

AhnLab SEM

2022. 3. 23 (chapter. 02)

- 자바스크립트 기본 -

인선미

2. 자료와 변수

- 1) 기본 자료형
- 2) 상수와 변수
- 3) 자료형 변환



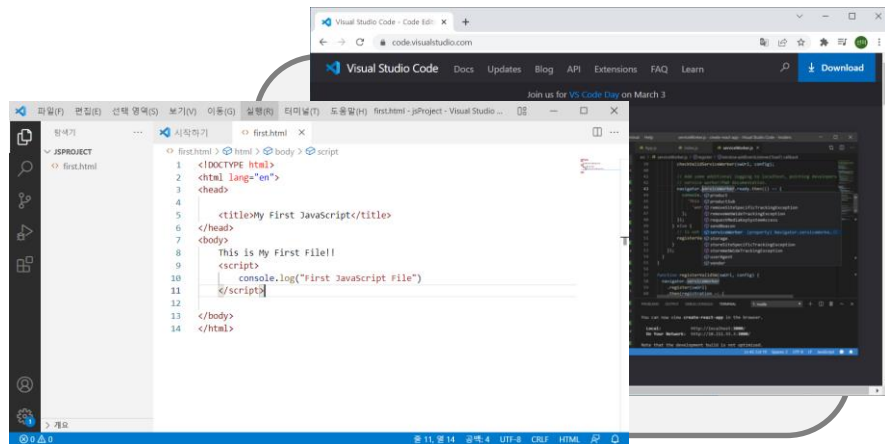
지난시간에...



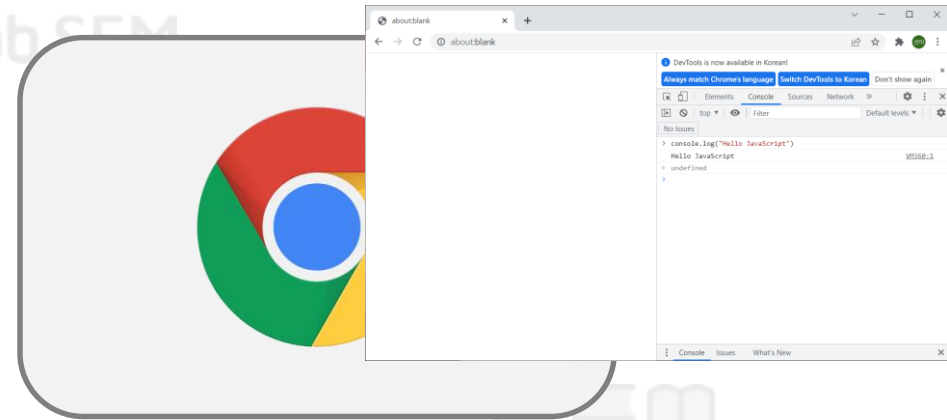
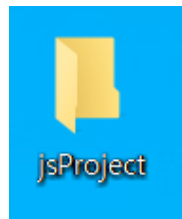
지난 시간에...

개발환경과 코드 실행

- 코드 실행기 : 구글 크롬 -> 개발자 도구 실행 (Ctrl + Shift + i 혹은 F12)
- 코드 에디터 : 비주얼 스튜디오 코드 Visual Studio Code



코드 에디터
(코드 작성)



코드 실행기
(웹 브라우저: 코드 실행)

지난 시간에...

주석

- 프로그램 코드를 설명할 때 사용하며 프로그램 진행에는 전혀 영향을 주지 않음
- HTML 태그 주석 : `<!-- -->`로 문자열을 감싸 생성
- 자바스크립트 주석

[방법1] `//`를 입력하는 것으로 **한 줄 주석**을 표현(`//` 뒤의 문장 : 실행되지 않음)

[방법2] `/* */`를 입력하여 **여러 줄 주석**을 표현(`/*`와 `*/` 사이에 있는 모든 문장 : 실행되지 않음)

지난 시간에...

출력

- 자바스크립트는 다른 프로그래밍 언어와 비교해서 출력 방법이 많고 복잡한 편
- 간단한 표현식 결과 확인하기
- 구글 크롬의 주소창에 **about:blank**를 입력해 빈 페이지로 들어가 단축키
- Ctrl + Shift+ I (알파벳 ‘아이’)를 눌러서 개발자 환경을 띄우기
- [Console] 탭을 클릭해 구글 크롬 개발자 도구에 진입. 값을 입력하면 곧바로 그 결과가 출력 됨

지난 시간에...

알림창 출력

- alert(메시지)
- alert("안녕하세요?");

확인창 출력

- confirm(메시지)
- let reply = confirm("정말 배경 이미지를 바꾸겠습니까?");

콘솔창 출력

- console.log(메시지)

지난 시간에...

프롬프트 창 입력

- prompt(메시지)
- prompt(메시지, 기본값)
- const name = prompt("이름을 입력하세요", "홍길동")

웹 브라우저 화면에 출력

- document.write(메시지)
- document.write("<h1>어서오세요</h1>")

지난 시간에...

