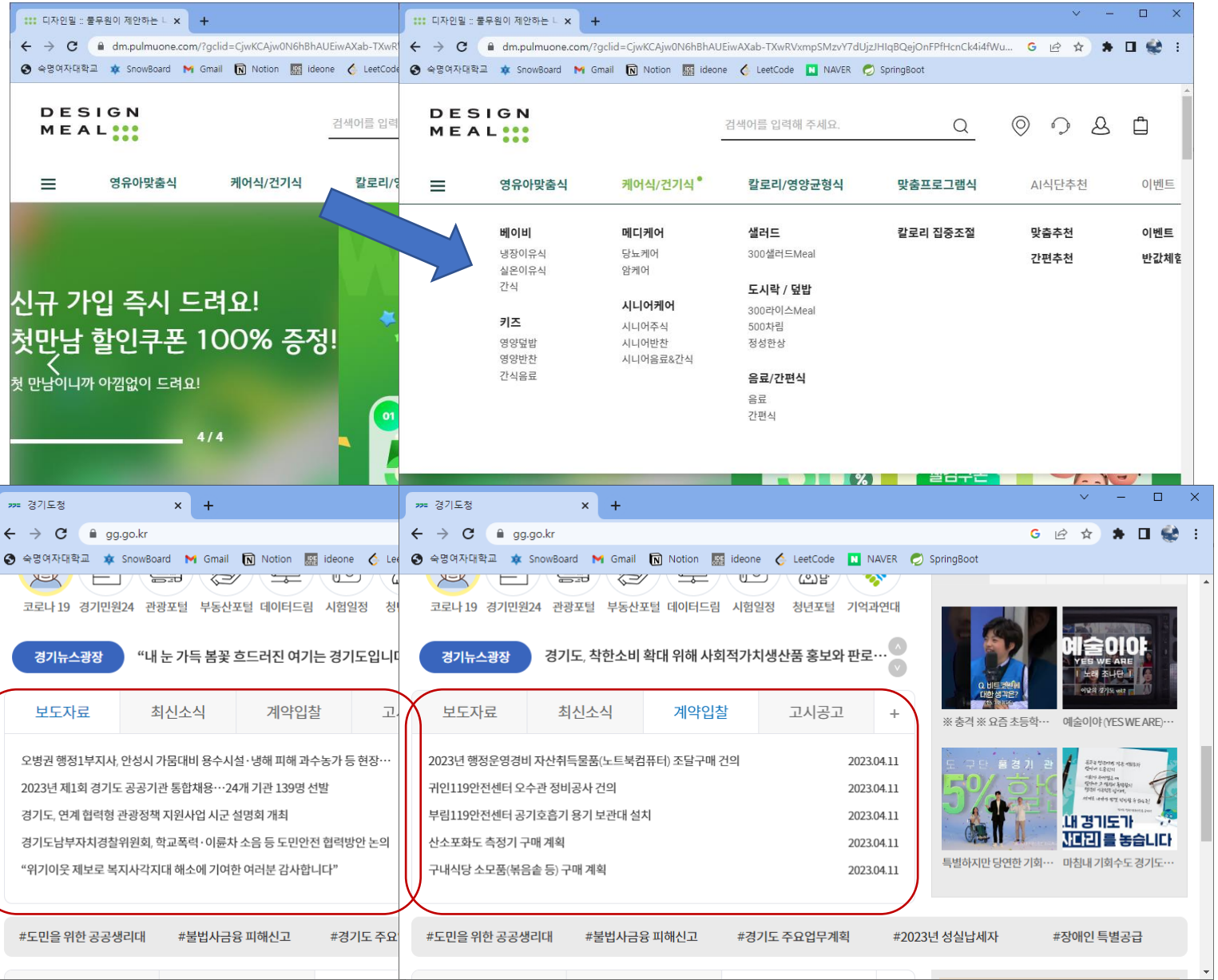
자바스크립트 기본 문법 1

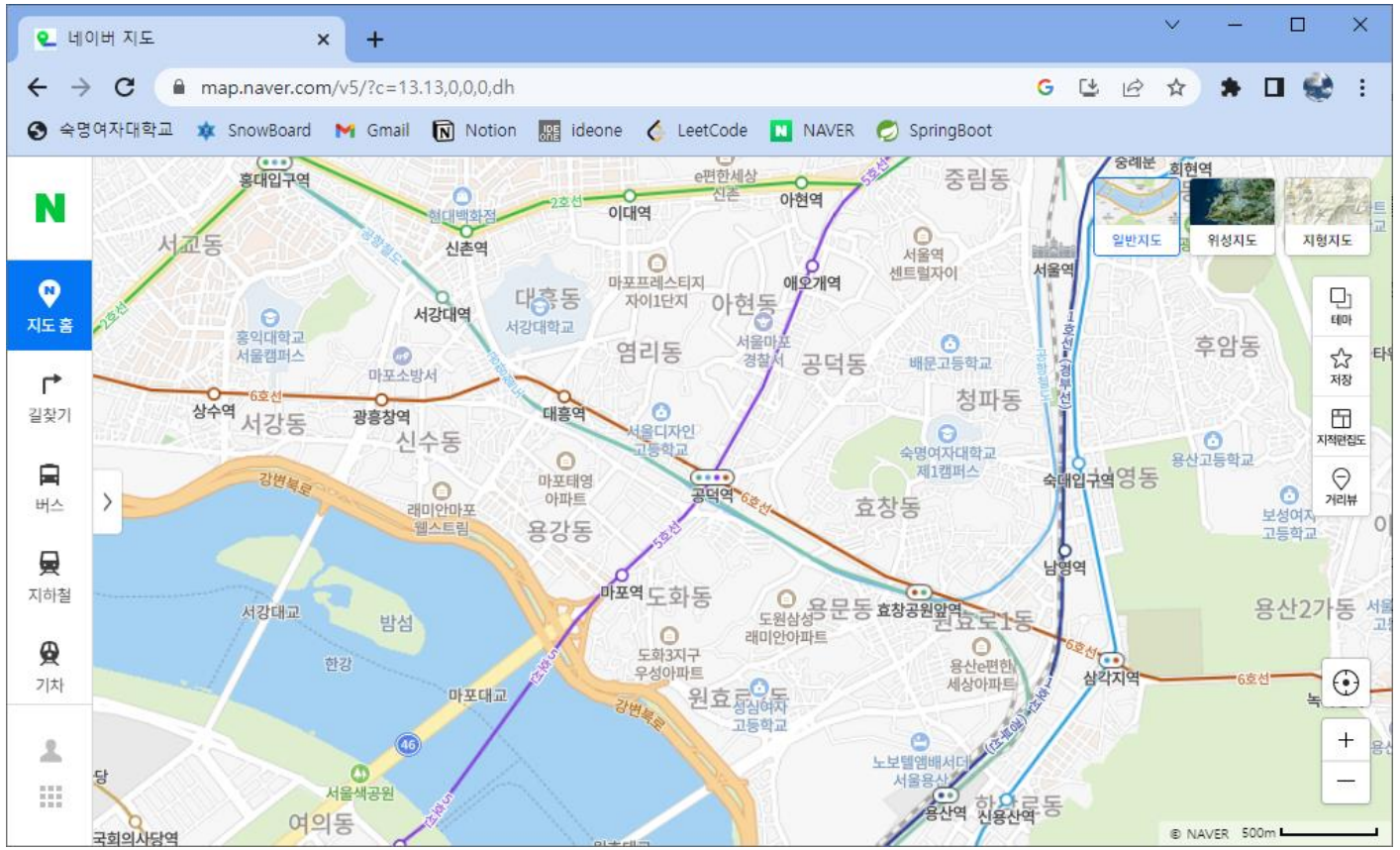
자바스크립트로 무엇을 할까

- 웹 요소 제어
  - 웹 요소를 가져와서 필요에 따라 스타일을 변경하거나 움직이게 할 수 있음
  - 웹 사이트 UI 부분에 많이 활용
    - 예) 마우스 포인터를 올렸을 때 펼쳐지는 메뉴
    - 한 화면에서 탭을 눌러 내용만 바꾸도록 하는 콘텐츠



자바스크립트로 무엇을 할까

- 웹 애플리케이션을 만듭니다.
  - 최근의 웹 사이트는 사용자와 실시간으로 정보를 주고 받으며 애플리케이션처럼 동작
  - 예) 온라인 지도의 길찾기 서비스, 데이터 시각화 서비스, 공개된 API를 활용한 다양한 서비스



자바스크립트로 무엇을 할까

다양한 라이브러리를 사용할 수 있습니다

- 웹을 중심으로 하는 서비스가 늘어나면서 브라우저에서 처리해야 할 일이 늘어남 → 라이브러리와 프레임워크가 계속 등장
- 예) 시각화를 위한 d3.js, 머신러닝을 위한 tensorflow.js, DOM 조작을 위한 jQuery 등
- 예) 웹 애플리케이션 개발을 위한 React, Angular, Vue 등

