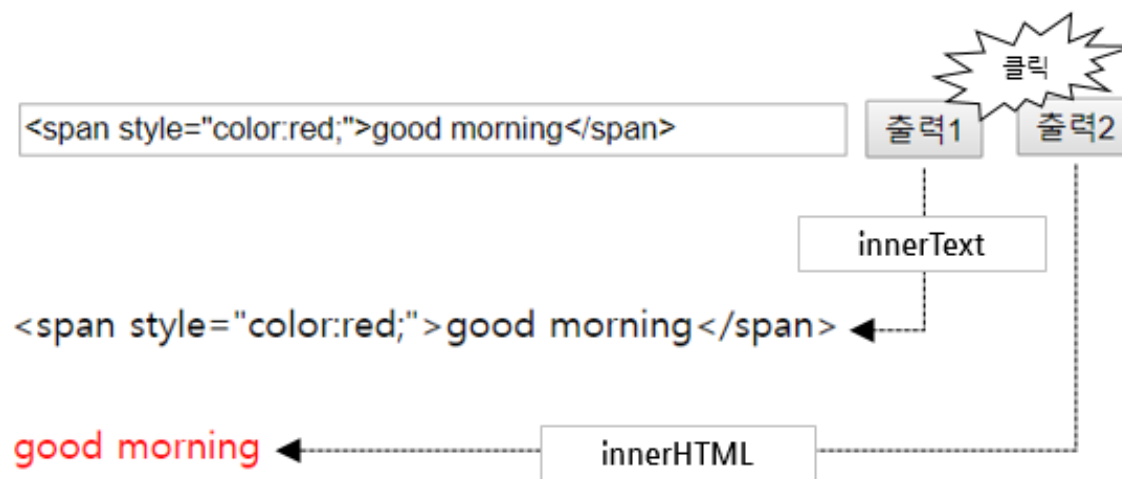
원거리 선택자

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.

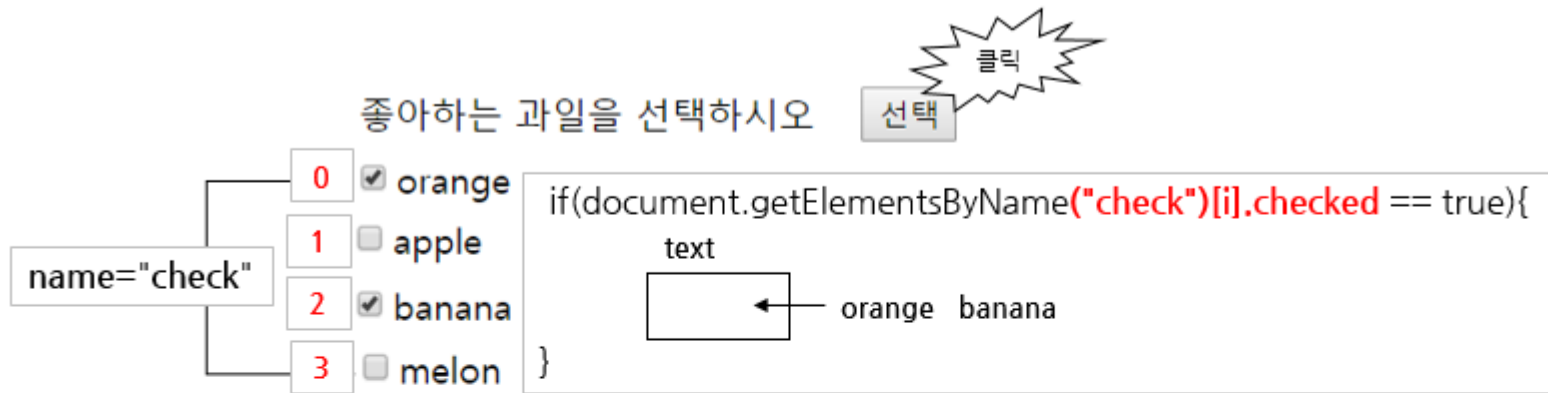
속성

속성	설명
innerText	요소의 내용을 Text로 설정하거나 반환한다.
innerHTML	요소의 내용을 HTML로 설정하거나 반환한다

[ex03_37.html] getElementById로 id 값 추출하기



[ex03_38.html] getElementByName으로 좋아하는 과일 찾기

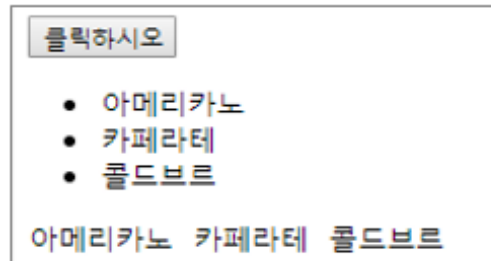
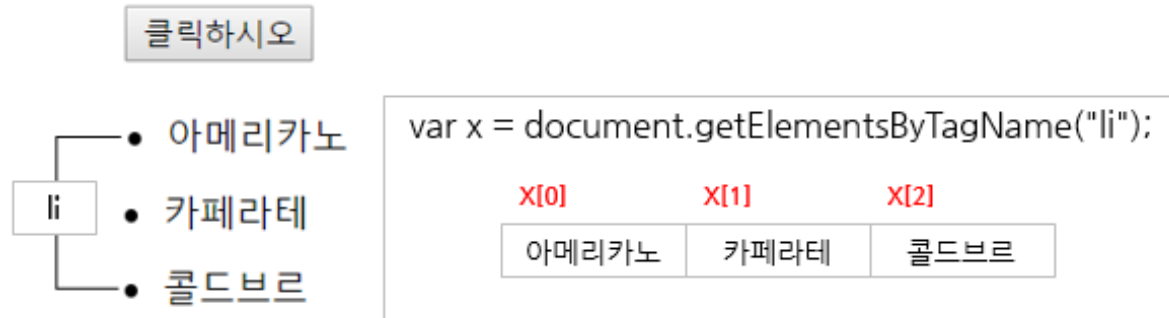


좋아하는 과일을 선택하시오

- ☒ orange
- ☐ apple
- ☒ banana
- ☐ melon

내가 좋아하는 과일은 orange banana 입니다

[ex03_39.html] getElementsByTagName 으로 Tag값 추출하기



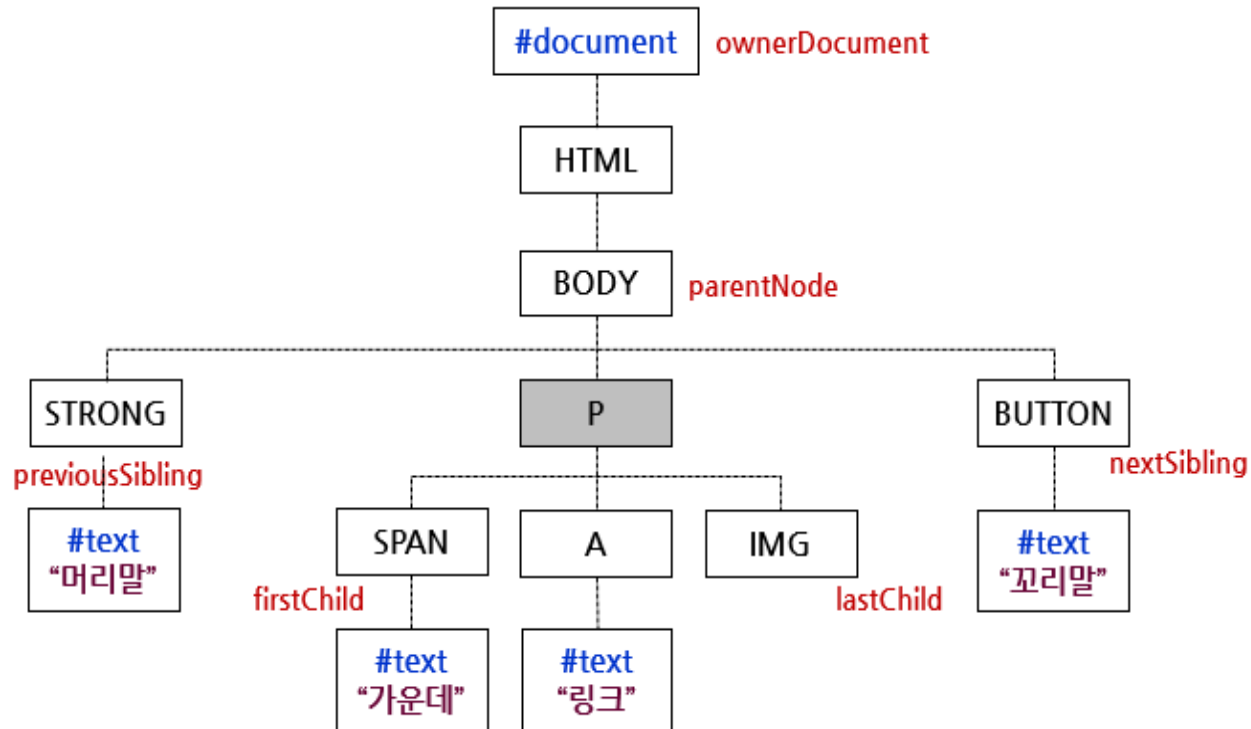
근거리 선택자

속성	설명
ownerDocument	최상위 노드를 선택한다.
parentNode	선택요소의 부모 노드를 선택한다.
childNodes	선택요소의 모든 자식 노드(배열객체로 저장)를 선택한다.
children	선택요소의 자식 노드(배열객체로 저장)만 선택한다.
firstChild	선택요소의 첫 번째 자식 노드만 선택한다.
lastChild	선택요소의 마지막 자식 노드만 선택한다.
previousSibling	선택요소의 이전에 오는 형제 노드를 선택한다.
nextSibling	선택요소의 다음에 오는 형제 노드를 선택한다.

DOM 객체 메서드

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

[ex03_40.html] 근거리 선택자 이해



머리말



가운데 [링크](#)

꼬리말

topNode : #document
firstChild : SPAN
lastChild : IMG
parentNode : BODY
previousSibling : STRONG
nextSibling : BUTTON

[ex03_41.html] 근거리 선택자 활용

- 아메리카노
- 카페라테
- 콜드브르

1번째 데이터 = 아메리카노
3번째 데이터 = 카페라테
5번째 데이터 = 콜드브르

3.2.5 이벤트(Event) 모델

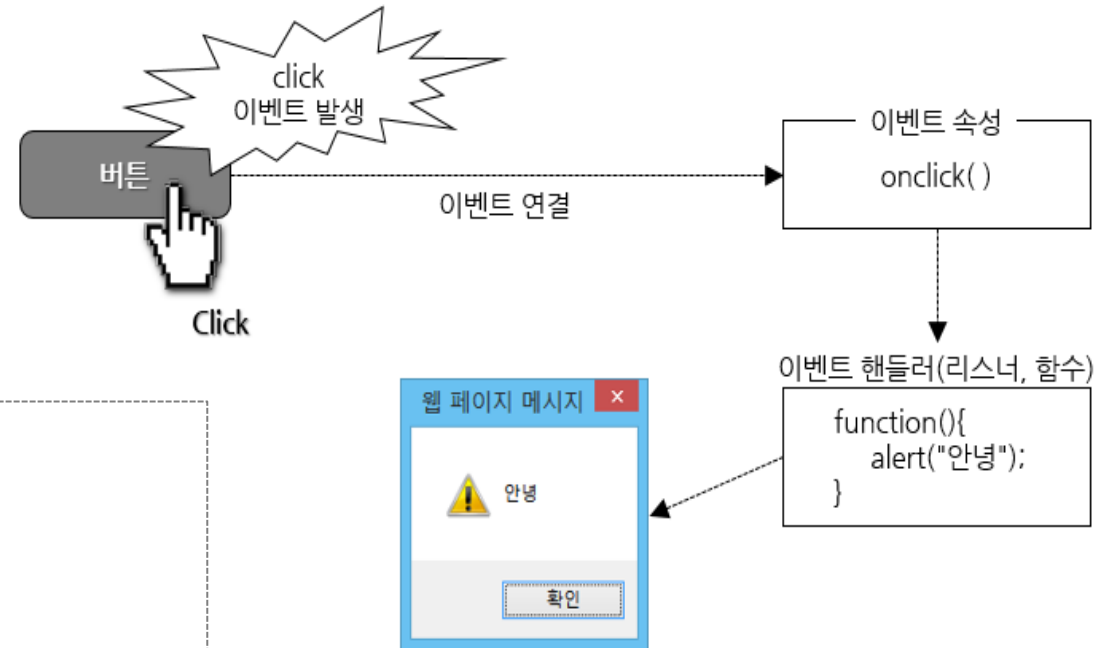
- 이벤트(Event)란 키보드 입력, 마우스 클릭과 같이 사용자가 특정 행위를 했을 때 행위에 대한 결과 동작을 보여주는 것이다.

이벤트 모델 구성

이벤트 이름	이벤트 속성	이벤트 핸들러(리스너, 함수)
click	onclick	function(){ }

ex)

```
<html>
<head>
  <title>문서 구조</title>
</head>
<body>
  <input type="button" value="버튼" onclick="alert('안녕')"> </body>
</html>
```





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 뗄 때 발생한다.
keypress	onkeypress()	키보드에 문자가 입력될 때 발생한다.

포커스 이벤트

이벤트	이벤트 속성	이벤트 발생 시기
focus	onfocus()	포커스를 얻었을 때 발생한다.
blur	onblur()	포커스를 잃었을 때 발생한다.
focusin	onfocusin()	포커스를 얻기 직전에 발생한다.
focusout	onfocusout()	포커스를 잃기 직전에 발생한다.

