

JavaScript 객체

내장객체

내장 객체와 BOM객체

- 자바스크립트 자체에서 지원하는 객체
 - String, Math, Date, Array 객체
 - eval, parseInt, parseFloat 등의 메소드
- 웹 브라우저 환경에서 지원하는 객체 (BOM)
 - parents, frame 등의 윈도우 객체
 - 흔히 사용하는 document, form 객체 등

Array 객체

- 배열

- 동일한 타입의 데이터 집합
- 첨자(index): 음수가 아닌 정수값, 문자열

James	Merry	Jane	Tom	Diana
name[0]	name[1]	name[2]	name[3]	name[4]

- 선언과 초기화

```
var 배열명 = new Array();
```

```
var months = new Array();
```

```
months[0]="1월";
```

```
months[1]="2월";
```

```
months[2]="3월"
```

```
var array_name = new Array(100);
```

```
var weekday = new Array("일", "월", "화");
```

Array 객체

```
<html> <head> <title>Array() 객체의 생성 및 출력</title>
```

```
<script>
```

```
    var book = new Array(6);  
    book[0] = "자바스크립트";  
    book[1] = "자바";  
    book[2] = "HTML & CSS";  
    book[3] = "Visual Basic.Net";  
    book[4] = "C#";  
    book[5] = "컴퓨터 개론";  
    book[7] = "JSP";
```

```
</script> </head>
```

```
<body bgcolor="lavender">
```

```
<h2>book Array</h2>
```

```
<p style="font-size:150%">
```

```
    <script>
```

```
        for(var i in book){  
            document.write("book[" + i + "] "+ book[i] + "<br/>");  
        }
```

```
    </script>
```

```
</p>
```

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

book Array

book[0] 자바스크립트

book[1] 자바

book[2] HTML & CSS

book[3] Visual Basic.Net

book[4] C#

book[5] 컴퓨터 개론

book[7] JSP

Array 객체

- 배열을 빠르게 선언 및 초기화

```
var myarray=["홍길동", "이순신", "강감찬"]
```

```
var array1 = new Array(10);
```

```
var array2 = [10];
```

```
<html>  <head>  <title>문자열 선언 방법</title>
  <script>
    var pet = [ "개", "고양이", "원숭이", "사슴" ];
  </script>
</head>
<body bgcolor="lavender">
  <h2>동물 객체 선언</h2>
  <script>
    for (var i in pet) {
      document.write("pet[" + i + "] " + pet[i] + "<br/>");
    }
  </script>
</body></html>
```

동물 객체 선언

```
pet[0] 개
pet[1] 고양이
pet[2] 원숭이
pet[3] 사슴
```

Array 객체

- Array 속성

속성	기능 설명
length	객체의 개수를 수치로 나타냄
constructor	객체의 생성자를 참조
prototype	속성과 메소드를 추가하여 배열 선언을 확장

```
<html>
<head>   <title>Array 객체</title> </head>
<body>
  <h2>수식 배열 생성</h2>
  <script>
    var years = new Array(10);
    for(var i=0; i < years.length; i++ ){
      years[i] = i + 2003;
      document.write "[" + i + "] = " + years[i]
                      + "년 <br/>");
    }
  </script>
</body>
</html>
```

수식 배열 생성

```
[0] = 2003년
[1] = 2004년
[2] = 2005년
[3] = 2006년
[4] = 2007년
[5] = 2008년
[6] = 2009년
[7] = 2010년
[8] = 2011년
[9] = 2012년
```

Array 객체

```
<html>
  <head>
    <title>Array 객체</title>
  </head>
  <body>
    <h2>색상 배열 객체</h2>
    <script>
      var colors = new Array("red", "green", "blue", "purple");
      for(var i in colors){
        document.write("<font color='" + colors[i] + "'>");
        document.write("colors[" + i + "] = " + colors[i] + "<br />");
      }
    </script>
  </body>
</html>
```

색상 배열 객체

```
colors[0] = red
colors[1] = green
colors[2] = blue
colors[3] = purple
```

Array 객체

```
<html> <head> <title>배열의 속성</title>
```

```
<script >
```

```
var book = new Array();
```

```
book[0] = "자바스크립트 프로그래밍";
```

```
book[1] = "HTML & CSS";
```

```
book[2] = "JAVA 프로그래밍";
```

```
book[3] = "컴퓨터 구조";
```

```
book[4] = "엑셀과 파워포인트";
```

```
book[5] = "JSP";
```

```
</script>
```

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

```
<h2>배열의 속성 종류</h2>
```

```
<big>
```

```
<script>
```

```
document.write("book 배열의 갯수는 " + book.length + "개 입니다 <br/>");
```

```
document.write("book 생성자 내용 : "+ book.constructor + "<br />");
```

```
book.length=0;
```

```
document.write("첫번째 책의 이름은 " + book[0] );
```

```
document.write("<br />prototype : "+ book.prototype);
```

```
</script>
```

```
</big> </body></html>
```