서버를 구성하고 서버용 프로그램을 만들 수 있습니다

- node.js : 프런트엔드 개발에 사용하던 자바스크립트를 백엔드 개발에서 사용할 수 있게 만든 프레임워크

웹 문서 안에 자바스크립트 작성

- <script> 태그와 </script> 태그 사이에 자바스크립트 소스 작성

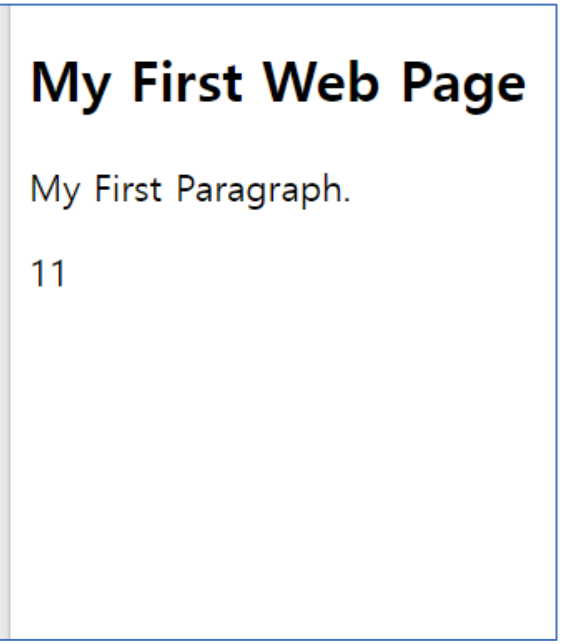
```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

- 웹 문서 안의 <head> 태그 또는 <body> 태그 안에 위치
  - 주로 </body> 태그 앞에 작성 권장
- 외부 파일로 작성
  - 확장자 .js
  - 외부 스크립트 사용 `<script src="myScript.js"></script>`

자바스크립트 출력

- HTML 요소 내부에 내용 작성
  - innerHTML

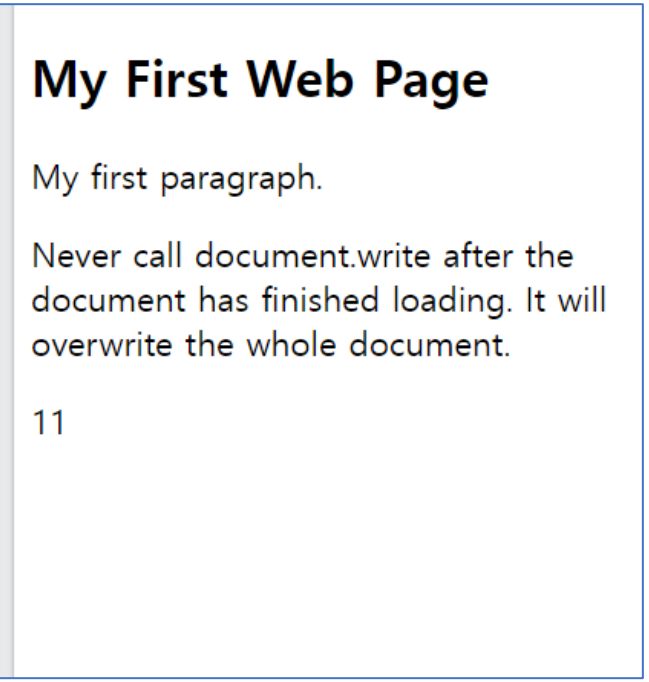
```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```



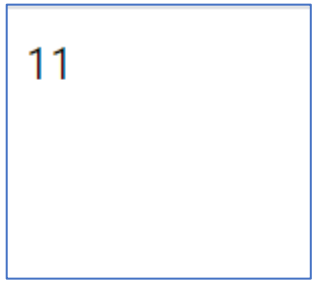
자바스크립트 출력 (계속)

- HTML 출력
  - document.write()

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<p>Never call document.write after the
document has finished loading.
It will overwrite the whole document.
</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```



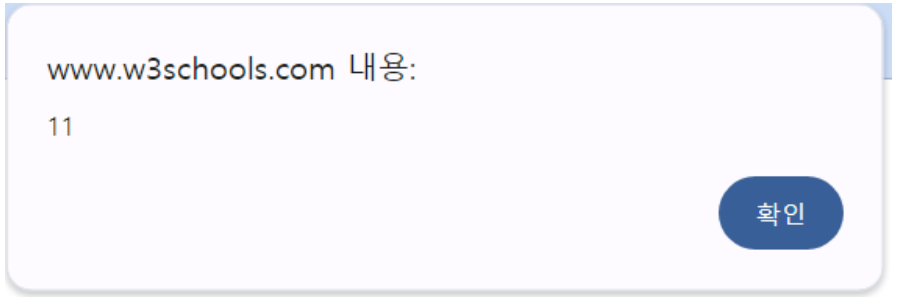

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">
Try it</button>

</body>
</html>
```



자바스크립트 출력 (계속)

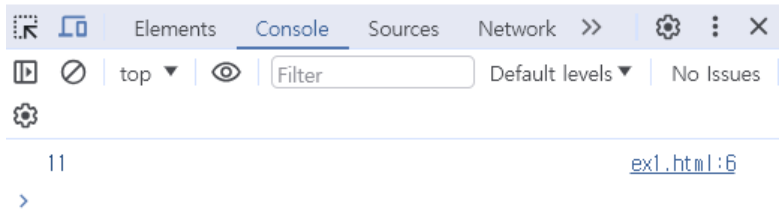
- alert 창으로 출력
  - window.alert()
    - window 생략 가능

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```



- 브라우저 콘솔에 출력
  - console.log()

```
<!DOCTYPE html>
<html>
<body>

<h2>Activate Debugging</h2>

<p>F12 on your keyboard will activate debugging.</p>
<p>Then select "Console" in the debugger menu.</p>
<p>Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