폼 이벤트

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

유저 인터페이스(UI) 이벤트

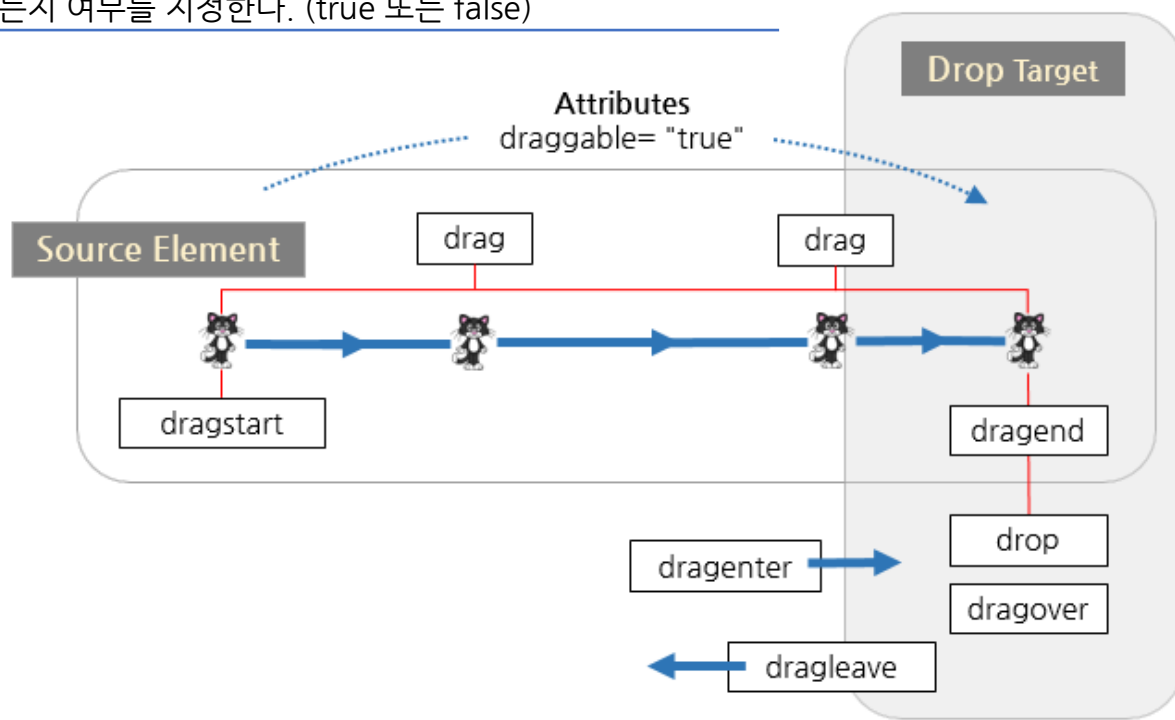
이벤트	이벤트 속성	이벤트 발생 시기
load	onload()	웹페이지의 로드가 완료되었을 때 발생한다. (처음 페이지 요청).
unload	onunload()	웹페이지가 빠져나갈 때 발생한다. (새로운 페이지 요청)
error	onerror()	스크립트에 오류가 있을 때 발생한다.
resize	onresize()	브라우저의 창 크기를 변경할 때 발생한다.
scroll	onscroll()	페이지를 위, 아래로 스크롤 할 때 발생한다.
select	onselect()	요소를 선택할 때 발생한다. (text, textarea는 제외)

터치 이벤트

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

드래그 이벤트

이벤트	이벤트 핸들러	이벤트 발생 시기
drag	ondrag()	요소가 드래그 되고 있는 동안 소스 객체에서 발생한다.
dragend	ondragend()	요소의 드래그가 끝나면 소스 객체에서 발생한다.
dragenter	ondragenter()	요소가 타겟 객체의 태그 영역에 들어갈 때 타겟 객체에서 발생한다.
dragleave	ondragleave()	요소가 타겟 객체의 태그 영역에 나갈 때 타겟 객체에서 발생한다.
dragover	ondragover()	요소를 타겟 객체 영역에서 드래그 하는 동안 타겟 객체에서 발생한다.
dragstart	ondragstart()	요소를 드래그하기 시작할 때 소스 객체에서 발생한다.
drop	ondrop()	요소를 타겟 객체 영역에서 놓으면 타겟 객체에서 발생한다.
draggable		요소를 드래그 할 수 있는지 여부를 지정한다. (true 또는 false)



이벤트 모델

- 이벤트 모델이란 문서 객체에 이벤트를 연결하는 것이다.
- 총 4가지 방법의 이벤트가 있다.

이벤트 모델 종류

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

❖ 인라인 이벤트 모델

- HTML 태그에 JavaScript 코드를 추가해서 이벤트 핸들러를 연결한다.
- 하나의 이벤트에 하나의 핸들러만 연결이 가능하다.

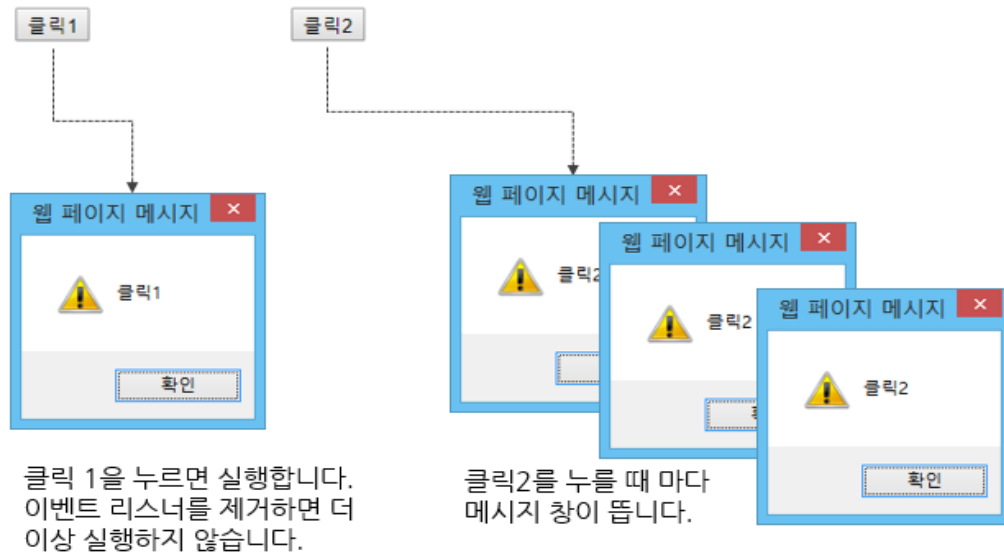
[ex03_42.html] 인라인 이벤트 모델의 사용



❖ 고전 이벤트 모델

- 문서 객체의 이벤트를 사용해서 이벤트 핸들러에 연결하는 방법이다.
- 하나의 이벤트에 하나의 핸들러만 연결할 수 있다.

[ex03_43.html] 고전 이벤트 모델의 사용



❖ 인터넷 익스플로러 이벤트 모델

- 한 번에 하나의 이벤트 핸들러만 연결할 수 있는 Level 0의 단점을 보완하기 위해 만들어진 방법이다.
- 한 번에 여러 가지 이벤트 핸들러를 연결할 수 있다.

인터넷 익스플로러 이벤트 메서드

메서드	설명
attachEvent(eventProperty, eventListener)	이벤트를 연결한다.
detachEvent(eventProperty, eventListener)	이벤트를 삭제한다.

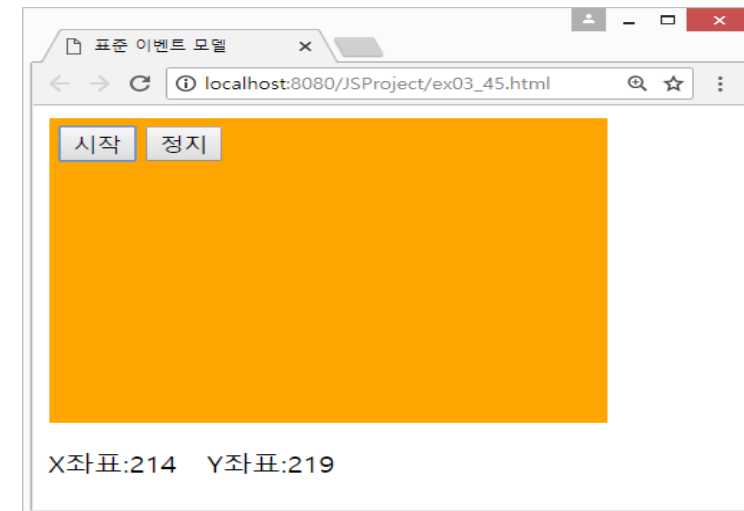
❖ 표준 이벤트 모델

- 객체나 요소의 메서드로 이벤트 핸들러를 전달하는 방식이다.
- 한 번에 여러 가지 이벤트 핸들러를 연결할 수 있다.

표준 이벤트 모델

메서드	설명
<code>addEventListener(eventName, handler, [useCapture])</code>	특정 이벤트를 등록하고 해당 함수를 실행한다.
<code>removeEventListener(eventName, handler)</code>	등록한 이벤트 리스너를 제거한다.

[ex03_44.html] 표준 이벤트 메서드



3.2.6 JSON (JavaScript Object Notation)

- JSON객체는 클라이언트와 서버의 통신에서 데이터를 주고받을 때 주로 이용한다.
- 저중량(lightweight) 데이터를 교환하는 형식으로 컴퓨터 시스템이 파싱하고 생성하기 쉽다.

JSON의 특징

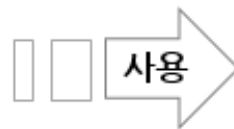
1. JSON은 특정 언어에 종속되지 않는 완벽한 독립적인 텍스트 형식을 가지고 있다.
2. JSON 객체의 데이터는 이름(key) / 값(value)의 쌍(pair)으로 구성한다.
3. {(left brace)로 시작해서 (right brace)}로 끝난다.
4. Key와 value 사이에는 콜론(:)으로 구분한다.
5. 각각의 데이터와 데이터 사이에는 콤마(,)로 구분한다.
6. key는 문자열로 만들기 때문에 이중 따옴표(" ") 또는 단일 따옴표(' ')로 표시한다.

[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']

객체와 배열의 경우 다음과 같이 작성하면 된다.

객체인 경우

```
var book={ "code" : 101 , "title" : JAVA , "page" : 350 }
```

배열인 경우

```
var book=[{ "code" : 101 , "title" : JAVA , "page" : 350 } ,  
           {"code" : 102 , "title" : JSP , "page" : 500 }]
```

[ex03_45.html] JSON 객체 생성

emp

0x1000

이름 (key)	값 (value)
name	민들레
age	25
tel	111-1111
address	서울

emp.name 또는 emp['name']으로 접근합니다.

JSON표기법

이름:민들레
나이:25
전화:111-1111
주소:서울

[ex03_46.html] JSON 배열

emp

0x1000

0번째 (0x1000)	
이름 (key)	값 (value)
name	kim
age	30

1번째 (0x2000)	
이름 (key)	값 (value)
name	kim
age	30

2번째 (0x3000)	
이름 (key)	값 (value)
name	kim
age	30

JSON 표기법

이름 : kim
나이 : 30

이름 : lee
나이 : 40

이름 : park
나이 : 20

emp[0].name 또는 emp[0]['name'] 으로 접근합니다.

[참고]

with 키워드를 이용하여 "객체명.이름"에서 객체명을 생략하는 방법이 있다.

with 키워드

객체명을 with로 선언하고 코드를 줄여 준다.

With 키워드를 사용하지 않은 코드

```
var view="";  
view += "이름:" + info.name;  
view += "나이:" + info.age;  
view += "점수:" + info.score;  
document.write(view);
```

With 키워드를 사용한 코드

```
var view="";  
with(info){  
    view += "이름:" + name;  
    view += "나이:" + age;  
    view += "점수:" + score;  
}  
document.write(view);
```

이름:가나다 나이:23 점수:75

[ex03_47.html] JSON 로그인

emp
0x1000

이름 (key)	값 (value)
id	abcd
pw	1234
name	홍길동
point	1250

아이디와 비밀번호가 맞는 경우

emp.id == a
abcd == abcd

&&

emp.pw == b
1234 == 1234

JSON로그인 x

localhost:8080/JSProject/...

[로그인 화면]

abcd 아이디를 입력하십시오

1234 비밀번호를 입력하십시오

Login

홍길동님 환영합니다.
현재 포인트는 1250점 입니다.

아이디는 맞지만 비밀번호는 틀린 경우

emp.id == a
abcd == abcd

&&

emp.pw == b
1234 == 1111

JSON로그인 x

localhost:8080/JSProject/...

[로그인 화면]

abcd 아이디를 입력하십시오

1111 비밀번호를 입력하십시오

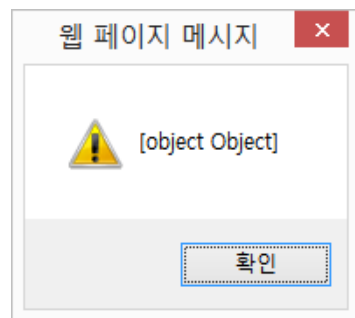
Login

아이디 또는 비밀번호가 틀립니다.

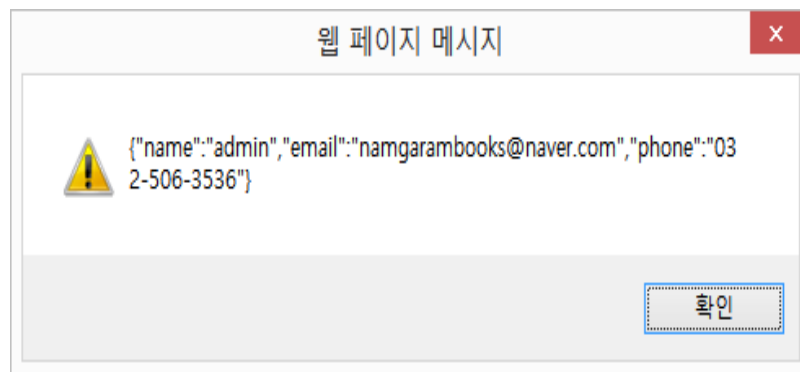
JSON 메서드

메서드	설명
parse()	string 객체를 JSON 객체로 변환해 준다.
stringify()	JSON 객체를 String 객체로 변환해 준다.

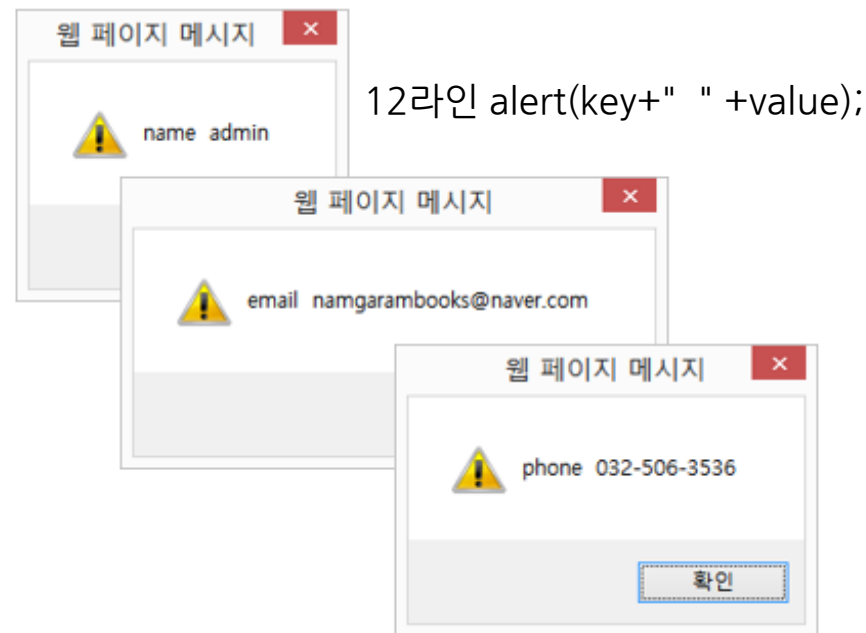
[ex03_48.html] JSON stringify와 parse



6라인의 alert(user);



9라인 alert(userStr);



3.2.7 정규 표현식

- 정규 표현식(Regular Expression)이란, 특정한 규칙을 가진 문자열의 집합으로 조합된 문자열이 특정 규칙에 맞게 작성되었는지를 찾아내는 검색 패턴이다.
- 개인 정보 보호에 관한 보안정책에 의해 회원가입을 할 때 아이디와 비밀번호의 설정을 복잡하게 조합해야만 가입을 할 수 있도록 규정했다.

