



CHAPTER 8.

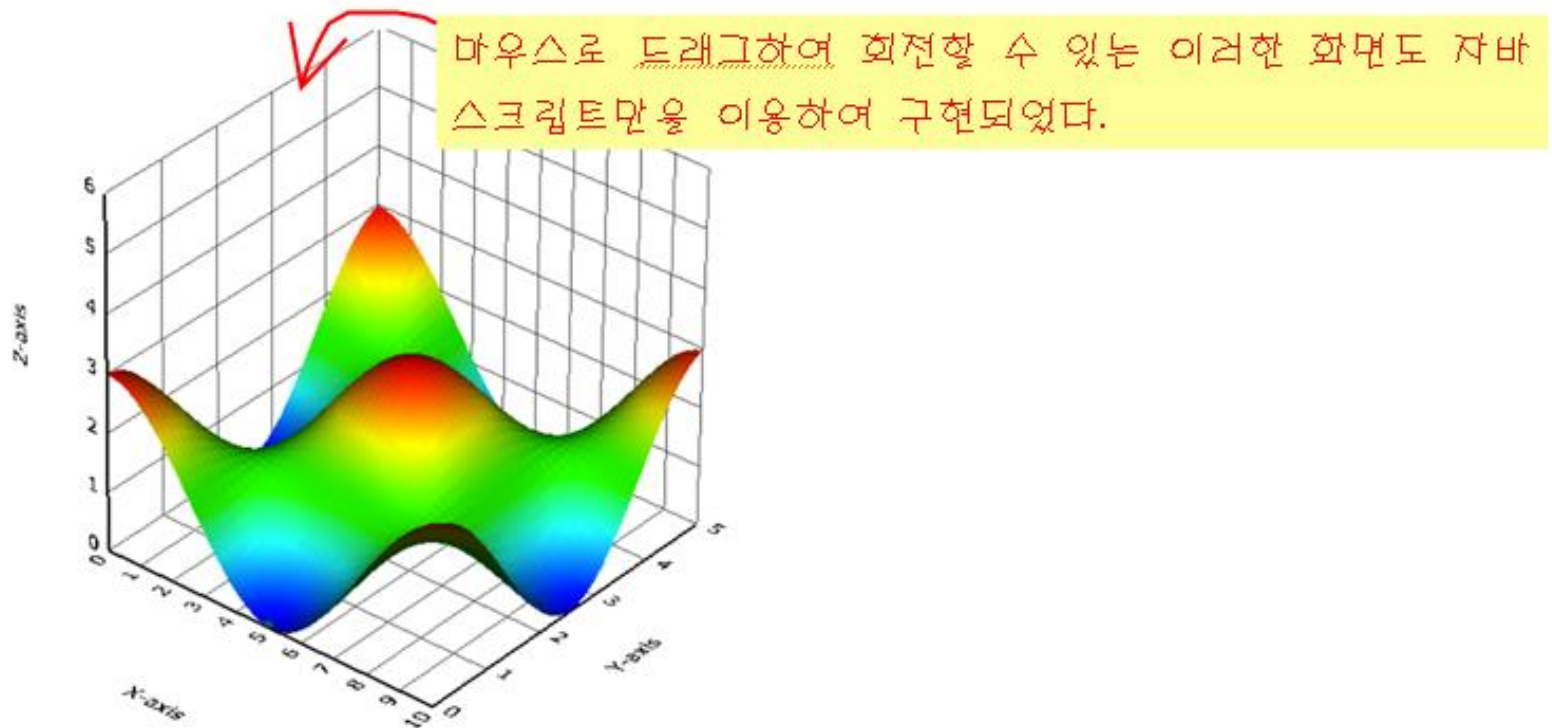
자바 스크립트 기초





자바 스크립트 소개

- 자바스크립트(javascript): 동적인 웹 페이지를 작성하기 위하여 사용되는 언어
- 웹의 표준 프로그래밍 언어
- 모든 웹브라우저들은 자바스크립트를 지원





HTML5 기술의 핵심





자바 vs 자바 스크립트

특징	자바 언어	자바스크립트
언어 종류	소스 파일을 컴파일하여 실행하는 컴파일 언어이다.	브라우저가 소스 코드를 직접 해석하여 실행하는 인터프리트 언어이다.
실행 방식	자바 가상 기계 위에서 실행한다.	브라우저 위에서 실행된다.
작성 위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수 선언	변수의 타입을 반드시 선언해야 함	변수의 타입을 선언하지 않아도 사용 가능



자바 스크립트 역사

- 넷스케이프의 브렌던 아이크(Brendan Eich)가 개발
- 처음에는 라이브스크립트(LiveScript)
- 최신 버전은 자바스크립트 1.8.5 (2010/7/27)
- ECMA(European Computer Manufacturer's Association)이 ECMAScript라는 이름으로 표준을 제정-> ECMA-262



브렌던 아이크(1961~)는 미국의 프로그래머이며, 프로그래밍 언어 자바스크립트의 창조자이다. 모질라 코퍼레이션에서 최고 기술 책임자를 맡고 있다.



자바 스크립트 특징

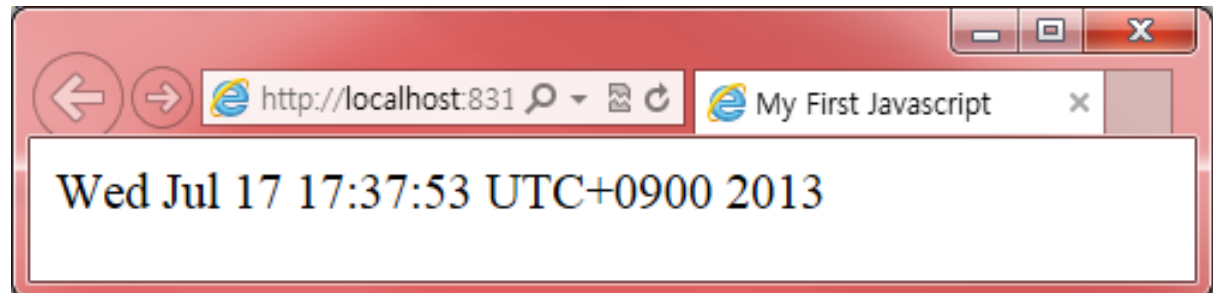
- 인터프리트 언어: 컴파일 과정 불필요
- 동적 타이핑(dynamic typing): 변수의 자료형을 선언하지 않고도 변수 사용 가능
- 구조적 프로그래밍 지원: if-else, while, for와 같은 제어 구조를 지원
- 객체 기반
- 함수형 프로그래밍 지원: 함수도 객체로 취급. 함수는 속성과 메소드를 갖는다.
- 프로토타입-기반(prototype-based): 상속을 위해 클래스 대신 프로토타입을 이용



