

JavaScript

1. 선언문(Declaration)

선언문이란 HTML 페이지 내에 자바스크립트를 작성하기 위해 '여기서부터 자바스크립트를 작성할 것이다'라고 스크립트 개시를 선언하는 것.

```
<script type="text/javascript">  
    스크립트 실행문;  
</script>
```

- head 및 body 요소 내에 작성할 수 있음.

```
<script type="text/javascript">  
    document.write("hello javascript");  
</script>
```

- document를 객체(Object)라 부르고, write()는 메서드(Method)라고 부른다.
ex) 복사기.복사해라();
- 자바스크립트는 다양한 객체들로 구성되며, 각 객체마다 다양한 메서드를 갖고 있음.
자바스크립트 ==> 객체 기반 언어
- 한 줄 실행문을 작성했다면, 문장의 마침표에 해당하는 세미콜론(;)으로 마무리

- 외부형 선언문

```
<head>  
    <script type="text/javascript" src="js/note.js"></script>  
    // 외부형 js 파일을 소스로 불러온다는 의미의 외부형 선언문  
</head>
```

```
document.write("hello javascript");  
// 실행문 내용만 작성
```

- 외부형 선언문의 장점
 - 1) 관리가 편하다.
 - 2) 소스를 숨길 수 있다.
 - 3) 라이브러리(library) 언어를 사용할 수 있다. (jQuery, jQuery-mobile, Sencha touch 등)

2. 변수 및 데이터형(Variables & Data Type)

- 변수란 '변화하는 값(Data)'을 저장하는 공간
- 변수에는 한 번에 1개의 데이터(Data)만 저장할 수 있음.
새로운 데이터가 입력되면 기존의 값은 삭제됨.

```
var 변수명;  
var 변수명1, 변수명2, 변수명3;  
var 변수명=데이터 값;
```

```
<script type="text/javascript">  
    var num=10;           // 'num'이라는 변수에 숫자형 데이터 값인 10을 저장  
    var num=num/2;        // 'num'변수에 num/2의 결과인 5가 새롭게 저장  
    document.write(num);  
</script>
```

- 변수명 작성 시 주의사항

- 1) 대소문자 구분
- 2) 변수명 맨 앞에는 영문, _(underscore), \$ 등만 가능 (숫자가 맨 앞에 올 수 없음)
- 3) 변수명에는 영문, \$, _(underscore), 숫자만 가능 (%와 같은 특수 문자는 올 수 없음)

```
<script type="text/javascript">  
    var firstNum=100;  
    var nextNum=20;  
    var result=firstNum+nextNum  
    document.write(result); //출력값 : 120;  
</script>
```

- 변수에 저장 가능한 데이터의 종류

- 1) 문자(String)형 데이터

```
<script type="text/javascript">  
    var txt1="자바스크립트";  
    var txt2='2013'  
</script>
```

데이터가 " "(큰따옴표) 또는 ' '(작은따옴표)로 둘러싸인 데이터. 문자로 인식됨.

- 2) 숫자(Number)형 데이터

```
<script type="text/javascript">  
    var num=100;  
</script>
```

데이터가 숫자로 정의되는 데이터

3) 논리(Boolean)형 데이터

<pre><script type="text/javascript"> var result=true; </script></pre>
데이터가 true(참) 또는 false(거짓)으로 정의되는 데이터

4) 널(Null)형 데이터

<pre><script type="text/javascript"> var result=null; </script></pre>
데이터가 null(비어있음)로 정의되는 데이터

3. 연산자(Operator)

1) 산술연산자

피연산자가 2개 필요하며 '이항 연산자'라고도 부름.(사칙연산)

산술연산자	+	-	*	/	%
설명	더하기	빼기	곱하기	나누기	나머지

<pre><script type="text/javascript"> var a=10; var b=3; var result; result=a+b; document.write(result+"
"); result=a-b; document.write(result+"
"); result=a*b; document.write(result+"
"); result=a/b; document.write(result+"
"); result=a%b; document.write(result+"
"); </script></pre>

<유효성 검사 오류를 막는 CDATA선언>

- 내부형으로 작성된 스크립트에서 html 태그를 포함하게 되면 비록 기능은 정상적으로 작동할 지라도 브라우저의 유효성 검사기능들은 오류로 지적함.

