

웹 프로그래밍 강의노트

9. JavaScript 기초

2023년 2학기

목차

- 자바스크립트 시작하기
- 자바스크립트 기본 문법
- 자바스크립트 제어문
- 자바스크립트 반복문

자바스크립트 시작하기

자바스크립트 개요와 특징

자바스크립트 작성하기

자바스크립트 실행 및 디버깅

자바스크립트 개요

- 개요 및 특징

- 동적인 웹 문서 제작과 웹 응용프로그램 개발을 위해 필수적으로 사용됨
 - 자바 애플릿, CGI 스크립트 대체 가능
- C/C++이나 자바 언어 등에 비해서 작성 및 실행이 매우 간편함
- 인터프리터 (interpreter) 방식

- 자바스크립트 연혁

- 라이브스크립트라는 이름으로 넷스케이프사에서 개발 시작
- 1995년 썬 (Sun, 현재 오라클)사와 공동 개발 → 자바스크립트 (JavaScript)로 개명
- 현재 표준
 - ECMA(European Computer Manufacturers Association): ECMA-262 혹은 ISO 16262
 - ECMAScript라고도 불리움

객체 기반의 자바스크립트

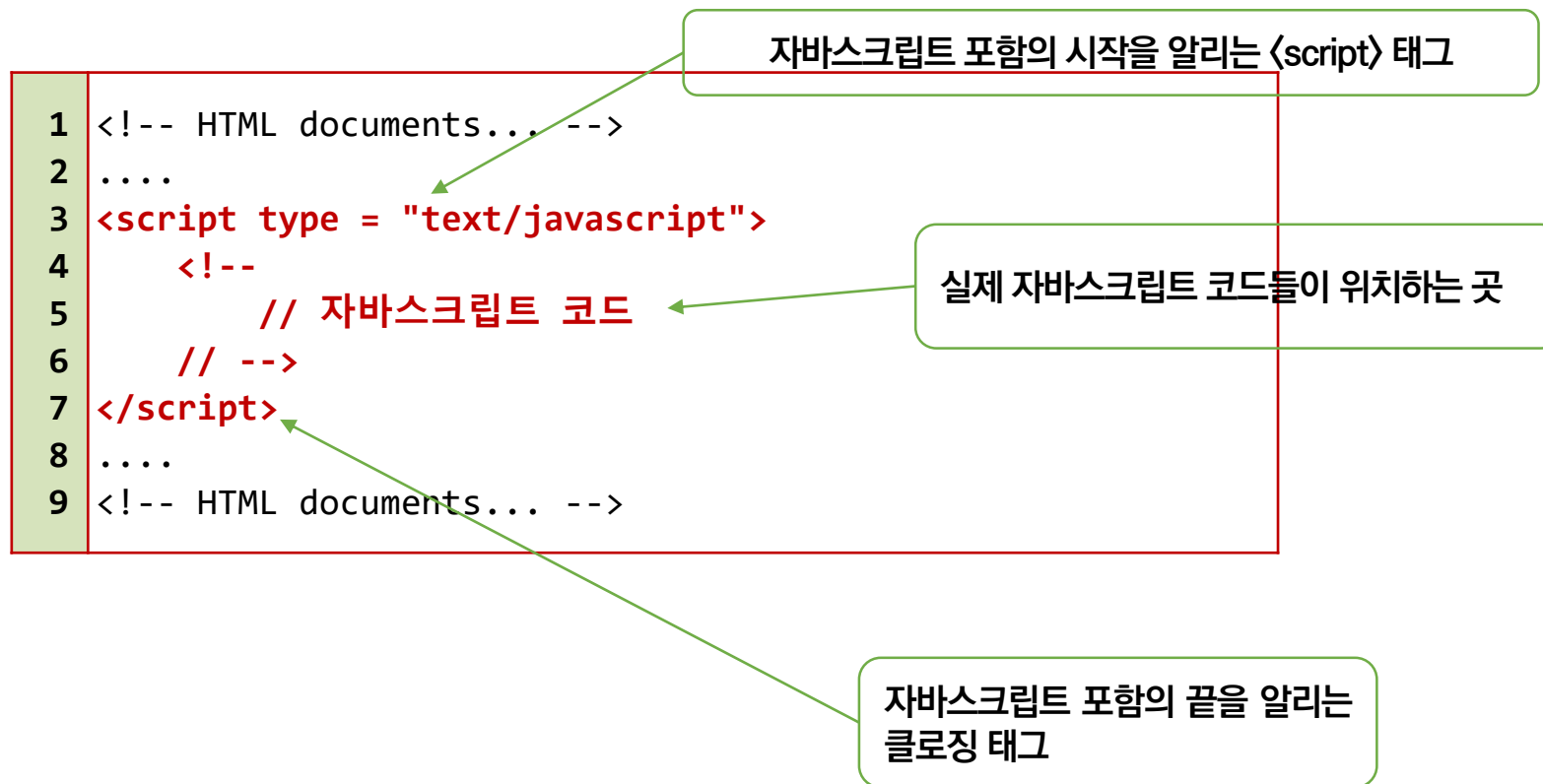
- 자바 언어의 영향을 받아서 문법적으로 비슷한 형태를 가지는 공통점이 있으나 자바 언어와는 다음과 같이 차이점을 가진다