첫 번째 예제

js.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title> My First Javascript </title>
</head>
<body>
  <script>
    var now = new Date();
    document.write(now);
  </script>
</body>
</html>
```





자바 스크립트의 용도

- 이벤트에 반응하는 동작을 구현
- **AJAX(Asynchronous JavaScript and XML)**
- **HTML** 요소들의 크기나 색상을 동적으로 변경
- 게임이나 애니메이션
- 사용자가 입력한 값들을 검증



자바 스크립트의 위치

- 내부 자바스크립트
- 외부 자바스크립트
- 인라인 자바스크립트



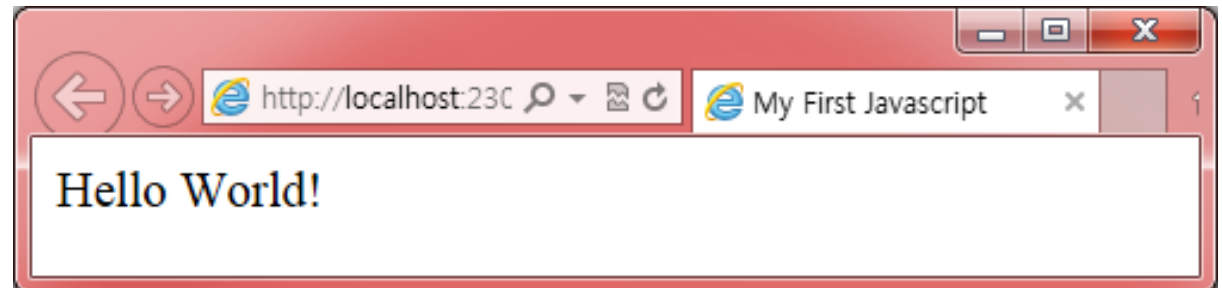
내부 자바 스크립트

```
<!DOCTYPE HTML>
<html>
<head>

  <title>My First Javascript </title>

  <script>
    document.write("Hello World!");
  </script>

</head>
<body></body>
</html>
```



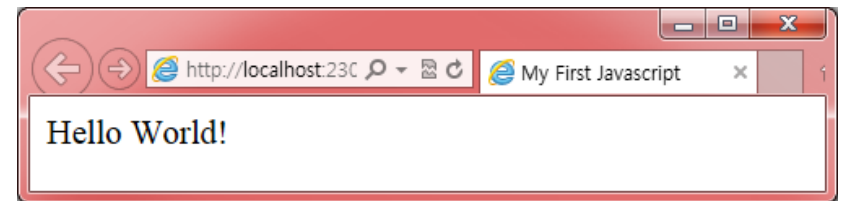


외부 자바 스크립트

```
<!DOCTYPE html>
<html>
<head>
  <script src="myscript.js"></script>
</head>
<body>
</body>
</html>
```

myscript.js

```
document.write("Hello World!");
```

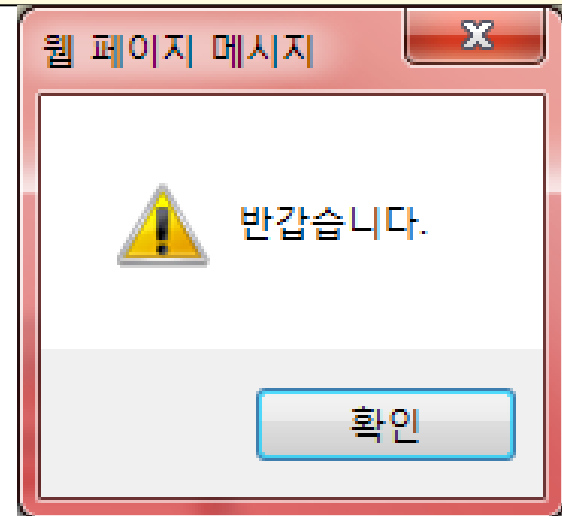
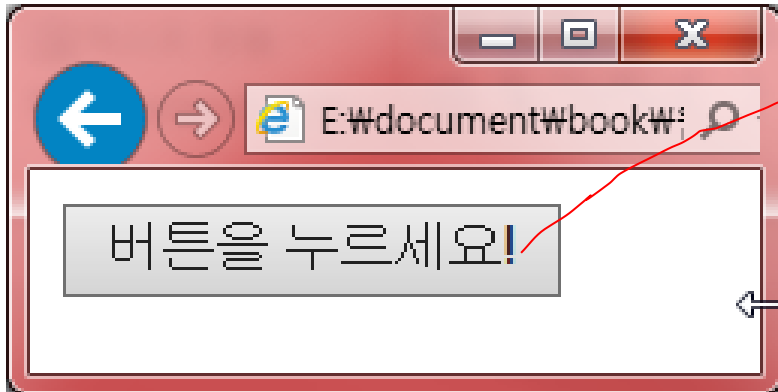




인라인 자바 스크립트

inline.html

```
<!DOCTYPE html>
<html>
<body>
  <button type="button" onclick="alert('반갑습니다.')">버튼을
  누르세요!</button>
</body>
</html>
```





문장

- 자바스크립트 문장(statement)들은 웹 브라우저에게 내리는 명령

1. 문서에 "Hello World!"를 추가하시오.
2. 화면에 경고창을 띄우시오.
3. 변수를 하나 만드시오.
4. ...



```
document.write("Hello World!");  
alert("warning!!");  
var count;  
...
```

순차적
으로
실행된
다.



변수

- 변수(**variable**)는 데이터를 저장하는 상자
- `var` 키워드를 사용하여 선언(**declare**)한다.

변수 선언할 때 자료형이 없다!

변수 이름이다. 스크립트 안에서 유일해야 한다.

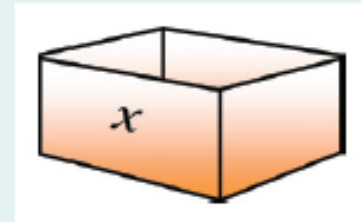
var

x

;

var 키워드는 변수가 선언된다는 것을 나타낸다.

세미콜론은 하나의 문장이 끝났음을 나타낸다.



자바스크립트는 대소문자를 구별한다.

HTML은 대소문자를 구별하지 않는다.

자바스크립트에서 **HTML**의 태그와 속성 이름은 소문자로 적어야 한다.



예제

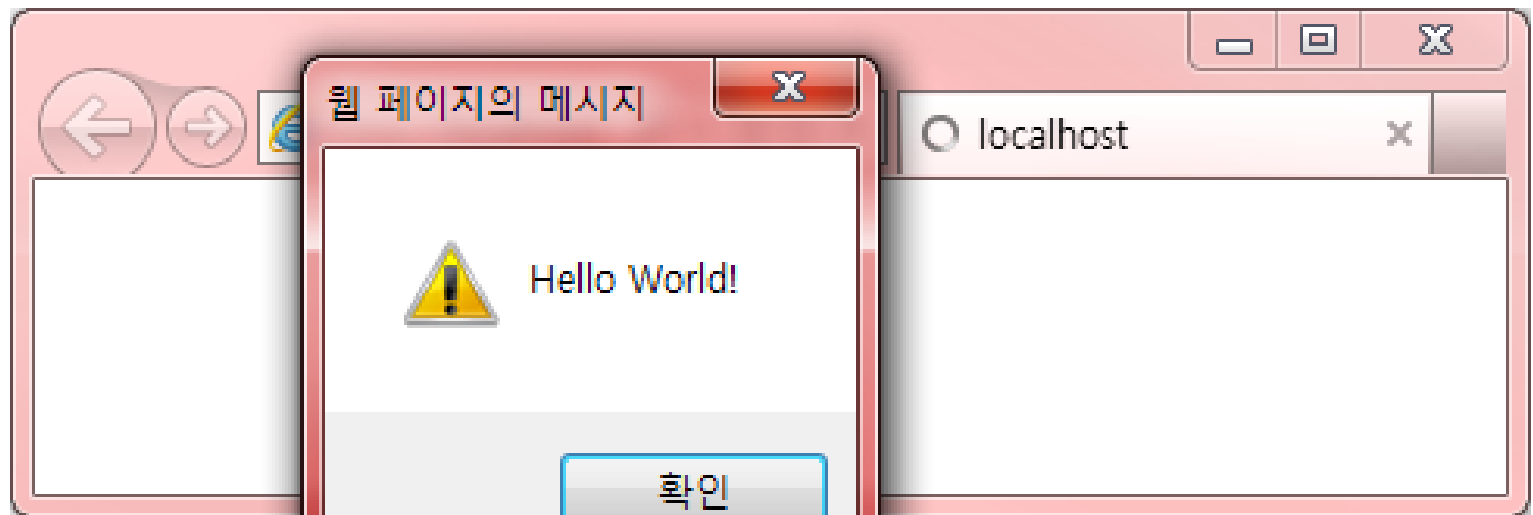
- 자바 스크립트 변수는 어떤 형태의 값도 모두 저장할 수 있다.