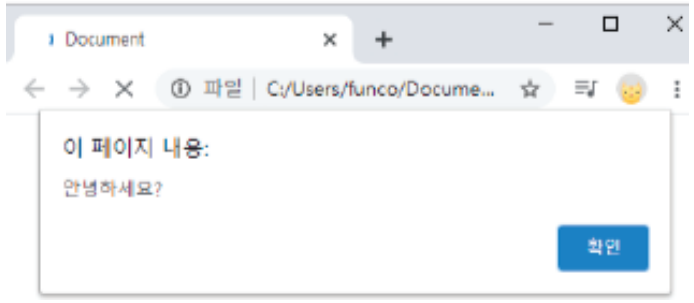


기본 입출력 방법_창

▪ 알림 창 출력

기본형 `alert(메시지)`

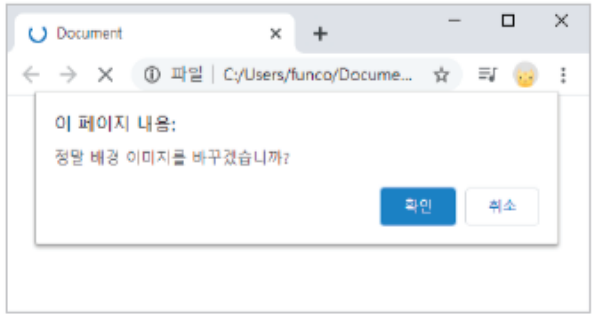
- '확인' 버튼이 있는 메시지 창 표시



▪ 확인 창 출력

기본형 `confirm(메시지)`

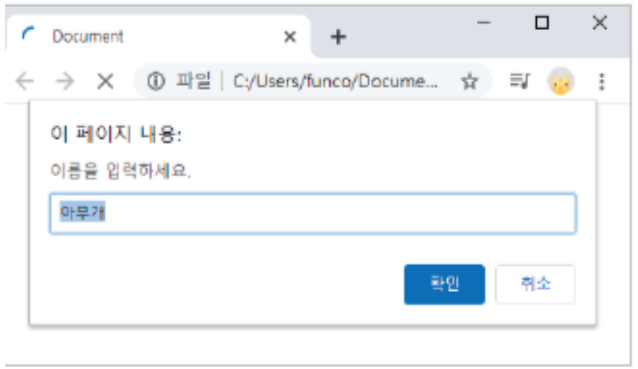
- '확인' 과 '취소' 버튼이 있는 창 표시
- 클릭하는 버튼에 따라 프로그램 동작



▪ 프롬프트 창에서 입력받기

- 텍스트 필드가 있는 창 표시
- 사용자 입력 값을 가져와 프로그램에서 사용

기본형 `prompt(메시지) 또는 prompt(메시지, 기본값)`



자바스크립트 출력 (계속)

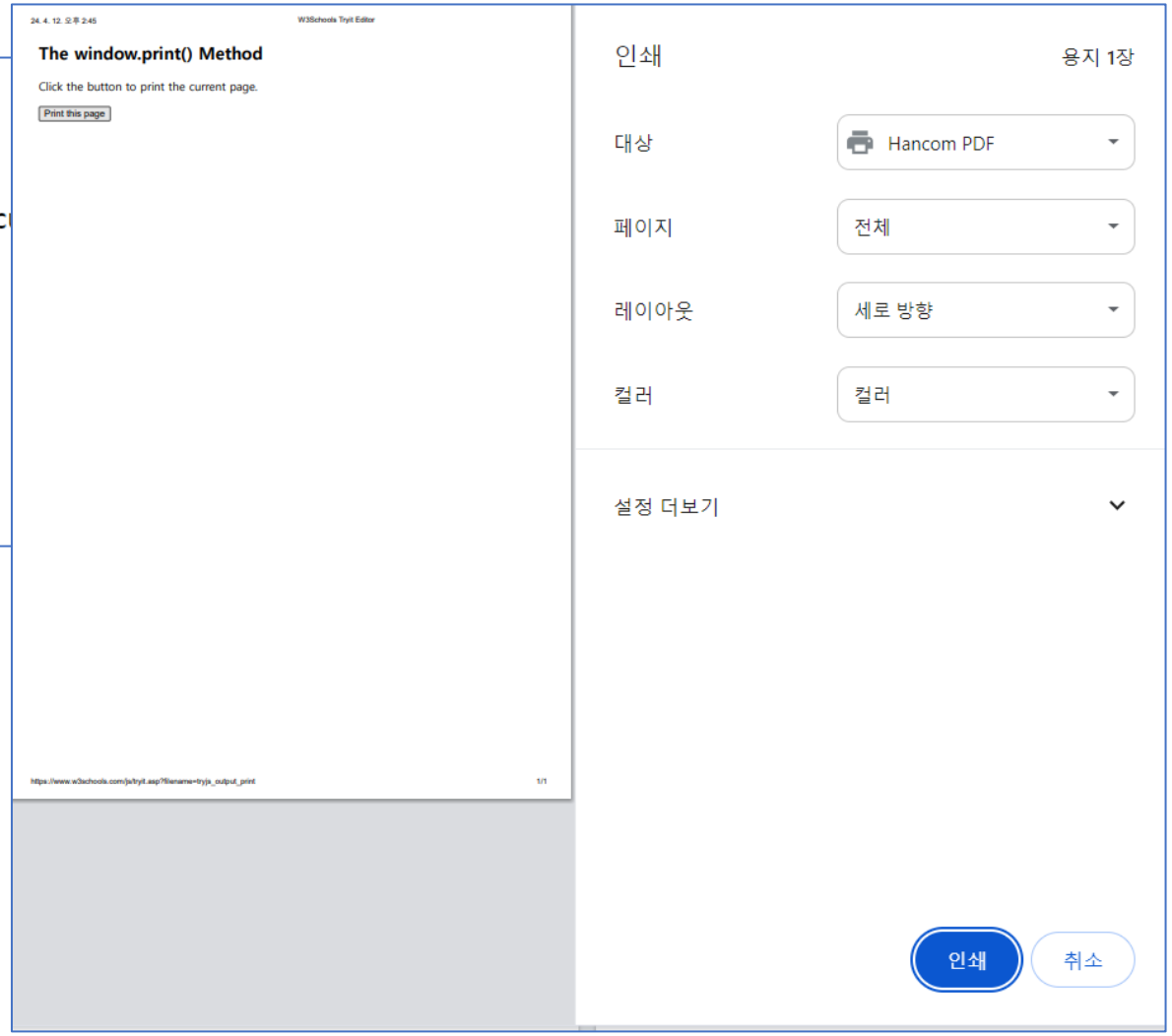
- 인쇄

```
<!DOCTYPE html>
<html>
<body>

<h2>The window.print() Method</h2>

<p>Click the button to print the current
page.</p>

<button onclick="window.print()">Print
this page</button>

</body>
</html>
```

