

내장 객체 - String -

String

❖ 검색

메서드	설명
charAt(index)	index 위치의 문자를 구한다. index가 문자열의 범위를 벗어나면 빈 문자열이 리턴된다.
charCodeAt(index)	index 위치의 문자에 대한 유니코드를 구한다.
indexOf(searchvalue,start)	부분 문자열의 위치를 검색한다. start는 검색 시작 위치이며 생략시 0이 적용되어 처음부터 검색한다. 없을 경우 -1을 리턴한다.
lastIndexOf(searchvalue,start)	부분 문자열의 위치를 역방향에서 검색한다. start는 검색 시작 위치이며 생략시 문자열의 제일 끝이 적용된다. 없을 경우 -1을 리턴한다.
concat(s1, s2, ...)	여러 개의 문자열을 연결한다. + 연산자와 동일하다.
toLowerCase()	소문자로 변환한다.
toUpperCase()	대문자로 변환한다.
replace(searchvalue,newvalue)	문자열을 대체한다. 정규식도 사용할 수 있다.
search(searchvalue)	부분 문자열 또는 정규식을 검색하여 그 위치를 리턴한다.

String

❖ 검색

메서드	설명
match(regex)	정규식으로 검색하여 일치하는 결과를 배열로 리턴한다. 발견되지 않으면 null을 리턴한다.
substring(from, to)	두 위치 사이의 부분 문자열을 추출한다. to를 생략하면 뒤쪽 모든 문자열을 추출한다.
slice(start,end)	start 위치에서 end 위치까지 부분 문자열을 추출한다. 음수로 끝에서부터의 위치를 지정할 수 있다.
substr(start,length)	start에서 시작하여 length 길이만큼 부분 문자열을 추출한다. 길이를 생략하면 뒤쪽 모든 문자열을 추출한다.
split(separator,limit)	구분자로 구분된 문자열을 분리하여 배열로 리턴한다. limit는 최대 몇개까지 리턴할 것인가를 지정한다.
trim()	앞 뒤의 공백을 제거한다.

String

❖ 09_1_01_charat.html

```
<body>
  <script>
    var s = "우리나라 대한민국 좋은나라";
    document.write("charAt(2) = " + s.charAt(2) + "<br>");
    document.write("charCodeAt(2) = " + s.charCodeAt(2) + "<br>");
    document.write("indexOf('나라') = " + s.indexOf("나라") +
"<br>");
    document.write("lastIndexOf('나라') = " + s.lastIndexOf("나라")
+ "<br>");
  </script>
</body>
```

String

❖ 09_1_02_search.html

```
<body>
  <script>
    var s = "독도는 일본땅";
    document.write("s = " + s + "<br>");
    document.write("search = " + s.search("일본") + "<br>");
    document.write("replace = " + s.replace("일본", "한국") +
"<br>");
  </script>
</body>
```

String

❖ 09_1_03_tocase.html

```
<body>
  <script>
    var s = "아름다운 Korea!";
    var u = s.toUpperCase();
    var l = s.toLowerCase();

    document.write("s = " + s + "<br>");
    document.write("toUpperCase() = " + u + "<br>");
    document.write("toLowerCase() = " + l + "<br>");
  </script>
</body>
```

String

❖ 09_1_04_casecompare.html

```
<body>
  <script>
    var a = "JavaScript";
    var b = "JAVASCRIPT";

    if (a == b) {
      document.write("== 비교 : 두 문자열이 같다.<br>");
    }
    if (a.toUpperCase() == b.toUpperCase()) {
      document.write("대소문자 무시 비교 : 두 문자열이 같다.<br>");
    }
  </script>
</body>
```

String

❖ 09_1_05_trim.html

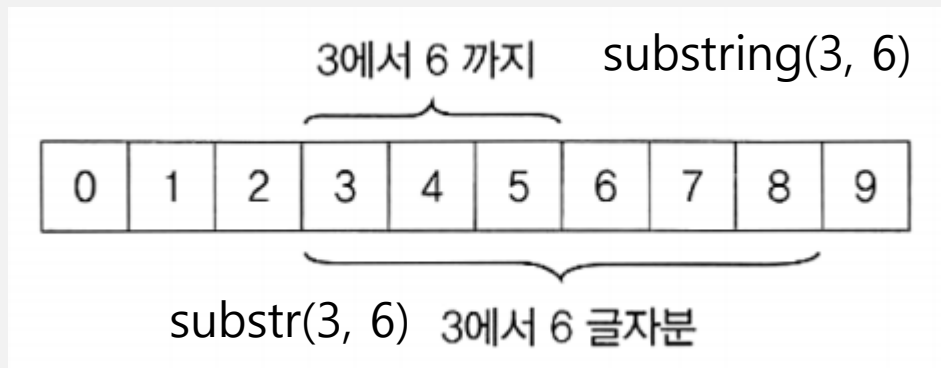
```
<body>
  <script>
    var s = "  우리나라 대한민국  ";
    var t = s.trim();

    document.write("s = " + s + ", 길이 = " + s.length + "<br>");
    document.write("t = " + t + ", 길이 = " + t.length + "<br>");
  </script>
</body>
```