```
var x;  
x = "Hong";  
x = 10;
```



예제

```
<script>  
  //예제입니다.  
  var x;  
  x = "Hello World!";  
  alert(x);  
  /*  
    다중 주석은 /* */를 사용합니다.  
  */  
</script>
```





자료형

- 수치형(number)
- 문자열(string)
- 부울형(boolean)
- 객체형(object)
- undefined

변수 선언할 때는 자료형이 필요 없지만 특정한 값이 변수에 저장되는 순간 자료형이 내부적으로 결정된다.

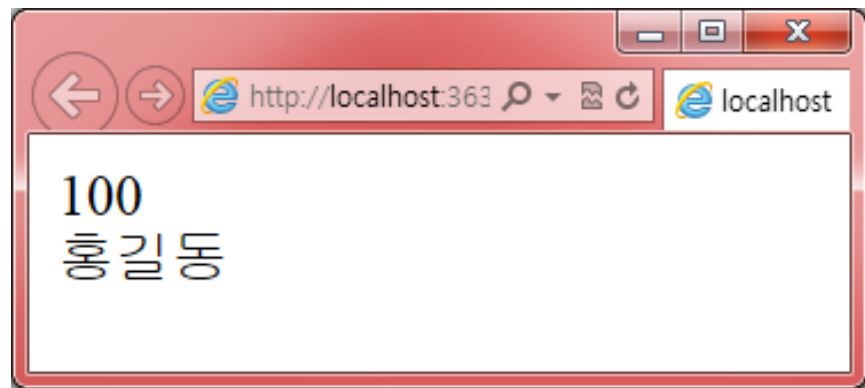


예제

```
<script>
  var s;

  s = 100;
  document.write(s + "<br>");

  s = "홍길동";
  document.write(s + "<br>");
</script>
```

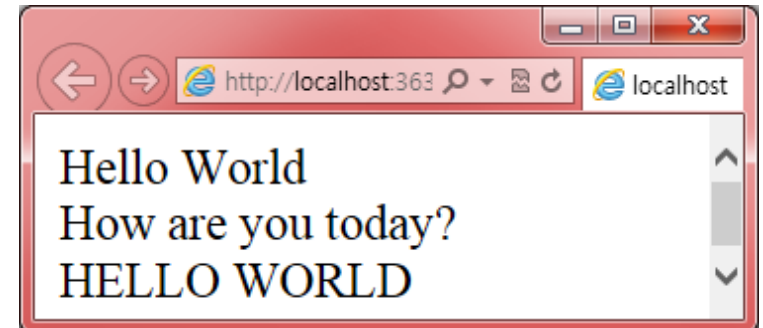




예제

```
<script>
  var s = "Hello World";
  var t = "How are you" + " today?";

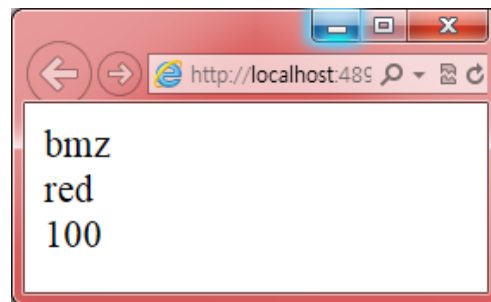
  document.write(s + "<br>");
  document.write(t + "<br>");
  document.write(s.toUpperCase() + "<br>");
</script>
```





객체형

- 객체(object)는 사물의 속성과 동작을 묶어서 표현하는 기법
- (예) 자동차는 메이커, 모델, 색상, 마력과 같은 속성도 있고 출발하기, 정지하기 등의 동작도 가지고 있다.
- `var myCar = { model: "bmz", color: "red", hp: 100 };`
- `document.write(myCar.model + "
");`
- `document.write(myCar.color + "
");`
- `document.write(myCar.hp + "
");`





연산자

연산자	설명	예	수식의 값
+	덧셈	$x=3+2$	5
-	뺄셈	$x=3-2$	1
*	곱셈	$x=3*2$	6
/	나눗셈	$x=3/2$	1.5
%	나머지	$x=3\%2$	1
++	증가	$++x$	x의 값 3 → 4
--	감소	$--x$	x의 값 3 → 2

++x와 x++의 차이

```
var x= 10;  
var y;  
y = ++x; //y 값은?  
y = x++; //y 값은?
```

문자열에서의 + 연산자

```
var x= 1+1;  
var y = "Car"+1;  
document.write(x+"<br>");  
document.write(y+"<br>");
```

조건연산자

```
max_value = (x>y)? x: y;
```



실습

- $x=10$ 일 때 $++x$ 와 $x++$ 의 결과를 웹브라우저에 출력하는 자바스크립트 코드를 작성하시오.
- $x=1$ 이고 $y="1"$ 일 때 $x==y$ 와 $x===y$ 의 실행결과를 웹브라우저에 출력하는 자바스크립트 코드를 작성하시오.



prompt() 함수

```
<script>  
  var age = prompt("나이를 입력하세요", "만나이로 입력합니다.");  
</script>
```



- prompt 함수는 사용자가 입력한 내용을 문자열로 반환하기 때문에 숫자를 원할 경우 문자열을 숫자로 변환하는 기능을 하는 “parseInt()” 함수를 사용한다.
- 참고로 숫자를 문자열로 변환시킬 때는 “string()” 함수를 사용



예제

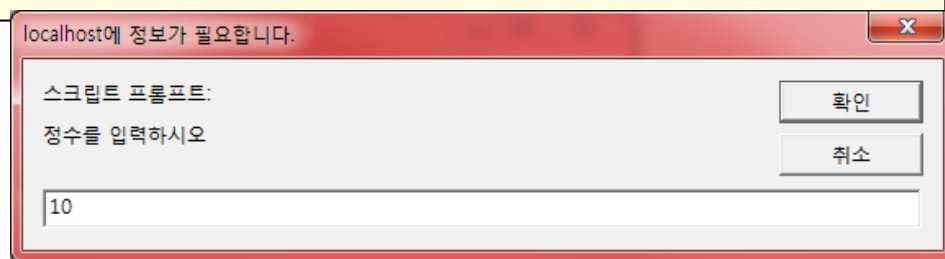
```
<script>
  var x, y;
  var input;

  input = prompt("정수를 입력하시오", "정수로");
  x = parseInt(input);

  input = prompt("정수를 입력하시오", "정수로");
  y = parseInt(input);

  document.write(x + y + "<br>");
</script>
```

웹브라우저에서 실행





예제

```
<html>
<head>
  <title>Calculator</title>
  <script>
    function calc() {

      var x = document.getElementById("x").value;

      var y = document.getElementById("y").value;
      var sum;

      sum = parseInt(x) + parseInt(y);
      document.getElementById("sum").value = sum;
    }
  </script>
</head>
```



예제