배열의 속성 종류

book 배열의 갯수는 6개 입니다

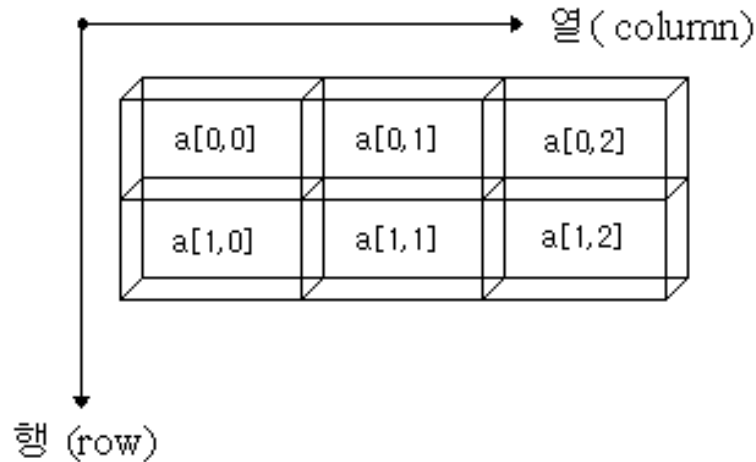
book 생성자 내용 : function Array() { [native code] }

첫번째 책의 이름은 undefined

prototype : undefined

Array 객체

- 2차원 배열
 - 배열명[n][m]



```
var array_name = new Array(new Array(77,88,99),  
                             new Array(50,60,99),  
                             new Array(99,88,78)  
                             );
```

또는

```
var array_name= [ [77,88,99],  
                  [50,60,99],  
                  [99,88,78] ];
```

Array 객체

- 2차원 배열

```
<html>
  <head>    <title>2 차원 배열</title>  </head>
  <body>
    <table border="2" bordercolor="blue">
      <caption>등급표</caption>
      <tr>
        <script>
          var grades= [ [77,88,99,75] ,
                          [50,60,99,89],
                          [99,88,78,92] ];
          for (var i=0; i < grades.length; i++) {
            for (var j=0; j < grades[i].length; j++) {
              document.write("<td bgcolor='#fff00'>" + grades[i][j] + "</td>");
            }
            document.write("</tr>");
          }
        </script>
      </table>
    </body>
  </html>
```

등급표

77	88	99	75
50	60	99	89
99	88	78	92

Array 객체

- 연관배열

- 배열의 색인(index)으로서 문자열 값을 사용
- 색인: 키(key), 값(value)-key와 연관되어 저장

```
<html>
```

```
<head>
```

```
<title>연관 배열</title>
```

```
</head>
```

```
<body>
```

```
<h2>색인 배열의 문자열</h2>
```

```
<script>
```

```
var states = new Array();
```

```
states["경기"] = "경기도";
```

```
states["전북"] = "전라북도";
```

```
states["부산"] = "부산직할시";
```

```
for( var i in states ) {
```

```
    document.write(" <b> 색인 : "+ i );
```

```
    document.write(" - 값 : " + states[i] + "<b><br/>");
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

색인 배열의 문자열

색인 : 경기 - 값 : 경기도

색인 : 전북 - 값 : 전라북도

색인 : 부산 - 값 : 부산직할시

Array 객체

- Array 메소드

메소드	기능
<code>arr.concat(arr1[,...])</code>	arr배열에 arr1를 결합
<code>arr.join([separator])</code>	배열값들이 separator로 구분되어 이어진 문자열로 반환(디폴트: ,)
<code>arr.pop()</code>	arr배열의 last요소를 삭제한 후 그 값을 리턴하고 배열의 크기를 줄임
<code>arr.push(elem1[,...])</code>	arr배열 last요소 뒤에 하나 또는 여러개의 요소를 넣고 새로운 배열의 길이를 리턴
<code>arr.shift()</code>	arr배열의 첫번째 요소를 삭제, 배열의 길이를 감소시킨 후, 삭제된 요소를 리턴
<code>arr.unshift(elem1[,...])</code>	arr 배열 처음에 요소들을 추가
<code>arr.slice(begin[,end])</code>	begin이 지정하는 index~end가 지정하는 index를 제외한 그 사이의 모든 원소를 배열로 반환, end는 생략시 배열의 맨 마지막 index
<code>arr.splice(idx, howMany, [elem1][,...])</code>	idx위치에 howMany만큼 삭제, elem1 등을 삭제한 위치에 추가
<code>arr.sort([sortFunc])</code>	sortFunc에 정의된 방법으로 정렬, 생략시 알파벳순
<code>arr.reverse()</code>	배열 요소를 역순으로 재배치(첫번째 요소: 마지막, 마지막 요소:처음).

Array 객체

- concat() 메소드