String

❖ 09_1_06_substring.html

```
<body>
  <script>
    var s = "0123456789";
    document.write("substring = " + s.substring(3, 6) + "<br>");
    document.write("substr = " + s.substr(3, 6) + "<br>");
    document.write("slice = " + s.slice(3, 6) + "<br>");
    document.write("slice = " + s.slice(3, -2) + "<br>");
  </script>
</body>
```



String

❖ 09_1_07_slicestring.html

```
<body>
  <script>
    var music1 = "사랑과 우정 사이.mp3";
    var music2 = "오직 하나뿐인 그대.mp3";

    document.write("음악1 = " + music1.slice(0,-4) + "<br>");
    document.write("음악2 = " + music2.slice(0,-4) + "<br>");
  </script>
</body>
```

String

❖ 09_1_08_split.html

```
<body>
  <script>
    var citys = "서울,부산,대전,광주";
    var parts = citys.split(",");
    for (var city in parts) {
      document.write("도시명 : " + parts[city] + "<br>");
    }
  </script>
</body>
```

String

❖ 09_1_09_fromcharcode.html

```
<body>
  <script>
    var s = String.fromCharCode(45824, 54620, 48124, 44397);
    document.write("s = " + s + "<br>");
    var t = "\ub300\ud55c\ubbfc\uad6d";
    document.write("t = " + t + "<br>");
  </script>
</body>
```

String

❖ 09_1_10_taginsert.html

```
<body>
  <script>
    var s = "글자의 속성";
    document.write("굵게 : " + s.bold() + "<br>");
    document.write("기울임 : " + s.italics() + "<br>");
    document.write("빨간색 : " + s.fontcolor("red") + "<br>");
    document.write("큰 글자 : " + s.fontsize("6") + "<br>");

    var soen = "SoEn으로 이동";
    var link = soen.link("http://www.soen.kr");
    document.write(link + "<br>");
    alert(link);
  </script>
</body>
```

내장 객체

- Core 객체 -

Core 객체

❖ Math

- 상수 속성

속성	설명
PI	원주율이다.
E	오일러의 상수이다.
LN2	2의 자연 로그
LN10	10의 자연 로그
LOG2E	2의 로그
LOG10E	10의 로그
SQRT1_2	1/2의 제곱근
SQRT2	2의 제곱근

❖ 09_2_01_mathconst.html

```
<body>
  <script>
    document.write("PI : " + Math.PI + "<br>");
    document.write("E : " + Math.E + "<br>");
    document.write("LN2 : " + Math.LN2 + "<br>");
    document.write("LN10 : " + Math.LN10 + "<br>");
    document.write("LOG2E : " + Math.LOG2E + "<br>");
    document.write("LOG10E : " + Math.LOG10E + "<br>");
    document.write("SQRT1_2 : " + Math.SQRT1_2 + "<br>");
    document.write("SQRT2 : " + Math.SQRT2 + "<br>");
  </script>
</body>
```


메서드	설명
abs(x)	절대값
sin(x), cos(x), tan(x)	삼각함수. 각도는 라디안으로 지정한다.
asin(x), acos(x), atan(x), atan2(x)	삼각함수의 역함수
floor(x)	직전 정수값. 수직선상의 왼쪽 값
ceil(x)	직후 정수값. 수직선상의 오른쪽 값
round(x)	반올림
sqrt(x)	제곱근
log(x)	로그
exp(x)	지수
min(x,y,...).	인수중에 작은 값을 리턴한다.
max(x,y,...)	인수중에 큰 값을 리턴한다.
pow(x,y)	거듭승. x의 y승
random()	난수. 0~1사이의 난수를 리턴한다.