자료(data)

- 프로그래밍에서 프로그램이 처리할 수 있는 모든 것

자료형(data type)

- 자료 형태에 따라 나눠 놓은 것
- 숫자(number), 문자열(string), 불(Boolean)

지난 시간에...



변수



- 데이터를 담을 수 있는 공간
- 변수에 값을 대입할 때 데이터 타입이 결정됨
- let 키워드를 사용해 변수 선언
- **let** 이름 = 값

```
> let pi = 3.141592
```

```
< undefined
```

```
> let r = 10
```

```
< undefined
```

```
> pi * r * r
```

```
< 314.1592
```

지난 시간에...



상수



- 항상 같은 수(값에 이름을 한 번 붙이면 값을 수정할 수 없음)
- `const` 키워드 사용
- **const** 이름 = 값

```
> const pi = 3.141592
```

```
< undefined
```

```
> pi
```

```
< 3.141592
```

```
> const r = 10
```

```
< undefined
```

```
> 2*pi*r
```

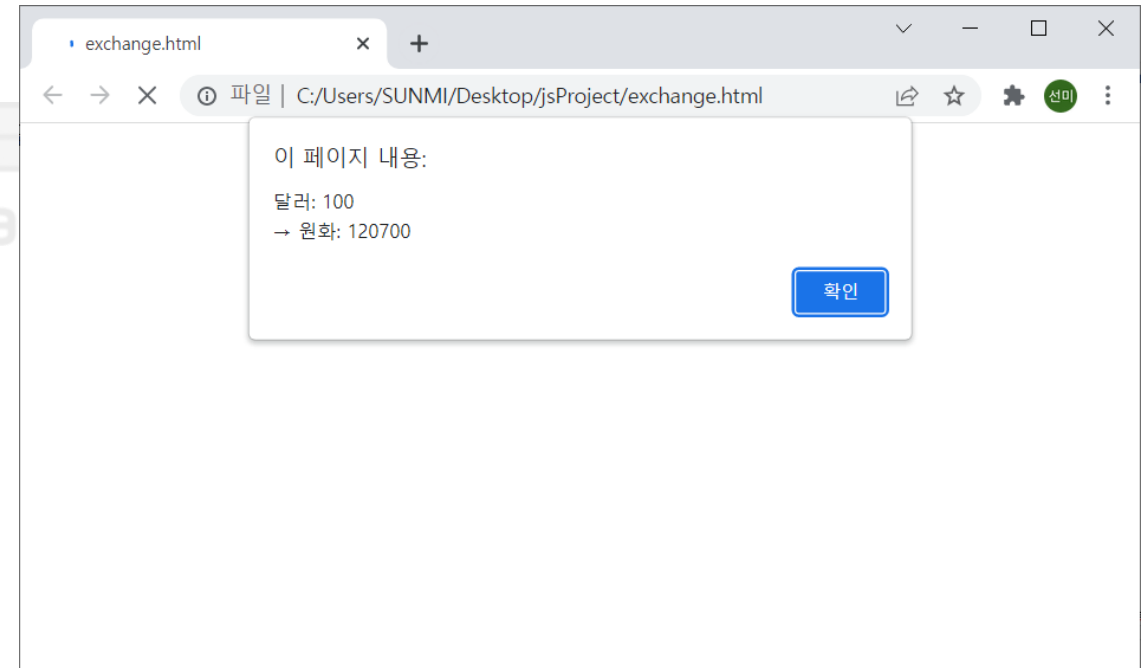
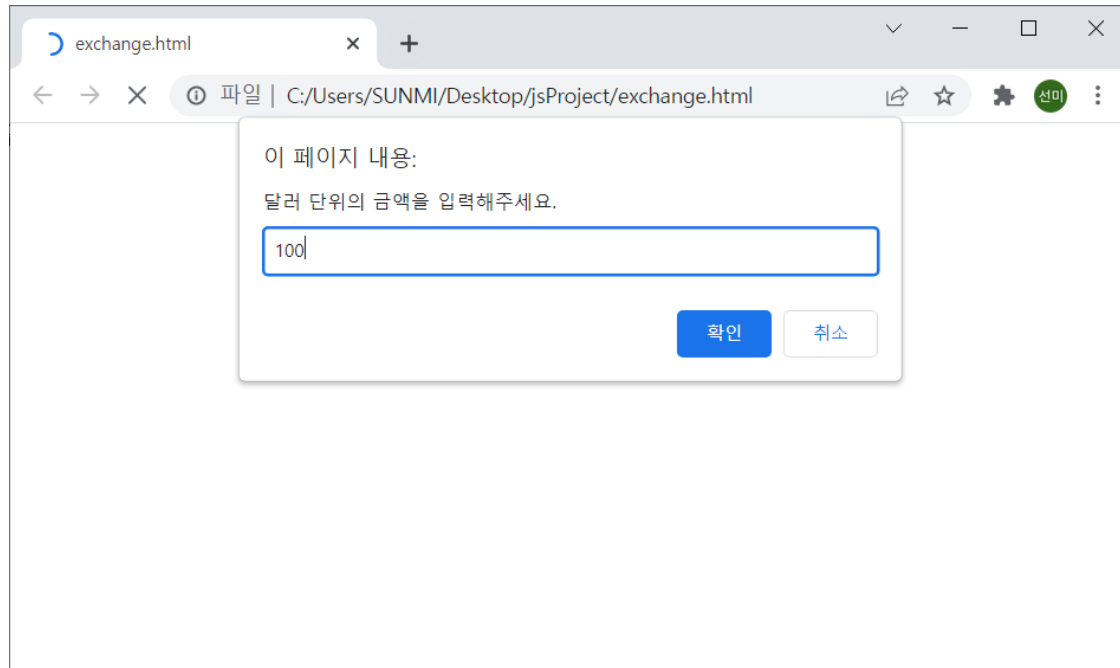
```
< 62.83184
```