정규 표현식을 만드는 방법은 2가지가 있다.

1. RegExp 객체를 이용한 생성
2. 정규 표현식 리터럴을 이용한 생성

[형식]

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

```
var 객체명 = /정규표현식/[Flag];
```

정규 표현식의 패턴

표현식	설명	표현식	설명
^	문자열의 시작을 의미한다.		or를 의미한다.
\$	문자열의 종료를 의미한다.	\d	숫자[0-9]와 동일한다.
.	임의의 한 문자이다.	\s	공백 문자이다.
*	앞 문자가 없을 수도 있고 많을 수도 있다.	\S	공백 문자가 아님을 의미한다.
+	반복을 표현하며 앞 문자가 한번 이상 반복함을 의미한다.	\w	단어를 만들 수 있는 알파벳 또는 숫자이다.
?	존재 여부를 표시하며 앞 문자가 없거나 있습니다.	\W	알파벳, 숫자를 제외한 문자이다.
{ }	횟수 또는 범위를 나타냅니다.	\D	숫자를 제외한 모든 문자이다
()	그룹을 의미합니다.	\w	\w 다음에 특수문자가 오면 그 문자 자체를 의미한다.
[]	문자의 집합이나 범위를 나타내며 두 문자 사이는 - 기호로 범위를 나타냅니다.		

Flag종류

Flag	설명
g	문자열 내의 모든 패턴을 찾아 준다.
i	문자열의 대, 소문자를 구별하지 않는다.
m	문자열의 줄바꿈이 일어나도 찾아준다.

간단한 사용 방법 예제

ex) $^[0-9]*\$$

^ 으로 우선 문자열의 시작을 알린다.

[0-9] 는 대괄호 사이에 두 숫자를 넣어 범위를 지정할 수 있다.

* 를 넣으면 문자 수는 상관없다.

\$ 로 문자열의 종료를 알린다.

[ex03_49.html] 정규 표현식


[아이디 입력 조건]

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



아이디

웹 페이지 메시지

 다시입력

[ex03_50.html] 정규 표현식


[날짜 입력 조건]

연도 4자리, 월이나 일은 1자리이거나 2자리로 한다.
예시: 2018-05-24



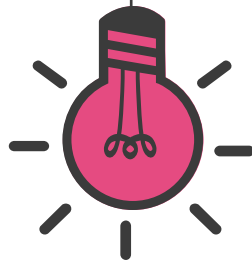
입사연월일

웹 페이지 메시지

 입사축하

03

JavaScript와 표준 API 활용



3.3.1 Audio와 Video API

- HTML5에서는 Audio 또는 Video 태그를 삽입하고 재생과 관련된 제어를 수행하는 API를 제공한다.

❖ Media를 이용한 미디어 제어

- 이전의 HTML은 음악이나 동영상을 재생하기 위해서 외부 기술인 플러그인은 다음과 같다.
 - ✓ Window의 미디어 플레이어(Media Player)
 - ✓ Adobe의 플래시 플레이어(Flash Player)
 - ✓ Apple의 퀵 타임(QuickTime Player)
 - ✓ Microsoft의 실버라이트(Silverlight)



재생 플러그인

- 하지만, HTML5에서도 Audio나 Video와 같은 멀티미디어를 자체적으로 지원한다.

❖ Audio 주요 메서드 • 음악을 쉽게 재생할 수 있다.

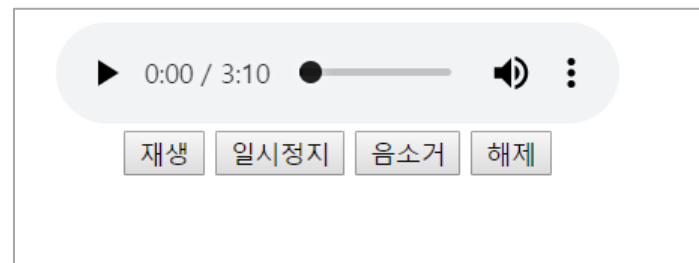
멀티미디어 태그의 기본 메서드

메서드	설명
play()	동영상 또는 음악을 재생한다.
pause()	동영상 또는 음악을 잠시 멈춘다.

Audio 태그 속성

속성	사용 방법	설명
controls	controls="controls"	재생(play), 일시정지(pause) 등 같은 조작 패널을 화면에 보이게 할지 여부를 설정한다.
loop	loop="loop"	음악 트랙이 끝난 후 자동으로 반복할 것인지를 설정한다.
autoplay	autoplay="autoplay"	페이지가 로딩 되자마자 자동으로 트랙을 재생할지를 설정한다.
preload	preload="auto metadata none"	음악을 재생하기 전에 준비할 사항을 설정한다. (auto: 페이지 로딩을 마치고 즉시 로딩 / metadata: 트랙 제목만 로딩 / none: 페이지가 로딩되어도 파일은 로딩 안됨)
src	src="경로"	재생 파일의 위치를 설정하는 속성이다.
volume	volume="숫자"	음악의 소리 크기를 설정한다.
currentTime	currentTime="초"	음악의 현재 위치를 표시한다.

[ex03_51.html] 오디오



❖ Video 주요 메서드 • 동영상을 쉽게 재생할 수 있다.

Video 태그 속성

속성	사용 방법	설명
height	height="픽셀"	동영상의 높이를 지정한다.
width	width="픽셀"	동영상의 넓이를 지정한다.
controls	controls="controls"	동영상 재생 시 조작 패널의 화면에 보이게 할지 여부를 설정한다.
loop	loop="loop"	동영상을 자동 반복할지를 설정한다.
autoplay	autoplay="autoplay"	동영상을 자동 재생할지를 설정한다.
preload	preload="auto metadata none"	동영상을 재생하기 전에 준비할 사항을 설정한다. (auto: 페이지 로딩을 마치고 즉시 로딩, metadata: 크기, 첫 프레임, 트랙 리스트, 길이 등을 미리 로드, none: 페이지가 로딩되어도 파일은 로딩 안됨)
src	src="경로"	동영상 재생 파일의 위치를 설정하는 속성이다.
muted	muted="muted"	동영상 재생 시 소리를 음소거 한다.
poster	poster="경로"	동영상을 기다리는 동안 화면에 보이는 이미지를 설정한다.

[ex03_52.html] 비디오



3.3.2 Canvas API

- Canvas는 지정한 영역 안에서 그래픽을 자유롭게 그릴 수 있는 기능이다.

Canvas의 메서드

메서드	설명
<code>beginPath()</code>	그리기를 위한 경로를 시작하거나 현재 경로를 다시 설정한다.
<code>getContext()</code>	이미지와 그래픽을 조작하기 위한 메서드와 속성을 제공한다.

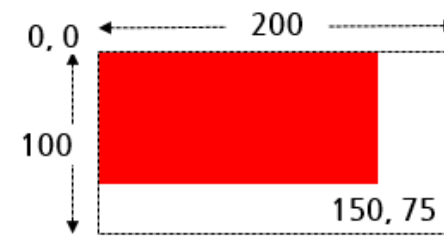
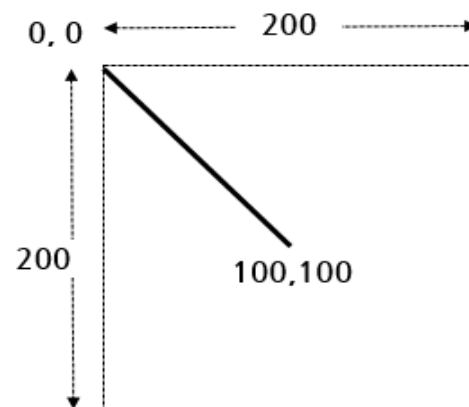
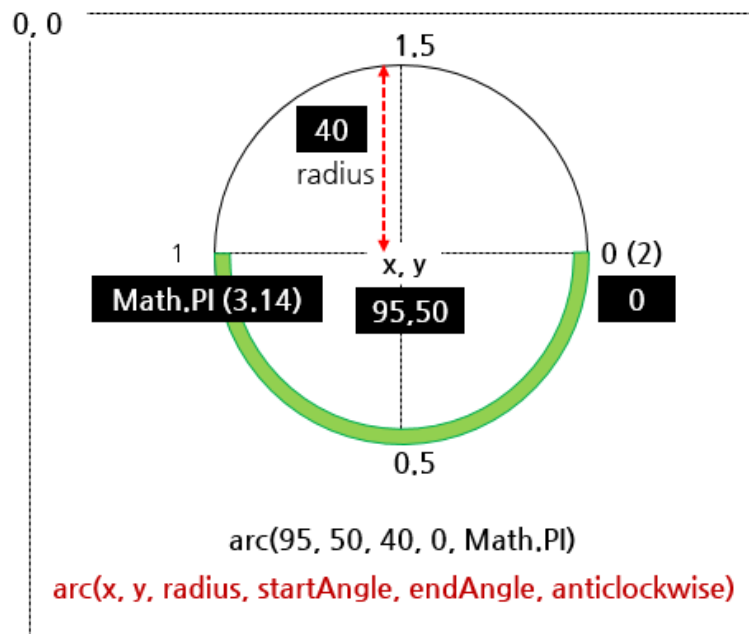
Canvas의 선 관련 메서드

메서드	설명
<code>moveTo(x, y)</code>	라인의 시작점을 선언한다.
<code>lineTo(x, y)</code>	라인의 종료점을 선언한다.
<code>stroke()</code> 또는 <code>fill()</code>	선을 그리기 시작한다.

Canvas 그래픽 관련 메서드

메서드	설명
<code>strokeRect(startX, startY, width, height)</code>	"채워지지 않은 " 사각형을 그립니다.
<code>fillRect(startX, startY, width, height)</code>	"채워진" 사각형을 그립니다.
<code>clearRect(startX, startY, width, height)</code>	지정한 사각형 영역을 지웁니다.
<code>fillStroke()</code>	설정된 값이 앞으로 그려지는 모든 선 색상의 기준이 됩니다. (기본값: #000)
<code>arc(x, y, radius, startAngle, endAngle, anticlockwise)</code>	원 위의 점들 중 startAngle에서 endAngle까지 연결한 선을 그립니다. anticlockwise(기본값: false)을 true로 설정하면 시계 반대 방향으로 그립니다.
<code>fillStyle</code>	설정된 값이 앞으로 그려지는 모든 채우기 색상의 기준이 됩니다. (기본값:#000)
<code>fillText(text, x, y)</code>	"채워진" 텍스트를 만듭니다.
<code>strokeText(text, x, y)</code>	"채워지지 않은 " 텍스트를 만듭니다.

[ex03_53.html] Canvas로 도형 그리기



[ex03_54.html] Canvas를 이용한 텍스트

Hello HTML5API

Hello HTML5API

[ex03_55.html] 텍스트와 이미지



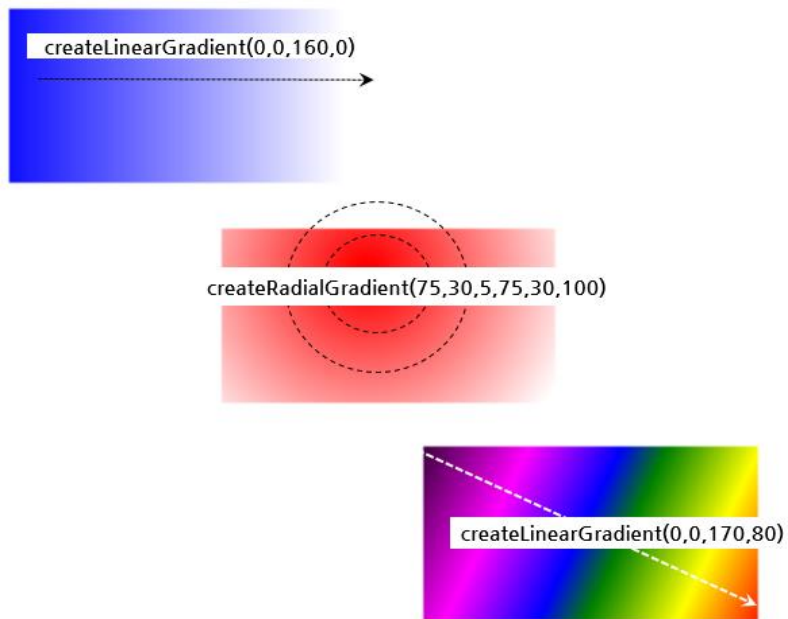
Welcome To My Blog

HTML5 Canvas

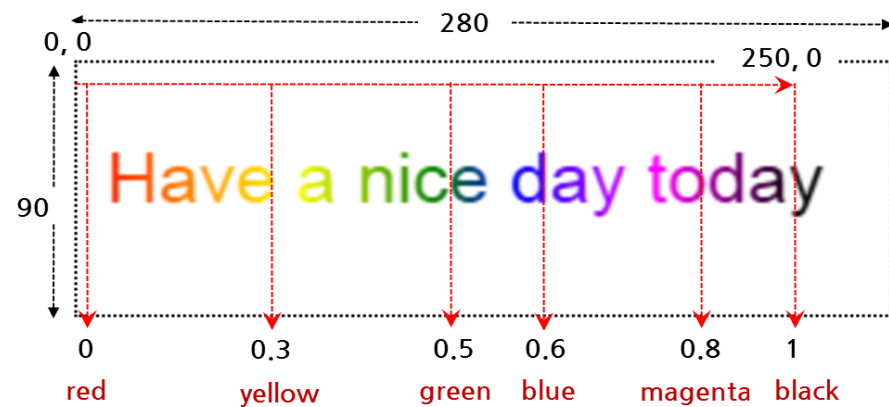
그래디언트 관련 메서드

메서드	설명
<code>createLinearGradient(x1, y1, x2, y2)</code>	선형 그래디언트
<code>createRadialGradient(x1,y1,r1,x2,y2,r2)</code>	원형 그래디언트
<code>addColorStop(offset, color)</code>	그래디언트 경계색 지정

[ex03_56.html] 선형 그래디언트와 원형 그래디언트



[ex03_57.html] 선형 그래디언트를 응용한 문자열



3.3.3 Drag & Drop API

- Drag & Drop은 쇼핑몰에서 상품을 카트에 담을 때 많이 사용한다.
- 메일이나 게시판에서 파일을 첨부할 때도 사용한다.

Drag & Drop 메서드

메서드	설명
setData(type, data)	끌어다 놓기 위해 새로운 type을 정의하고 끌어올 data를 저장한다.
getData(type)	setData에서 정의한 type을 가져온다.

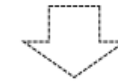
박스 안으로 드래그 하시오

```
id="box"
ondrop="drop(event)" ondragover="allowDrop(event) "
```

```
draggable="true" ondragstart="drag(event)"
```



w3schools.com id="logo"



박스 안으로 드래그 하시오

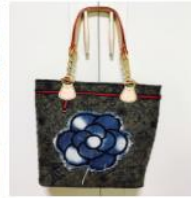


w3schools.com

<그림 출처= www.w3schools.com>

[ex02_58.html] 장바구니

원하는 상품을 카트에 담으세요



3.3.4 웹 스토리지(Web Storage) API

- 웹 스토리지는 브라우저(또는 클라이언트) 내부에 스토리지라는 저장소를 제공해주는 기능이다.
- 사용자의 컴퓨터에 텍스트로 저장되어 있어서 쉽게 접근할 수 있고 내용 확인이 가능하다.