	자바스크립트	자바 언어
실행 방식	웹 브라우저에서 바로 자바스크립트 코드를 해석하고 바로 실행 (스크립트/인터프리터 기반 언어)	자바 프로그램을 컴파일 후 변환된 object code를 자바가상머신에서 실행하는 방식 (컴파일 기반 언어)
성격	객체기반(object-based)	객체지향(object-oriented)
작성 형태	HTML 파일 내에 포함되어 작성됨	별도의 자바 프로그램 파일로 작성
변수형 선언 및 타입 검사	변수의 선언이 따로 필요 없으며 타입 검사도 매우 느슨함	변수의 선언이 필요하며 변수 타입의 검사가 매우 엄격함

자바스크립트 작성하기

- 자바스크립트 코드
 - HTML 파일 없이 웹브라우저에서 독립적으로 실행 될 수 없음
 - 반드시 HTML 파일 내에 포함되어 있어야 한다
- HTML 파일 내에 포함 시키는 두가지 방식
 - 웹문서 내장 방식
 - 외부 파일 참조 방식
 - CSS를 다룰 때도 내장하거나 외부 파일을 사용할 수 있었음

웹문서 내장 방식



외부 파일 참조 방식

- `<script>` 태그의 `src` 속성의 값으로 자바스크립트 파일의 경로를 지정

```
1 <!-- HTML documents... -->
2 ....
3 <script type="text/javascript" src="myscript.js">
4 </script>
5 ....
6 <!-- HTML documents... -->
```

`<script>` 요소의 `src` 속성의 값으로 실제 자바스크립트 파일의 경로를 지정할 수 있다.

- 자바스크립트 파일의 URL 경로 지정 가능

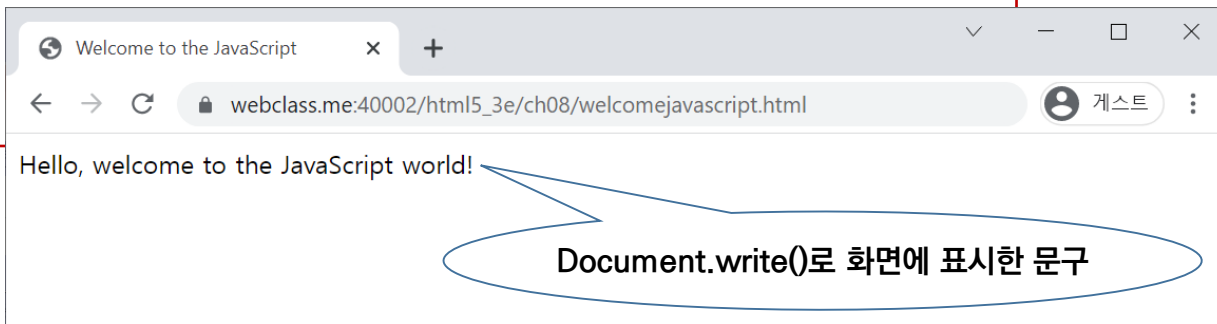
```
1 <!-- HTML documents... -->
2 ....
3 <script type="text/javascript"
4         src="http://webclass.me/html5/ch08/remove_script.js">
5 </script>
6 ....
7 <!-- HTML documents... -->
```

`<script>` 요소의 `src` 속성의 값으로 자바스크립트 파일이 위치한 URL 경로를 지정할 수도 있다.

간단한 자바스크립트 예제

```
1 <!DOCTYPE html>
2 <!-- hello.html
3   간단한 인사말을 화면에 표시하는 HTML/자바스크립트 기본 예제
4 -->
5 <html>
6 <head>
7 <title> Welcome to the JavaScript </title>
8 </head>
9 <body>
10  <script type="text/javascript">
11    <!--
12      document.write("Hello, welcome to the JavaScript world!");
13    // -->
14  </script>
15 </body>
16 </html>
```

웹브라우저에서 보여지는 문서상에 Hello, welcome to the JavaScript world!라는 문구를 출력하는 명령문



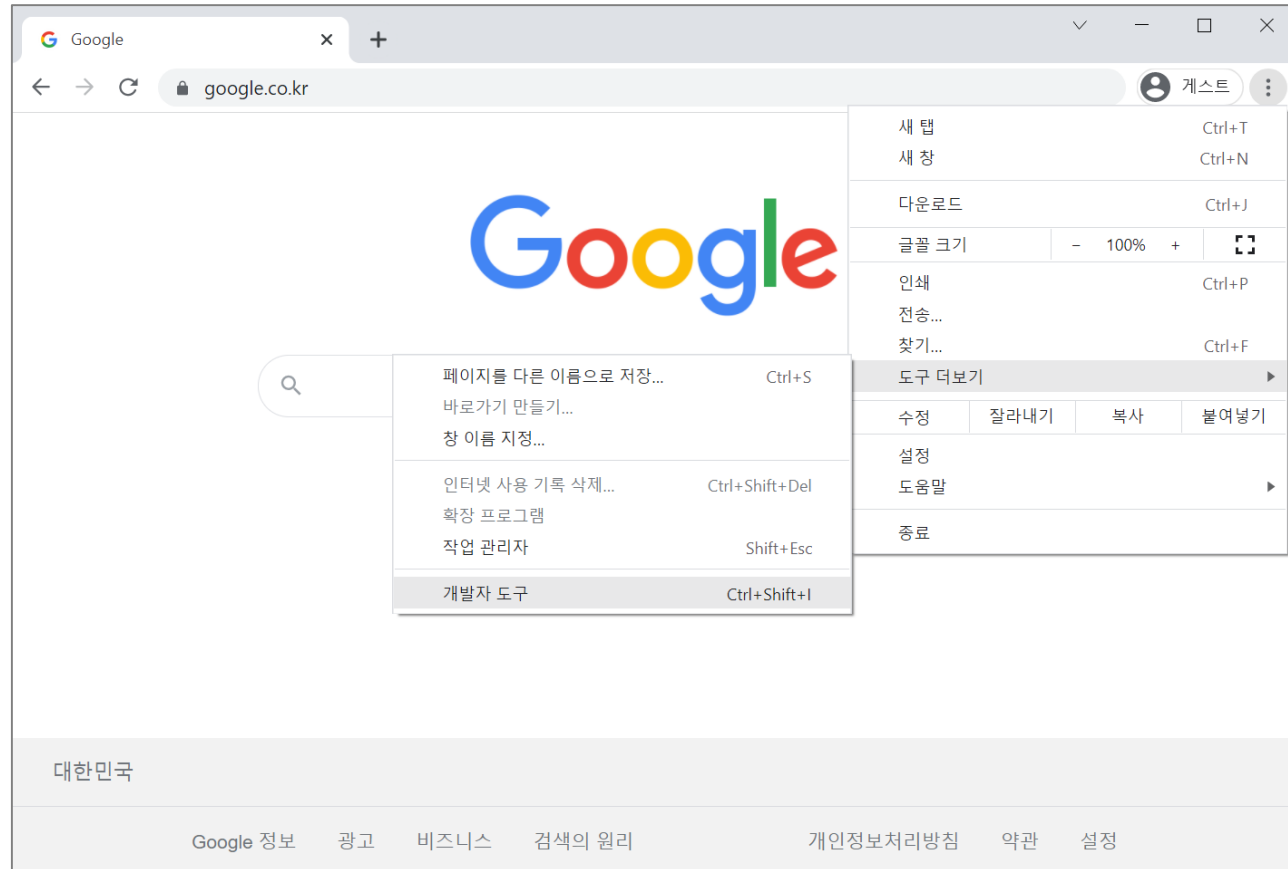
- “Hello, welcome to the JavaScript world!”라는 문구를 출력