Core 객체

❖ 09_2_02_mathfunc.html

```
<body>
  <script>
    for(var d = 0; d <= 90; d += 15) {
      var r = d * Math.PI / 180;
      document.write(d + "(" + r.toFixed(2) + " rad) : " +
        "sin = " + Math.sin(r).toFixed(2) +
        ", cos = " + Math.cos(r).toFixed(2) +
        ", tan = " + Math.tan(r).toFixed(2) + "<br>");
    }
  </script>
</body>
```

Core 객체

❖ 09_2_03_minmax.html

```
<body>
  <script>
    var value = Number(prompt("점수를 입력하세요.", "80"));
    value = Math.min(100, Math.max(value, 0));
    // if (value < 0) value = 0;
    // if (value > 100) value = 100;

    document.write("입력한 점수는 " + value + "입니다.<br>");
  </script>
</body>
```

Core 객체

❖ 09_2_04_mathabs.html

```
<body>
  <script>
    var a = 21;
    var b = 28;
    var c = 0;

    if (a > b) {
      c = a - b;
    } else {
      c = b - a;
    }
    document.write("두 값의 차이는 " + c + "입니다.");
  </script>
</body>
```

Core 객체

❖ 09_2_04_mathabs2.html

```
<body>
  <script>
    var a = 21;
    var b = 28;
    document.write("두 값의 차이는 " + Math.abs(a-b) + "입니다.");
  </script>
</body>
```

Core 객체

❖ 09_2_05_mathrandom.html

```
<body>
  <script>
    document.write("무작위 수 : " + Math.random() + "<br>");
    document.write("1~10 사이 : " + Math.floor(Math.random() * 10)
+ "<br>");
  </script>
</body>
```

Core 객체

❖ Date

`new Date()`

`new Date(y,m,d,h,m,s,ms)`

`new Date(milliseconds);`

`new Date(dateString);`

Core 객체

❖ Date 메서드

메서드	설명
getFullYear	1970년 이후의 경과 년. 사용을 권장하지 않는다.
getMonth	월. 0이 1월이다.
getDate	날짜
getDay	요일. 0이 일요일이다.
getHours	시
getMinutes	분
getSeconds	초
getMilliseconds	1/1000초

Core 객체

❖ Date 메서드

메서드	설명
getTime	1970년 1월 1일 이후 경과 시간(1/1000초 단위). 이 형태의 시간을 에폭 타임이라고 한다.
setTime	에폭 타임을 설정한다.
toString	날짜를 문자열로 변환한다. 현지 시간을 사용하지만 형식이 지역화되지 않아 읽기는 불편하다.
toDateString	날짜만 문자열로 변환한다.
toTimeString	시간만 문자열로 변환한다.
toLocaleString	지역화된 날짜와 시간을 출력한다.
toLocaleDateString	지역화된 날짜를 출력한다. 한국에서는 년월일순으로 표시된다.
toLocaleTimeString	지역화된 시간을 출력한다. 한국에서는 오전(후) h:m:s로 표시된다.

Core 객체

❖ Date 정적 메서드

메서드	설명
now	현재 에폭 타임을 구한다.
parse(s)	문자열을 분석하여 에폭 타임을 구한다.
UTC(y,m,d,h,m,s,ms)	국제 표준시의 에폭 타임을 구한다.

❖ 09_2_06_datetostring.html

```
<body>
  <script>
    var now = new Date();
    document.write("toString : " + now.toString() + "<br>");
    document.write("toLocaleDateString : " + now.toLocaleDateString() + "<br>");
    document.write("toLocaleTimeString : " + now.toLocaleTimeString() + "<br>");
    document.write("toLocaleString : " + now.toLocaleString() + "<br>");
  </script>
</body>
```

❖ 09_2_07_dateformat.html

```
<body>
  <script>
    var now = new Date();
    document.write("현재 시간은 " +
      now.getFullYear() + "년 " +
      (now.getMonth() + 1) + "월 " +
      now.getDate() + "일 " +
      now.getHours() + ":" +
      now.getMinutes() + "입니다."
    );
  </script>
</body>
```

Core 객체

❖ 09_2_08_calcddate.html

```
<body>
  <script>
    var now = new Date();
    var epoch = now.getTime();
    epoch += (86400 * 1000 * 3);
    now.setTime(epoch);
    document.write("사흘 후는 " + now.toLocaleDateString() +
                  "입니다.");
  </script>
</body>
```

Core 객체

❖ 09_2_09_epochtime.html

```
<body>
  <script>
    var birth = new Date(1970,8,1);
    var now = new Date();
    var ellapse = (now.getTime() - birth.getTime()) /
                  (86400 * 1000);
    document.write("오늘은 태어난지 " + Math.floor(ellapse) +
                  "일 째입니다.");
  </script>
</body>
```