- 내부형 스크립트에서는 html 코드를 포함해야 할 때에는 유효성 검사 오류를 막기 위해 'CDATA 선언'을 함께 작성해야 함. (외부형 파일로 따로 만들어 불러들일 경우에는 필요 없음)
- 'CDATA 선언문'은 내부형 자바스크립트 내에 HTML 태그가 포함되었을 경우, 이들을 모두 문자형 데이터로 인식시켜 이후 문서의 body 부분에 잘 출력되도록 돕는 역할을 함.

```
<script type="text/javascript">
//<![CDATA[
    실행문 내용들
//]]>
</script>
```

2) 비교연산자

두 개의 피연산자 데이터를 비교할 때 사용.

비교하여 잘못되었으면 false(거짓)를 올바르게 true(참)값을 반환함.

비교연산자	설명	
A>B	A가 B보다 크다	
A>=B	A가 B보다 크거나 같다	
A<B	A가 B보다 작다	
A<=B	A가 B보다 작거나 같다	
A==B	A와 B가 같다	데이터 형(type) 상관없음
A!=B	A와 B가 다르다	데이터 형(type) 상관없음
A===B	A와 B가 같다	데이터 형(type)이 일치할 때 true
A!==B	A와 B가 다르다	데이터 형(type)이 불일치할 때 true

```
<script type="text/javascript">
//<![CDATA[
    var a=10;
    var b=3;
    var result;
    result=a>b;
    document.write(result+"<br />");
    result=a<=b;
    document.write(result+"<br />");
    result=a==b;
    document.write(result+"<br />");
    result=a!=b;
    document.write(result+"<br />");
//]]>
</script>
```

3) 대입연산자

연산된 데이터를 지정된 변수에 대입할 때 사용.

대입연산자	사용	설명
=	A=10	A=10
+=	A+=B	A=A+B
-=	A-=B	A=A-B
=	A=B	A=A*B
/=	A/=B	A=A/B
%=	A%=B	A=A%B

```
<script type="text/javascript">
//
    var num1=10;
    var num2=20;

    num1+=5;
    document.write(num1+"&lt;br /&gt;");
    num1+=num2;
    document.write(num1+"&lt;br /&gt;");
    num1-=5;
    document.write(num1+"&lt;br /&gt;");
    num1*=5;
    document.write(num1+"&lt;br /&gt;");
    num1/=5;
    document.write(num1+"&lt;br /&gt;");
    num1%=4;
    document.write(num1+"&lt;br /&gt;");
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="470 957 525 974" data-label="Page-Footer"><p>- 5 -</p></div>
```

4) 증감연산자

증감연산자	++	--
설명	1씩 증가	1씩 감소

```
<script type="text/javascript">
//<![CDATA[
    var num1=10;
    var num2=20;

    num1++;
    document.write(num1+"<br />");
    num2--;
    document.write(num2+"<br />");
//]]>
</script>
```

5) 논리연산자

피연산자로 논리데이터인 true 또는 false가 올 수 있음.

논리연산자	(논리합)	&&(논리곱)	!(논리부정)
설명	or(또는)	and(그리고)	not(부정)

- 논리합 연산자는 피연산자에 true가 한 개만 있어도 true값을 반환
- 논리곱 연산자는 피연산자 중 false가 한 개라도 있으면 false값을 반환
- 논리 부정은 변수에 true가 저장되어 있으면 false를 false가 저장되어 있으면 true값을 반환