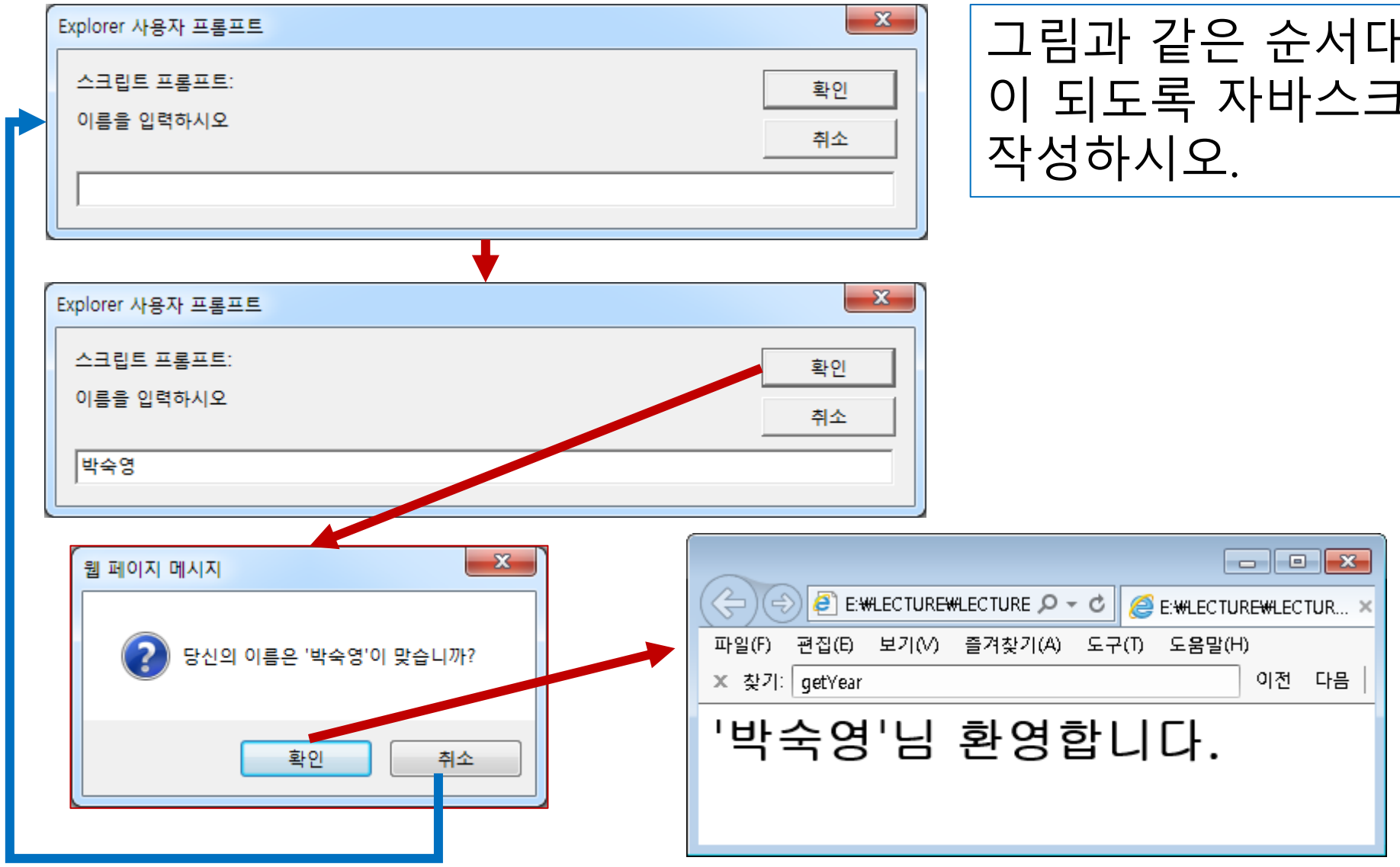
The window.print()