```
<html>
  <head>
    <title>concat() 메소드</title>
  </head>
  <body>
    <script>
      var names1=new Array("서울", "경기", "강원" );
      var names2=new Array("충남", "충북");
      document.write("첫 번째 배열 : "+ names1 + "<br/>");
      document.write("두 번째 배열 : "+ names2 +
        "<br/><br/>");
      document.write("결합 후 배열 : ");
      names1 = names1.concat(names2);
      document.write(names1);
    </script>
  </body>
</html>
```

첫 번째 배열 : 서울,경기,강원
두 번째 배열 : 충남,충북

결합 후 배열 : 서울,경기,강원,충남,충북

name1.join(" + ")

Array 객체

- pop()/push() 메소드

```
<html>
<head>
  <title>pop()/push() 메소드 </title>
</head>
<body>
  <script>
    var names=new Array("서울", "경기", "강원", "대전");
    document.write("<b>본래 배열 : "+ names + "<br/><br/>");
    var newstring=names.pop();
    document.write("popped 요소 : "+ newstring);
    document.write("<br/>새로운 배열 : "+ names + "<br/>");
    var pushed = Array("충남","충북")
    document.write("<br/>pushed 요소: "+ pushed+ "</b>");
    names.push(pushed);
    document.write("<br/><b>새로운 배열 : "+ names + "</b>");
  </script>
</body>
</html>
```

본래 배열 : 서울,경기,강원,대전

popped 요소 : 대전

새로운 배열 : 서울,경기,강원

pushed 요소: 충남,충북

새로운 배열 : 서울,경기,강원,충남,충북

Array 객체

- shift()/unshift() 메소드

```
<html>
<head>
  <title>shift() / unshift() 메소드</title>
</head>
<body>
  <script>
    var names=new Array("서울", "경기", "강원" );
    document.write("<b>본래의 배열 : "+ names + "<br /><br />");
    names.shift();
    document.write("shift 메소드 후의 배열 : " + names);
    names.unshift("충남","충북");
    document.write("<br /><br />unshift 메소드 후의 배열 : " + names);
  </script>
</body>
</html>
```

본래의 배열 : 서울,경기,강원

shift 메소드 후의 배열 : 경기,강원

unshift 메소드 후의 배열 : 충남,충북,경기,강원

Array 객체

- slice()/splice() 메소드

```
<html>
```

```
<head>
```

```
<title>slice() / splice() 메소드</title>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
var names=new Array("서울", "경기", "강원", "충남", "충북");
```

```
document.write("<b>본래의 배열 : "+ names + "<br />");
```

```
var sliceArray = names.slice(2, 4);
```

```
document.write("slice(2,4) 후의 배열 : ");
```

```
document.write(sliceArray);
```

```
var names=new Array("서울", "경기", "강원", "충남");
```

```
document.write("<br /><br />본래의 배열 : “ + names + "<br />");
```

```
names.splice(1, 2, "전남","전북","대전");
```

```
document.write("splice(1,2) 후의 배열 : “ + names + "</b>");
```

```
</script>
```

```
</body>
```

```
</html>
```

본래의 배열 : 서울,경기,강원,충남,충북

slice(2,4) 후의 배열 : 강원,충남

본래의 배열 : 서울,경기,강원,충남

splice(1,2) 후의 배열 : 서울,전남,전북,대전,충남

Date 객체

- 날짜와 시간을 다루는 객체
- 날짜는 1970년 1월 1일에 시작되는 UNIX 날짜에 기준

```
new Date("Month dd, yyyy hh:mm:ss");  
new Date("Month dd, yyyy");  
new Date(yy, month, dd [, hh[, mm[, ss[,ms]]]]);  
new Date(milliseconds); //1970/01/01 이후의 밀리초
```

```
mydate = new Date(); // 컴퓨터의 현재 날짜와 시간  
mydate = new Date("October 15, 2015 09:25:00");  
mydate = new Date("October 15, 2015");  
mydate = new Date(15,10,15);  
mydate = new Date(15,10,15,9,25,0);  
mydate = new Date(500);
```

- Date 객체 속성

속성	기능 설명
constructor	객체의 생성자를 참조
prototype	속성과 메소드를 추가하여 Date 객체를 확장

Date 객체

- Date 객체 메소드

메소드	기능 설명
<code>getDate()</code>	1부터 31일까지 특정 달의 날짜를 반환
<code>getDay()</code>	요일(0-6)을 숫자로 반환, 0은 일요일, 1은 월요일
<code>getFullYear()</code>	4자리 숫자의 연도를 반환
<code>getHours()</code>	시간(0-23)을 반환
<code>getMilliseconds()</code>	Millisecond(000~999)를 반환
<code>getMinutes()</code>	분(0-59)을 반환
<code>getMonth()</code>	월(0-11)을 반환 ; 0은 1월, 1은 2월
<code>getSeconds()</code>	초(0-59)를 반환
<code>getTime()</code>	1970년 1월 1일 이후의 시, 분, 초를 반환

Date 객체

- Date 객체 메소드

메소드	기능 설명
parse()	매개변수로 제공된 날짜의 시간을 00 : 00 : 00으로 반환
setDate(value)	일 (1-31)을 지정한다.
setFullYear()	연도를 4자리 숫자로 지정한다.
setHours()	시간 (0-23)을 지정한다.
setMilliseconds	밀리초를 지정한다.
setMonth(month,date)	시간을 월로 지정한다.
setSeconds()	초를 지정한다.
setTime()	시간을 지정한다.
setYear()	년을 지정한다.
toLocaleString	11/09/99 12:43:22 형태로 시간과 날짜를 문자로 지정
toSource()	Date() 객체의 원형을 반환한다.
toString()	시간과 날짜를 문자열로 반환한다.

Date 객체

- Date 객체 메소드