```
<body>
  <h3>덧셈 계산기</h3>
  <form name="myform" action="..." method="POST">
    첫번째 정수:
    <input id="x" /><br />
    두번째 정수:
    <input id="y" /><br />
    합계:
    <input id="sum" /><br />
    <input type="button" value="계산" onclick="calc();" />
  </form>
</body>
</html>
```

웹브라우저에서 실행

덧셈 계산기

첫번째 정수: 10

두번째 정수: 20

합계: 30

계산

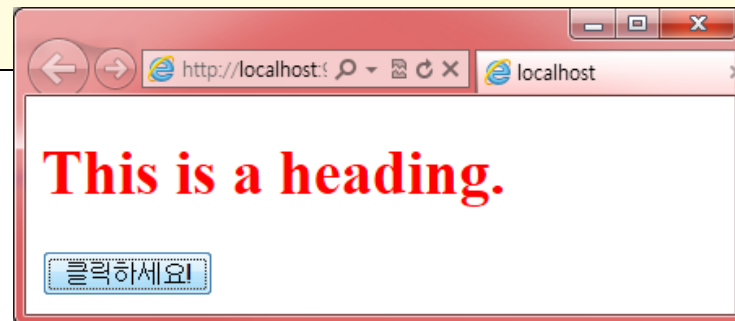


HTML 요소에 접근하기

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

  <h1 id="test">This is a heading.</h1>
  <script>
    function func() {
      e = document.getElementById("test");
      e.style.color = "red";
    }
  </script>
  <button type="button" onclick="func()">클릭하세요!</button>
</body>
</html>
```

웹브라우저에서 실행





실습

- 교재 296페이지의 예제와 교재 297페이지를 응용하여 두 개의 정수를 입력 받은 후 두 수를 곱한 결과를 웹브라우저에 출력하는 계산기 함수를 작성하시오. 단 계산 결과는 빨간색으로 표시되어야 함.

input X:

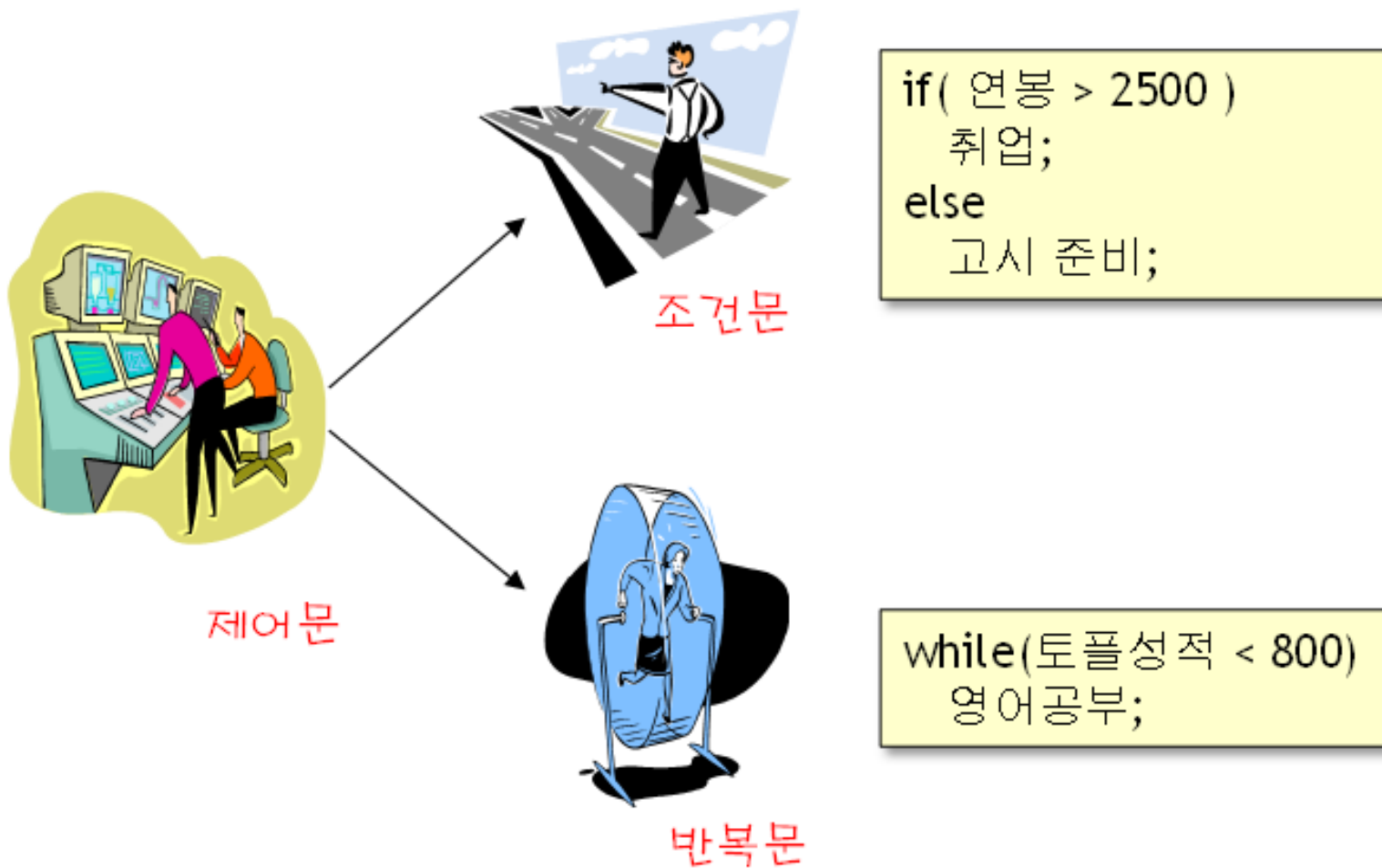
input Y:

X x Y = :

[chap8 Calc.html](#)



제어문



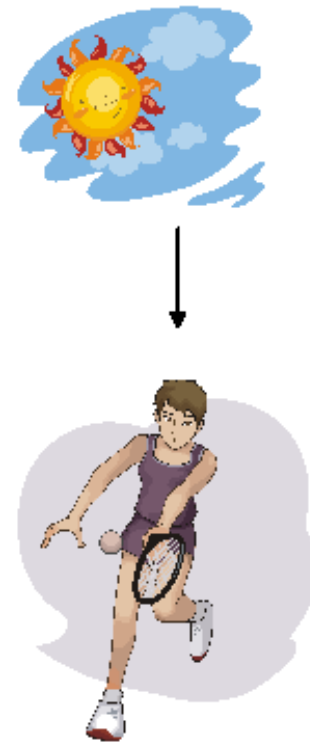
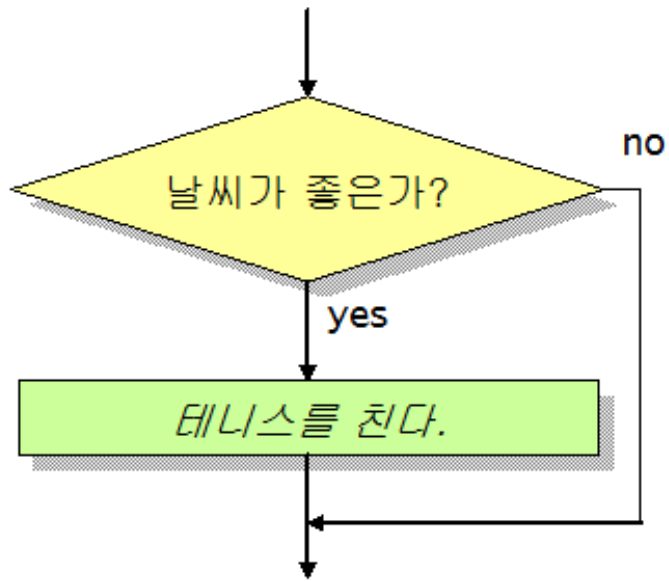