자바스크립트 실행 및 디버깅

- HTML 파일 내의 자바스크립트 코드
 - 웹 브라우저가 자체 인터프리터를 이용해 그 스크립트 코드를 해석하고 실행
- 자바스크립트를 실행하는 동안 오류가 발생하는 경우
 - 치명적 오류가 아니라면 기본적으로 웹 브라우저는 그 오류를 무시하고 진행
 - 개발 단계에서 개발자가 확인하고 수정하는 것이 바람직
 - 한 번 배포된 웹 페이지 내의 오류가 일으키는 문제를 되돌리기 쉽지 않다

자바스크립트 오류 확인

- [Chrome 설정 및 관리] 버튼(:)을 누른 후 [도구 더보기] → [개발자 도구] → [콘솔]을 실행



자바스크립트 기본 문법

자바스크립트 기본 변수

기본 연산자와 변수 형 변환

화면 출력 및 키보드 입력

자바스크립트 기본 변수

- 기본 변수 타입
 - 대부분의 경우 자바스크립트 변수는 사용전에 미리 선언할 필요가 없음
 - 타입도 지정할 필요가 없음
- 내부적인 변수의 다섯가지 기본 형식

Number, String, Boolean, Undefined, Null

- 숫자의 표현 형태

125 1.25 0.125 .125 125. 12.e5 1.2e-5 12E5 12e5 .12e5

- 정수든 실수든 관계없이 **내부적으로 숫자는 모두 실수로 저장됨**

자바스크립트 기본 변수 타입

기본 변수 타입	변수 값	비고
Number	정수, 실수 등 숫자 값을 가짐	숫자 (Number)와 문자열 (String) 타입간에는 숫자 값에 대해 자동 형변환을 제공한다.
String	연속된 글자들로 이루어진 문자열 (공백도 가능함). 문자열의 시작과 끝은 작은 따옴표 (') 혹은 겹따옴표 (")로 지정	
Boolean	true 혹은 false	조건식에서 사용
Undefined	undefined 만 가능	변수의 값이 아직 지정되지 않은 경우 undefined 값을 가진다
Null	null 만 가능	

- 자바스크립트 변수형은 typeof() 연산자를 이용해서 확인 가능
 - typeof(123) → "Number"를 반환
 - typeof("123") → "String"을 반환

자바스크립트 변수 선언

- 변수를 사전에 선언 없이 사용하는 것이 가능
 - 전역 변수로 사용할 때는 미리 선언되어 있어야 함
 - 변수 타입을 고려하지 않고 선언해서 사용하면 되므로 편리
- 별도의 변수 타입이 없으며 var 타입 한가지만 제공됨
 - 변수에 실제로 어떤 값이 저장될 때 그 값에 따라 내부적으로 변수 타입이 정해진다
- 변수의 선언 방식
 - 대소문자를 구분함

```
var 변수명;  
    혹은  
var 변수명 = 변수값;  
    혹은  
변수명 = 변수값;
```

자바스크립트 변수 선언 예제

문자열, 정수, 실수 등 모든 변수 타입에 대해 var로 선언한다.

```
var index, name = "모바일 웹";  
var start = 0, end = 100.0;  
var message, condition, sender, receiver;
```

```
var a = "3";  
var b = 2;
```

```
c = b + 3 + a;    // c값은 "53"이 됨, 더하기 후 문자열 붙이기 연산  
d = a + b;        // d값은 "3"+"2"="32", 문자열연산(Concatenation)
```