미션> 환전 프로그램



달러를 입력하면 원화로 바꿔주는 코드를 작성하세요



미션

두 수를 입력 받아 $+$, $-$, $*$, $/$, $\%$ 결과를 출력해 주세요



미션



두 수를 입력 받아 첫번째 수부터 두번째 수 개수만큼 1씩 증가하며 출력해주세요

두 수를 입력 받아 첫번째 수부터 두번째 수 개수만큼 1씩 감소하며 출력해주세요



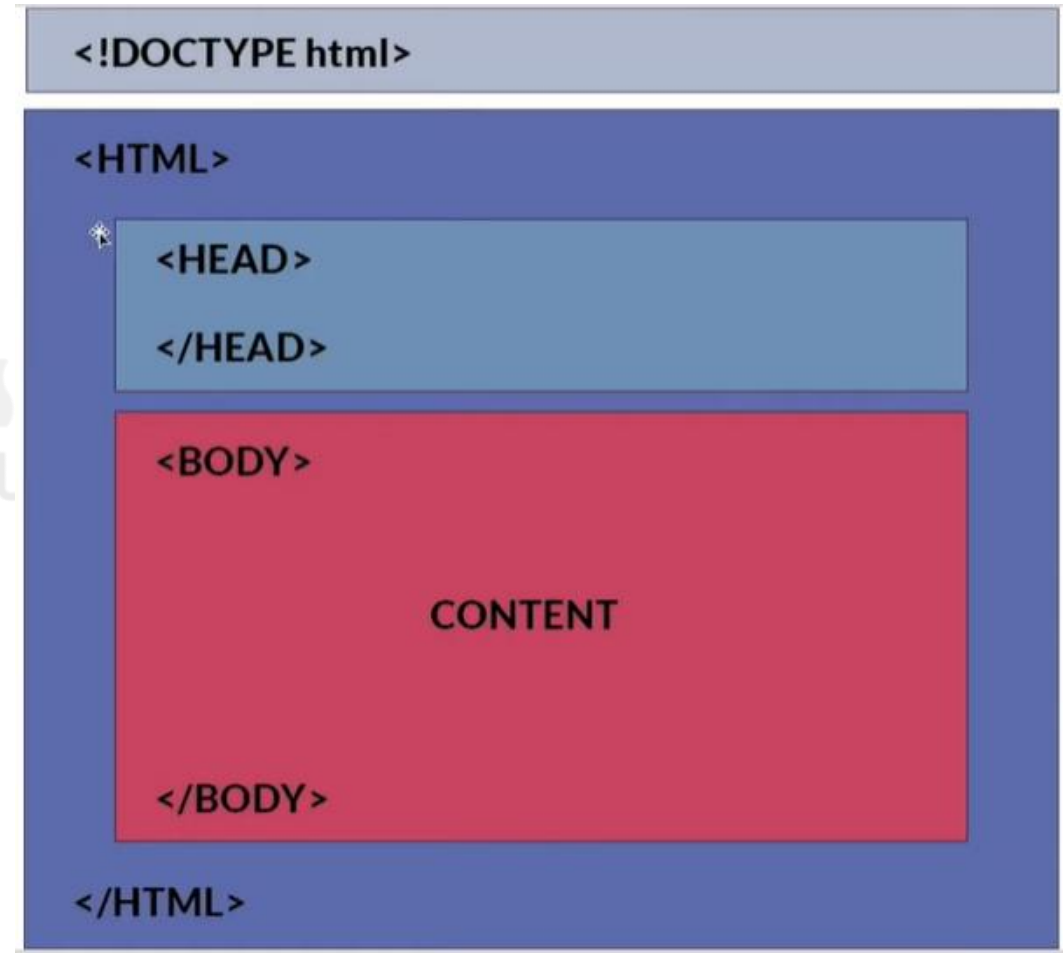
HTML 문서의 기본 구조와 JavaScript

AhnLab SEM

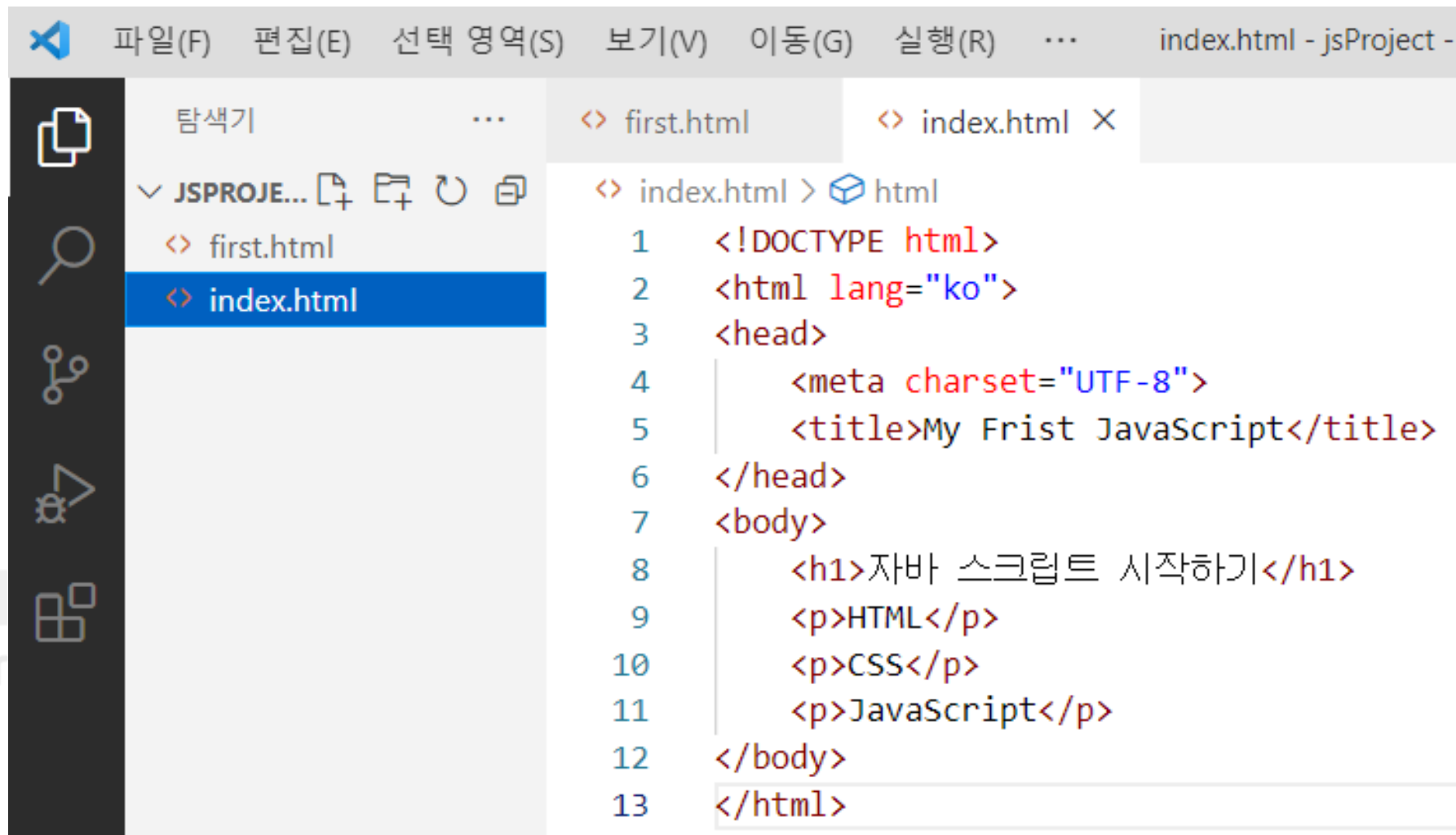
HTML 문서 기본 구조

HTML 구조


- 정해진 형식에 맞추어 입력
- 태그; < > 기호로 묶인 부분
- 태그 사이에 작성한 내용만 출력



HTML 문서 기본 구조



The screenshot shows a web development IDE with a menu bar at the top containing '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '실행(R)', and a dropdown menu. The title bar indicates the current file is 'index.html - jsProject -'. On the left, a file explorer shows a project named 'JSPROJE...' with two files: 'first.html' and 'index.html', with 'index.html' selected. The main editor displays the code for 'index.html' with line numbers 1 through 13. The code is as follows:

```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4      <meta charset="UTF-8">
5      <title>My Frist JavaScript</title>
6  </head>
7  <body>
8      <h1>자바 스크립트 시작하기</h1>
9      <p>HTML</p>
10     <p>CSS</p>
11     <p>JavaScript</p>
12 </body>
13 </html>
```