조건문의 종류

- if 문
- if...else 문
- switch 문



if 문



```
if (time<12)
{
    greeting="Good Morning!";
}
```



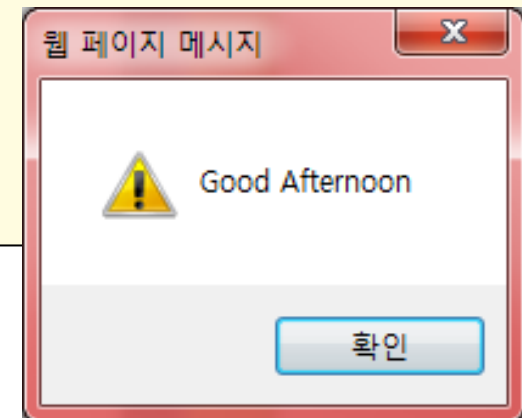
if-else 문

```
if (time<12)
{
    msg="Good Morning!";
}
else{
    msg="Good Afternoon!";
}
```




연속적인 if 문

```
<script>
  var msg = "";
  var time = new Date().getHours();
  if (time < 12) {                                // 12시 이전이면
    msg = "Good Morning";
  }
  else if (time < 18) {                            // 오후 6시 이전이면
    msg = "Good Afternoon";
  }
  else {                                           // 그렇지
    msg = "Good evening";                        // 않으면(오후 6시 이후이면)
  }
  alert(msg);
</script>
```





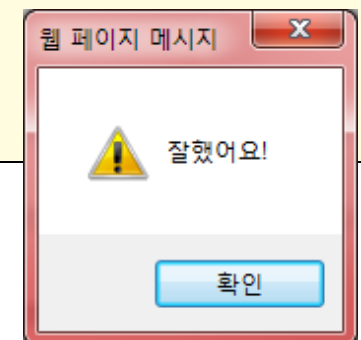
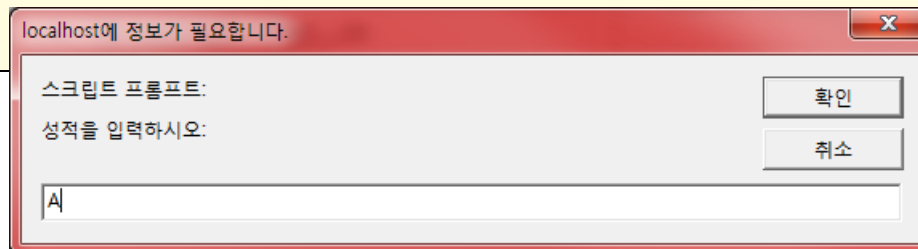
실습

- 사용자로부터 점수를 입력 받아 90점 이상은 A, 80~90점은 B, 70~80점은 C, 70점 이하는 D를 출력하는 스크립트를 작성하시오(prompt 사용, if 사용).



switch 문

```
<script>
  var grade = prompt("성적을 입력하시오:", "A-F사이의 문자로");
  switch (grade) {
    case 'A': alert("잘했어요!");
              break;
    case 'B': alert("좋은 점수군요");
              break;
    case 'C': alert("괜찮은 점수군요");
              break;
    case 'D': alert("좀더 노력하세요");
              break;
    case 'F': alert("다음학기 수강하세요");
              break;
    default: alert("알수없는 학점입니다.")
  }
</script>
```





숫자 게임 예제

```
<html>
<head>
  <title></title>
  <script>
    var computerNumber = 53; // 정답
    var nGuesses = 0;        // 추측 횟수
    function guess() {
      var result = "";       // 결과 메시지
      // 사용자가 입력한 값을 받아서 변수 number에 대입한다.
      var number = parseInt(document.getElementById("user").value);
      nGuesses++;            // 추측 횟수를 증가시킨다.
      if (number == computerNumber) result = "성공입니다.";
      else if (number < computerNumber) result = "낮습니다.";
      else result = "높습니다.";
      document.getElementById("result").value = result;
      document.getElementById("guesses").value = nGuesses;
      return true;
    }
  </script>
</head>
```



숫자 게임 예제

```
<body>
```

```
  <h2>숫자 맞추기 게임</h2>
```

이 게임은 컴퓨터가 생성한 숫자를 맞추는 게임입니다. 숫자는 1부터 100 사이에 있습니다.

```
  <form>
```

숫자:

```
    <input type="text" id="user" size="5">
```

```
    <input type="button" value="확인" onclick="guess();">
```

추측횟수:

```
    <input type="text" id="guesses" size="5">
```

힌트:

```
    <input type="text" id="result" size="16">
```

```
  </form>
```

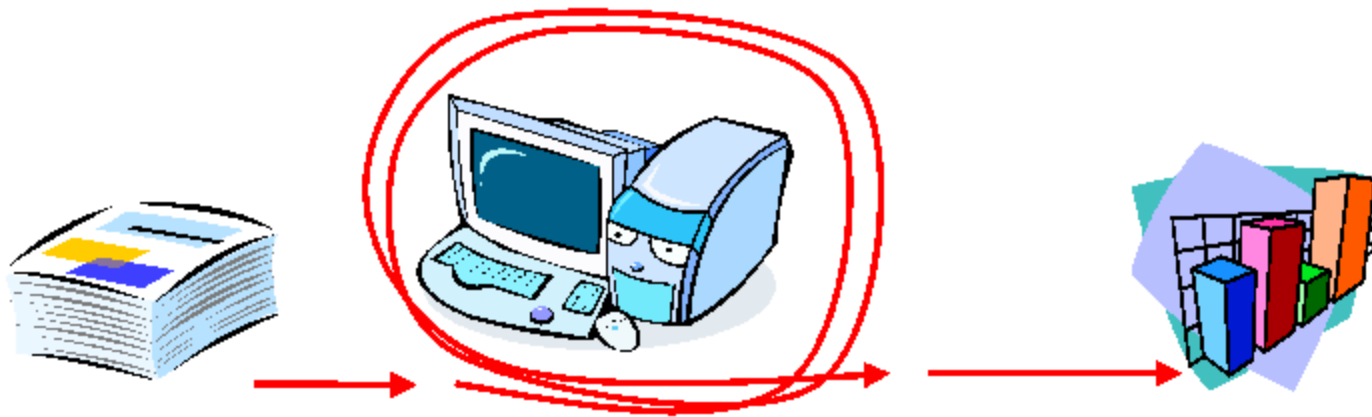
```
</body>
```

```
</html>
```



반복문

- 같은 처리 과정을 여러 번 되풀이하는 것





반복문의 종류

- **while** – 지정된 조건이 참이면 반복 실행한다.
- **for** – 정해진 횟수 동안 코드를 반복 실행한다.



while 문

```
var i = 0;
```

```
while( i < 10 ) ←반복 조건, 변수 i가 10보다 작으면 반복 계속
```

```
{
```

```
    document.write(i+" <br>"); ←반복되는 문장, i를 화면에 출력한다.
```

```
    i++; ←한 번의 반복마다 i를 하나 증가시킨다.
```

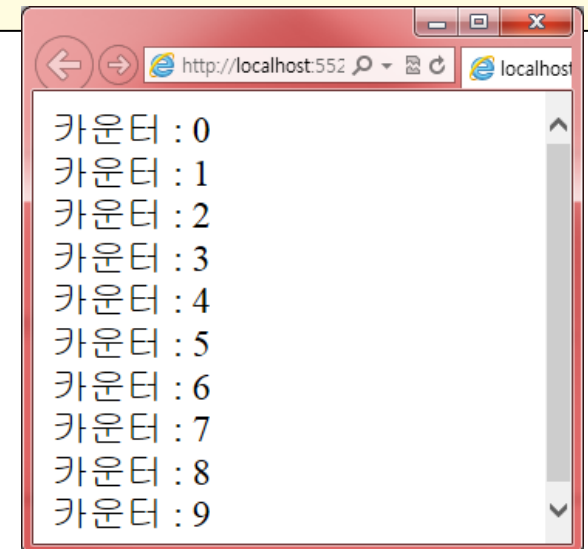
이 문장이 없으면 무한히 반복한다.

```
}
```




while 문

```
<script>
  var i = 0;
  while (i < 10) {
    document.write("카운터 : " + i + "<br />");
    i++;
  }
</script>
```





