

6-3장. JavaScript 기본동작

(1) 역사

네스케이프사가 사이트의 처리능력을 높이기 위해 개발한 스크립트 언어
처음엔 Live Script라고 발표되었다가 선마이크로시스템의 자바의 기능을 결합
자바스크립트라 부르게 되었다.

* 스크립트란?

컴파일되지 않고 Application이 실행되는 동안 Line 단위로 해석되는 명령어나 문장들의 집합

(2) 특징

- 브라우저 안에서만 쓰이는 언어
- Html태그로는 부족한 동적인 효과를 내준다.
- Html태그에 쓰이는 명령어를 다시 응용해서 사용하게 된다.

(3) 자바와 자바스크립트의 차이

- 자바스크립트 : 넷스케이프에서 제작. 특별한 개발환경이 없다.
Client-side-script(클라이언트인 브라우저에서 소스자체가 해석된다.)
객체기반(Based)언어. 단 클래스를 지원하지 않는다.
변수형을 따로 선언할 필요가 없다.
동적Binding, 실행시 객체 참조가 체크된다.
- 자바 : 선마이크로시스템에서 제작. Visual J++이나 Visual Cafe, Latte의 개발환경
Server-side-script(서버에서 먼저 컴파일되고 그코드를 클라이언트에서 해석한다.)
객체지향(Oriented)언어. 자바애플릿은 클래스의 상속을 지원한다.
변수형을 반드시 선언해야 한다.
정적Binding. 컴파일시 객체참조가 체크된다.

(4) 기본구조

```
<script type="text/javascript"> ----> 스크립트의 종류 표기  
소스코드  
</script>
```

(5) 주석문 : 프로그램 실행에는 아무런 영향을 미치지 않는다.

```
// -----> 한줄 주석  
/* */ -----> 두줄 이상의 주석 : debugging의 용도로도 쓰임
```

(6) HTML문서내에서의 위치

가급적 <HEAD>...</HEAD>사이에 위치하는 것이 좋다
그러나, 페이지내 다른 객체와 어울려 작동시 또는 HEAD에서는 작동하지 않는 경우
BODY의 어떤 곳에도 사용가능

(7) 스크립트문 작성시 주의점

- 대소문자를 구분한다.
- 한 line (실행단위)을 끝내려면 ;(세미콜론)으로 닫아준다.
----> 요즘은 ;없이도 enter로 line구분이 가능하게 되었다.
- 한줄에 2문장을 기술할 때는 ;를 생략할 수 없다.

1. HTML 페이지에 결과 출력하기

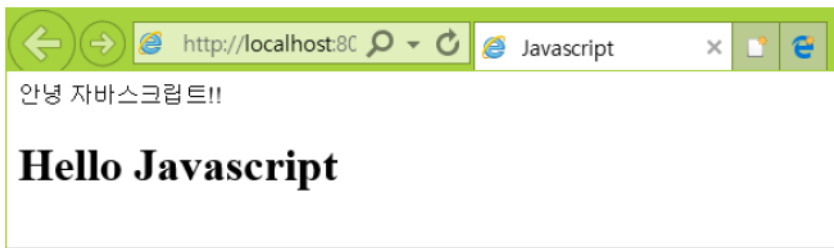
- HTML 페이지의 <body> 태그 안을 document(문서)라고 부른다.
- <body> 안에 새로운 내용을 출력
document.write("출력할 내용");
- <body> 태그안에 출력되는 내용이므로, 다른 HTML 태그를 포함하여 출력할 수 있다.

예제1. HTML 페이지에 결과 출력하기

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam01.html

<실행 결과>



2. 변수

- 변수를 사용하기 위해서는 “선언”과 “할당”의 두 영역으로 구분된다.
- 변수의 자료형은 값을 할당할 때 자동으로 결정된다.