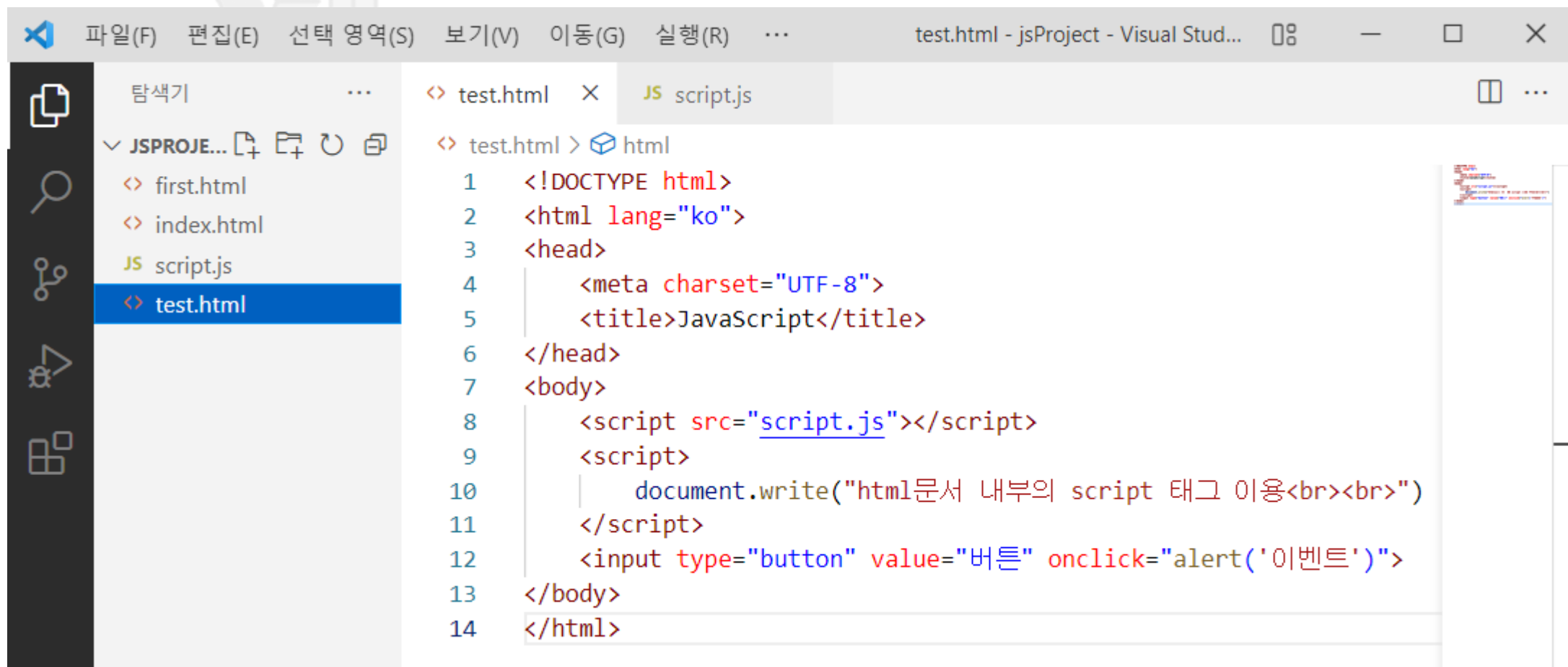
HTML 문서 기본 구조

```
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="UTF-8">
5     <title>My Frist JavaScript</title>
6   </head>
7   <body>
8     <h1>자바 스크립트 시작하기</h1>
9     <p>HTML</p>
10    <p>CSS</p>
11    <p>JavaScript</p>
12  </body>
13 </html>
```

HTML 문서에서 자바스크립트의 위치

1. 웹 문서 안에 <script> 태그로 자바스크립트 작성
2. 외부 스크립트 파일로 연결해서 작성
3. 인라인 자바스크립트(이벤트)