연산의 우선순위에서
b+3이 우선

b의 값 즉 숫자 2을 문자열
"2"으로 변환한 후 a의 값
"3"과 문자열 붙이기 연산
수행

- 모든 변수 타입에 대해 var 타입으로만 선언
 - 문자열, 정수, 실수 등

자바스크립트 기본 연산자

종류	연산자	설명	비고
사칙연산	+ - * / %	더하기, 빼기, 곱하기, 나누기, 나머지	*, /, %가 +, - 보다 연산 우선순위가 높다
대입 연산자	+ = - = * = / = % =	왼편 변수에 우측 값을 연산 후 왼편 변수에 대입	
증감 연산자	++ --	기존 변수 값에 +1 혹은 -1 연산 수행	
논리연산자	> <	왼편의 값이 크다, 작다	결과값이 true 혹은 false이다.
	>= <=	왼편의 값이 크거나 같다, 작거나 같다	
	== !=	양편이 같다, 다르다 (값만 비교)	
	=== !==	양편의 같다, 다르다 (값과 타입 모두 비교)	
	! &&	논리부정 (NOT), 논리합 (OR), 논리곱 (AND) 연산	

자바스크립트 문자열 붙이기 연산

- 문자열 붙이기 (Concatenation) 연산
 - '+' 연산자를 이용해서 두 문자열을 붙임

```
var first_name = "Steve";  
var last_name = "Jobs";  
  
var full_name1 = first_name + " " + last_name;  
// full_name1: "Steve Jobs"  
  
var full_name2 = last_name + ", " + first_name;  
// full_name2: "Jobs, Steve"  
  
var address = "서울시 " + "용산구 " + "청파동";  
// address: "서울시 용산구 청파동"
```

변수 형변환 (type conversion)

- 문자열 타입 → 숫자 타입
 - 문자열 변수를 `parseInt()` 혹은 `parseFloat()` 함수에 입력
- 숫자 타입 → 문자열타입
 - 숫자 형 변수에 `toString()` 메소드를 이용

NOTE: 메소드(Method): 객체에 미리 정의되어 포함되어 있는 함수

변수 형변환 예제

```
var length = 123, length_num, length_str;
```

```
length_num = length + 10;
```

```
length_str = length.toString() + 10;
```

```
document.write("Length in Number: " + length_num + " cm" + '<br />');
```

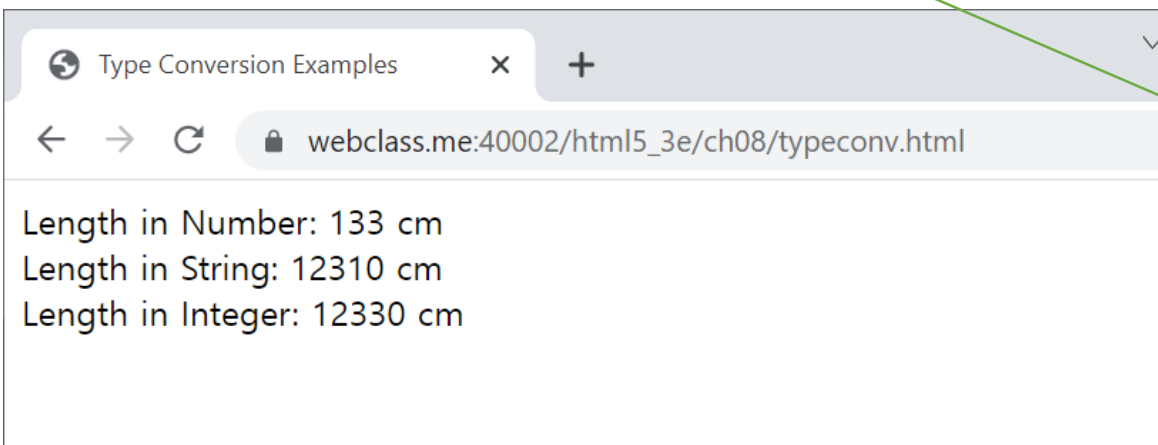
```
document.write("Length in String: " + length_str + " cm" + '<br />');
```

```
var num = parseInt(length_str) + 20;
```

```
document.write("Length in Integer: " + num + " cm");
```

length의 값을 toString()이라는 메소드를 이용해서 문자열 형태로 변환한다. 숫자 10을 문자열 "10"으로 변환 후 문자열 붙이기 연산이 수행되어 결과 값은 문자열 "12310"이 된다.

parseInt() 함수는 문자열을 숫자로 변환하므로 문자열 "12310"을 숫자 12310으로 변환하다. 다시 숫자 20과 더해지므로 결과값은 숫자 12330이 된다.