Core 객체

❖ RegExp

- 정규 표현식
 - `new RegExp(패턴, 플래그)` // 객체
 - `/패턴/플래그` // 리터럴

표현식	설명
[abc]	[] 괄호안의 임의의 한 문자와 대응된다.
[^abc]	[^] 괄호안의 문자가 아닌 문자와 대응된다.
[0-9]	임의의 숫자와 대응된다.
(a b)	둘 중 하나의 문자와 대응된다.
\w, \W	아스키 문자와 대응된다. [a-zA-Z0-9]와 같다. 대문자는 반대 조건이다.
\d, \D	숫자와 대응된다. [0-9]와 같다. 대문자는 반대 조건이다.
\s, \S	공백과 대응된다. 대문자는 반대 조건이다.
\b, \B	단어의 시작이나 끝과 대응된다.

Core 객체

❖ RedExp

○ 반복 패턴

표현식	설명
$\{a, b\}$	a번 이상, b번 이하의 횟수
$\{a, \}$	a번 이상
$\{a\}$	정확하게 a번
$?$	0번 또는 한번.
$+$	한번 이상
$*$	0번 이상

○ 위치 패턴

표현식	설명
\wedge	문자열의 시작 부분에 나타난다.
$\$$	문자열의 끝에 나타난다.

Core 객체

❖ 정규표현식

○ 플래그

플래그	설명
i	대소문자를 무시한다.
g	일치하는 모든 경우를 다 찾는다.
m	개행 문자가 있는 경우 ^와 \$가 각 줄의 시작과 끝을 인식한다.

Core 객체

❖ 정규 표현식

- 검색
 - `exec(string)`
 - 존재하지 않으면 `null` 리턴
 - `test(string)`
 - `true/false`로 리턴

Core 객체

❖ 09_2_10_regexp.html

```
<body>
  <script>
    var str = "test S123 s-45 s67M S-8m s-123M s-superTM";
    var reg = new RegExp("[sS]-[0-9]*T?M");
    var result = reg.exec(str);
    document.write(result + "<br>");
  </script>
</body>
```

Core 객체

❖ 09_2_11_regliteral.html

```
<body>
  <script>
    var str = "test S123 s-45 s67M S-8m s-123M s-superTM";
    var reg = /[sS]-[0-9]*T?M/;
    var result = reg.exec(str);
    document.write(result + "<br>");
  </script>
</body>
```

Core 객체

❖ 09_2_12_searchreg.html

```
<body>
  <script>
    var str = "test S123 s-45 s67M S-8m s-123M s-superTM";
    var result = str.search(/[sS]-[0-9]*T?M/);
    document.write(result + "<br>");
  </script>
</body>
```

❖ 09_2_13_testjumin.html

```
<body>
  <script>
    var str = "890629-1914920";
    var result = str.search(/[0-9]{6}-[1234][0-9]{6}/);
    if (result != -1) {
      document.write("유효한 주민 등록 번호입니다.");
    } else {
      document.write("주민 등록 번호가 올바르지 않습니다.");
    }
  </script>
</body>
```

Core 객체

❖ 09_2_14_replaceexp.html

```
<body>
  <script>
    var str = "내가 그린 기린 그림은 암 기린을 그린 기린 그림이다.";
    var result = str.replace( /기린\s/g, "코끼리 ");
    document.write(str + "<br>");
    document.write(result + "<br>");
  </script>
</body>
```

Core 객체

❖ 09_2_14_replaceexp2.html

```
<body>
  <script>
    var count = 1;
    var str = "생각이란 생각할수록 생각이 나므로 생각하지 말아야 할
생각은 " +
      "생각하지 않으려고 하는 생각이 옳은 생각이라고 생각합니다.";
    var result = str.replace(/생각/g, "[$&]");
    document.write(str + "<br>");
    document.write("-----<br>");
    document.write(result + "<br>");
  </script>
</body>
```


❖ 09_2_14_replaceexp3.html

```
<body>
<script>
  var count = 1;
  var str = "생각이란 생각할수록 생각이 나므로 생각하지 말아야 할 생각
은 " +
    "생각하지 않으려고 하는 생각이 옳은 생각이라고 생각합니다."
  var result = str.replace(/생각/g, function(value) {
    return value + "(" + count++ + ")";
  });
  document.write(str + "<br>");
  document.write("-----<br>");
  document.write(result + "<br>");
</script>
</body>
```