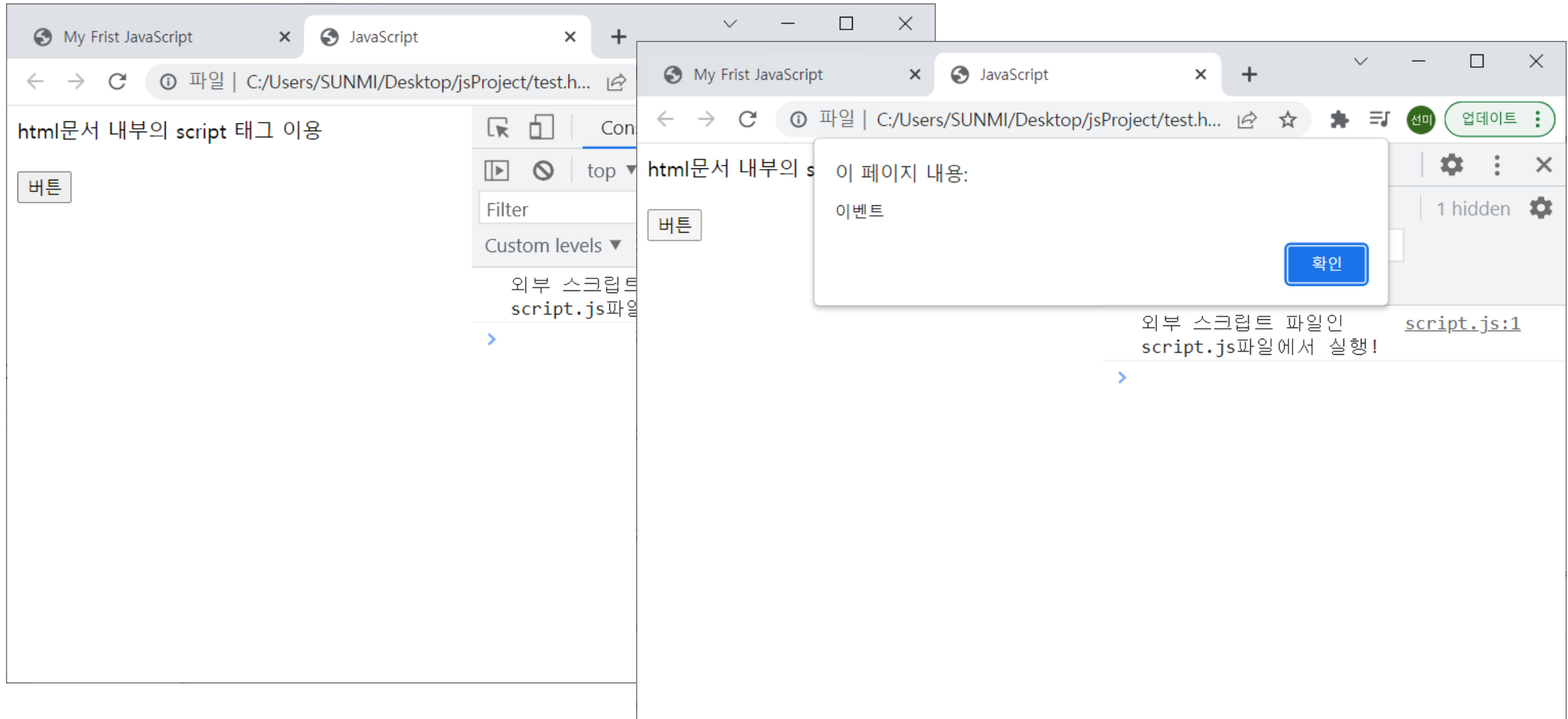
HTML 문서에서 자바스크립트의 위치



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project named 'JSPROJE...' with files 'first.html', 'index.html', 'script.js', and 'test.html'. The 'test.html' file is selected and open in the main editor. The file explorer shows the file structure: 'test.html' > 'html'. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <script src="script.js"></script>
  <script>
    document.write("html문서 내부의 script 태그 이용<br><br>")
  </script>
  <input type="button" value="버튼" onclick="alert('이벤트')">
</body>
</html>
```

HTML 문서에서 자바스크립트의 위치





자료형 변환



자료형 변환



자료형(data type)의 변환

- 자동 형 변환
- 숫자 타입에 문자열을 더하면 결과가 문자열, 문자열 연결연산자(+)
- 문자열에 숫자를 곱하면 숫자 타입

```
//숫자를 문자로 바꾸기  
let number=2;  
number += "";  
console.log(typeof(number));
```

```
//문자를 숫자로 바꾸기  
let txt = "2";  
txt*=1;  
console.log(typeof(txt));
```

자료형 변환



NaN

전역 `NaN` 속성은 Not-A-Number(숫자가 아님)를 나타냅니다.

`NaN` 을 반환하는 연산에는 다섯 가지 종류가 있습니다.

- 숫자로서 읽을 수 없음 (`parseInt("어쩌구")`, `Number(undefined)`)
- 결과가 허수인 수학 계산식 (`Math.sqrt(-1)`)
- 피연산자가 `NaN` (`7 ** NaN`)
- 정의할 수 없는 계산식 (`0 * Infinity`)
- 문자열을 포함하면서 덧셈이 아닌 계산식 (`"가" / 3`)



자료형 변환



자료형(data type)의 변환

- 사칙 연산자를 활용하여 숫자로 변환

```
let txt="11"-0;  
console.log(txt);  
console.log(typeof(txt));
```

```
let result= true-0;  
console.log(result);  
console.log(typeof(result));
```



자료형 변환



타입변환 함수 사용

- String()함수 사용해 다른 자료형을 문자열 자료형으로 변환
- Number()함수를 사용해 숫자 자료형으로 변환



자료형 변환

Boolean(자료) 변환

- 대부분 false 반환