<prompt 메서드>

- 사용자에게 간단한 창을 띄워 질문에 대한 답을 받아 저장하는 기능
- 입력 받은 데이터를 문자형 데이터로 반환.

```
prompt("질문내용") 또는 prompt("질문내용", "기본답변" );
ex) prompt("당신의 애완동물 이름은?", "똥이");
```

ex1) 소개팅 대상 모집 문제

20~30대 대상의 소개팅을 준비 중입니다. 남성은 모집이 완료되었지만 여성은 아직 모집이 되지 않았습니다. prompt 메서드를 활용하여 성별과 나이와 결혼여부를 물어봅니다. 20~30대 미혼 여성이라면 결과 값이 true로, 아니면 결과 값이 false로 나오게 만드세요.

```
<script type="text/javascript">
//<![CDATA[
    var gender=prompt("당신의 성별은?", "여성");
    var marry=prompt("결혼 여부는?", "no");
    var age=prompt("당신의 나이는?", "20");
    var result= gender=="여성" && marry=="no" && age>=20 && age<=39;
    document.write(result);
//]]>
</script>
```

ex2) 국어, 영어, 수학 3과목 평균 및 통과여부 문제

prompt 메서드를 활용하여 국어, 영어, 수학 점수를 입력 받아, 전체 평균이 70점 이상이고 각 과목별 점수가 60점 이상이면 true를 아니면 false를 출력하게 만드세요.

```
<script type="text/javascript">
//<![CDATA[
    var korNum=Number(prompt("당신의 국어 점수는?", "0"));
    var engNum=Number(prompt("당신의 영어 점수는?", "0"));
    var mathNum=Number(prompt("당신의 수학 점수는?", "0"));
    var avg=(korNum+engNum+mathNum)/3;
    var result= avg>=70 && korNum>=60 && engNum>=60 && mathNum>=60;
    document.write("당신의 평균 점수는 "+ avg +"점입니다.<br />");
    document.write(result);
//]]>
</script>
```

6) 문자결합연산자

- 두 개의 피연산자 데이터 중 1개 이상이 문자형 데이터일 경우 문자 결합으로 실행.
- 피연산자 데이터가 모두 숫자형 데이터라면 산술 연산자로 실행.

```
<script type="text/javascript">
//<![CDATA[
    var txt1="아름다운 ";
    var txt2="우리강산";
    document.write(txt1+txt2+"<br />"); //문자결합=문자+문자

    var txt3="은하철도 ";
    var txt4=999;
    document.write(txt3+txt4+"<br />"); //문자결합=문자+숫자

    var num1=200;
    var num2=300;
    var txt5=String(1000); //String() 메서드로 숫자를 문자형 데이터로 변환
    document.write(num1+num2+txt5+"<br />"); // 문자결합=숫자+숫자+문자

    var addText="2002월드컵";
    addText+=" 대한민국";
    //기존 문자데이터에 추가 결합 "2002월드컵 대한민국"
    addText+=" 짹짹짹짹~!!";
    //기존 문자데이터에 추가 결합 "짹짹짹짹~!!" 추가
    document.write(addText);
    //출력값: "2002월드컵 대한민국 짹짹짹짹~!!"
//]]>
</script>
```

7) 조건연산자

조건식의 결과가 true(참) 혹은 false(거짓)일 때 각각 다른 실행문을 수행

- 기본형식

조건식 ? 실행문1 : 실행문2

- 조건식의 결과 값이 true(참)이면 실행문1을, 거짓(false)이면 실행문2를 실행


```
<script type="text/javascript">
//<![CDATA[
    var num1=30;
    var num2=20;
    num1>num2 ? alert("true(참)입니다.") : alert("false(거짓)입니다.");
//]]>
</script>
```

<confirm 메서드>

- 확인, 취소창을 띄울 수 있음.
- **확인 버튼을 누를 경우 true값을 반환, 취소 버튼을 누를 경우 false값을 반환**

```
confirm("내용")
ex) confirm("당신은 반려견과 함께 살고 있습니까?");
```

```
<script type="text/javascript">
//<![CDATA[
    var result=confirm("고양이를 키워보실 생각이 있나요?");
    //확인: true   취소: false
    var theText=result ? "네, 있습니다." : "아니요, 무섭습니다.";
    document.write(theText);
//]]>
</script>
```