화면 출력

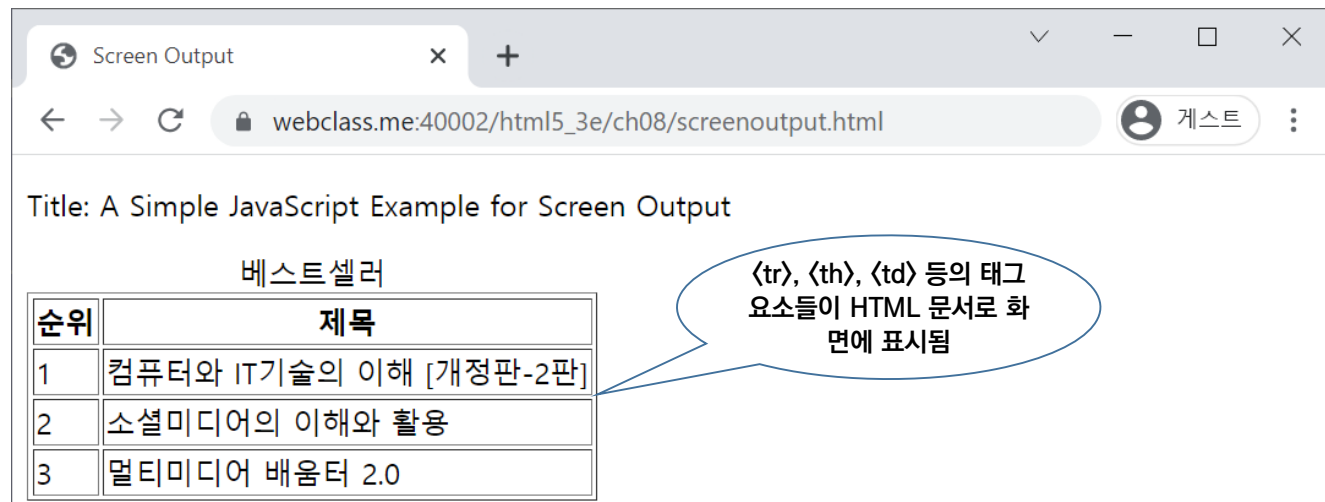
- `document.write()`라는 화면 출력 명령어
 - `document.write()`라는 명령어는 HTML 문서에 콘텐츠 추가
 - 콘텐츠가 삽입된 HTML 문서의 내용이 화면에 출력
 - HTML 태그를 추가할 경우에는 그 태그도 해석되어 화면에 출력
- HTML 문서는 Document라는 객체로 모델링 되어 있다
 - `document`라는 이름으로 접근
 - Document 객체의 `write()` 메소드

화면 출력 예제

```
var title1 = "컴퓨터와 IT기술의 이해 [개정판-2판]";  
var title2 = "소셜미디어의 이해와 활용 ";  
var title3 = "멀티미디어 배움터 2.0";
```

```
document.write("<caption> 베스트셀러 </caption>");  
document.write("<tr>");  
document.write("<th> 순위 </th>");  
document.write("<th> 제목 </th>");  
document.write("</tr>");  
document.write("<tr> <td> 1 </td> <td> " + title1 + " </td> </tr>");  
document.write("<tr> <td> 2 </td> <td> " + title2 + " </td> </tr>");  
document.write("<tr> <td> 3 </td> <td> " + title3 + " </td> </tr>");
```

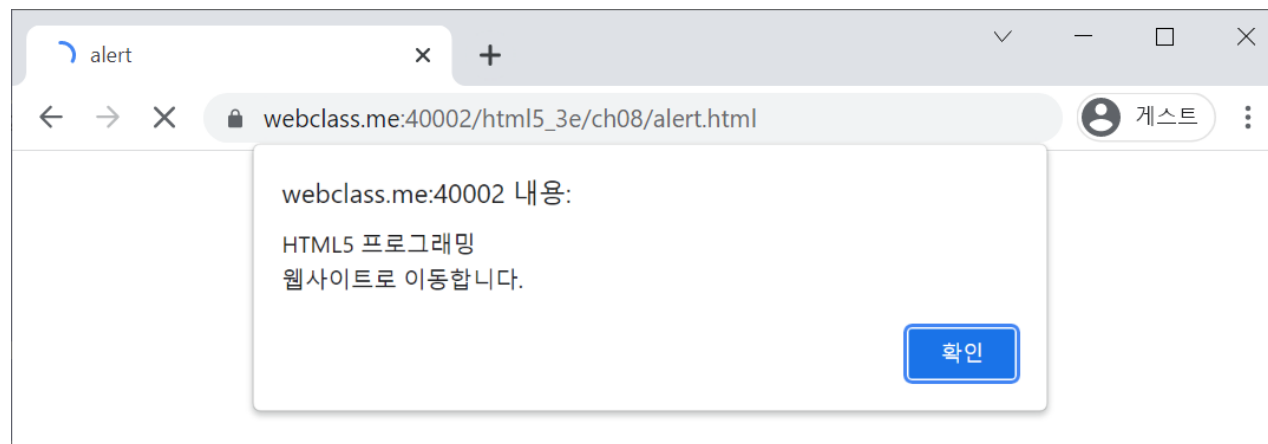
document.write() 메소드로 삽입된 문자열은 모두 HTML 문서 안에 삽입되어 일반 HTML 문서처럼 해석된다. 위의 예제처럼 테이블을 만들기 위한 태그들 (예: <th>, <td> 등)을 사용하여 HTML 문서를 만들어 낼 수 있다.