```
<html>
<head>
  <title>Time 과 Date</title>
</head>
<body>
  <h2>Date 와 Time</h2>
  <big>
    <script>
      var now = new Date();
      document.write("<b>Local time :</b> " + now + "<br />");
      var hours = now.getHours();
      var minutes = now.getMinutes();
      var seconds = now.getSeconds();
      var year = now.getFullYear();
      document.write("<b>연도는 : " + year + " </b><br />");
      document.write("<b>시간은 : " +
        hours + ":" + minutes + ":" + seconds + "</b>");
    </script>
  </big>
</body>
</html>
```

Date 와 Time

Local time : Fri Sep 25 2015 15:13:03 GMT+0900 (대한민국 표준시)

연도는 : 2015

시간은 : 15:13:3

Date 객체

- Date 객체 메소드
 - 시계 구현

```
<div id='clock'></div>
```

```
<script>
```

```
function setClock() {
```

```
    var now = new Date();
```

```
    var s = now.getHours() + ':' + now.getMinutes() + ':' + now.getSeconds();
```

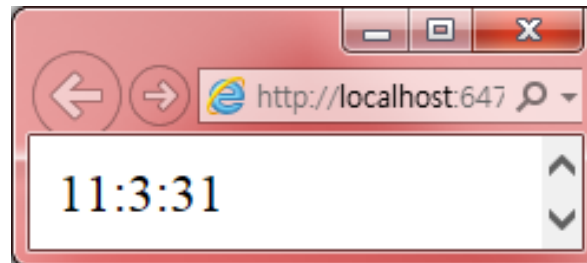
```
    document.getElementById('clock').innerHTML = s;
```

```
    setTimeout('setClock()', 1000);
```

```
}
```

```
setClock();
```

```
</script>
```



Date 객체

- Date 객체 메소드: 메소드 추가-prototype 이용

```
<html>
  <head> <title>Prototype 속성</title>
    <script>
      function weekDay() {
        var now = this.getDay();
        var names = new Array(7);
        names[0]="일요일";
        names[1]="월요일";
        names[2]="화요일";
        names[3]="수요일";
        names[4]="목요일";
        names[5]="금요일";
        names[6]="토요일";
        return(names[now]);
      }
      Date.prototype.DayOfWeek = weekDay;
    </script>
  </head>
```

Date 객체

```
<body>
  <font face="arial">
    <big>
      <div align="center">
        <script>
          var today=new Date();
          var year = today.getFullYear();
          var month = today.getMonth() + 1;
          var day = today.getDate();
          document.write(year + "년 " + month + "월 " + day + "일");
          document.write("은<b> " + today.DayOfWeek() + "</b> 입니다<br />");
        </script>
      </div>
    </big>
  </body>
</html>
```

2015년 9월 25일은 금요일 입니다

Math 객체

- 제곱근, 삼각함수, 대수함수들, 랜덤값(무작위 추출에 의한 수치) 등에 대한 라이브러리 제공
- 생성하지 않고 사용
- Math 객체 속성

속성	기능 설명
E	자연로그 밑으로 사용하는 오일러 상수
LN10	10의 자연로그
LN2	2의 자연로그
PI	파이()값
SQRT2	2의 제곱근index 위치의 글자 확인
SQRT1_2	1/2의 제곱근

Math 객체

```
<html>
<head>
  <title> Math 객체</title>
</head>
<body>
  <h3> Math 객체 사용 </h3> <hr />
  <SCRIPT>
    <!--
      document.write("오일러 상수 **>:"+Math.E+"<br/>");
      document.write("10의 자연로그 **>:"+Math.LN10+"<br/>");
      document.write("2 의 자연로그 **>:"+Math.LN2+"<br/>");
      document.write("....."+"<br/>");
      document.write("파이값      **>:"+Math.PI+"<br/>");
      document.write("2의 제곱근    **>:"+Math.SQRT2+"<br/>");
      document.write("1/2의 제곱근  **>:"+Math.SQRT1_2+"<br/>");
    //-->
  </SCRIPT>
</body>
</html>
```

Math 객체 사용