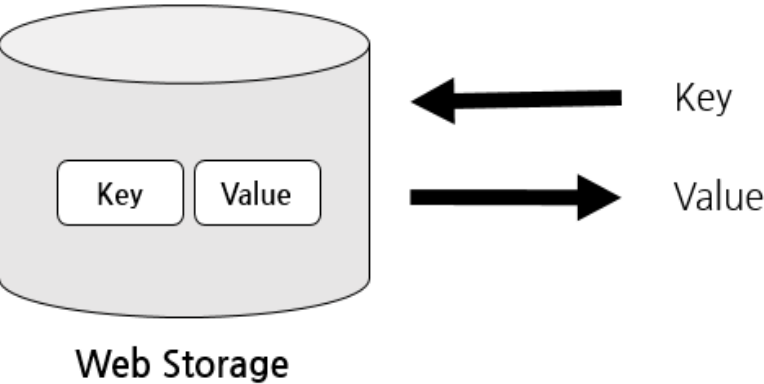
쿠키의 단점

- 사용 기한이 있는 저장소이고 4KB의 용량 제한이 있습니다.
- 서버에 요청할 때마다 쿠키를 전송해야 하므로 웹이 느려 지는 원인이 될 수 있다.

웹 스토리지의 장점

- 웹 스토리지는 약 5MB까지 저장 공간을 이용할 수 있다.
- 서버로 보내지 않기 때문에 사용자가 많은 양의 정보를 안전하게 보관할 수 있다.

Key와 Value로 Web Storage에 저장



웹 스토리지는 세션 스토리지(sessionStorage)와 로컬 스토리지(localStorage)가 있다.

	sessionStorage	localStorage
특징	1개의 세션만을 저장하는 데이터 객체입니다.	보관 기한이 없는 데이터를 저장하는 객체입니다.
사용기한	브라우저의 탭(또는 창)을 닫으면 객체에 저장된 데이터가 사라집니다.	브라우저의 탭(또는 창)을 닫거나 컴퓨터를 재 부팅해도 저장된 데이터는 사라지지 않습니다.

[ex03_59.html] 세션 스토리지

sessionStorage로 데이터 저장하기

웹 브라우저 탭이나 창을 닫으면 Counter는 리셋됩니다.
하지만 F5를 누르면 리셋되지 않습니다.

카운터 증가

[ex03_60.html] 로컬 스토리지

localStorage로 데이터 저장하기

브라우저 탭이나 창을 닫아도 Counter의 횟수는 초기화되지 않습니다.

카운터 증가

카운터 리셋

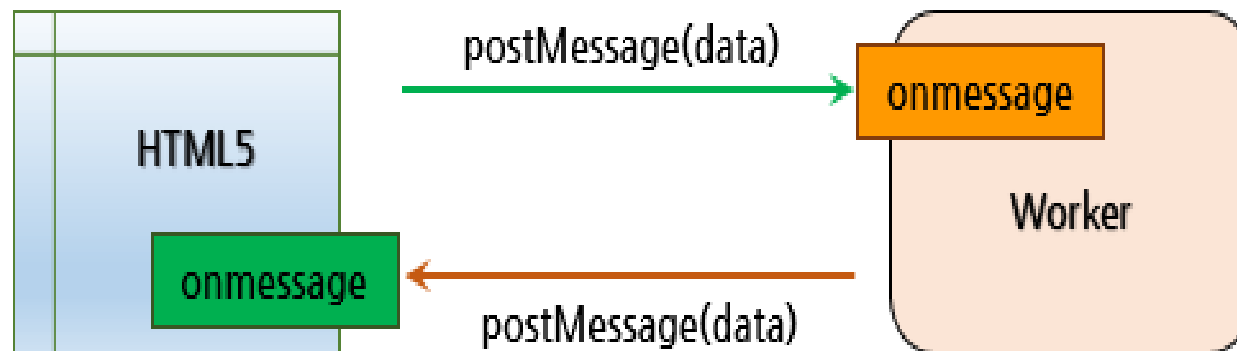
카운터의 현재 횟수는 1입니다!

3.3.5 웹 워커(Web Worker) API

- 웹 워커는 백그라운드에서 스레드(Thread)가 동작하게 해주는 JavaScript이다.
- 스레드란 응용 프로그램(프로세스) 내에서 실행하는 흐름의 단위를 말한다.

웹 워커의 활용 범위

1. 사용자 인터페이스를 방해하지 않고 지속적인 처리 작업을 해야 할 경우에 사용한다.
2. 복잡한 수학 계산이 많이 들어간 작업을 처리할 때 사용한다.
3. 로컬이나 원격지에 있는 리소스에 대한 액세스 작업을 할 경우에 사용한다.
4. 백그라운드에서 오랜 시간 작업해야 하는 경우에 사용한다.



웹 워커의 형식은 다음과 같다.

[형식]

```
var 객체명 = new Worker("파일명");
```

웹 워커 객체

이벤트 또는 메서드

설명

onmessage	워커로부터의 메시지를 전달받을 때 발생하는 이벤트이다.
onerror	워커가 예외를 발생시켰으나 내부적으로 처리되지 않는 경우 발생하는 이벤트이다.
postMessage()	데이터를 전송한다.
terminate()	워커 스레드를 강제로 종료한다.