Click the button to print the c

Print this page

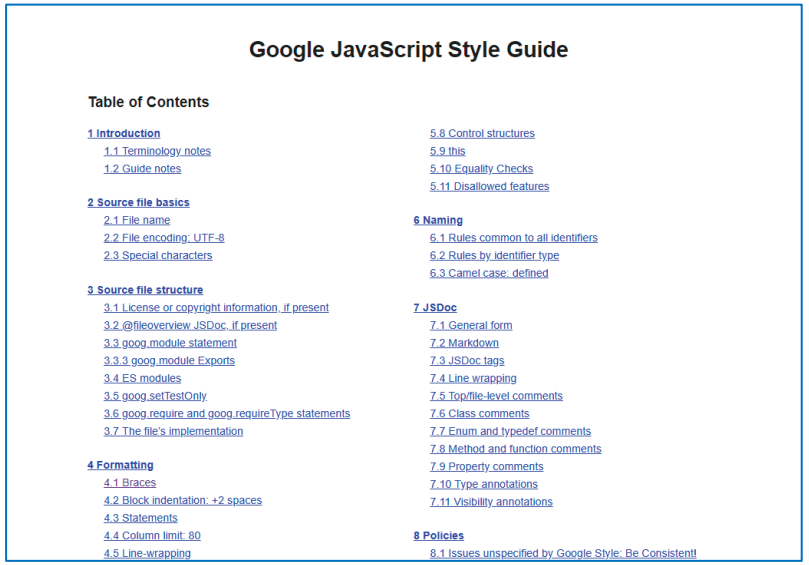


실습



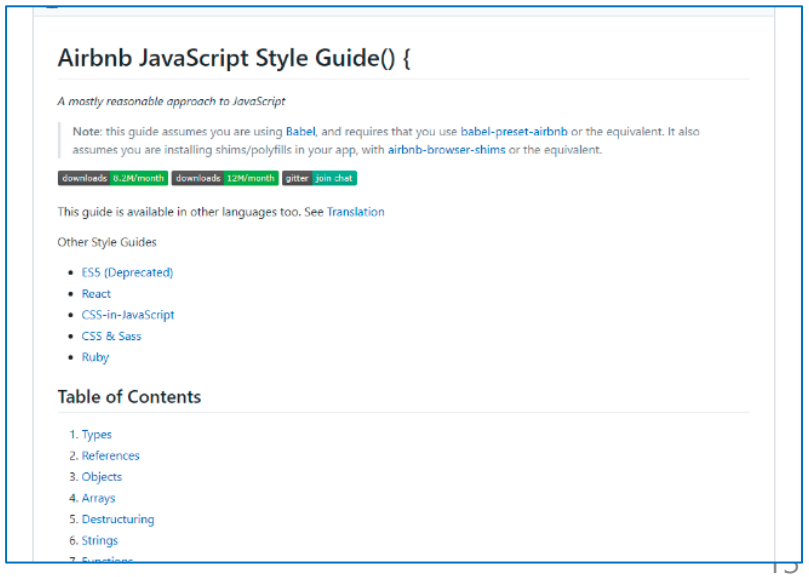
그림과 같은 순서대로 진행이 되도록 자바스크립트를 작성하시오.

- 코딩 규칙을 '스타일 가이드', '코딩 컨벤션', '코딩 스타일', '표준 스타일 ' 등으로 부름
- 코딩 규칙이 왜 필요할까?
  - 자바스크립트는 다른 프로그래밍 언어에 비해 데이터 유형이 유연해서 오류 발생이 잦다
  - 오픈 소스에 기여하거나 누군가와 공유할 소스라면 더욱 깔끔한 소스가 중요하다
  - 팀 프로젝트를 진행한다면 통일된 코딩 규칙이 필요하다
  - 코딩 규칙에 따라 작성된 웹 사이트는 유지 보수도 수월하고 그만큼 비용도 줄어든다
- 자바스크립트 스타일 가이드
  - 구글 자바스크립트 스타일 가이드 (google.github.io/styleguide/jsguide.html) 또는
  - 에어비앤비 자바스크립트 스타일 가이드(github.com/airbnb/javascript) 참고
  - 회사 프로젝트의 경우 팀 내에서 상의해서 결정



Google JavaScript Style Guide

Table of Contents



Airbnb JavaScript Style Guide() {

A mostly reasonable approach to JavaScript

Note: this guide assumes you are using Babel, and requires that you use babel-preset-airbnb or the equivalent. It also assumes you are installing shims/polyfills in your app, with airbnb-browser-shims or the equivalent.

downloads 8.2M/month downloads 12M/month gitter join chat

This guide is available in other languages too. See Translation

Other Style Guides

- ES5 (Deprecated)
- React
- CSS-in-JavaScript
- CSS & Sass
- Ruby

Table of Contents

자바스크립트 용어

- 식(expression)
  - 값을 만들어 낼 수 있다면 모두 식이 될 수 있다
  - 식은 변수에 저장된다
- 문(statement)
  - 문의 끝에는 세미콜론(;)을 붙여서 구분하는게 좋다
  - 넓은 의미에서 식이나 값을 포함할 수 있다
- 주석(Comments)
  - 한 줄 주석
    - //
  - 여러줄 주석
    - /* */