Core 객체

❖ JSON

- JavaScript Object Notation
- JSON 타입
 - 문자열
 - “홍길동”
 - 숫자
 - 1000
 - 10.5
 - Boolean
 - true/false
 - null
 - 객체
 - {name: “홍길동”, age:20}
 - 배열
 - [1, 2, 3, 4, 5, “홍길동”, “김세진”]

Core 객체

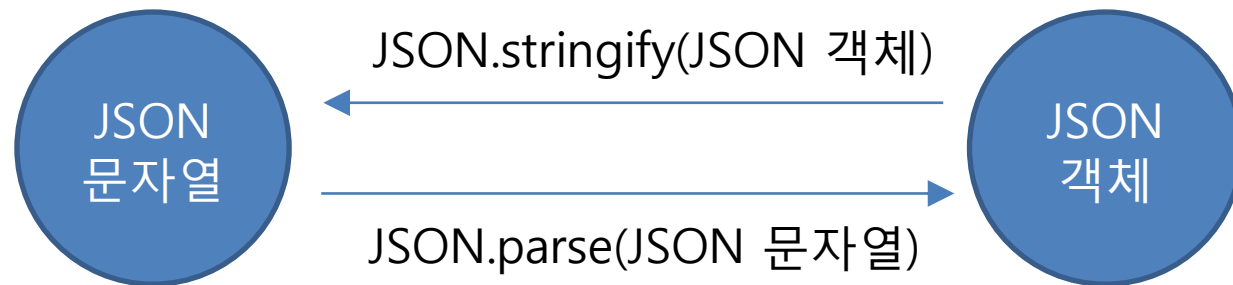
❖ JSON 문자열의 해석

- eval() 함수에 의한 JSON 문자열 해석
 - 방법
 - eval('JSON 문자열');
 - 문제점
 - 보안에 취약 : 일반 자바코드로 해석하기 때문에 실행문이 있는 경우
 - > 의도하지 않은 명령까지 실행
 - eval('{“x”:1}’) 실행시 에러 발생(실행 블록으로 해석)
 - > eval('(("{“x”:1})’)'로 실행해야 함
- JSON 문자열 파싱 라이브러리 이용
 - json2.js 라이브러리로 안전하게 해석 가능

Core 객체

❖ JSON 처리

- JSON2 라이브러리 이용



- 문자열은 반드시 큰 따옴표로 묶어야 함
- 프로퍼티명은 반드시 큰 따옴표로 묶어야 함(문자열로 표현)

Core 객체

❖ JSON 문자열 처리

- 문자열은 반드시 큰 따옴표로 묶어야 함
- 프로퍼티명은 반드시 큰 따옴표로 묶어야 함(문자열로 표현)

```
> var s = JSON.parse("'홍길동'"); // 작은 따옴표 문자열은 에러
```

```
Uncaught SyntaxError: Unexpected token
```

```
> var arr = JSON.parse("{age:1}"); // 프로퍼티명이 문자열이 아니면 에러
```

```
Uncaught SyntaxError: Unexpected token
```

Core 객체

❖ JSON 문자열 처리

```
// JSON 문자열에서 객체로 변환
> var s = '{"x":1, "y": 2, "val":"홍길동"}'; // JSON 문자열
> var obj = JSON.parse(s);
> console.log(obj.x);
1

// 객체에서 JSON 문자열로 변환
> JSON.stringify({x:1, y:2, val: '홍길동'});
{"x":1,"y":2,"val": "홍길동"}

// 배열의 JSON 문자열에서 배열 객체로 변환
> var arr = JSON.parse('[4, 3, 5]');
> console.log(arr);
[4, 3, 5]
> Array.isArray(arr);
true
```

❖ 09_2_15_jsonobject.html

```
<body>
  <script>
    var human = {
      name: "김상형",
      age: 29
    };
    var json = JSON.stringify(human);
    document.write(json + "</br />");

    var human2 = JSON.parse(json);
    document.write("name = " + human2.name + "<br>");
    document.write("age = " + human2.age + "<br>");
  </script>
</body>
```

Core 객체

❖ Audio

속성	설명
src	재생할 파일
volume	볼륨
currentTime	현재 재생 위치. 초단위이다.

Core 객체

❖ 09_2_16_audioobject.html

```
<body>
  <script>
    var audio = new Audio("saemaul.mp3");
    audio.play();
  </script>
</body>
```