```
오일러 상수 **>:2.718281828459045
10의 자연로그 **>:2.302585092994046
2 의 자연로그 **>:0.6931471805599453
.....
파이값 **>:3.141592653589793
2의 제곱근 **>:1.4142135623730951
1/2의 제곱근 **>:0.7071067811865476
```

Math 객체

- Math 객체 메소드

Math 메소드의 종류	기능 설명
<code>abs(su)</code>	절대값을 반환해 주는 메소드
<code>acos(su)</code>	아크코사인 값을 레디안 값으로 반환해 주는 메소드
<code>asin(su)</code>	아크사인 값을 레디안 값으로 반환해 주는 메소드
<code>atan(su)</code>	아크탄젠트 값을 레디안 값으로 반환해 주는 메소드
<code>ceil(su)</code>	su의 소수부분을 올린수를 반환해 주는 메소드
<code>exp(su)</code>	지수값을 반환해 주는 메소드
<code>floor(su)</code>	su의 소수부분을 버린수를 반환해 주는 메소드
<code>log(su)</code>	밑수 10의 로그값을 반환해 주는 메소드
<code>max(su1, su2)</code>	두 수 중 큰 수를 반환해 주는 메소드
<code>min(su1,su2)</code>	두 수 중 작은 수를 반환해 주는 메소드
<code>pow(su1, su2)</code>	역수값을 반환해 주는 메소드($su1^{su2}$)
<code>random()</code>	0과1 사이의 난수 발생값을 반환해 주는 메소드
<code>round(su)</code>	수 su를 반올림한 수를 반환해 주는 메소드
<code>sin(su)</code>	사인값을 반환해 주는 메소드
<code>sqrt(su)</code>	제곱근 값을 반환해 주는 메소드
<code>tan(su)</code>	탄젠트 값을 반환해 주는 메소드

Math 객체

- Math 객체 메소드

Math 메소드 사용

```
abs(-100) **>:100
asin(1) **>:1.5707963267948965
acos(1) **>:0
atan(1) **>:0.7853981633974483
```

```
exp(0) **>:1
log(2) **>:0.6931471805599453
max(1,2) **>:2
min(1,2) **>:1
```

```
pow(2,3) **>:8
random(0) **>:0.3753843621130426
round(11.4) **>:11
sqrt(100) **>:10
sin(0) **>:0
cos(0) **>:1
tan(1) **>:1.5574077246549023
```

```
<html>
```

```
<head> <title> Math 객체</title> </head>
```

```
<body> <h4> Math 메소드 사용 </h4> <hr/>
```

```
<SCRIPT>
```

```
with(Math) {
```

```
document.write("abs(-100) **>:"+abs(-100)+"<br/>");
```

```
document.write("asin(1) **>:"+asin(1)+"<br/>");
```

```
document.write("acos(1) **>:"+acos(1)+"<br/>");
```

```
document.write("atan(1) **>:"+atan(1)+"<br/>");
```

```
document.write("-----"+"<br/>");
```

```
document.write("exp(0) **>:"+exp(0)+"<br/>");
```

```
document.write("log(2) **>:"+log(2)+"<br/>");
```

```
document.write("max(1,2) **>:"+max(1,2)+"<br/>");
```

```
document.write("min(1,2) **>:"+min(1,2)+"<br/>");
```

```
document.write("-----"+"<br/>");
```

```
document.write("pow(2,3) **>:"+pow(2,3)+"<br/>");
```

```
document.write("random(0) **>:"+random()+"<br/>");
```

```
document.write("round(11.4) **>:"+round(11.4)+"<br/>");
```

```
document.write("sqrt(100) **>:"+sqrt(100)+"<br/>");
```

```
document.write("sin(0) **>:"+sin(0)+"<br/>");
```

```
document.write("cos(0) **>:"+cos(0)+"<br/>");
```

```
document.write("tan(1) **>:"+tan(1)+"<br/>");
```

```
} </SCRIPT> </body></html>
```

Math 객체

- Math 객체 메소드: 올림, 반올림, 절사
 - ceil() : 올림, ceil(5.02) -> 6, ceil(-5.02) -> -5
 - round() : 반올림, round(5.9) -> 6, round(5.02) -> 5
 - floor() : 절사(버림), floor(5.02) -> 5, floor(-5.02) -> -6

<html>

<head> <title>Math 객체</title> </head>

<body>

<h2>수의 반올림</h2>

<h3>

<script>

var num = 16.3;

document.write("선언된 수는 : ", num, "

");

document.write("Math.floor 의 결과는 : " +
Math.floor(num) + "
");

document.write("Math.ceil 의 결과는 : " +
Math.ceil(num) + "
");

document.write("Math.round 의 결과는 : " +
Math.round(num) + "
");

</script>

</h3>

</body>

</html>

수의 반올림

선언된 수는 : 16.3

The Math.floor 의 결과는 : 16

The Math.ceil 의 결과는 : 17

The Math.round 의 결과는 : 16

Math 객체

- Math 객체 메소드: 랜덤값-난수
 - random() 메소드에 의한 수의 발생: 0 ~ 1 사이의 수를 발생