예제2. 변수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam02.html

<실행 결과>



3. 연산자

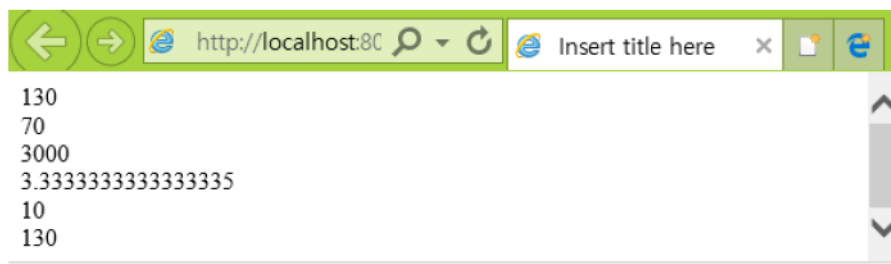
- 대입 연산자
- 산술 연산자
- 증감 연산자
- 비교 연산자
- 논리 연산자

예제3. 산술 연산자

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam03.html

<실행 결과>

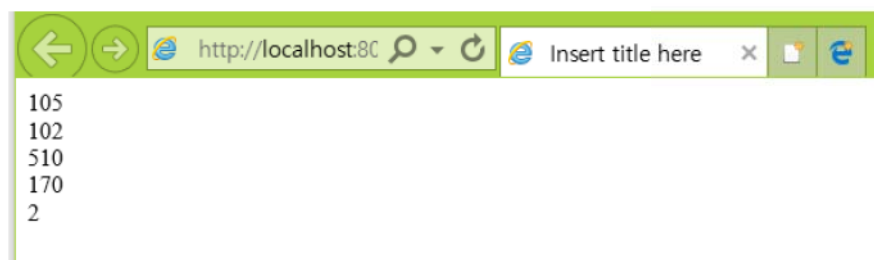


예제4. 대입 연산자

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam04.html

<실행 결과>

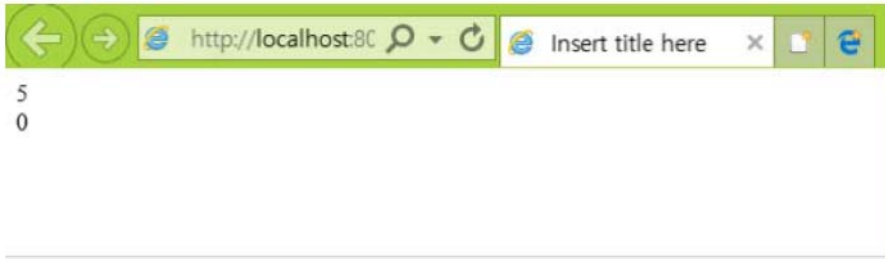


예제5. 증감 연산자

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam05.html

<실행 결과>



4. 제어문

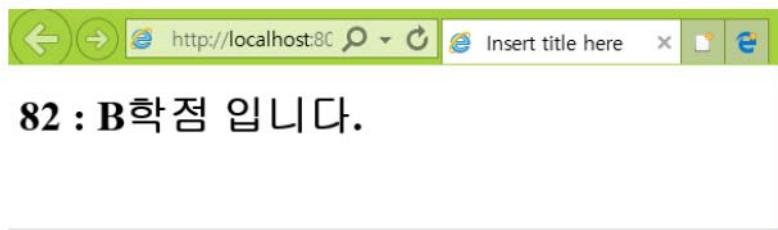
- 조건문 : if, switch
- 반복문 : for, while, do-while

예제6. if-else

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam06.html

<실행 결과>

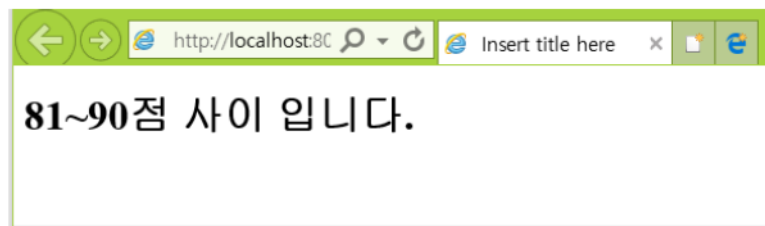


예제7. switch

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam07.html

<실행 결과>



예제8. for

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam08.html

<실행 결과>



5. 함수

- 어떤 기능을 구현해놓은 명령어들의 집합
- 함수의 사용방법은 “정의”와 “호출”의 두부분으로 나뉜다.