자바스크립트 키워든

Keyword	Description
var	Declares a variable
let	Declares a block variable
const	Declares a block constant
if	Marks a block of statements to be executed on a condition
switch	Marks a block of statements to be executed in different cases
for	Marks a block of statements to be executed in a loop
function	Declares a function
return	Exits a function
try	Implements error handling to a block of statements

변수 알아보기

변수란

- 변수(variable) : 값이 여러 번 달라질 수 있는 데이터
- 상수(constant) : 값을 한번 지정하면 바뀌지 않는 데이터

변수 선언의 규칙

- 변수 이름
  - 영어 문자, 언더스코어(_), 숫자를 사용한다
  - 첫 글자는 영문자, _기호, $기호를 사용한다. (숫자로 시작 불가)
  - 띄어쓰기나 기호는 허용하지 않는다
    예) now, _now, now25 (사용할 수 있음)
    예) 25now, now 25, *now (사용할 수 없음)
- 영어 대소문자를 구별하며 예약어는 변수 이름으로 사용할 수 없다
- ㄴ변수 이름은 의미있게 작성한다

변수 선언 키워드

- var
  - 1995~2015
  - 이전 브라우저를 지원하려면 사용
- let
  - 변수 선언
  - 2015~
- const
  - 상수 선언
  - 2015~

- 사용 예제
  - var sum=100;
  - var grade; // undefined
  - let total; // undefined
  - let name="sook";
  - const pi = 3.14;
  - const test = 5;

let과 var 비교

- let
  - block scope
  - 사용하기 전에 선언되어야 한다.
  - 동일 영역 내에 재선언 불가
- var
  - function scope
  - 동일 영역 내에 재선언 가능
    - 문제 발생 가능성 있음

	Scope	Redeclare	Reassign	Hoisted	Binds this
var	No	Yes	Yes	Yes	Yes
let	Yes	No	Yes	No	No
const	Yes	No	No	No	No

const 사용 예제

- 선언할 때 바로 값이 할당되어야 함
- 배열, 객체, 함수 등을 선언할 때 const 사용
  - 해당 배열, 객체 자체는 변경 못하지만
  - 상수 배열의 요소 변경 가능
  - 상수 객체의 속성 변경 가능

```
// You can create a constant array:
const cars = ["Saab", "Volvo", "BMW"];

// You can change an element:
cars[0] = "Toyota";

// You can add an element:
cars.push("Audi");
```

```
// You can create a const object:
const car = {type:"Fiat", model:"500", color:"white"};

// You can change a property:
car.color = "red";

// You can add a property:
car.owner = "Johnson";
```

산술 연산자

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

비교 연산자/논리 연산자/타입 연산자

▪ 비교 연산자

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

▪ 논리 연산자

Operator	Description
&&	logical and
||	logical or
!	logical not

▪ 타입 연산자

Operator	Description
typeof	Returns the type of a variable
instanceof	Returns true if an object is an instance of an object type

비트 연산자/연결 연산자

▪ 비트 연산자

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
|	OR	5 | 1	0101 | 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	left shift	5 << 1	0101 << 1	1010	10
>>	right shift	5 >> 1	0101 >> 1	0010	2
>>>	unsigned right shift	5 >>> 1	0101 >>> 1	0010	2

▪ 연결 연산자

- 둘 이상의 문자열을 합쳐서 하나의 문자열로 만드는 연산자
  - '+' 기호 사용

자료형

- 숫자
  - 정수 : 소수점 없는 숫자
  - 실수 : 소수점이 있는 숫자
- 문자열(string)
  - 작은따옴표(' ')나 큰따옴표(" ")로 묶은 데이터
- 논리형(boolean)
  - 참true이나 거짓false의 값을 표현하는 자료형. 불린 유형이라고도 함.
  - 조건을 확인해서 조건이 맞으면 true, 맞지 않으면 false라는 결괏값 출력
- undefined 유형
  - 자료형이 정의되지 않았을 때의 데이터 상태
  - 변수 선언만 하고 값이 할당되지 않은 자료형
- null 유형
  - 데이터 값이 유효하지 않은 상태
  - 변수에 할당된 값이 유효하지 않다는 의미
- 배열(array)
  - 하나의 변수에 여러개의 값을 저장
- 객체(object)
  - 함수와 속성을 함께 포함

조건문(if)

- if /if-else

```
if(조건){
    조건 결괏값이 true일 때 실행할 명령
}
```

```
if(조건){
    조건 결괏값이 true일 때 실행할 명령
}
else{
    조건 결괏값이 false일 때 실행할 명령
}
```

- 조건 연산자

```
(조건) ? true일 때 실행할 명령 : false일 때 실행할 명령
```

조건문(switch)

```
switch(조건)
{
    case 값1:
        조건이 값1일 때 실행할 명령
        break;
    case 값2:
        조건이 값2일 때 실행할 명령
        break;
    default:
        조건이 위의 case에 해당하지 않을 때 실행할 명령
        break;
}
```