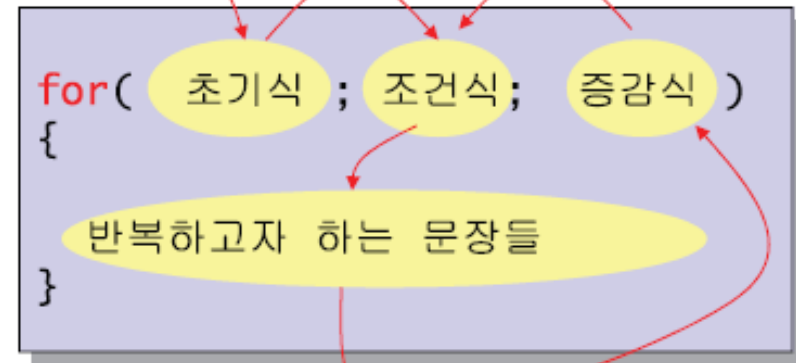
for 문

초기식 조건식 증감식

↓ ↓ ↓

```
for( i=0   ; i<10 ; i++ )  
{  
    document.write(i+"<br />");  
}
```

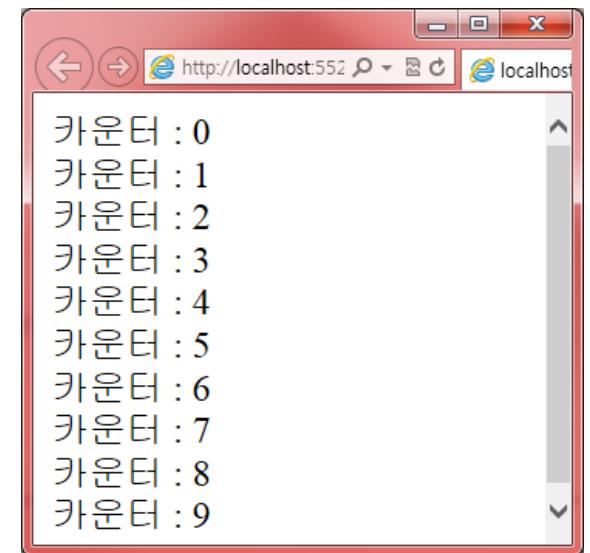
반복되는 문장





for 문

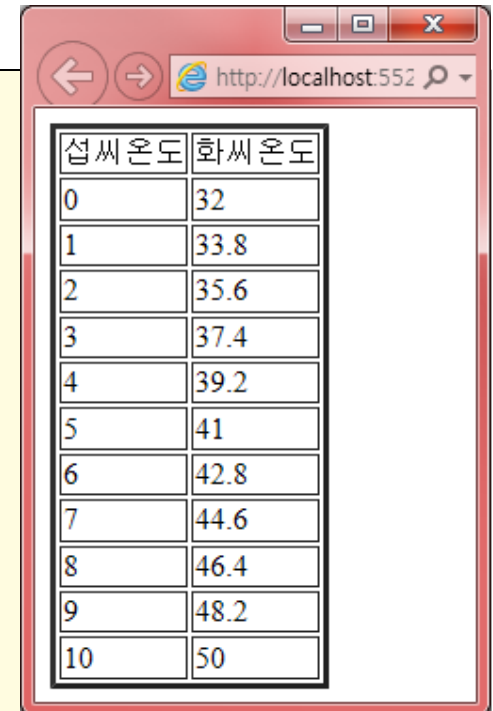
```
<script>
  var i = 0;
  for (i = 0; i < 10; i++) {
    document.write("카운터 : " + i + "<br />");
  }
</script>
```





예제

```
<html>
<head>
  <title>온도 변환기</title>
</head>
<body>
  <table border="3">
    <tr>
      <td>섭씨 온도</td>
      <td>화씨 온도</td>
    </tr>
    <script>
      for (celsius = 0; celsius <= 10; celsius = celsius + 1) {
        document.write("<tr><td>" + celsius + "</td><td>"
          + ((celsius * 9.0 / 5) + 32) + "</td></tr>");
      }
    </script>
  </table>
</body>
</html>
```



섭씨 온도	화씨 온도
0	32
1	33.8
2	35.6
3	37.4
4	39.2
5	41
6	42.8
7	44.6
8	46.4
9	48.2
10	50



중첩 반복문 예제

```
<script>
  document.write("<h1>구구단표</h1>");
  document.write("<table border=2 width=50%");

  for (var i = 1; i <= 9; i++) {
    document.write("<tr>");
    document.write("<td>" + i + "</td>");

    for (var j = 2; j <= 9; j++) {
      document.write("<td>" + i * j + "</td>");
    }

    document.write("</tr>");
  }
  document.write("</table>");
</script>
```

구구단표

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81



for 반복문 예제

```
for (var i = 0, len = fruits.length; i < len; i++) {  
    document.write(fruits[i] + "<br>");  
}
```

```
<script>  
    var myCar = {make: "BME", model: "X5", year: 2013};  
    var txt = "";
```

```
    for (var x in myCar) {  
        txt += myCar[x] + " ";  
    }
```

객체 내의 모든 속성에
대하여 반복

```
    document.write(txt);
```

```
</script>
```

[chap8 js for in.html](#)



break와 continue

- Break문: 반복 루프를 벗어나는 것이 목적이다.

```
<script>
  var msg = "";
  for (var i=0; i<10; i++){
    if (i==3) break;
    msg += i + "<br>";
  }
  document.write(msg);
</script>
```

- Continue문: 현재 실행하고 있는 반복 과정의 나머지를 생략하고 다음 반복을 시작한다.