4. 제어문(Control Statement)

1) 조건문

조건식을 만족(true)했을 때에만 실행문을 수행하는 제어문

가. if 문

조건식을 만족할 경우 중괄호({ })내의 실행문을 수행

```
if(조건식){
    실행문;
}
```

```
<script type="text/javascript">
//<![CDATA[
    var myPoint=prompt("시험 점수를 입력하세요.");
    if(myPoint>=70){
        document.write("축하합니다.<br />당신은 합격입니다.");
    };
//]]>
</script>
```

나. if~else 문

조건식을 만족(true)했을 때에는 if 문의 중괄호에 있는 실행문1을 실행하고, 만족하지 않을 경우(false)에는 else 문의 중괄호에 있는 실행문2를 실행. 출력결과가 2가지일 경우 사용.

```
if(조건식){
    실행문1;
} else{
    실행문2;
}
```

```
<script type="text/javascript">
//<![CDATA[
    var myPoint=prompt("시험 점수를 입력하세요.");
    if(myPoint>=70){
        document.write("축하합니다.<br />당신은 합격입니다.");
    } else{
        document.write("아쉽네요. <br />당신은 불합격입니다.");
    }
//]]>
</script>
```

다. if~else if 문

- 맨 위에 지정한 조건식1부터 아래로 내려가면서 순서대로 조건식 검사를 수행.
- 조건식을 검사하다가 조건식2에서 만족하게 되면 중괄호({ })에 든 실행문2를 수행하고 조건문 검사를 마침.
- 모든 조건식을 만족하지 못하면 마지막 else 문 중괄호 내에 기술된 실행문을 수행.
- 출력 결과가 3가지 이상일 경우 사용

```
if(조건식1){
    실행문1;
} else if(조건식2){
    실행문2;
} else if(조건식3){
    실행문3;
} else{
    실행문4;
}
```

```
<script type="text/javascript">
//<![CDATA[
    var myPoint=prompt("시험 점수를 입력하세요.");
    if(myPoint>=90){
        document.write("A학점");
    } else if(myPoint>=80){
        document.write("B학점");
    } else if(myPoint>=70){
        document.write("C학점");
    } else if(myPoint>=60){
        document.write("D학점");
    } else{
        document.write("F학점");
    }
//]]>
</script>
```

2) switch~case 문

- 지정한 변수에 저장된 데이터(data)와 정확히 일치되는 경우(case)가 있는지 검사하여 일치하는 데이터가 있을 경우 해당 실행문을 실행한 뒤 break문으로 반복을 끝냄.
- 일치하는 데이터가 없을 때에는 default에 해당하는 실행문을 수행한 뒤 문장을 마침.
- 조건문은 '데이터의 중복'의 경우라 하더라도 특정 '조건'을 허용할 때 사용하는 반면, switch~case문은 정확히 일치하는 경우(case)만 허용할 때 사용함.

ex) 조건문은 A>=90인 경우, 즉. A가 90이상이라는 조건 하에서는 A가 91이든 95이든 모두 만족하지만 switch~case문은 데이터 중복이 허용되지 않음.

```

var num=3;
switch(num){
    case 1:
        실행문1;
        break;
    case 2:
        실행문2;
        break;
    case 3:
        실행문3;
        break;
    default:
        실행문4;
}

```

```

<script type="text/javascript">
//<![CDATA[
var myArea=prompt("지역을 입력하세요","서울");
switch(myArea){
    case "서울":        //myArea에 값이 “서울”인 경우
        areaNum="02";
        break;
    case "경기":        //myArea에 값이 “경기”인 경우
        areaNum="031";
        break;
    case "부산":        //myArea에 값이 “부산”인 경우
        areaNum="051";
        break;
    default:            //myArea에 값이 “서울”인 경우
        alert("등록되지 않은 지역입니다")
}
document.write(myArea+"의 지역번호는 "+areaNum+"입니다.");
//]]>
</script>

```

3) 반복문

해당 조건식을 만족하는 동안 중괄호({ }) 안의 실행문을 반복적으로 수행.

