

자바스크립트 (웹 프로그래밍)

□ 자바스크립트 프로그래밍 기초

※ 자바스크립트 시작하기

○ 자바스크립트 개요와 특징

* 개요 및 특징 // ECMAScript 라고도 불림

- 동적인 웹 문서, 웹 응용프로그램을 위한 사용자 인터페이스 개발
- 작성 및 실행이 매우 간편함 / 인터프리터 (interpreter) 방식

○ 객체 기반의 자바스크립트

* 자바 언어와 차이점

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

○ 자바스크립트 작성

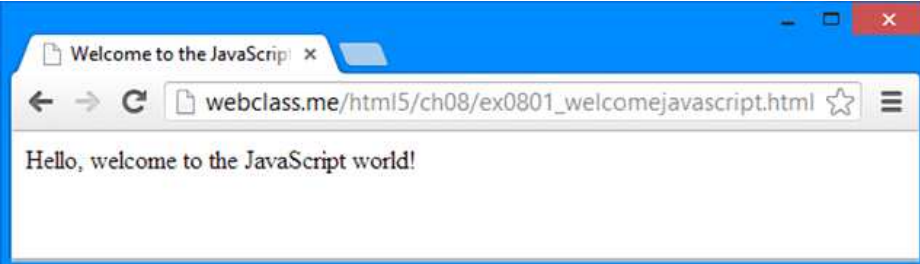
◎ 웹문서 내장 방식

/// 웹문서 내장 방식	
01	<!-- HTML documents... -->
02	...
03	<script type = "text/javascript">
04	<!--
05	// 자바스크립트 코드
06	// -->
07	</script>
08
09	<!-- HTML documents... -->

◎ 외부 파일 참조 방식

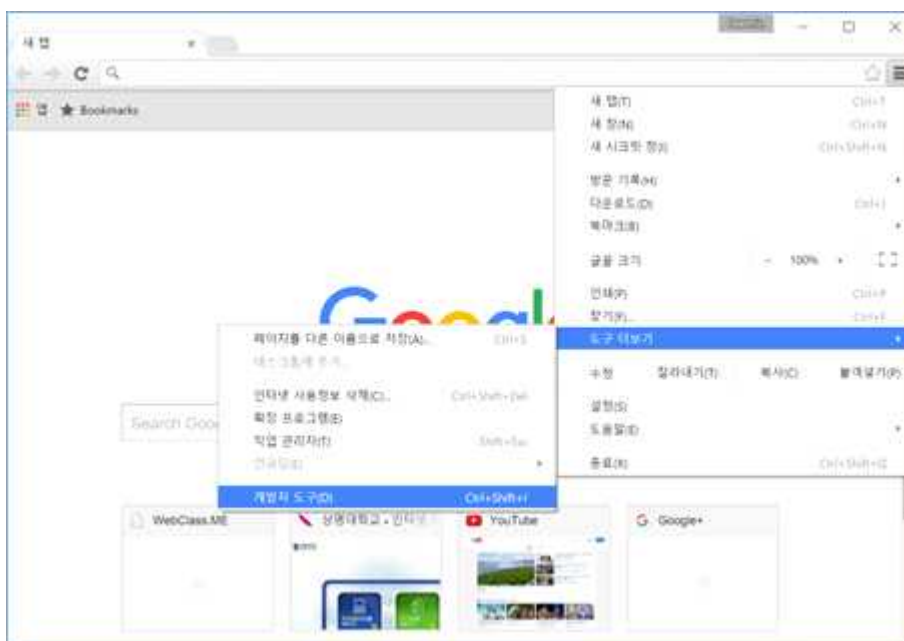
/// 외부 파일 참조 방식	
01	<!-- HTML documents... -->
02	...
03	<script type = "text/javascript" src="myscript.js"> </script>
04	...
05	<script type = "text/javascript"
	src="http://weblclass.me/html5/javascript_example.js"> </script>
07
08	<!-- HTML documents... -->