[ex03_61.html] 웹 워커 / [worker.js] 웹 워커 JavaScript

개발자 도구로 Web Worker 테스트

Web Worker Test

test: 1
test: 2
test: 3
test: 4
test: 5

Start Stop

DevTools - localhost:8080/JSProject/ex03_63.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>웹 워커</title>
6 <script>
7 var worker;
8 function init(){
9   var btnStart=document.getElementById('btnStart');
10  var btnStop=document.getElementById('btnStop');
11  var demo=document.getElementById('demo');
12
13  btnStart.addEventListener('click', startWorker);
14
```

Line 7, Column 12

Console

호출 메시지: test: 1 ex03_63.html:28
호출 메시지: test: 2 ex03_63.html:28
호출 메시지: test: 3 ex03_63.html:28
호출 메시지: test: 4 ex03_63.html:28
호출 메시지: test: 5 ex03_63.html:28

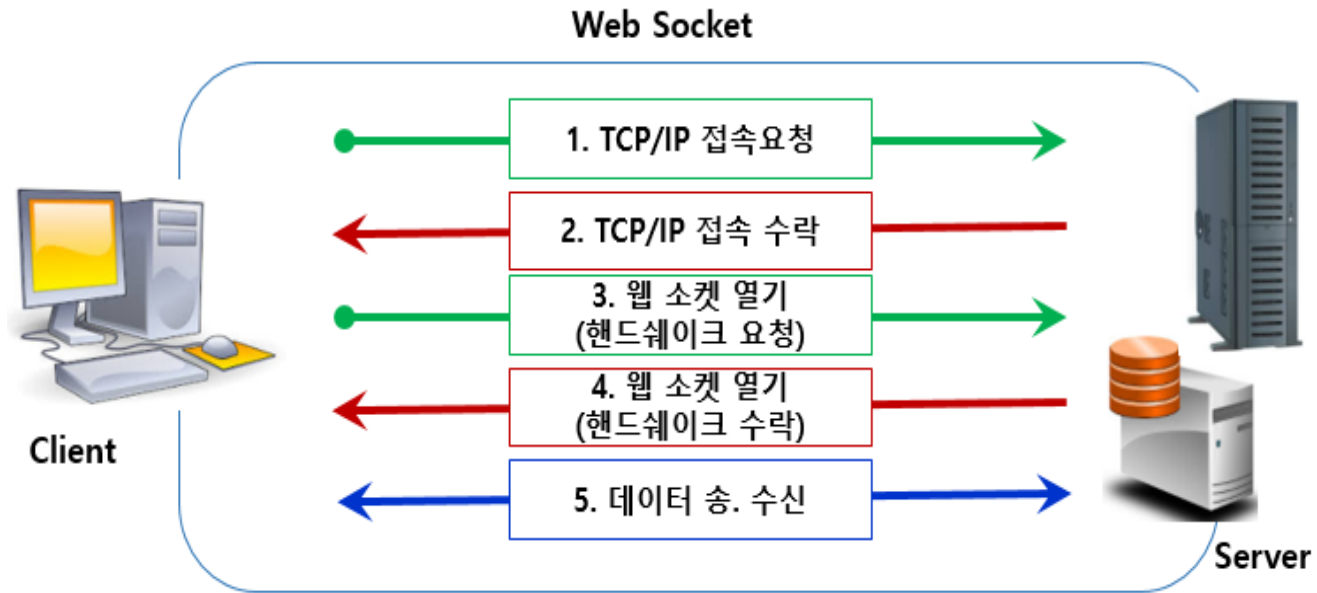


Web Worker Test

Start Stop

3.3.6 웹 소켓(Web Socket) API

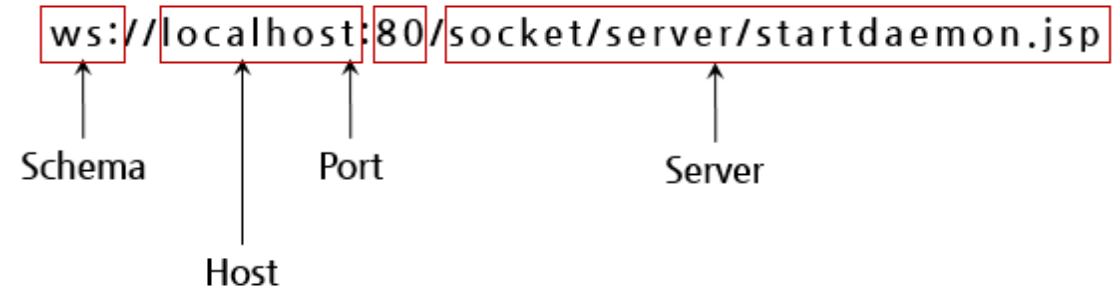
- 웹 소켓은 서버와 클라이언트 간에 실시간으로 데이터를 교환할 수 있게 해주는 기술이다.
- 서버와 클라이언트에 연결 객체를 생성하면 지속적인 양방향 통신이 가능하다.



웹 소켓의 형식은 다음과 같다.

[형식]

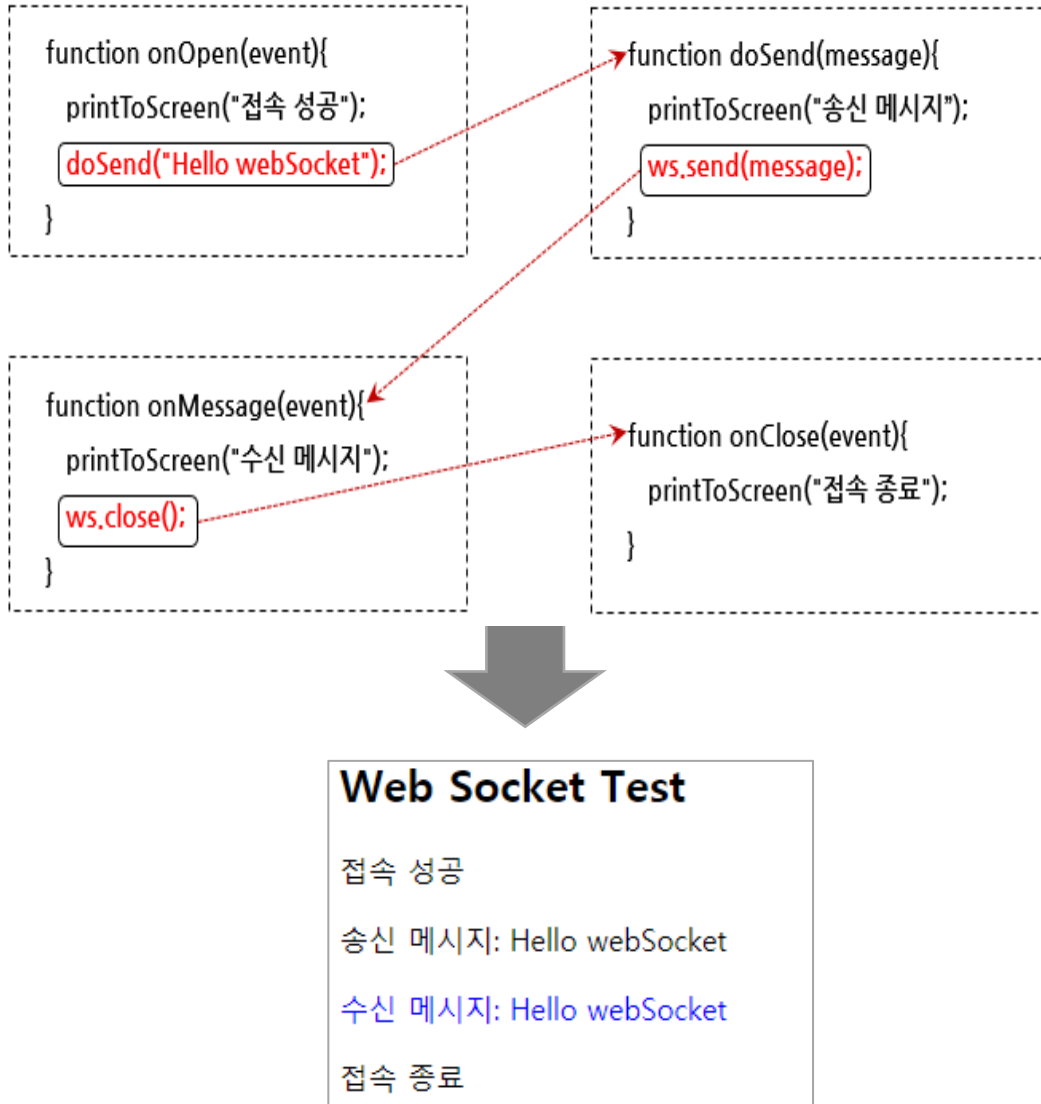
```
var 객체명 = new WebSocket("ws://사용자 도메인/페이지");
```



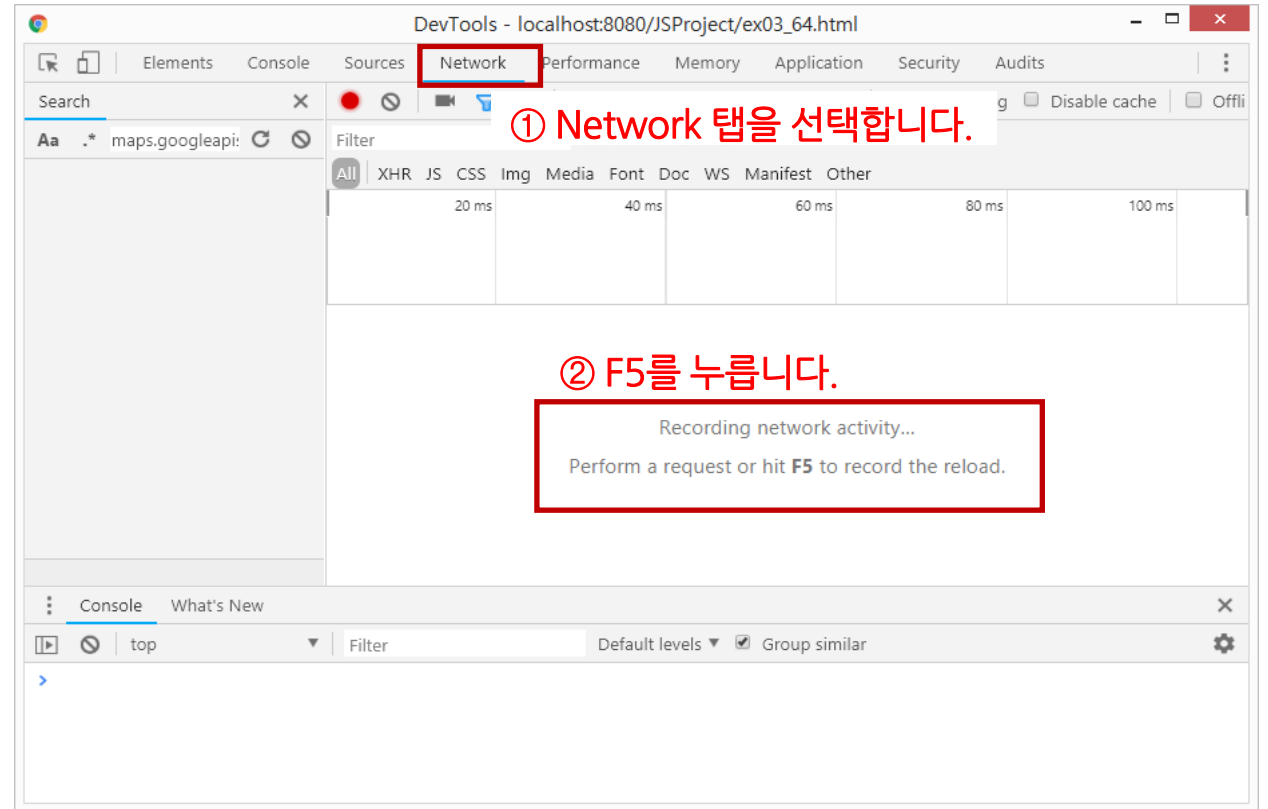
웹 소켓 객체

이벤트 또는 메서드	설명
onopen	서버와 연결이 설정되면 발생하는 이벤트이다.
onclose	서버와 연결이 끊어지면 발생하는 이벤트이다.
onmessage	서버에서 push(전송)하는 데이터를 받는 이벤트이다.
onerror	에러가 발생할 때 받는 이벤트이다.
send()	데이터를 송신하는 메서드이다.
close()	객체 종료 메서드이다.

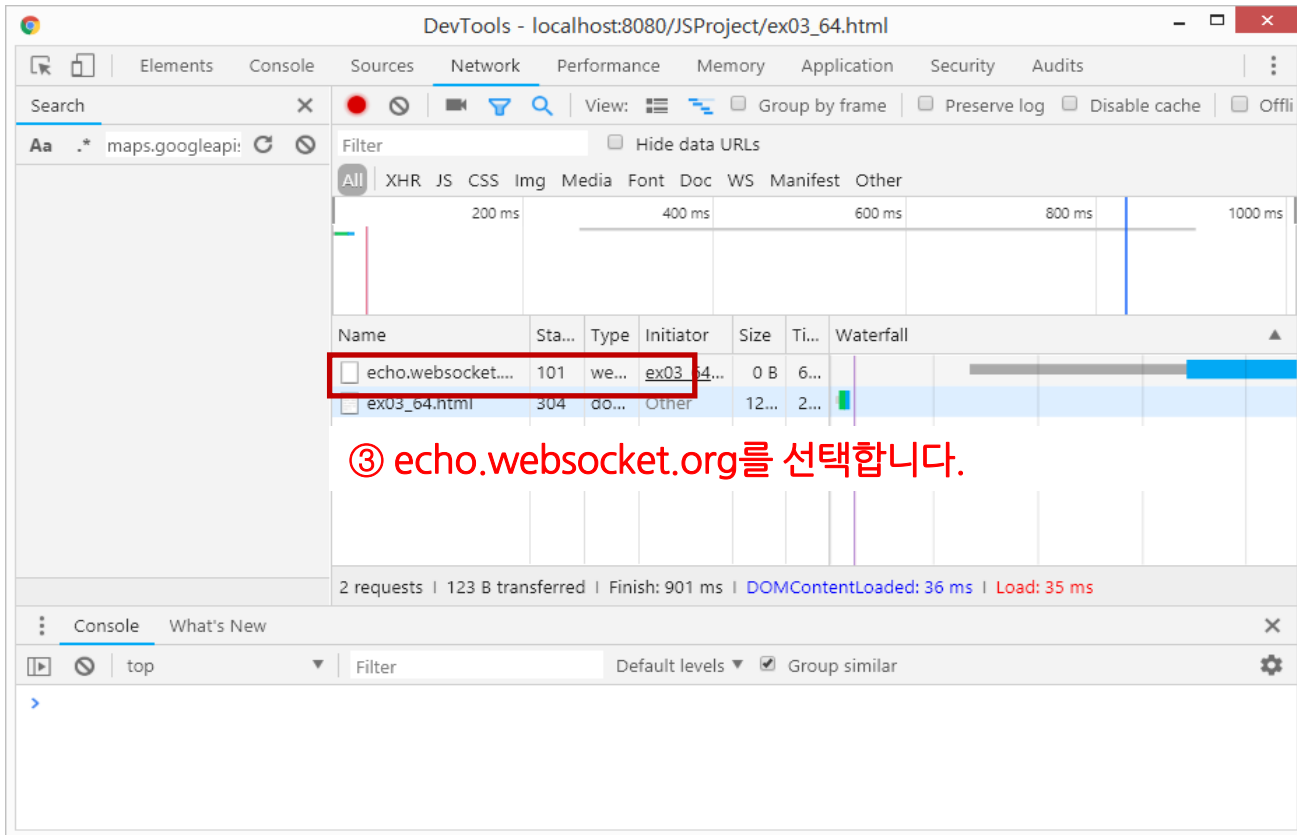
[ex03_62.html] 웹 소켓



개발자 도구의 Network 탭



접속 사이트 선택



DevTools - localhost:8080/JSProject/ex03_64.html

Search: Aa .* maps.googleapi: [refresh] [stop]

Filter: [x] Hide data URLs

View: [list icon] [waterfall icon] [preserve log icon] [disable cache icon] [offline icon]

Tab: All | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

Name	Sta...	Type	Initiator	Size	Ti...	Waterfall
echo.websocket....	101	we...	ex03_64...	0 B	6...	[bar]
ex03_64.html	304	do...	Other	12...	2...	[bar]

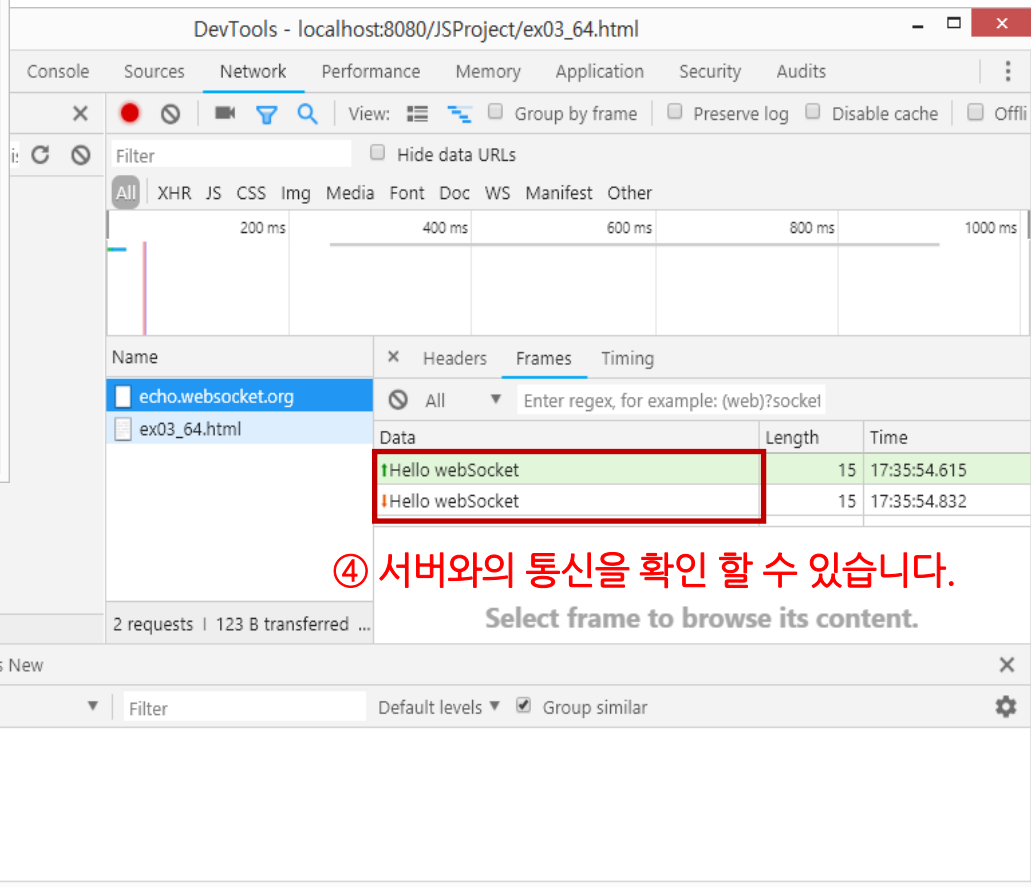
2 requests | 123 B transferred | Finish: 901 ms | DOMContentLoaded: 36 ms | Load: 35 ms

Console: What's New

top | Filter | Default levels | [x] Group similar

③ echo.websocket.org를 선택합니다.

클라이언트와 서버 간 통신 확인



DevTools - localhost:8080/JSProject/ex03_64.html

Console | Sources | Network | Performance | Memory | Application | Security | Audits

Filter: [x] Hide data URLs

View: [list icon] [waterfall icon] [preserve log icon] [disable cache icon] [offline icon]

Tab: All | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

Name	Headers	Frames	Timing
echo.websocket.org	[x]	All Enter regex, for example: (web)?socket	
ex03_64.html	Data	Length	Time
	!Hello webSocket	15	17:35:54.615
	!Hello webSocket	15	17:35:54.832

2 requests | 123 B transferred ...

Select frame to browse its content.

Console: What's New

top | Filter | Default levels | [x] Group similar

④ 서버와의 통신을 확인 할 수 있습니다.

3.3.7 지오로케이션(Geolocation) API

- 지오로케이션은 JavaScript API를 이용해서 사용자의 위치 정보를 제공하는 API이다.

위치 정보를 받아 오기 위해 알아야 할 사항

1. 위치를 수집하는 방식

Wi-Fi 네트워크와 IP 주소, GPS를 이용하여 현재의 위치 정보를 받아온다.

2. 개인 정보 보호

사생활 침해 등의 이유로 사용자의 동의를 받아야 합니다.

지오로케이션 메서드