```
> Boolean(0);
```

```
< false
```

```
> Boolean(NaN);
```

숫자가 아님

```
< false
```

```
> Boolean(null);
```

빈 값

```
< false
```

```
> Boolean("");
```

빈 문자열

```
< false
```

```
> let name;
```

```
< undefined
```

변수값 미정의

```
> Boolean(name);
```

```
< false
```

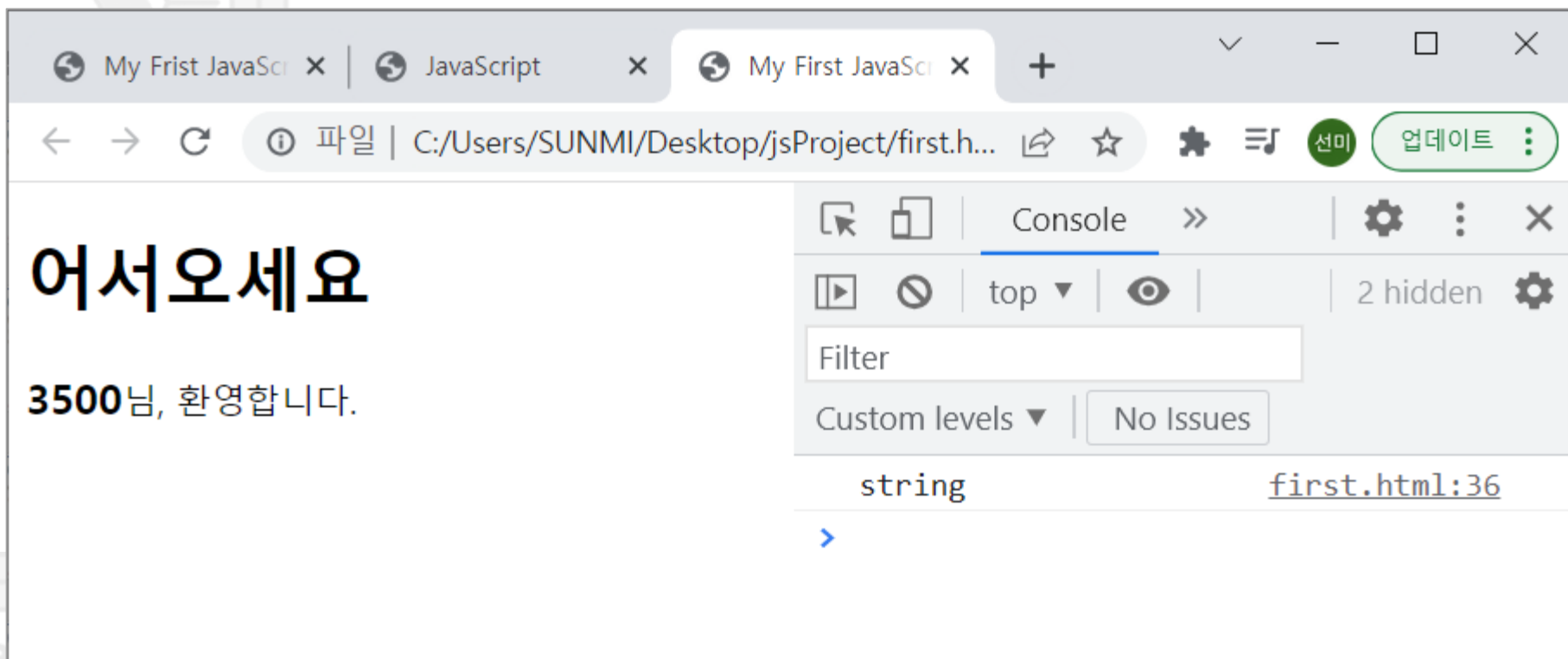
미션> 입력되는 자료형 확인



```
<script>
  document.write("<h1>어서오세요</h1>");