(1) 정의

```
function 함수이름() {  
    명령문;  
}
```

(2) 호출

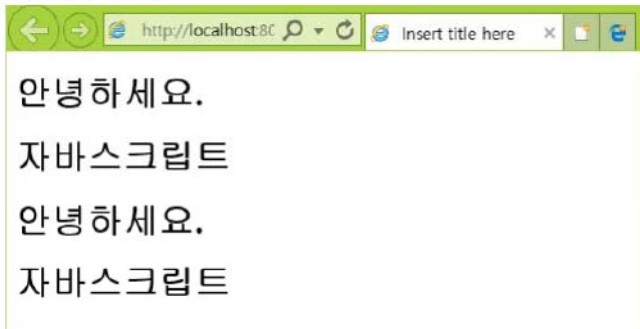
```
함수이름();
```

예제9. 함수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam09.html

<실행 결과>

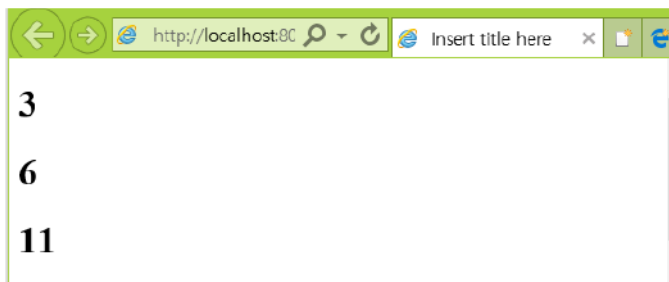


예제10. 파라미터를 갖는 함수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam10.html

<실행 결과>



예제11. return을 사용하는 함수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam11.html

<실행 결과>



6. 내장함수

- 웹페이지 작업을 하면서 자유롭게 호출하여 사용할 수 있는, 미리 정의되어 있는 함수

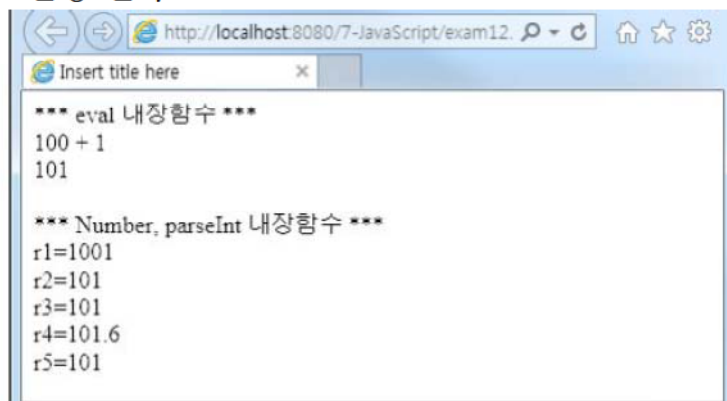
eval(문자열)	문자열을 수식으로 변환
Number(문자열)	문자열을 숫자로 변환
parseInt(문자열)	문자열을 정수로 변환
isNaN(문자열)	문자열이 숫자형식이 아니면 true, 숫자형식이면 false
alert(문자열)	문자열을 대화상자로 출력
confirm(문자열)	문자열을 확인/취소 대화상자로 출력
prompt(문자열)	문자열이 제목으로 표시되는, 값을 입력받기 위한 대화상자 호출