```
navigator.geolocation.getCurrentPosition(success, [error], [options]);
```

메서드는 기기가 있는 곳의 현재 위치를 얻어 온다.

속성	설명
success	현재의 위치를 반환하는 콜백 함수이다.
error	오류가 발생하면 실행하는 콜백 함수이다.
option	위치 옵션이다.

[ex03_63.html] 지오로케이션

현재 위치가 궁금한가요?

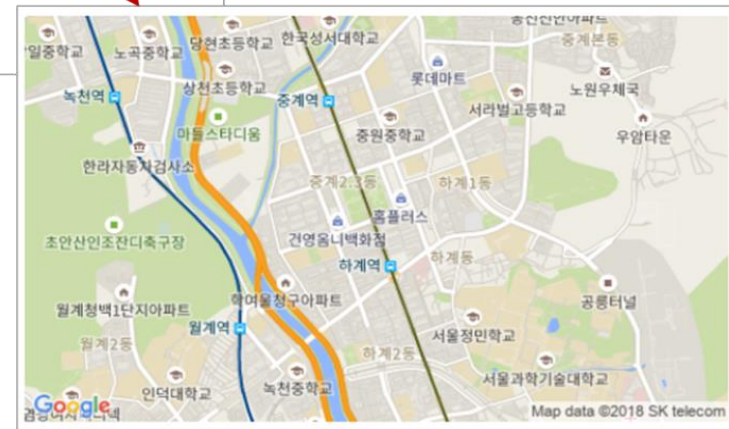
위도/경도

위도: 37.639285199999996
경도: 127.0678171

[ex03_64.html] 지오로케이션

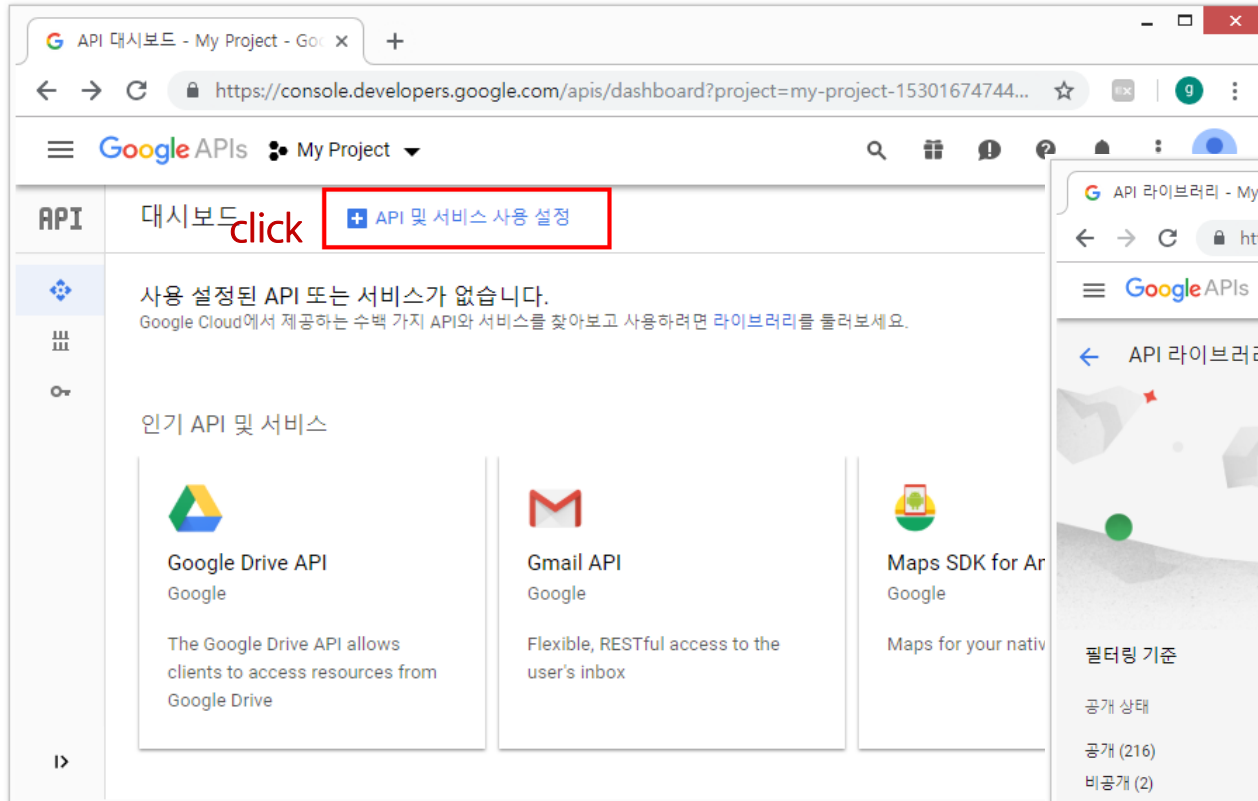
이곳의 지도가 궁금한가요? 지도보기

웹 브라우저는 Internet Explorer로 실행 하기 바랍니다.
Chrome은 보안상 지도를 출력하지 않습니다.

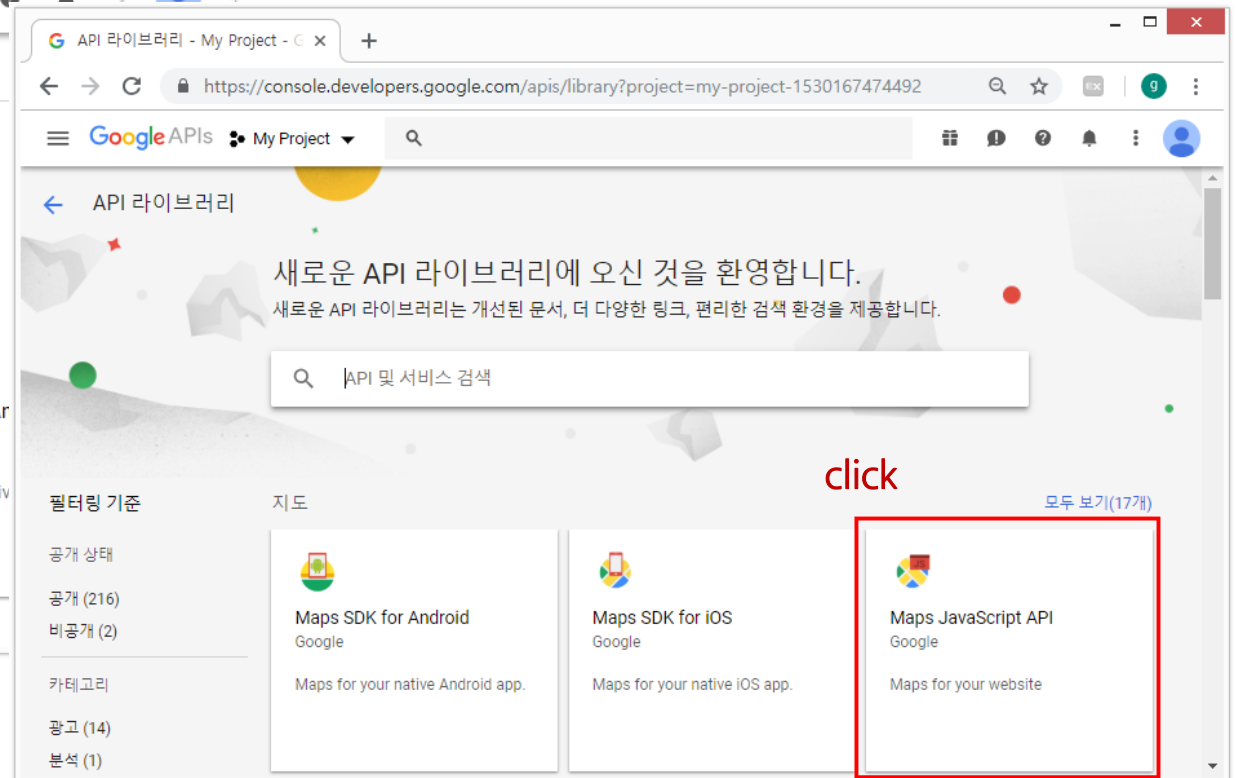


❖ Maps JavaScript API 인증키 받아오기

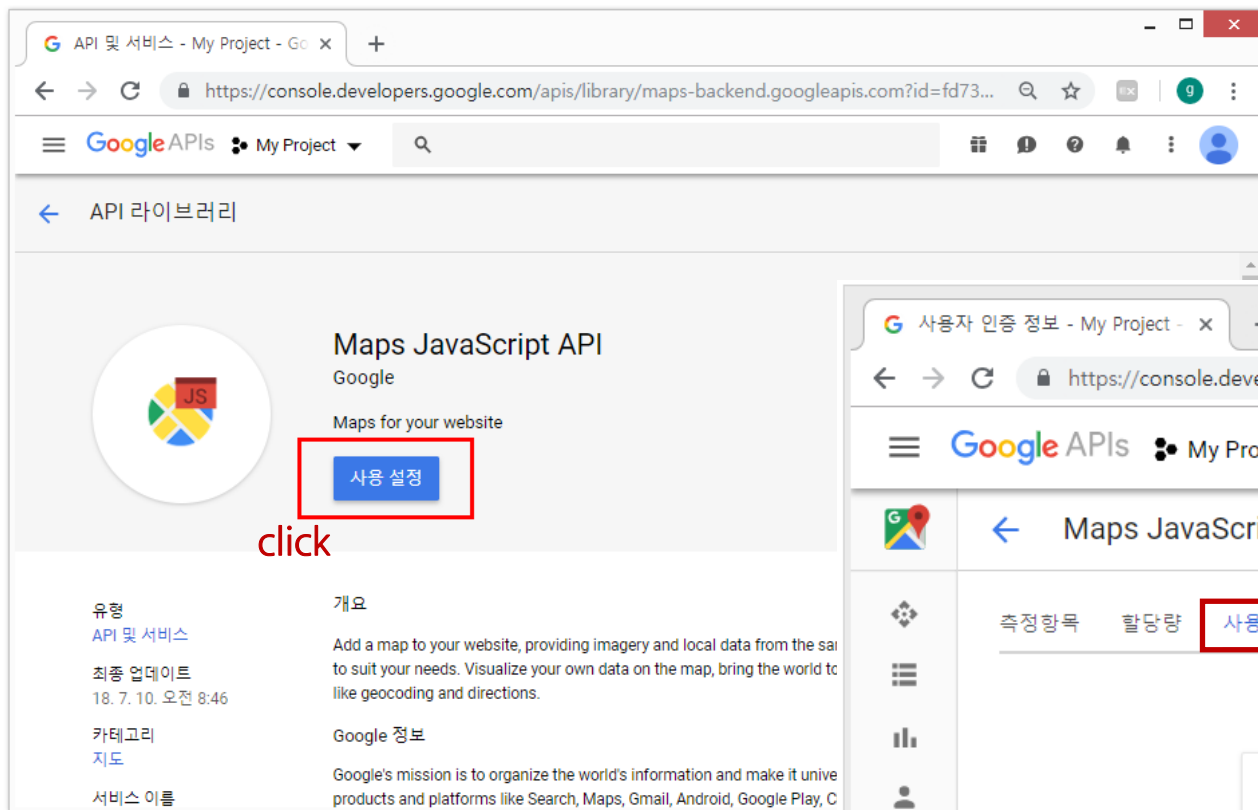
API 및 서비스 사용 설정



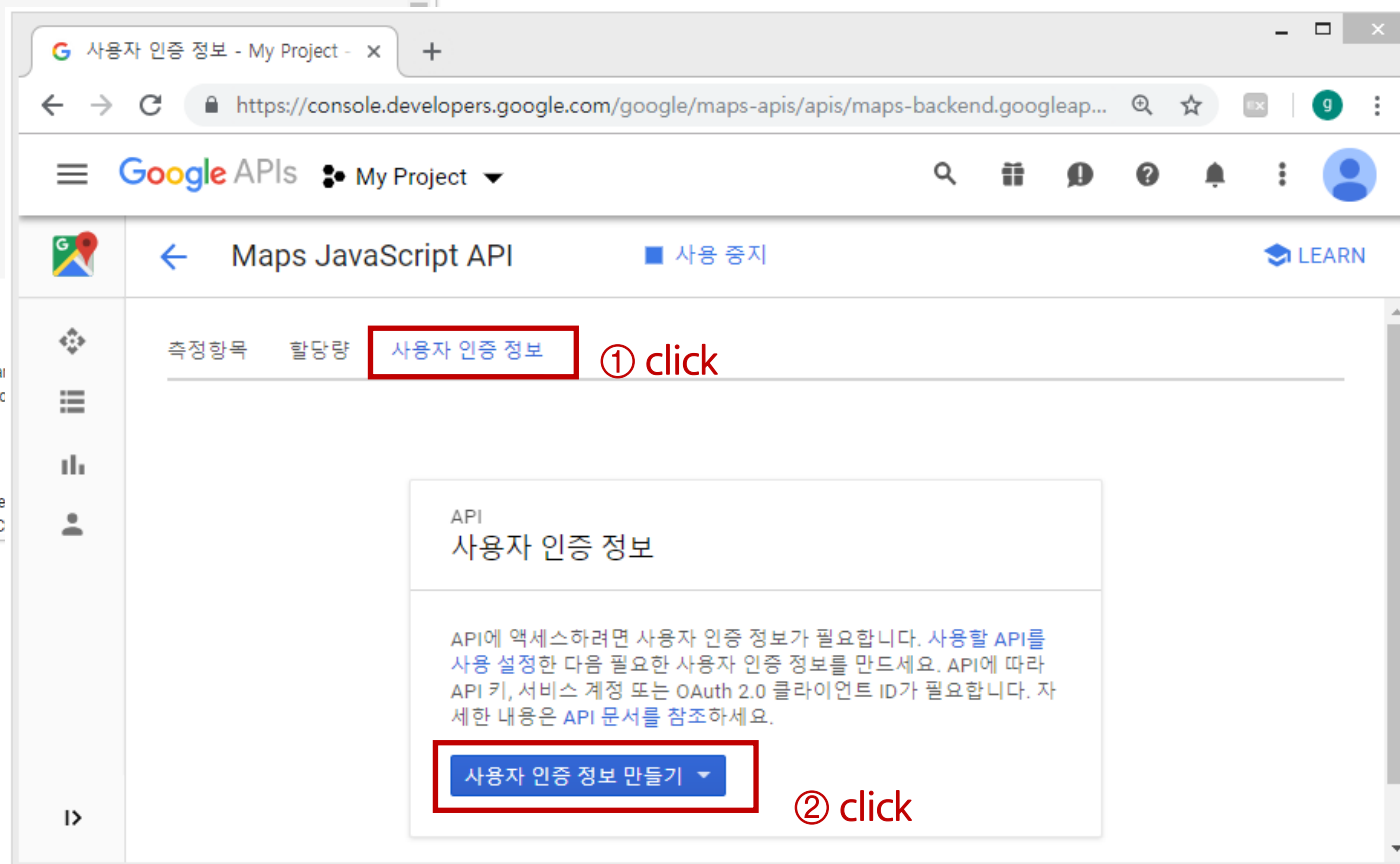
사용할 Map의 종류 선택



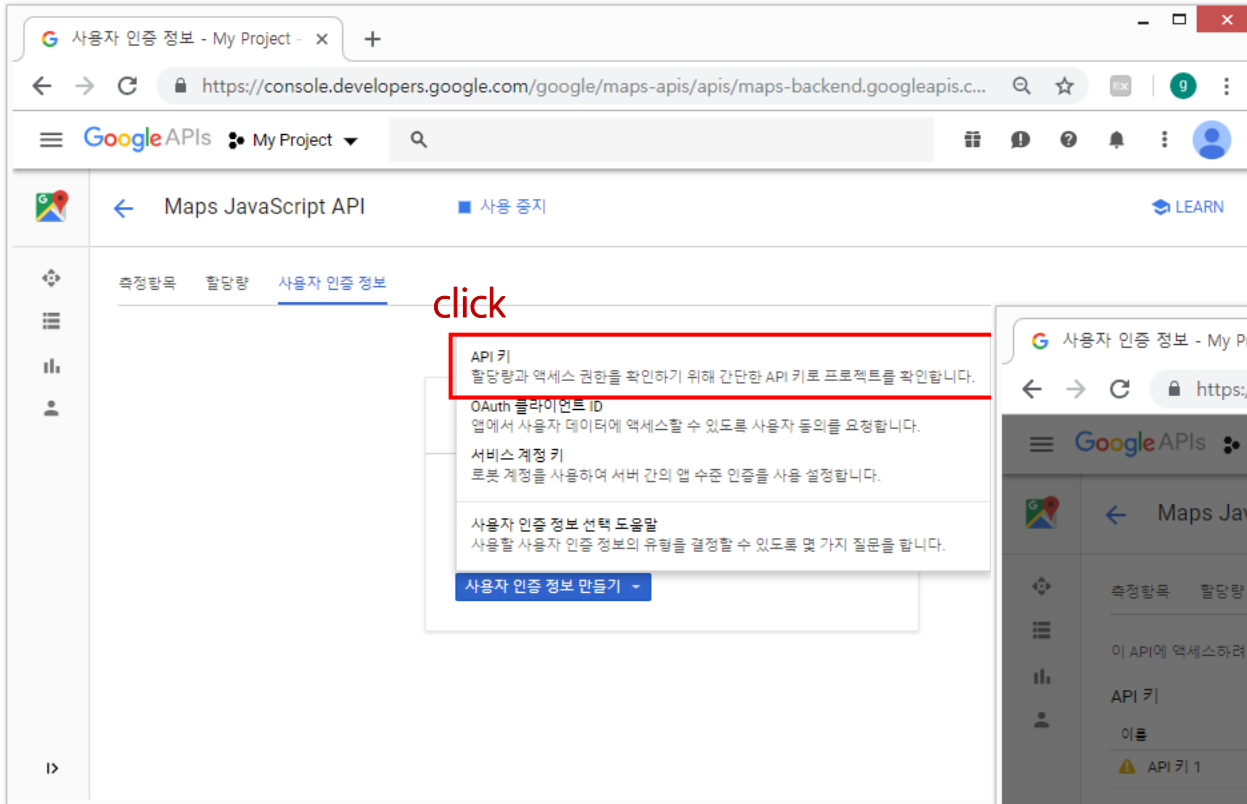
Maps JavaScript API 사용 설정



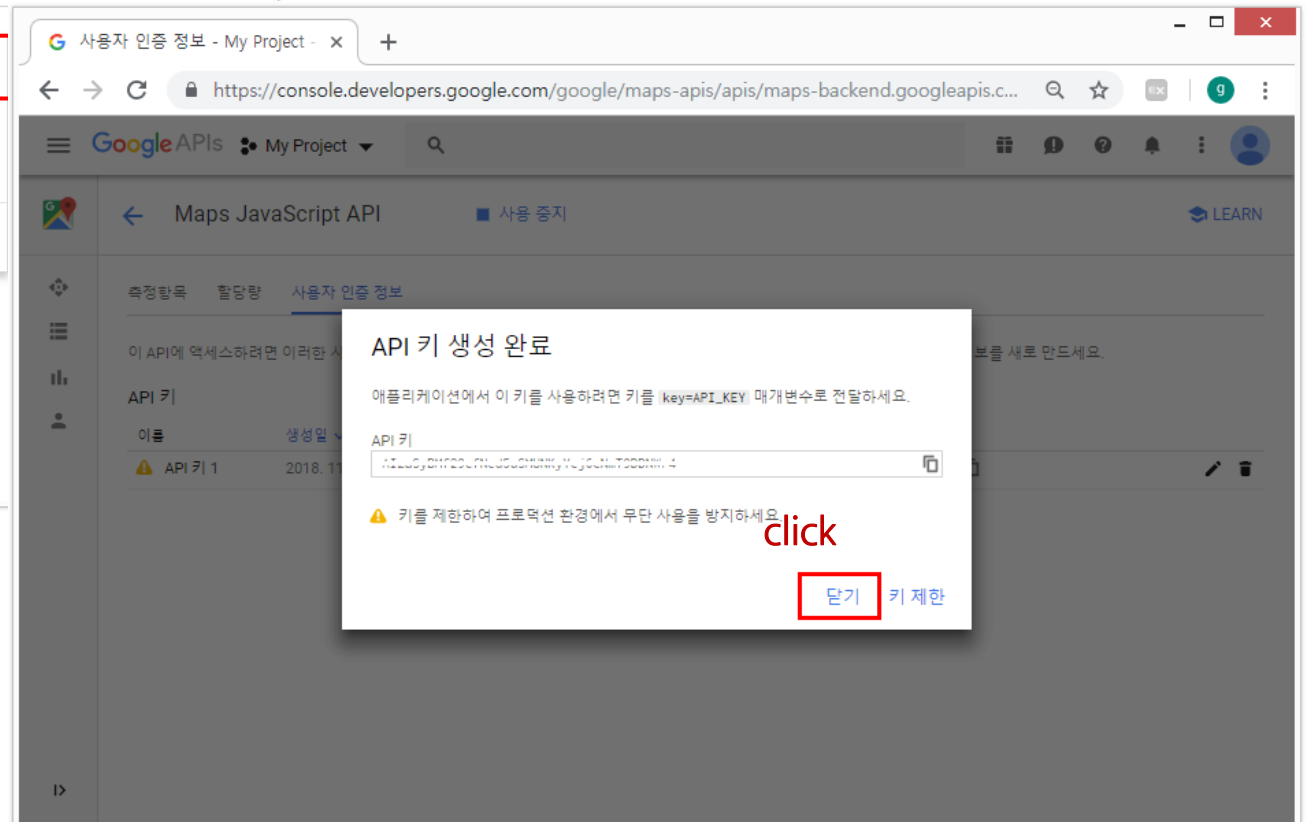
사용자 인증 정보 만들기



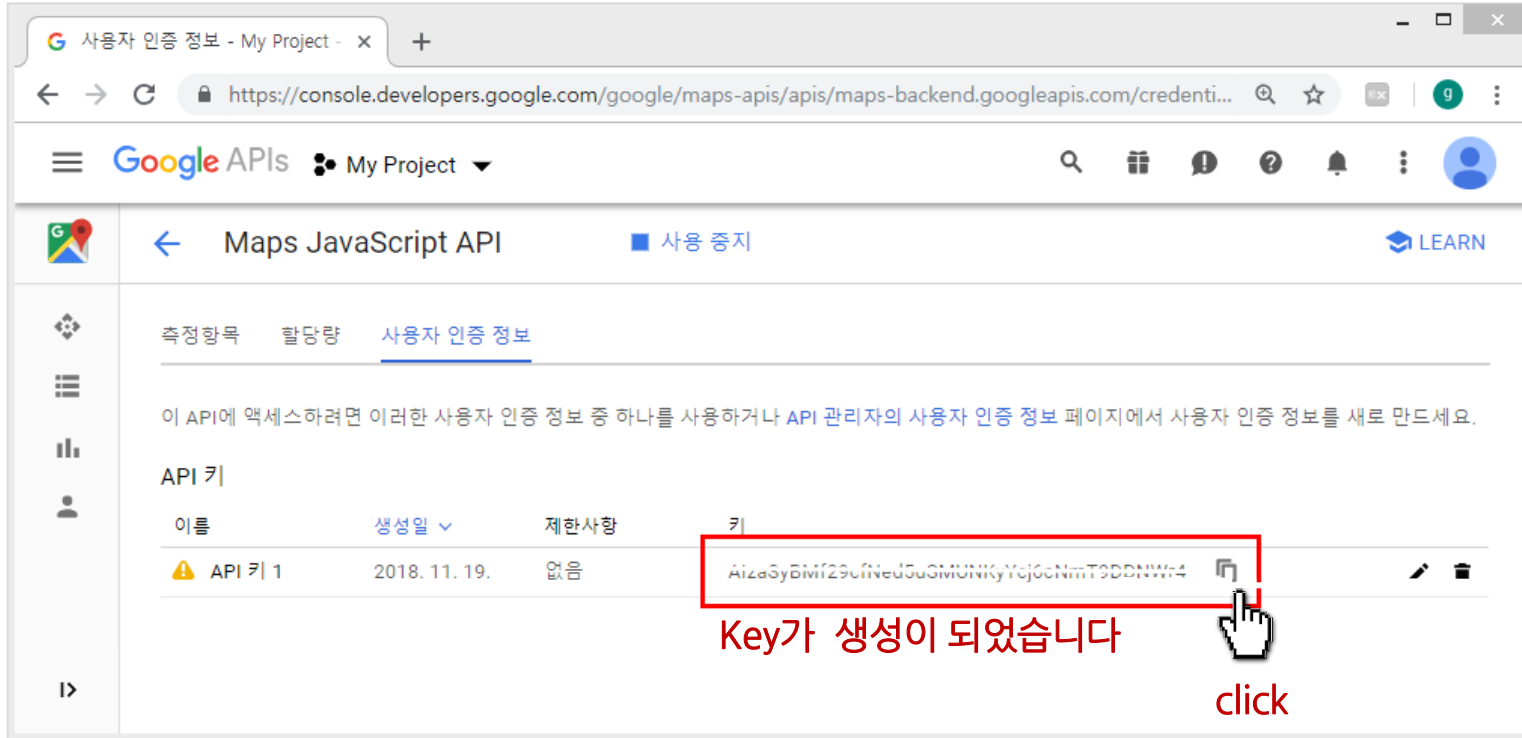
API 키 만들기



API키 생성 완료



사용자 API 키 생성



사용자 인증 정보 - My Project - x

https://console.developers.google.com/google/maps-apis/apis/maps-backend.googleapis.com/credenti...

Google APIs My Project

Maps JavaScript API 사용 중지 LEARN

측정 항목 할당량 사용자 인증 정보

이 API에 액세스하려면 이러한 사용자 인증 정보 중 하나를 사용하거나 API 관리자의 사용자 인증 정보 페이지에서 사용자 인증 정보를 새로 만드세요.

API 키

이름	생성일	제한사항	키
API 키 1	2018. 11. 19.	없음	AizaSyBmf29cfNeu5uGMUNKyYcj6cNnT9DDNw4