자리 배치도 만들기

seat-1.html - WebWorkspace - Visual Studio Code

seat-1.html

doit_sources > 14 > seat-1.html > ...

```
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>자리 배치도</title>
7  </head>
8  <body>
9    <h1>자리 배치도</h1>
10   <script>
11     var memNum = prompt("입장객은 몇 명인가요?");  // 전체 입장객
12     var colNum = prompt("한 줄에 몇 명씩 앉습니까?");  // 한 줄에 앉을 사람
13
14     if (memNum % colNum === 0)
15       rowNum = parseInt(memNum / colNum);
16     else
17       rowNum = parseInt(memNum / colNum) + 1;
18
19     document.write("모두 " + rowNum + "개의 줄이 필요합니다.");
20   </script>
21 </body>
22 </html>
```

main* 0 0 줄 1, 열 1 공백: 2 UTF-8 with BOM CRLF HTML

127.0.0.1:5500/doit_s...

127.0.0.1:5500 내용:

입장객은 몇 명인가요?

75

확인 취소

자리 배치도

127.0.0.1:5500/doit_s...

127.0.0.1:5500 내용:

한 줄에 몇 명씩 앉습니까?

10

확인 취소

자리 배치도

127.0.0.1:5500/doit_s...

숙명여자대학교 SnowBoard Gmail Notion ideone

자리 배치도

모두 8개의 줄이 필요합니다.

반복문

- for 문

```
for(초깃값; 조건; 증가식){
    실행할 명령
}
```

- while 문

```
while(조건){
    실행할 명령
}
```

- do ~ while 문

```
do{
    실행할 명령
}while(조건);
```

- continue 문
  - 반복문의 다음 스텝으로 건너띔

- break 문
  - 반복문 하나 빠져나가는 문장

for 문

- for in Loop

```
for (key in object) {
  // code block to be executed
}
```

```
const numbers = [45, 4, 9, 16, 25];
let txt = "";
for (let x in numbers) {
  txt += numbers[x];
}
```

- Array.forEach()

```
const numbers = [45, 4, 9, 16, 25];

let txt = "";
numbers.forEach(myFunction);

function myFunction(value, index, array) {
  txt += value;
}
```

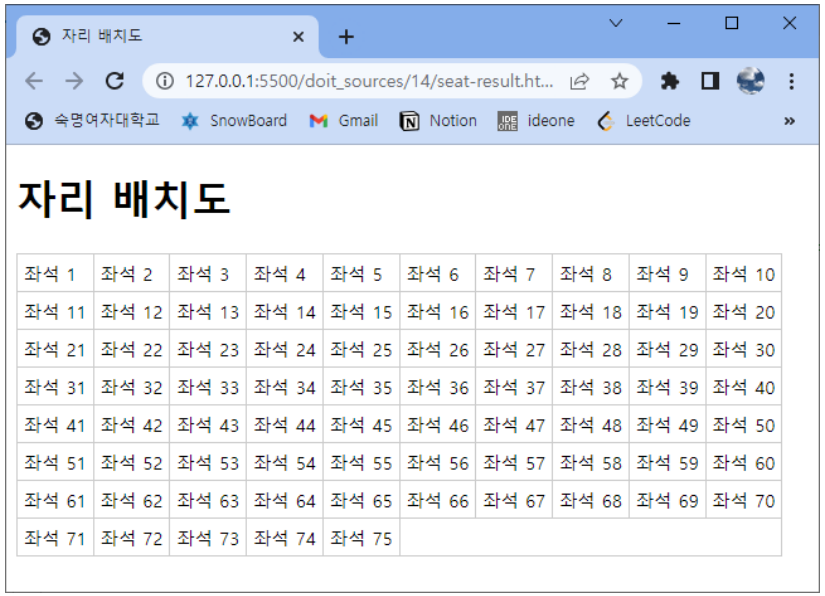
- for of Loop

```
for (variable of iterable) {
  // code block to be executed
}
```

```
const cars = ["BMW", "Volvo", "Mini"];

let text = "";
for (let x of cars) {
  text += x;
}
```

자리 배치도 만들기 2

```
 7   <style>
 8     table, td {
 9       border:1px solid #ccc;
10       border-collapse: collapse;
11     }
12     td {
13       padding:5px;
14       font-size:0.9em;
15     }
16   </style>
```




```
20  <script>
21    var i, j;
22    var memNum = prompt("입장객은 몇 명인가요?");  // 전체 입장객
23    var colNum = prompt("한 줄에 몇 명씩 앉습니까?");  // 한 줄에 앉을 사람
24
25    if (memNum % colNum == 0)
26      rowNum = parseInt(memNum / colNum);
27    else
28      rowNum = parseInt(memNum / colNum) + 1;
29
30    // document.write("모두 " + rowNum + "개의 줄이 필요합니다.");
31
32    document.write("<table>");
33    for (i = 0; i < rowNum; i++) {
34      document.write("<tr>");
35      for (j = 1; j <= colNum; j++) {
36        seatNo = i * colNum + j;     // 좌석 번호
37        if (seatNo > memNum) break;
38        document.write("<td> 좌석 " + seatNo + " </td>");
39      }
40      document.write("</tr>");
41    }
42    document.write("</table>");
43  </script>
```