* 간단한 자바스크립트 예제

/// 기본 자바스크립트 예제	
01	<!DOCTYPE html>
02	<!-- hello.html
03	간단한 인사말을 화면에 표시하는 HTML/자바스크립트 기본 예제
04	-->
05	<html>
06	<head>
07	<title> Hello JavaScript </title>
08	</head>
09	<body>
10	<script type="text/javascript">
11	<!--
12	document.write("Hello! JavaScript!");
13	// -->
14	</script>
15	</body>
16	</html>
출력	

○ 자바스크립트 실행 및 디버깅

* [Chrome 설정 및 관리] 버튼(☰) -> [도구 더보기] -> [개발자 도구] -> [콘솔] 실행



※ 자바스크립트 기본 문법

○ 자바스크립트 기본 변수

* 기본 변수 타입

- 대부분의 경우 미리 선언할 필요가 없음
- 타입도 지정할 필요가 없음

* 내부적인 변수의 다섯가지 기본 형식

- Number, String, Boolean, Undefined, Null

* 숫자의 표현 형태 (내부적으로 숫자는 모두 실수로 저장)

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

○ 자바스크립트 기본 변수 타입

기본 변수 타입	변수 값	비고
Number	정수, 실수 등 숫자 값을 가짐	
String	연속된 글자들로 이루어진 문자열(공백도 가능), 문자열의 시작과 끝은 작은 따옴표 (') 혹은 겹따옴표 (")로 지정	숫자 (Number)와 문자열 (String)타입 간에는 숫자 값에 대해 자동 형변환을 제공한다.
Boolean	true 혹은 false	조건식에서 사용
Undefined	undefined 만 가능	
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 선언해야함

* 별도의 변수 타입이 없으며 var 선언 한가지만 제공

* 변수에 값이 저장될 때, 그 값에 따라 내부적으로 변수 타입 결정

- 문자열, 정수, 실수 등 타입 상관 없음
 // ex) var 변수명; 혹은 var 변수명 = 변수 값;
- 변수의 선언 방식 : 대소문자 구분

/// 변수 선언 예제	
01	var index, name = "모바일 웹";
02	var start = 0, end = 100.0;
03	var message, condition, sender, receiver;
04	
05	var a = "3";
06	var b = 2;
07	c = b + 3 + a; // c 값은 "53"이 됨, 더하기 후 문자열 붙이기 연산
08	d = a + b; // d 값은 "3"+"2"="32", 문자열 연산(Concatenation)

○ 기본 연산자

종류	연산자	설명	비고
사칙연산	+ - * / %	더하기, 빼기, 곱하기, 나누기, 나머지	-내부적으로 실수 값으로 변환 후 처리
대입 연산자	+= -= *= /= %=	왼편 변수에 우측 값을 연산 후 왼편 변수에 대입	-자바 언어와 동일한 연산 우선순위
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	

* 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'
```


○ 변수 형 변환

* 문자열 타입 -> 숫자 타입

◎ parseInt (문자열 변수) 혹은 parseFloat (문자열 변수) 함수

* 숫자 타입 -> 문자열 타입

◎ toString (숫자 형 변수) 메소드를 이용

/// 변수 형 변환 예제	
01	var length = 123, length_num, length_str;
02	
03	length_num = length + 10;
04	length_str = length.toString() + 10;
05	
06	document.write("Length in Number: " + length_num + "cm" ");
07	document.write("Length in String: " + length_str + "cm" ");
08	
09	var num = parseInt(lenth_str) + 20;
10	document.write("Length in Integer: " + num + "cm");
출력	

○ 화면 출력

◎ **document.write()** : 화면 출력 명령어

* HTML 문서에 콘텐츠를 추가하여 화면에 출력

- HTML 태그를 추가할 경우에는 그 태그도 해석되어 화면에 출력

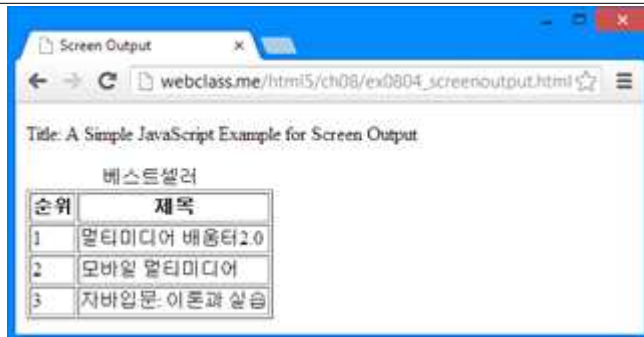
* HTML 문서는 Document라는 객체로 모델링 되어 있음

- document라는 이름으로 접근 => Document 객체의 write() 메소드

/// 화면 출력 예제