예제12. eval, Number, parseInt 내장함수

Dynamic Web Project : 7-JavaScript

HTML File : /WebContent/exam12.html

<실행 결과>

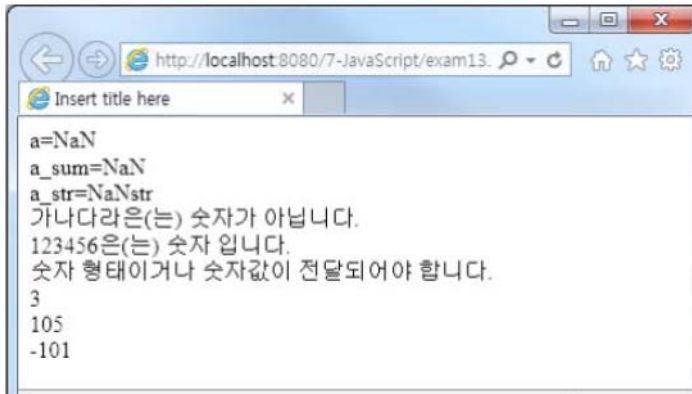


예제13. isNaN 내장함수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam13.html

<실행 결과>

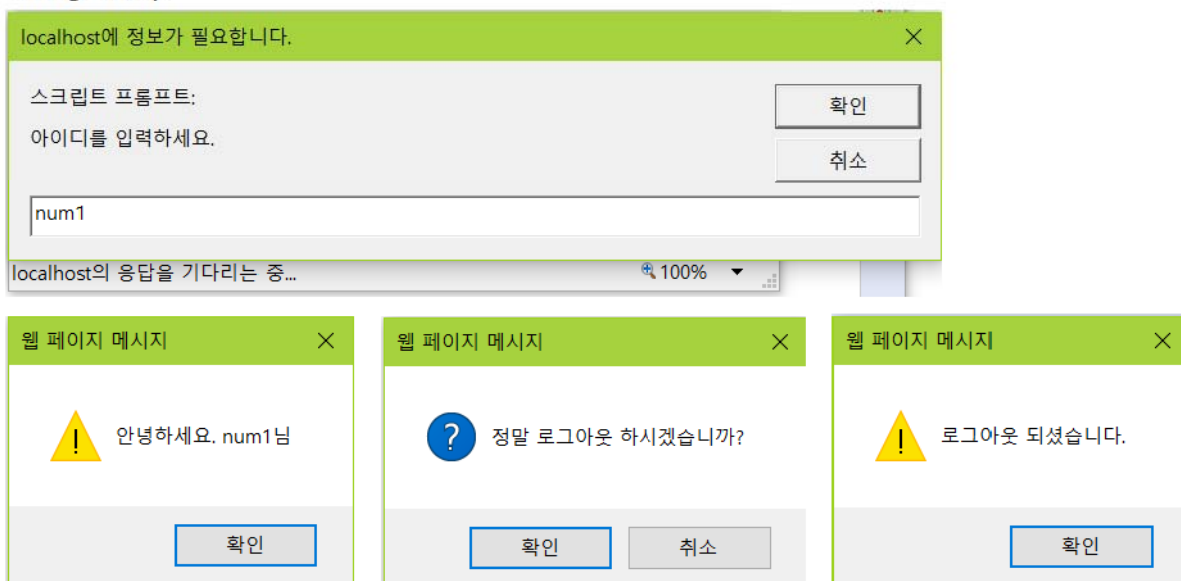


예제14. prompt, alert, confirm 내장함수

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam14.html

<실행 결과>



8. 이벤트 처리

- 특정한 상황이 발생했을 때, 사용자가 정의하는 특정한 동작(함수)들을 호출되도록 하는 것.
- Javascript를 사용하는 이유는 <body> 안의 어떠한 태그를 사용자의 이벤트에 반응하도록 제어하기 위해서 이다.