대화상자로 메시지 출력

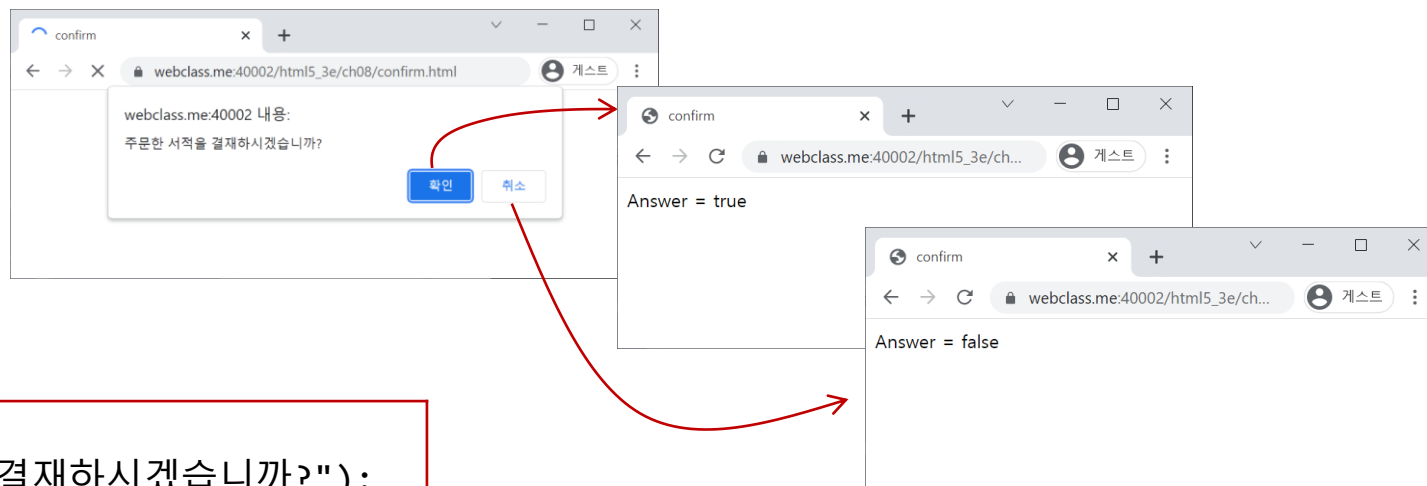
- 대화상자(dialog box)를 만들어 화면에 메시지를 출력하거나 키보드로부터 입력을 받을 수 있는 세가지 방법
- alert() 명령어
 - 사용자에게 경고사항이나 메시지를 전달
 - "확인" 버튼을 클릭하지 않으면 다음 자바스크립트 문장이 실행되지 않음

```
alert("HTML5 프로그래밍 \n웹사이트로 이동합니다.");
```



확인 입력 받기

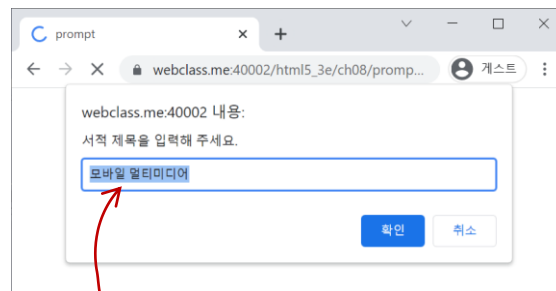
- confirm() 명령어
 - 사용자에게 Yes/No 선택을 입력받기 위해 사용하는 방식
 - 대화상자 내에 메시지를 표시하고 “확인”과 “취소” 버튼 표시
 - 버튼을 클릭할 때까지 실행을 대기
 - 확인 버튼을 누르면 true, 취소 버튼을 누르면 false를 반환



```
var answer = confirm("주문한 서적을 결재하시겠습니까?");  
document.write("Answer = " + answer + "<br/>");
```

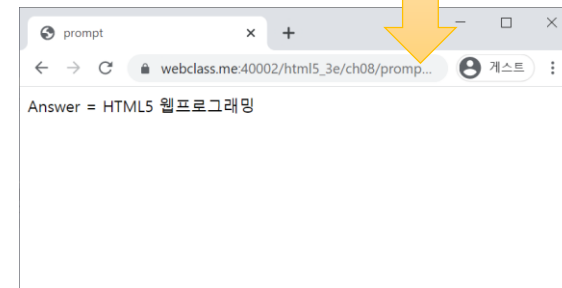
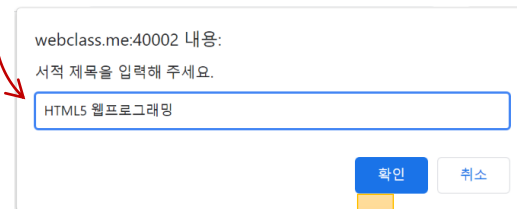

키보드로부터 문자열 입력 받기

- prompt() 명령어
 - 사용자로부터 키보드를 통해 문자열을 입력 받는다
 - 대화상자 내에 메시지와 초기입력값이 입력 상자와 함께 표시됨
 - “확인”을 누르면 입력된 문자열, “취소”를 누르면 null을 반환



초기입력값

사용자입력값



```
var answer = prompt("서적 제목을 입력해 주세요.", "모바일 멀티미디어");  
document.write("Answer = " + answer + "<br/>");
```

prompt()의 리턴값

- prompt()는 항상 문자열 형식을 리턴
- 사용자가 prompt 창에 숫자를 입력하더라도 "1", "2", "3"과 같이 문자열과 비교하는 이유는 prompt()가 리턴하는 값이 문자열 형식이기 때문

자바스크립트 제어문

자바스크립트 제어문

- 제어문으로 if 문과 switch 문을 제공

자바스크립트 제어문	문법 및 사용 형식	비고
if-then-else	<pre>if (조건식) { // 조건식의 값이 true일 때 실행될 문장 } else { // 조건식의 값이 false일 때 실행될 문장 }</pre>	실행될 문장이 한 개인 경우에는 { }를 생략할 수 있다.
switch	<pre>switch (expression) { case value_1: // expression값이 value_1일 때 실행될 문장 break; case value_2: // expression값이 value_2일 때 실행될 문장 break; case value_3: // expression값이 value_3일 때 실행될 문장 break; ... default: // case문에서 찾을 수 없을 때 실행될 문장 } </pre>	C/C++, 자바 언어와는 달리 (expression)에 정수형 이외의 타입도 사용할 수 있다. 예를 들면 문자열 형식의 값을 사용할 수도 있다.

if-else 문

```
document.write("<caption> 책 주문 입력 내용 </caption>");
document.write("<tr>");
document.write("<th> 제목 </th>");
document.write("</tr>");

var book1 = "1: 컴퓨터와 IT기술의 이해 [개정판-2판]";
var book2 = "2: 소셜미디어의 이해와 활용";
var book3 = "3: 멀티미디어 배움터 2.0";

var book_list = book1 + "\n" + book2 + "\n" + book3;
var choice = prompt("책 번호를 입력하세요...\n" + book_list, "1");

if (choice == "1")
    title = book1;
else if (choice == "2")
    title = book2;
else if (choice == "3")
    title = book3;
else {
    alert("리스트에 없는 책을 선택하셨습니다.");
    title = "";
}

document.write("<tr>");
document.write("<td>" + title + "</td>");
document.write("</tr>");
```

if-else 문

```
document.write("<caption> 책 주문 입력 내용 </caption>");  
document.write("<tr>");  
document.write("<th> 제목 </th>");  
document.write("<tr>");
```

실행 첫 화면

Control & Loop Statements

webclass.me:40002/html5_3e/ch08/ifexample.h...