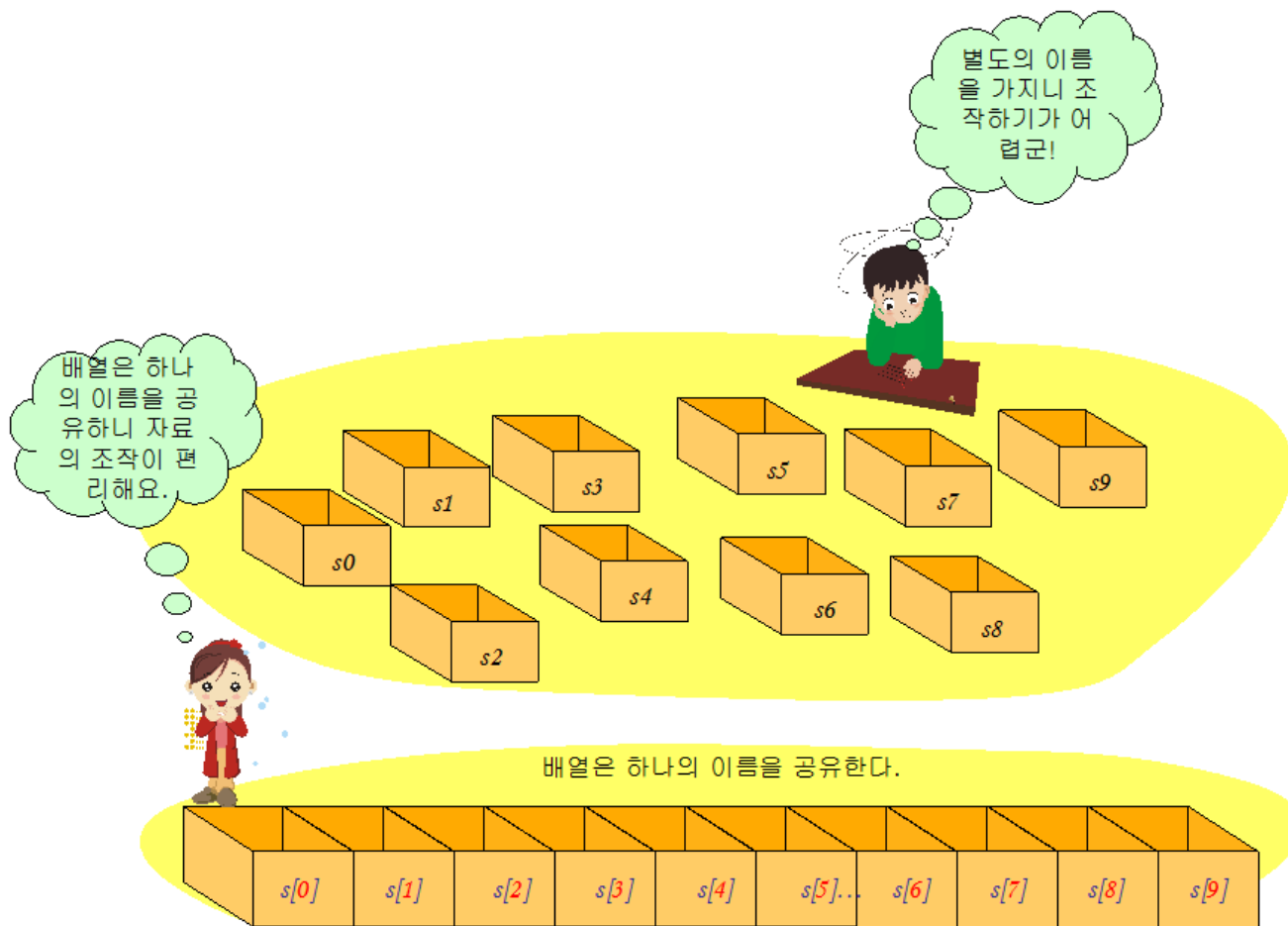
```
<script>
  var msg = "";
  for (var i=0; i<10; i++){
    if (i==3) continue;
    msg += i + "<br>";
  }
  document.write(msg);
</script>
```

0
1
2
4
5
6
7
8
9



배열

- 많은 값을 저장할 수 있는 공간이 필요할 때 배열을 사용한다.
- 서로 관련된 데이터를 차례로 접근하여서 처리할 수 있다.





배열을 생성하는 2가지 방법

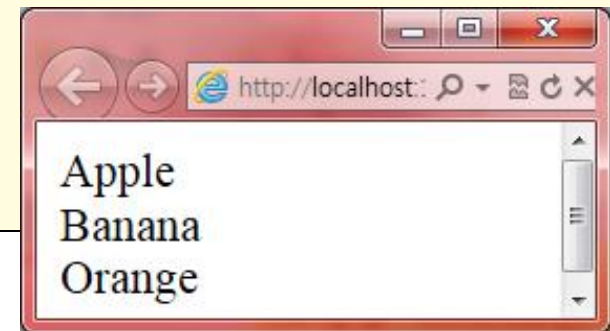
- 리터럴로 배열 생성
 - `var fruits = ["apple", "banana", "peach"];`
- Array 객체로 배열 생성
 - `var fruits=new Array("apple", "banana", "orange");`



예제

```
<!DOCTYPE html>
<html>
<body>
<script>
  var i;
  var fruits = new Array();
  fruits[0] = "Apple";
  fruits[1] = "Banana";
  fruits[2] = "Orange";

  for (i = 0; i < fruits.length; i++) {
    document.write(fruits[i] + "<br>");
  }
</script>
</body>
</html>
```





예제

```
var comp = new Array();  
comp[0] = "Apple";  
comp[1] = new Date();  
comp[2] = 3.14;
```

`Array.pop()`: 배열의 마지막 요소를 반환한 후 해당 요소를 배열에서 제거한다.
`Array.push()`: 배열에 새로운 요소를 추가한다(**LIFO** 스택 방식)

`Array.length`: 배열의 크기를 알 수 있다.



연관 배열

```
<!DOCTYPE html>
<html>
<body>
  <form name="myForm">
    필드1<input type="text" name="a0"><br />
    필드2<input type="text" name="a1"><br />
    필드3<input type="text" name="a2"><br />
    <input type="button" value="초기화" onclick="init();">
  </form>
  <script>
    function init() {
      for (var i = 0; i < 3; i++) {
        document.myForm["a" + i].value = i;
      }
    }
  </script>
</body>
</html>
```

자세한 설명은
10장에서

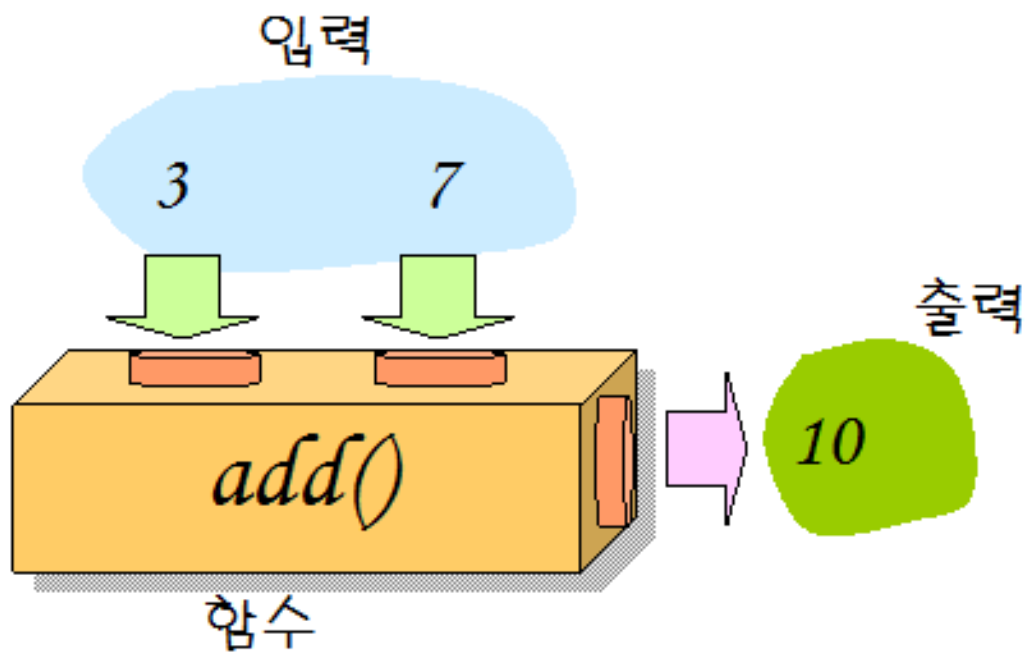
필드1	0
필드2	1
필드3	2

초기화



함수

- 함수는 입력을 받아서 특정한 작업을 수행하여서 결과를 반환하는 블랙 박스



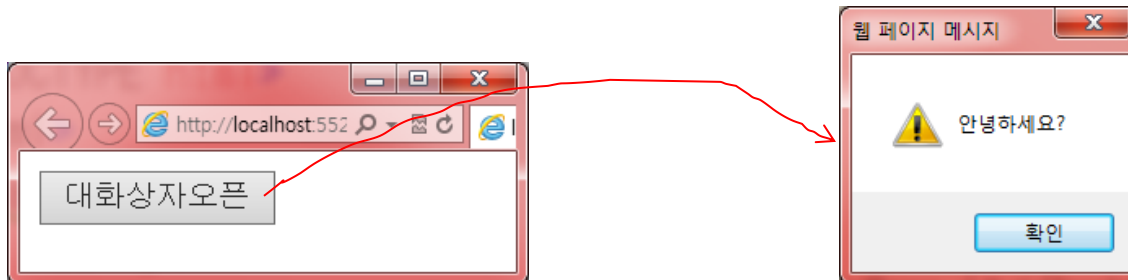
함수는 이름을 가지며 입력을 받아서 특정한 작업을 실행하고 결과를 반환합니다.