```
<html>
<head>
  <title>Random Numbers</title>
</head>
<body>
  <p style="font-size:120%;color:black">
    <script>
      for(i=0; i < 10;i++){
        document.write(Math.random(),"<br/>");
      }
      document.write("<br/>");
      for(i=0; i < 20; i++){
        document.write(Math.floor(Math.random() * 10 ) +" ");
      }
    </script>
  </p>
</body>
</html>
```

0.4665943654275121
0.652185848590034
0.7874957570441441
0.710008961847828
0.6593734011395835
0.09473358203144066
0.6271503245773429
0.5763158919971099
0.5651634813304256
0.35809631974834716

5 5 4 6 5 0 9 6 2 5 3 8 0 0 2 8 0 6 4 6

Wrapper 객체

- 원시자료형(string, number과 boolean)에 대한 객체
 - string 객체, number 객체, boolean 객체

String 객체

- 원시자료형 string의 Wrapper 객체

```
var title="JavaScript Programming";  
var title=new String("JavaScript Programming");
```

```
<html>  
  <head>    <title>String 객체</title>  </head>  
  <body bgcolor=yellow>  
    <big>    <h2>기본형과 String 객체</h2>  
      <script>  
        var first_string = "The winds of war are blowing.";  
        var next_string = new String("There is peace in the valley.");  
        document.write("첫번째 문자열의 형은 <em> "+ typeof(first_string));  
        document.write(".</em><br />두번째 문자열의 형은<em> "+  
          typeof(next_string) + "<br />");  
      </script>  
    </big>  
  </body>  
</html>
```

기본형과 String 객체

첫번째 문자열의 형은 *string*.
두번째 문자열의 형은 *object*.

String 객체

- String 객체 속성

속성	기능 설명
length	문자열의 길이를 정수로 나타냄
constructor	String 객체를 작성한 function을 반환
prototype	문자열에 속성과 메소드를 추가하여 정의

```
<html>
<head> <title>Prototype 속성</title> </head>
<body bgcolor=yellow>
  <script>
    function ucLarge() {
      var str=this.bold().fontcolor("green"); this.toUpperCase().fontsize("22");
      return(str);
    } String.prototype.ucLarge=ucLarge;
  </script>
  <div align="center">
    <script>
      var str="동물(Animal)을 사랑합시다!!"; document.write(str.ucLarge());
      document.write("<b>(길이 : "+ str.length +" )" + "<br />" );
    </script>
    
  </div> </body> </html>
```

동물(ANIMAL)을 사랑합시다!! (길이 : 19)



String 객체

- String 객체 메소드

메소드	HTML의 태그 기능
String.anchor(Name)	String
String.big()	<big>String</big>
String.blink()	<blink>String</blink>
String.bold()	String
String.fixed()	<tt>String</tt>
String.fontcolor(color)	String
String.fontSize(size)	String
String italics()	<i>String</i>
String.link(URL)	String
String.small()	<small>String</small>
String.strike()	<strike>String</strike>

String 객체

- String 객체 메소드

```
<html>
<head>
  <title>String object</title>
</head>
<body bgcolor="yellow">
  <big>
    <h2>String 객체의 메소드</h2>
    <script >
      var str1 = new String("Hello world!");
      var str2 = "안녕하세요 자바스크립트입니다!!.";
      document.write(str1) + "<br />";
      document.write(str1.fontcolor("blue")+"<br />");
      document.write(str1.fontSize(8).fontcolor("red").bold()+ "<br />");
      document.write(str1.big()+ "<br />");
      document.write("Good-bye, ".italics().bold().big() + str2 + "<br />");
    </script>
  </big>
</body>
</html>
```

String 객체의 메소드

Hello world!Hello world!

Hello world!

Hello world!

Good-bye, 안녕하세요 자바스크립트입니다!!.

String 객체

- String 객체 메소드

메소드	HTML의 태그 기능
<code>str.charAt(idx)/charCodeAt(idx)</code>	idx 위치의 문자를 반환
<code>str.concat([str1[, ..., strn]])</code>	문자열들을 결합
<code>fromCharCode(codes)</code>	컴마로 분리된 문자코드의 문자열을 생성
<code>str.indexOf(str1[,pos])</code>	pos위치에서부터 str1의 내용을 검색해 위치를 반환
<code>str.lastIndexOf(str1[,pos])</code>	pos위치에서부터 str1의 내용을 검색해 마지막 위치를 반환
<code>str.replace(searchVal, replaceStr)</code>	searchVal(정규식 or 문자열)을 검색하여 replaceStr으로 교체
<code>str.search(regExp)</code>	정규식 검색에서 일치하는 첫 위치를 반환
<code>str.slice(start[,end])</code>	start~end-1 위치에 해당하는 부분문자열
<code>str.split([separator[,limit]])</code>	분리자를 이용해 단어로 구분한 substr의 배열
<code>str.substr(start[,len])</code>	start~start+len 위치에 해당하는 부분문자열
<code>str.substring(start[,end])</code>	start~end-1 위치에 해당하는 부분문자열
<code>str.startsWith(str1[,pos])</code>	pos에서부터 검색해서 str1로 시작하면 true
<code>str.endsWith(str1[,pos])</code>	pos에서부터 검색해서 str1로 끝나면 true
<code>str.toString()</code>	문자열로 변환하여 반환
<code>str.trim()</code>	선행 및 후행 공백과 줄 종결자 문자를 제거
<code>str.valueOf()</code>	객체의 값을 문자열로 반환

String 객체

- String 객체 메소드

<html>

<head> <title>문자열 다루기</title> </head>

<body>

<h2>string 취급 관련 메소드</h2> string 취급 관련 메소드

<script>