게스트

webclass.me:40002 내용:

책 번호를 입력하세요...

1: 컴퓨터와 IT기술의 이해 [개정판-2판]
2: 소셜미디어의 이해와 활용
3: 멀티미디어 배움터 2.0

1

확인 취소

기본 입력값

2를 입력한 경우 실행 결과

Control & Loop Statements

webclass.me:40002/html5_3e/ch08/ifexample.h...

게스트

A Simple JavaScript Example for Control and Loop Statements

책 주문 입력 내용

제목
2: 소셜미디어의 이해와 활용

switch 문

- expression의 값과 각 case의 값을 비교하여 일치하는 case에 적힌 문장을 실행
- expression에 정수형 이외의 타입도 사용할 수 있다
 - 문자열 형식의 값도 사용 가능

```
switch (expression) {  
  case value_1:  
    // expression값이 value_1일 때 실행될 문장  
    break;  
  case value_2:  
    // expression값이 value_2일 때 실행될 문장  
    break;  
  case value_3:  
    // expression값이 value_3일 때 실행될 문장  
    break;  
  ...  
  default:  
    // case문에서 찾을 수 없을 때 실행될 문장  
}
```

switch 문예제

```
document.write("<caption> 책 주문 입력 내용 </caption>");
document.write("<tr>");
document.write("<th> 제목 </th>");
document.write("</tr>");

var book1 = "IT: 컴퓨터와 IT기술의 이해 [개정판-2판]";
var book2 = "SNS: 소셜미디어의 이해와 활용";
var book3 = "Multimedia: 멀티미디어 배움터 2.0";

var book_list = book1 + "\n" + book2 + "\n" + book3;
var choice = prompt("키워드를 입력하세요...\n" +
book_list, "");

switch (choice) {
    case "IT":
        title = book1; break;
    case "SNS":
        title = book2; break;
    case "Multimedia":
        title = book3; break;
    default:
        alert("리스트에 없는 책을 선택하셨습니다.");
        title = "";
}

document.write("<tr>");
document.write("<td>" + title + "</td>");
document.write("</tr>");
```


자바스크립트 반복문

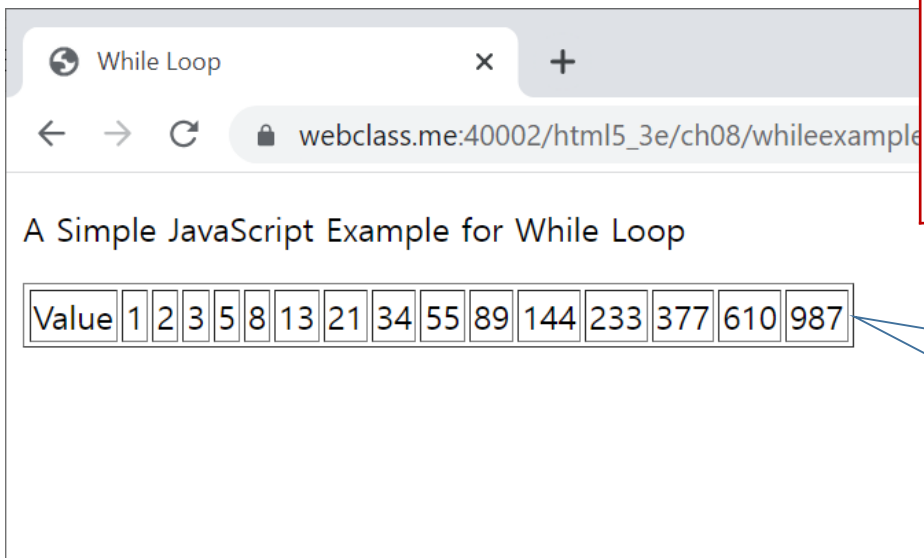
자바스크립트 반복문

- while, for, do-while 문을 제공

자바스크립트 반복문	문법 및 사용 형식	비고
while	<pre>while (조건식) { // 조건식의 값이 true일 동안 반복해서 실행될 문장 }</pre>	실행될 문장의 개수가 하나인 경우에는 { }를 생략할 수 있다.
for	<pre>for (초기화 문장; 조건식; 증감문) { // 조건식의 값이 true일 동안 반복해서 실행될 문장 }</pre>	
do-while	<pre>do { // 조건식의 값이 true일 동안 반복해서 실행될 문장 } while (조건식)</pre>	

while 반복문 예제

- 피보나치 수열 값이 1000 이하일 동안 반복해서 출력하는 예제



```
<table border="1">
<script type="text/javascript">
    document.write("<tr>");
    document.write("<td> Value </td>");

    fib1 = 1;
    fib2 = 1;
    while (fib2 < 1000) {
        document.write("<td>");
        document.write(fib2);
        document.write("</td>");
        tmp = fib1 + fib2;
        fib1 = fib2;
        fib2 = tmp;
    }
    document.write("</tr>");
</script>
</table>
```