예제

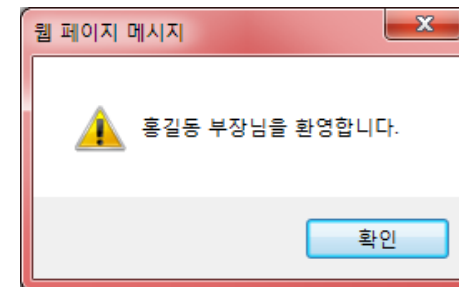
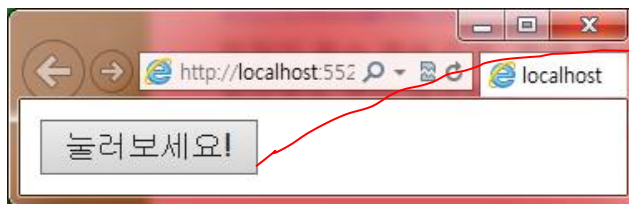
```
<!DOCTYPE html>
<html>
<head>
  <script>
    function showDialog() {
      alert("안녕하세요?");
    }
  </script>
</head>
<body>
  <button onclick="showDialog()">대화상자오픈</button>
</body>
</html>
```





인수와 매개 변수

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function greeting(name, position) {
      alert(name + " " + position + "님을 환영합니다.");
    }
  </script>
</head>
<body>
  <button onclick="greeting('홍길동', '부장')">눌러보세요!</button>
</body>
</html>
```





무명 함수 (anonymous function)

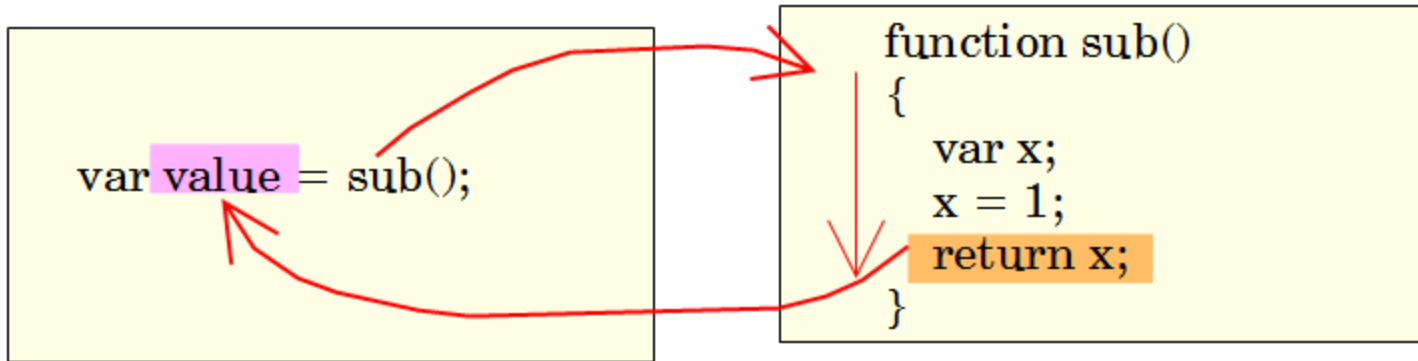
- 함수에 이름을 주지 않고 만들어서 사용하는 함수

```
var greeting = function()  
    {  
        alert("안녕하세요")  
    };  
  
greeting();
```




함수의 반환값

- return 문장을 사용하여 외부로 값을 반환





지역 변수와 전역 변수

- 지역 변수

- 함수 안에서 선언된 변수
- 함수 안에서만 사용 가능
- 다른 함수에서도 같은 이름으로 선언 가능
- 함수가 종료되면 소멸된다.

```
function add(a, b) {  
  var sum = 0;  
  sum = a + b;  
  return sum;  
}
```

지역 변수로서 add() 내부에서만 사용이 가능하다.

- 전역 변수

- 함수 외부에서 선언된 변수
- 웹 페이지 상의 모든 스크립트와 모든 함수는 전역 변수를 사용할 수 있다.

```
var sum = 0;
```

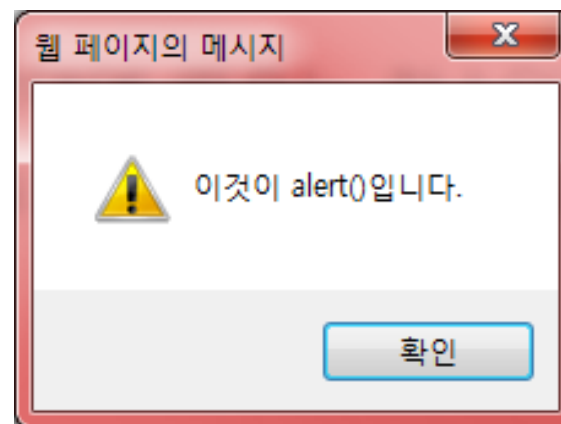
전역 변수로서 누구나 사용할 수 있다.

```
function add(a, b) {  
  sum = a + b;  
  return sum;  
}
```



alert() 함수

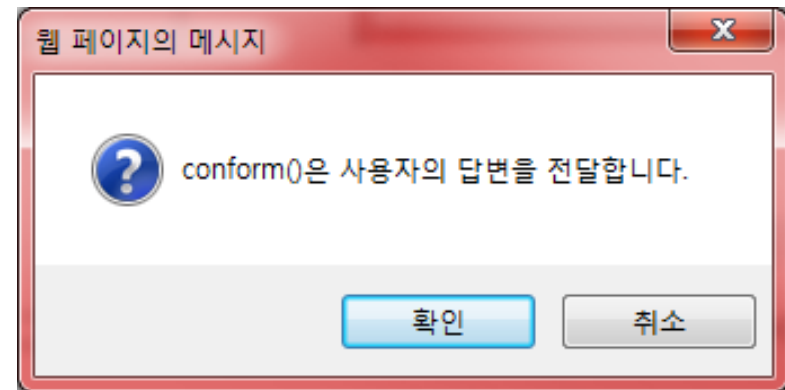
```
<script>  
  alert("이것이 alert()입니다.");  
</script>
```





confirm() 함수

```
<script>  
  var user = confirm("confirm()은 사용자의 답변을 전달합니다.");  
</script>
```





prompt() 함수

```
<script>  
  var age = prompt("나이를 입력하세요", "만나이로 입력합니다.");  
</script>
```





실습

- 다음 예제와 같이 실행하면 행렬(크기 4)의 값을 사용자로부터 입력 받은 후 사용자가 행렬의 인덱스를 입력하면, 해당 행렬 값을 더하여 화면에 출력하는 JavaScript 코드를 작성하시오.
 - 296페이지, 314 페이지 예제 참고
 - `prompt()` 사용



Q & A