(1) 자주 사용되는 이벤트의 종류

onBlur	문서나 윈도우, 프레임셋, 폼 요소에서 현재 입력 포커스가 사라질 때
onClick	링크, 이미지맵 영역, 폼 요소가 클릭되었을 때
onFocus	문서나 윈도우, 프레임셋, 폼 요소에서 현재 입력 포커스가 놓였을 때
onLoad	이미지, 문서, 프레임이 로드될 때
onMouseOver	태그의 요소 위로 마우스 커서가 들어갈 때
onMouseOut	태그의 요소 위에서 마우스 커서가 빠져나갈 때

(2) 제어대상 지정

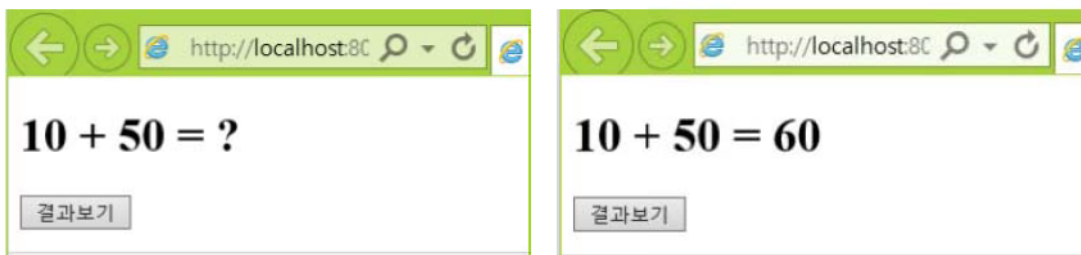
- 태그의 id로 제어할 특정 태그를 지정한다.
- document.getElementById("아이디") 로 자바스크립트로 태그를 가져온다.

예제15. 이벤트 처리 1

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam15.html

<실행 결과>

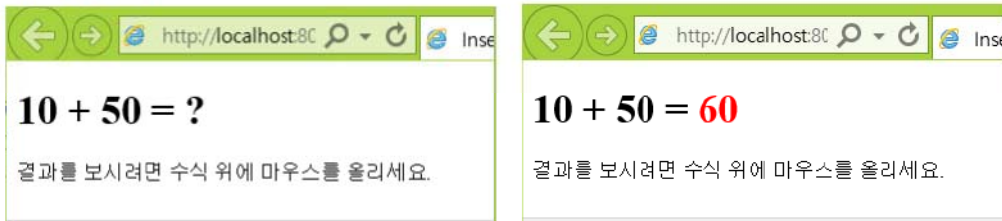


예제16. 이벤트 처리 2

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam16.html

<실행 결과>



예제17. 이벤트 처리 3

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam17.html

<실행 결과>



예제18. Javascript 구문의 실행 시점

- Javascript 구문의 실행 시점

- 1) HTML페이지는 웹 브라우저에 의해서 맨 처음부터 한 줄씩 읽혀서 실행된다.
- 2) doPrint()라는 함수를 호출하라는 명령어를 만나면 그 즉시, 해당 함수를 호출한다.
- 3) doPrint() 함수 안에서는 id속성값이 "myid"인 태그요소를 제어하려고 한다.
- 4) 하지만 웹 브라우저에게는 아직 HTML태그가 읽혀진 것이 아닌 시점이기 때문에 그 요소를 찾지 못하고 에러가 난다.
- 5) doPrint() 함수가 온전하게 실행되기 위해서는 이 함수에서 제어하려는 "myid"요소가 웹 브라우저에게 인식된 이후에 호출되어야 한다.

- 결론 : 제어하려는 태그가 인식된 이후에 Javascript 함수를 호출해야 한다는 제약은, Javascript소스와 HTML태그가 반드시 순서를 지켜가면서, 섞여서 코딩해야 한다는 제약을 가져온다. (스파게티 소스)
onload 이벤트는 이러한 제약을 해결하고 Javascript 코드와 HTML 태그를 분리할 수 있게 해 준다.