결과값이 1000보다
작을 동안 출력

for 반복문

- 문법

```
for (초기화 문장; 조건식; 증감문) {  
    // 조건식의 값이 true일 동안 반복해서 실행될 문장  
}
```

- 예) 1에서 10까지 출력하고 싶을 경우

```
for(num = 1; num <=10; num ++){  
    document.write(num + " ");  
}
```

for 반복문 예제

```
document.write("<h3> 책 주문 입력 내용 </h3>");

var book1 = "IT: 컴퓨터와 IT기술의 이해 [개정판-2판]";
var book2 = "SNS: 소셜미디어의 이해와 활용";
var book3 = "Multimedia: 멀티미디어 배움터 2.0";

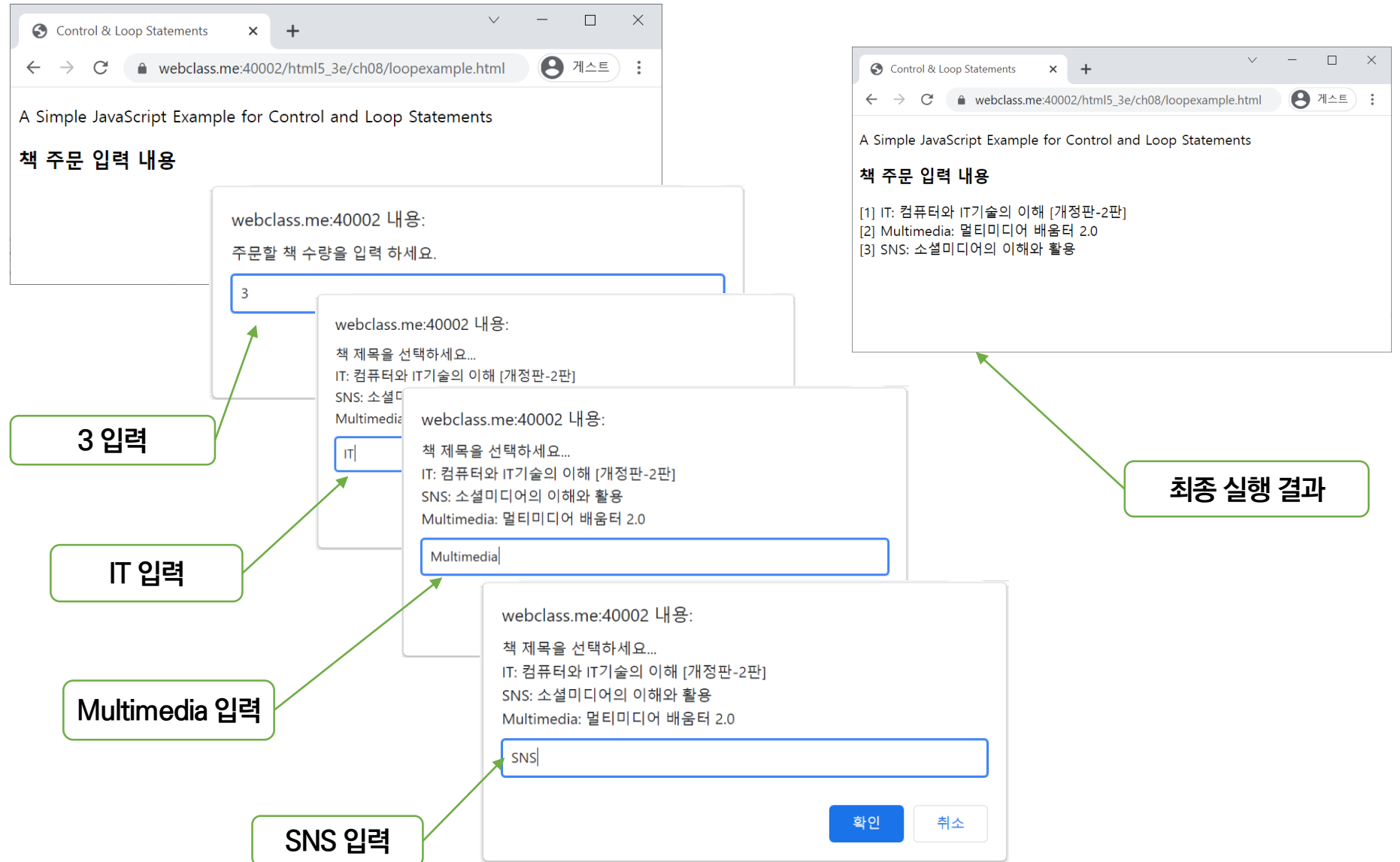
var n = prompt("주문할 책 수량을 입력 하세요.", "1");

for(i = 0; i <n; i++) {
    var book_list = book1 + "\n" + book2 + "\n" + book3;
    var choice = prompt("책 제목을 선택하세요...\n" + book_list, "");

    if (choice == "IT")
        title = book1;
    else if (choice == "SNS")
        title = book2;
    else if (choice == "Multimedia")
        title = book3;
    else {
        alert("리스트에 없는 책을 선택하셨습니다.");
        title = "";
    }

    document.write "[" + (i+1) + "]" + title + "<br/>");
}
```

for 반복문 예제 실행결과



do-while 반복문

- 문법

```
do {  
    // 조건식의 값이 true일 동안 반복해서 실행될 문장  
} while (조건식)
```

- 예) 1에서 10까지 출력하고 싶을 경우

```
num = 1;  
do {  
    document.write(num + " ");  
    num = num + 1;  
} while (num <= 10)
```