Key가 생성이 되었습니다

click

❖ Google Map API 연동

- Google Map API는 위치 기반 지도를 서비스하기 위한 지도 기반 앱을 만든다.

Geolocation API 속성

속성	설명
center	위도와 경도 설정한다.
zoom	지도에서 표시의 크기를 나타낸다(설정 범위 0~22).
mapTypeId	지도 유형이다. google.maps.MapTypeId.ROADMAP: 도로 지도 뷰를 표시 (기본 지도 유형) google.maps.MapTypeId.SATELLITE: 위성 지도 표시를 표시 google.maps.MapTypeId.HYBRID: 일반 뷰와 위성 뷰를 섞어서 표시 google.maps.MapTypeId.TERRAIN: 지형 정보를 기반으로 실제 지도를 표시
google.maps.LatLng	위도와 경도를 나타낸다
google.maps.Map	지도를 나타낸다.
google.maps.Marker	마커를 표시한다. setMap() 메서드를 호출하여 마커를 추가한다.

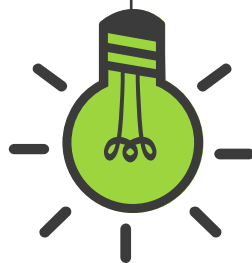
[ex03_65.html] 지오로케이션



[ex03_66.html] 마커 드래그와 말 풍선

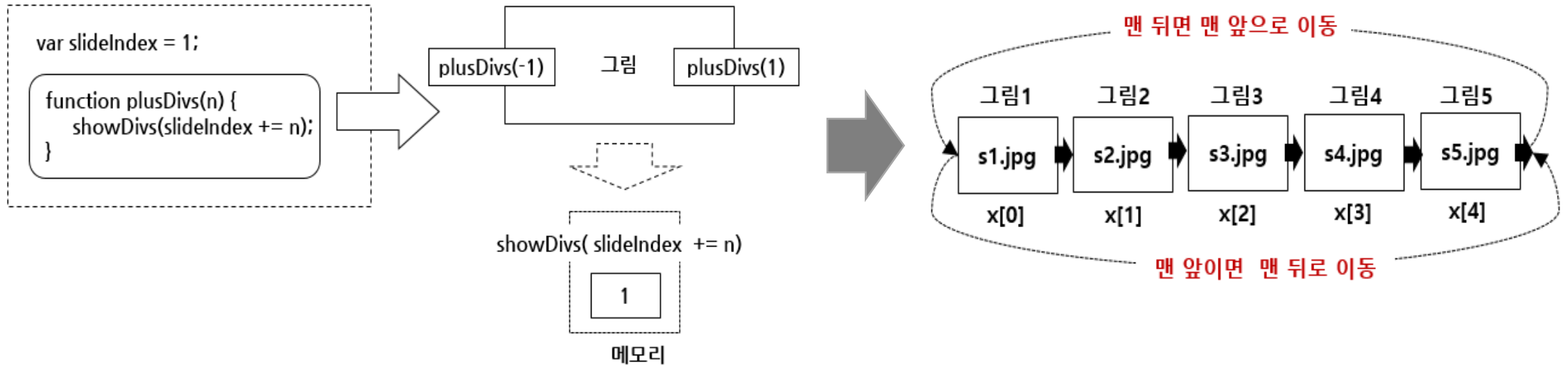


04 실습 3



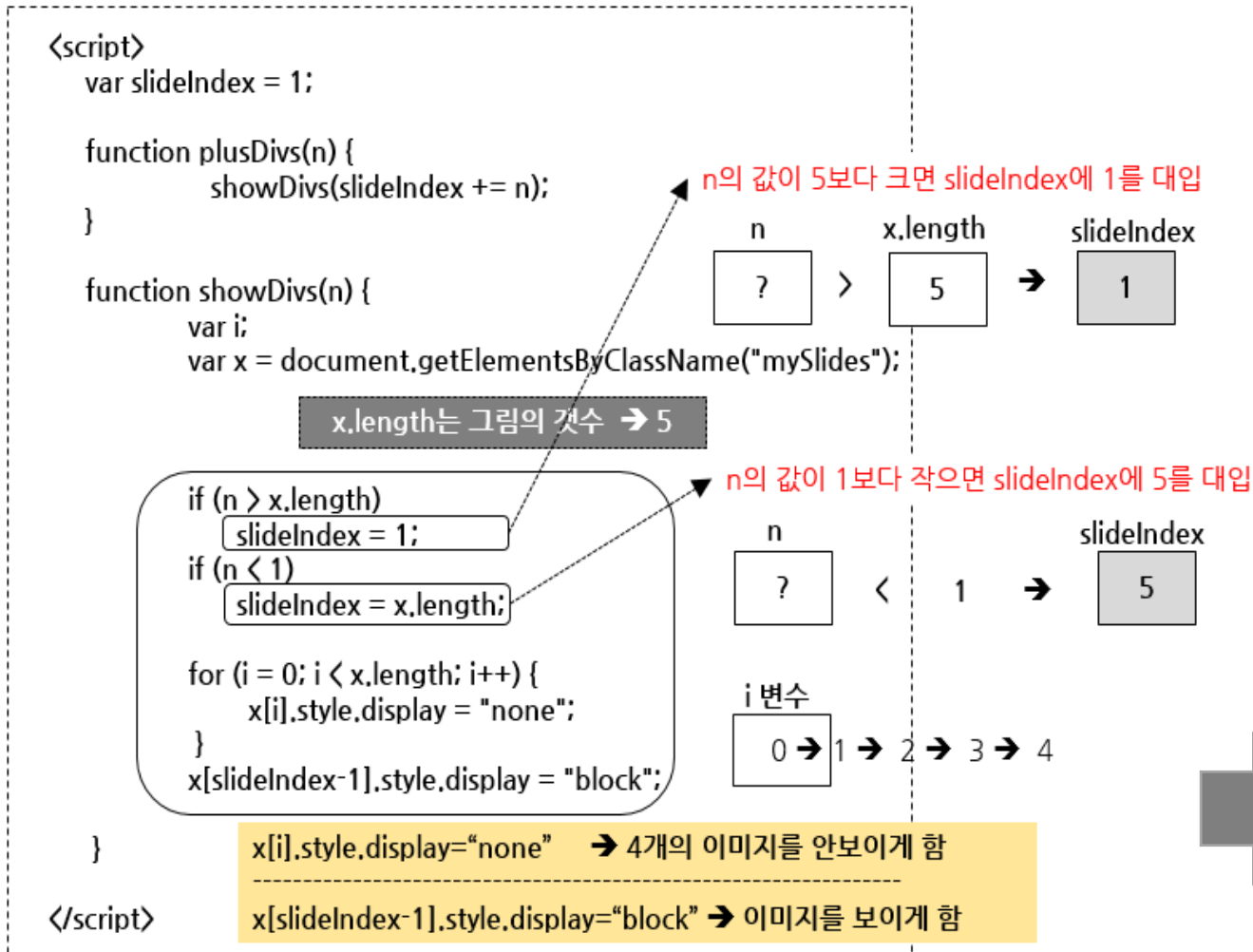
3.1.1 롤링 데모 화면 만들기

- 여러 장의 사진을 클릭 이벤트로 한 장씩 넘기는 프로그램을 만들어 본다.



양쪽 버튼 클릭 시 호출하는 함수 알고리즘

슬라이드 이동 순서



[demo.html] 수동 슬라이드

Manual Slides

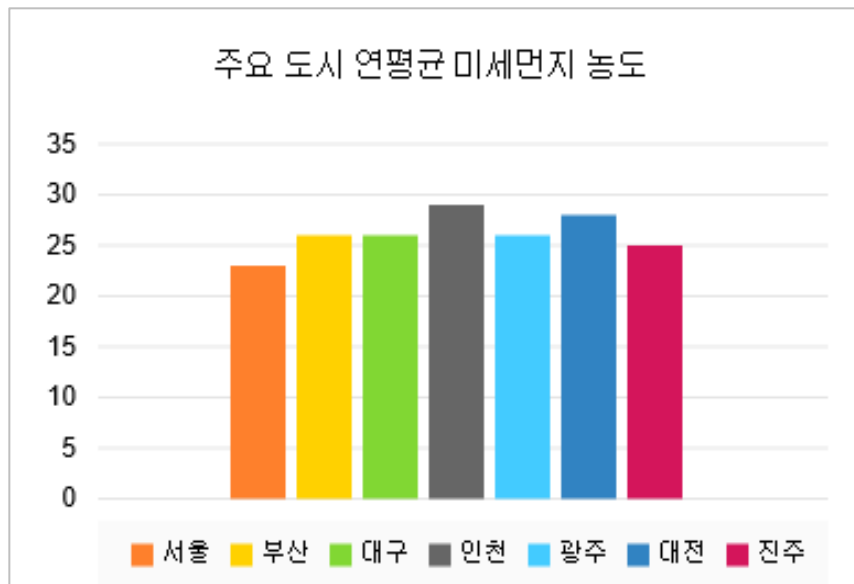


수동 슬라이드 구동 원리

3.1.2 막대 그래프 그리기

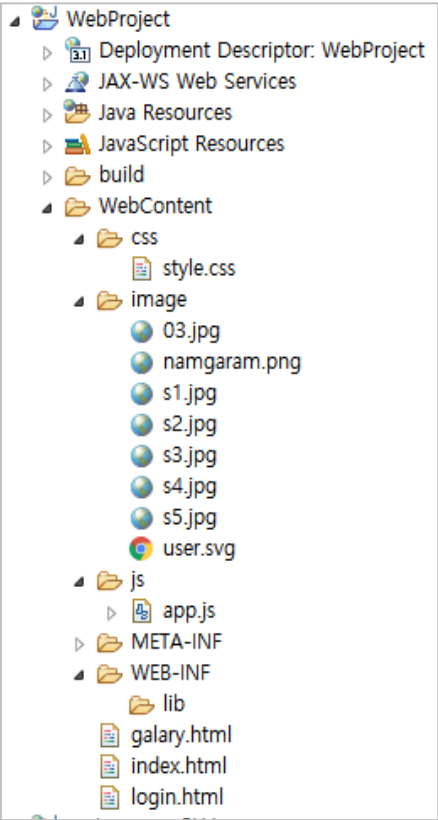
- Canvas API를 사용해서 그래프를 만들어 본다.
- 그래프는 빅데이터에서 데이터 시각화를 위해 많이 사용하는 기술이다.

[bargraph.html] 주요 도시 미세먼지 그래프 / [app.js]



3.4.3 프로젝트

프로젝트 설계도



화면 설계

- 파일은 index.html, login.html, gallery.html로 총 3개이다.

화면	파일명	설명
HOME	index.html	HOME 화면