```
01 var title1 = "멀티미디어 배움터 2.0";
02 var title2 = "모바일 멀티미디어";
03 var title3 = "자바입문: 이론과 실습";
04
05 document.write("<caption> 베스트셀러 </caption>");
06 document.write("<tr>");
07 document.write("<th> 순위 </th>");
08 document.write("<th> 제목 </th>");
09 document.write("</tr>");
10 document.write("<tr> <td> 1 </td> <td> " + title1 + "</td> </tr>");
11 document.write("<tr> <td> 2 </td> <td> " + title2 + "</td> </tr>");
12 document.write("<tr> <td> 3 </td> <td> " + title3 + "</td> </tr>");
```

출력



○ 대화상자로 메시지 출력

* 대화상자 (dialog box)로 입력을 받는 세 가지 방법

- alert(), prompt(), confirm()

◎ **alert()** 명령어

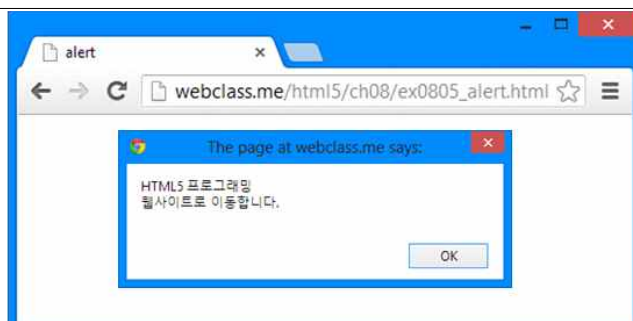
- 사용자에게 경고 사항이나 메시지를 전달

- “확인” 버튼을 클릭하지 않으면 다음 자바스크립트 문장이 실행되지 않음

/// alert() 명령어 예제

```
01 alert("HTML5 프로그래밍 \n 웹사이트로 이동합니다.");
```

출력



◎ confirm() 명령어

- 사용자에게 Yes/No 선택을 입력 => “확인”과 “취소” 버튼
- “확인” 버튼을 누르면 true, 취소 버튼을 누르면 false를 반환

/// alert() 명령어 예제	
01	var answer = confirm("주문한 서적을 결제하시겠습니까?");
02	document.write("Answer = " + answer + " ");
출력	

◎ prompt() 명령어

- 사용자에게 키보드를 통해 문자열을 입력
- “확인” 버튼을 누르면 입력된 문자열, “취소”를 누르면 null을 반환

/// alert() 명령어 예제	
01	var answer = prompt("서적 제목을 입력해 주세요.", "모바일 멀티미디어");
02	document.write("Answer = " + answer + " ");
출력	

※ 자바스크립트 제어문 및 반복문

※ 자바스크립트 함수

□

※

○

◎

*

-

/// 속성 선언	
	<p>선택자 속성선언 .</p> <p>선택자 { 속성:값, 속성:값, ... }</p> <p>ex) h3 {color:red; font-style:italic;}</p>

///	
01	
02	
03	
04	
05	
출력	