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam18.html

<실행 결과>

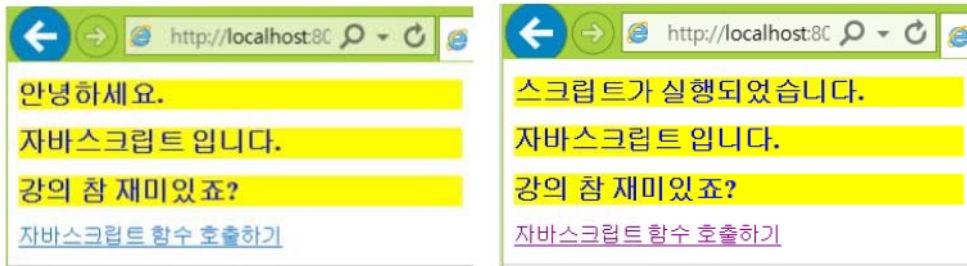


예제19. id 인식

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam19.html

<실행 결과>



9. 객체

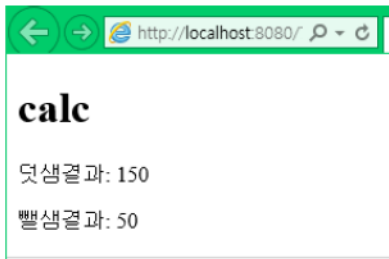
- 변수와 함수의 집합
- Javascript는 Prototype 기반의 객체지향언어이다.
 - => 객체를 원형(prototype) 형태로 생성하고, 그 안에 기능을 위한 함수나 변수를 추가하는 방법으로 그룹화 하는 개념

예제20. 객체

Dynamic Web Project : 7-JavaScript

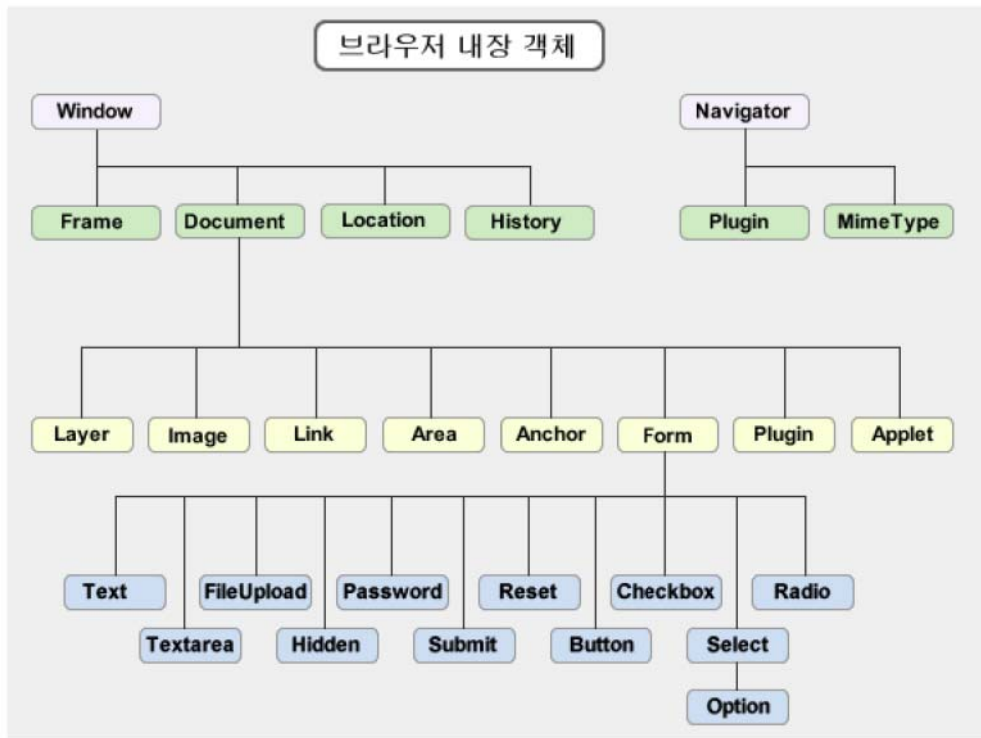
HTML File : /WebContent/exam20.html
/WebContent/calc.js

<실행 결과>



10. 내장객체

- 모든 웹 브라우저 개발사들이 공통기능의 필요성을 인지하고, 사전에 객체이름과 함수이름들을 통일하여 구현한 뒤, 브라우저 안에 내장시켜 놓았다.
- 이러한 객체를 활용하는 것 만으로 많은 기능들을 직접 구현하는 수고를 덜 수 있다.