로그인	login.html	로그인 화면
갤러리/갤러리 보기	gallery.html	갤러리 화면/이미지 슬라이드 보기

header

nav

section

footer

페이지 제작

localhost:8080/WebProject/index.html

Home Sitemap

Home 갤러리 게시판 멀티미디어 상품구매 찾아오시는길 회원가입 로그인

그래프 1

주요 도시 연평균 미세먼지 농도



서울 부산 대구 인천 광주 대전 울산

이미지



그래프 2

North South



미세먼지 농도

예보구간	중음	보통	악간나쁨	나쁨	테이블	
예측 농도	0~30	31~80	81~120	121~200	201~300	301~
노상차량			장시간 실외 활동 가급적 자제	무리한 실외활동 자제요청 (특히 호흡기, 심질환자, 노약자)	실외활동 제한	실내 생활
보행자				장시간 무리한 실외활동 자제	실외활동 자제	실외활동 자제

다운로드 개인정보처리방침 저작권지침및신고 이메일무단수집거부

Home : Location Log : Tag Log : Media Log : Guestbook : Admin : New Post

COPYRIGHT © 남가람복스's BLOG IS POWERED BY DAUM / DESIGNED BY TISTORY

페이지 제작

localhost:8080/WebProject/login.html

HomeSitemap

Home갤러리게시판멀티미디어상품구매찾아오시는길회원가입로그인

로그인

아이디

비밀 번호

로그인

소셜 로그인하기

페이스북 ID로 로그인

트위터 ID로 로그인

네이버 ID로 로그인

도움이 필요하신가요?

아이디 찾기

비밀번호 찾기

다운로드

개인정보처리방침

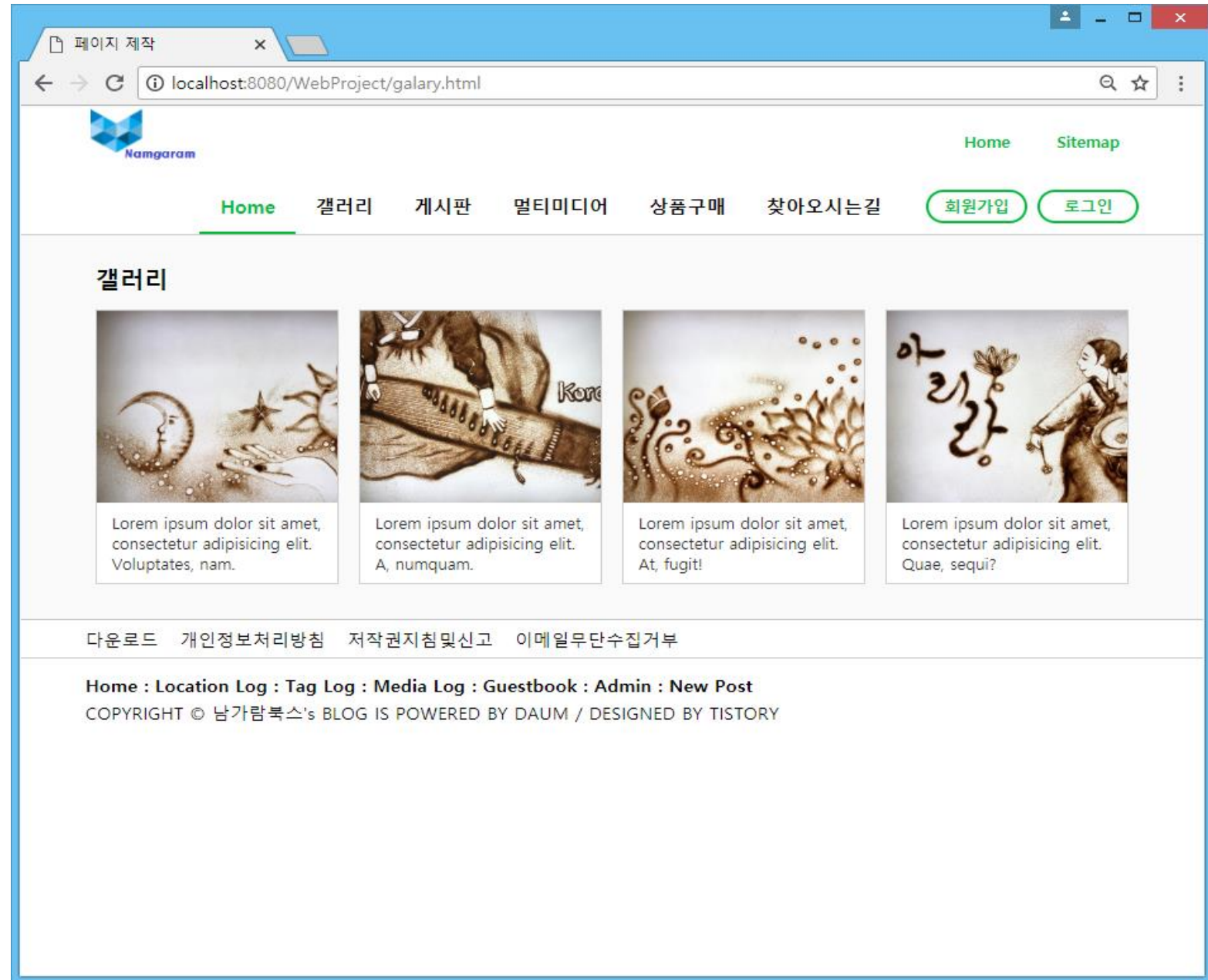
저작권지침및신고

이메일무단수집거부

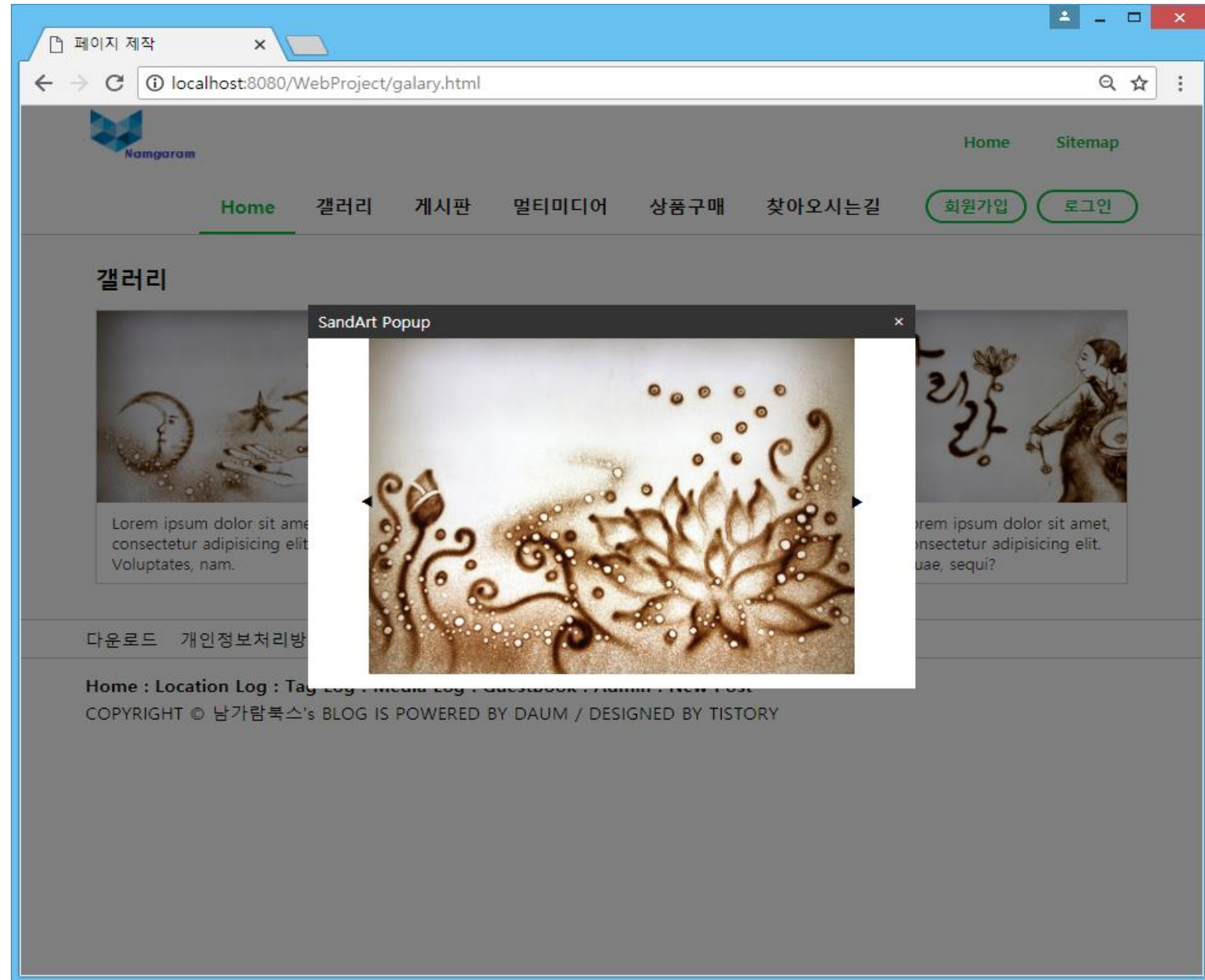
Home : Location Log : Tag Log : Media Log : Guestbook : Admin : New Post

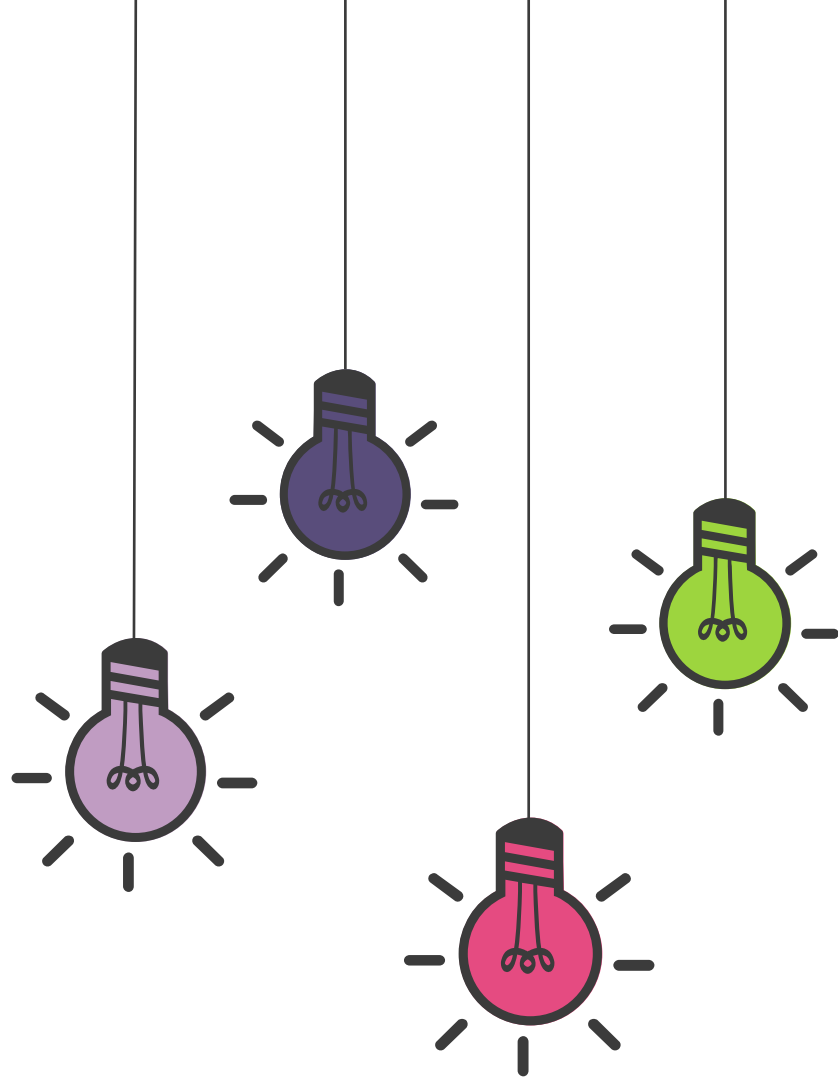
COPYRIGHT © 남가람복스's BLOG IS POWERED BY DAUM / DESIGNED BY TISTORY

localhost:8080/WebProject/login.html#



갤러리 보기





감사합니다

THANK YOU FOR WATCHING