```
function break_tag(){  
    document.write("<br />");  
}
```

```
document.write("<h3>");
```

```
var str1 = new String("Time and tide wait for no man.");
```

```
document.write("문자열 :<em> "+ str1 );
```

```
document.write("</em> <br>처음 'm'의 위치 : " + str1.indexOf("m"));
```

```
document.write(" - 마지막 'm'의 위치 : " + str1.lastIndexOf("m"));
```

```
break_tag();
```

```
document.write("<em>str1.substr(4,5)</em>위치의 반환 값 : <em> "  
                + str1.substr(4,5));
```

```
break_tag();
```

```
document.write("<b> 대문자 변환 : " + str1.toUpperCase());
```

```
document.write("</h3>");
```

</script>

</body></html>

문자열 : *Time and tide wait for no man.*

처음 'm'의 위치 : 2 - 마지막 'm'의 위치 : 26

*str1.substr(4,5)*위치의 반환 값 : *and*

대문자 변환 : TIME AND TIDE WAIT FOR NO MAN.

String 객체

- String 객체 메소드

```
<html>
<head>  <title>문자열 발췌</title> </head>
<body bgcolor=yellow>
<big>
문자열 발췌<br/>
<script >
  var str = "ljm3151@hanmail.net";
  document.write("<br/>메일 ID : <em> " + str.substr(0,7) + "</em>.<br />");
  var namesarr = str.split("@" );
  document.write( "사용자 이름 : <em> " + namesarr[0] + "</em>.<br />");
  document.write( "메일 서버 : <em> " + namesarr[1] + "</em>.<br />");
  document.write( "메일의 첫 글자 : <em>" +str.charAt(0)+ "</em>.<br />");
  document.write( "메일의 마지막 글자 : <em>" + str.charAt(str.length - 1)
    + "</em>.<br />");
</script>
</big>
</body>
</html>
```

문자열 발췌

메일 ID : *ljm3151*.

사용자 이름 : *ljm3151*.

메일 서버 : *hanmail.net*.

메일의 첫 글자 : *l*.

메일의 마지막 글자 : *t*.

String 객체

- String 객체 메소드

<html>

<head> <title>Search 와 Replace</title> </head>

<body bgcolor=yellow>

Search 와 Replace 메소드

<small>

<script>

```
var str = "ljm3151@hanmail.net";
```

```
document.write( "본래 문자열 : " + str + "<br/>");
```

```
document.write( "새로운 문자열 : " + str.replace("ljm","kwc") + "<br/>");
```

```
var index=str.search("hanmail");
```

```
document.write("search() 메소드 ₩"hanmail₩" 의 위치 : " + index + "<br/>");
```

```
var mysubstr=str.substr(index,7);
```

```
document.write("수정전 ₩"hanmail₩" 후 ₩"daum₩" <br/>");
```

```
document.write(str.replace(mysubstr,"daum")+ "<br/>");
```

</script>

</small>

</body>

</html>

Search 와 Replace 메소드

본래 문자열 : ljm3151@hanmail.net

새로운 문자열 : kwc3151@hanmail.net

search() 메소드 "hanmail" 의 위치 : 8

수정전 "hanmail" 후 "daum"

ljm3151@daum.net

Number 객체

- 기본적인 수치 값에 대한 Wrapper 객체

```
var number = new Number(numeric value); // 객체  
var number = numeric value; // 기본 데이터형
```

```
var n = new Number(65.7);  
var n = 65.7;
```

- Number 객체의 속성

속성	기능 설명
constructor	Number 객체의 prototype의 function을 반환한다.
MAX_VALUE	가장 큰 수를 반환한다.
MIN_VALUE	가장 작은 수를 반환한다.
NaN	숫자가 아닌 값
NEGATIVE_INFINITY	음수의 무한대 값을 나타낸다.(overflow를 반환)
POSITIVE_INFINITY	무한대 값을 나타낸다.(overflow를 반환)
prototype	객체에 속성과 메소드를 추가할 수 있게 한다

Number 객체

- Number 객체의 속성