(1) 값의 처리를 위한 내장객체

Date	시스템의 현재 날짜, 시각을 조회하거나 계산하기 위한 기능을 제공
Array	같은 종류의 변수를 하나로 묶기 위한 배열에 관련된 기능을 제공
String	하나의 문자열은 독립된 객체로 생성된다. 문자열 안에서 특정 글자가 시작하는 위치, 문자열 안에서 원하는 내용만 추출하는 등의 기능을 제공
Math	삼각함수, 지수, 로그 등 수학과 관련된 각종 고급함수를 제공

(2) 브라우저의 제어를 위한 내장객체

window	브라우저 창에 대한 모든 상황을 제어하는 최상위 객체. 모든 Javascript의 브라우저 제어관련 내장객체는 window 객체의 하위에 존재한다.
location	URL 정보를 제어하는 객체. 페이지 이동, 현재 주소 조회, 새로 고침 등의 기능을 제공
history	웹 브라우저에 기록되어 있는 히스토리 정보를 제어
navigator	브라우저의 종류를 판별한다.
screen	브라우저 화면에 대한 정보를 알려준다. 변수만 포함하고 있으며, 메소드는 포함하고 있지 않다.

(3) HTML문서를 제어하기 위한 내장객체

document	문서에 대한 정보, 즉 HTML문서의 각 요소들을 제어하기 위한 기능을 갖는다.
image	태그에 대한 속성을 제어하는 객체
form	입력양식 컴포넌트를 위한 개별 객체들을 포함한다.
frame	웹 페이지 안에 다른 웹 페이지를 포함하는 Frameset과 iframe을 제어하는 기능을 제공

예제21. Array 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam21.html

<실행 결과>

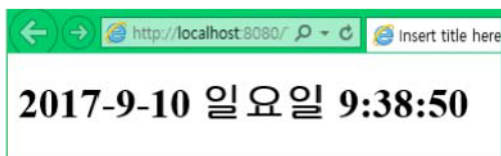


예제22. Date 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam22.html

<실행 결과>



예제23. window 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/open.html
/WebContent/exam23.html

<실행 결과>



예제24. location 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam24.html

<실행 결과>



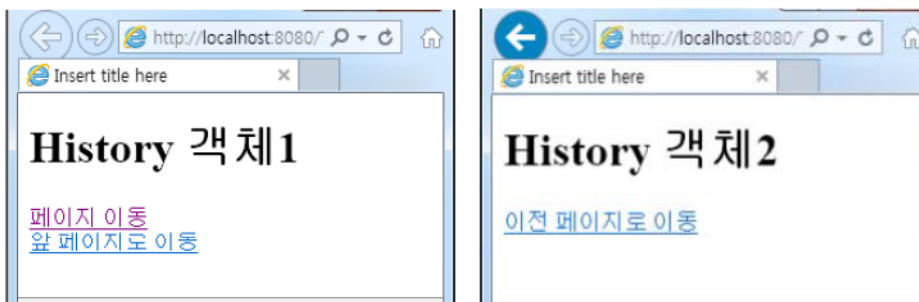
예제25. history 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam25.html

/WebContent/examHistory.html

<실행 결과>

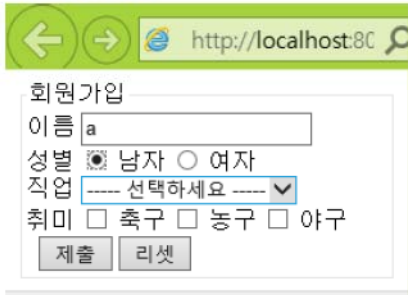


예제26. document 내장객체

Dynamic Web Project : 6-JavaScript

HTML File : /WebContent/exam26.html

<실행 결과>



회원가입

이름

성별 ☒ 남자 ☐ 여자

직업

취미 ☐ 축구 ☐ 농구 ☐ 야구