  let name = prompt("이름을 입력하세요");
  document.write("<b><big>" + name + '</big></b>님, 환영합니다. ');
  console.log(typeof(name));
</script>
```



미션> 입력되는 자료형 확인

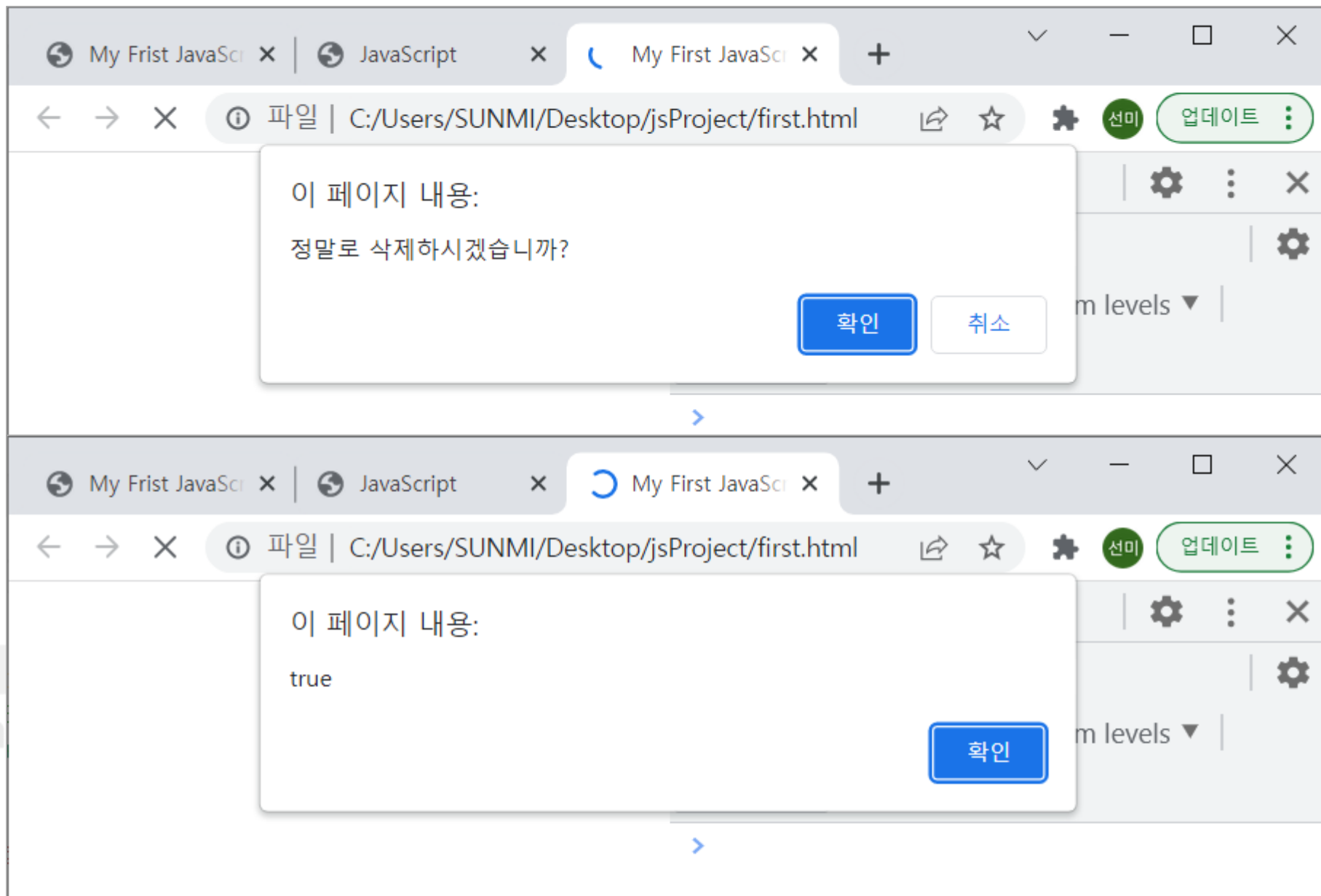


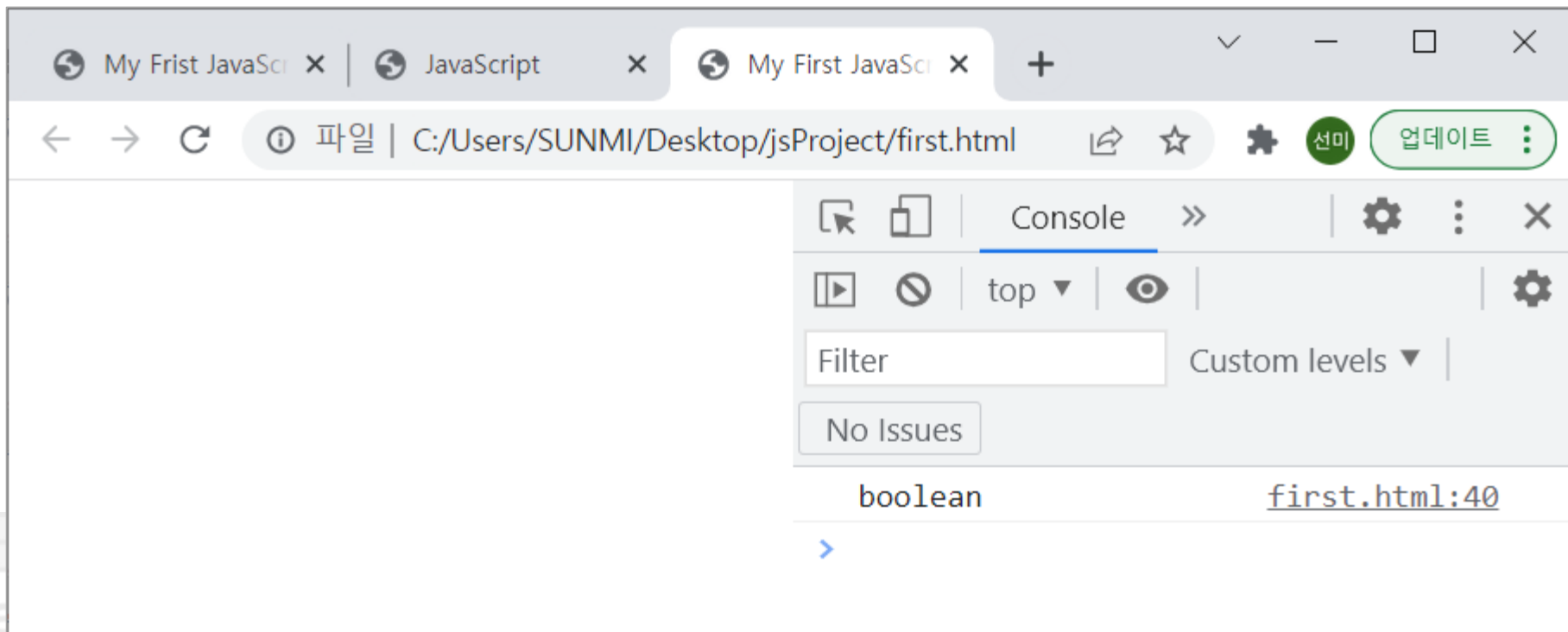
미션> 입력되는 자료형 확인



```
let rlt = confirm("정말로 삭제하시겠습니까?");  
alert(rlt);  
console.log(typeof(rlt));
```







감사합니다.