```
<html>
```

```
<head>    <title>Number 객체의 속성</title>  </head>
```

```
<body bgcolor='yellow'>
```

```
  <h2>Number 속성</h2>
```

```
  <script>
```

```
    var largest = Number.MAX_VALUE;
```

```
    var smallest = Number.MIN_VALUE;
```

```
    var num1 = 20;  // 기본 값
```

```
    var num2 = new Number(13);  // Number 객체의 생성
```

```
    document.write("<br/>최대 수치 값 : " + largest+ "<br/>");
```

```
    document.write("최소 수치 값 "+ smallest + "<br/>");
```

```
    document.write("수치 값의 문자열 (base 2) : "+ num1.toString(2));
```

```
    document.write("<br/>수치 값의 문자열 (base 8) : "+ num2.toString(8));
```

```
    document.write("<br/>-25 Sqrt : "+ Math.sqrt(-25) + "<br/>");
```

```
  </script>
```

```
</body>
```

```
</html>
```

Number 속성

최대 수치 값 : 1.7976931348623157e+308

최소 수치 값 5e-324

수치 값의 문자열 (base 2) : 10100

수치 값의 문자열 (base 8) : 15

-25 Sqrt : NaN

Number 객체

- Number 객체의 메소드

메소드	기능 설명
toString()	Number 객체를 문자열로 반환
toFixed()	소수점 뒤의 digit 숫자로 반환
toExponential()	숫자를 exponential notation으로 반환
toPrecision()	x 길이로 숫자를 변환
valueOf()	Number 객체의 기본 숫자로 변환

소수 값의 변환

기본 값 : 22.425456
2 digit 소수 값 : 22.43
3 digit 소수 값 : 22.425

```
<html>
<head>
  <title>Number 객체</title>
</head>
<body bgcolor='yellow'>
  <h2>Formatting Numbers</h2>
  <script>
    var n = new Number(22.425456); // var n = 22.425456;
    document.write("<b>기본 값 : " + n + "<br/>");
    document.write("2 digit 소수 값 : " + n.toFixed(2) + "<br/>");
    document.write("3 digit 소수 값 : " + n.toFixed(3) + "</b><br/>");
  </script>
</body>
</html>
```


Boolean 객체

- 참일 때는 true를, 거짓일 때는 false를 가지는 객체

```
var object = new Boolean(value);
```

```
var b1 = new Boolean(5);  
var b2 = new Boolean(null);
```

Boolean 객체의 종류

boolean 0 : false

boolean 1 : true

빈 문자열 boolean 값 : false

null 값 boolean : false

NaN 값 boolean : false

```
<html>
```

```
<head>    <title>Boolean 객체</title>  </head>
```

```
<body bgcolor='aqua'>
```

```
Boolean 객체의 종류<br/><br/>
```

```
<small>    <script>
```

```
var bool1 = new Boolean( 0); var bool2 = new Boolean(1);
```

```
var bool3 = new Boolean(""); var bool4 = new Boolean(null);
```

```
var bool5 = new Boolean(NaN);
```

```
document.write(" boolean 0 : "+ bool1 + "<br/>");
```

```
document.write(" boolean 1 : "+ bool2 + "<br/>");
```

```
document.write("빈 문자열 boolean 값 : "+ bool3+ " <br/>");
```

```
document.write(" null 값 boolean : "+ bool4+ " <br/>");
```

```
document.write("NaN 값  boolean : "+ bool5 + "<br/>");
```

```
</script>    </small>
```

```
</body>
```

```
</html>
```

Function 객체

- 함수를 정의하는 방법
 - keyword function을 사용하는 방법: 일반적
 - Function 객체를 사용하는 방법: JavaScript의 모든 함수-실제 Function 객체

```
funName = new Function( [arg1, [... argN,]] body );
```

```
var nameOfFun = new Function (args, statements_as_string);  
var addemUp = new Function ( "a", "b", "return a + b;" );  
document.write(addemUp(10, 5));
```

- 속성

속성	기능 설명
length	반환되는 인수의 수가 반환
prototype	객체에 속성과 메소드를 추가

- 메소드

메소드	기능 설명
apply()	다른 function의 메소드를 사용
call()	다른 객체의 메소드를 사용

Function 객체

```
<html>
  <head>
    <title>Function Object</title>
  </head>
  <body bgcolor='lightgreen'>
    <div align="center">
      익명의 Function과 생성자
    <script>
      var sum = new Function("a","b", "return a + b; ");
      window.onload = new Function ( "document.bgColor='yellow'");
      document.write( "합계는 = " + sum(5,10)+ "<br>");
      document.write( "배경색은 노랑입니다<br>");
    </script>
    </div>
  </body>
</html>
```

익명의 Function과 생성자

합계는 = 15
배경색은 노랑입니다