가. while 문

- 특정 조건식을 만족하는 동안에 지정된 실행문을 반복적으로 수행

```
var i=초기값;
while(조건식){
    실행문;
    증감식;
}
```

```
<script type="text/javascript">
//<![CDATA[
var i=1;
while(i<=10){
    document.write(i+"<br />");
    i++;
}
//]]>
</script>
```

ex1) 10부터 1까지 내림차순으로 2의 배수(짝수)만 출력해 보세요.

```
<script type="text/javascript">
//<![CDATA[
var i=10;
while(i>=1){
    if(i%2==0){
        document.write(i+"<br />");
    }
    i--;
}
//]]>
</script>
```

나. for 문

```
for(var k=초기값; 조건식; 증감식){  
    실행문;  
}
```

```
<script type="text/javascript">  
//<![CDATA[  
for(var k=1; k<=10; k++){  
    document.write(k+"<br />");  
}  
//]]>  
</script>
```

ex1) head 태그를 사용하여 다음과 같이 출력하시오.(HTML 태그 중 일부를 변수로 사용하기)

```
<script type="text/javascript">  
//<![CDATA[  
for(var k=6; k>=1; k--){  
    document.write("<h"+k+">h"+k+" 재미있는 자바스크립트</h"+k+">");  
}  
//]]>  
</script>
```

ex2) 구구단 중 2단을 출력하시오.

```
<script type="text/javascript">  
//<![CDATA[  
var result;  
var dan=2;  
document.write("<h1>"+dan+"단</h1>");  
for(var k=1; k<=9; k++){  
    result=k*dan;  
    document.write("2 x "+k+" = "+ result +"<br />");  
}  
//]]>  
</script>
```

다. 2중 for 문

ex2) 2단부터 9단까지 구구단을 출력하시오.

```
<script type="text/javascript">
//<![CDATA[
for(var dan=2; dan<=9; dan++){
    document.write("<h1>" + dan + "단</h1>");
    for(var k=1; k<=9; k++){
        document.write(dan + " x " + k + " = " + (dan*k) + "<br />");
    }
}
//]]>
</script>
```

라. break문

- 반복문의 실행문들은 수행과정에서 break문을 만나면 해당 반복문을 종료함.
- 반복문을 정지시킬 때 사용.

```
<script type="text/javascript">
//<![CDATA[
for(var k=1; k<=10; k++){
    document.write(k + "<br />");
    if(k==5) break;
}
//]]>
</script>
```

마. continue 문

반복문이 continue 문을 만나면 더 이상의 실행문을 수행하지 않고 바로 증감식으로 돌아가 실행을 반복됨.

```
<script type="text/javascript">
//<![CDATA[
for(var k=1; k<=10; k++){
    if(k%2==1) continue;
    document.write(k + "<br />");
}
//]]>
</script>
```

5. 객체(Object)

- 자바스크립트는 객체기반(Object based)의 스크립트 언어
- **객체** = 특정 기능을 수행하는 단위체
- 해당 객체가 지닌 각각의 기능 = **메서드**

ex) 휴대폰.통화하기(); //메서드
 휴대폰.색상; //속성

- 항상 객체를 중심으로 기술.
- 객체명 뒤의 점(.)을 경계로 하여 그 객체에 해당하는 메서드(기능) 또는 속성을 기술
- 내장 객체 : 자바스크립트 내에 내장된 객체
 ex) Date, Array, Math 등
- 브라우저 객체(BOM) : 각 웹 브라우저에서 지원되는 객체
 ex) window, document(DOM), location 등
- 사용자 정의 객체 : ex) 함수

※ 객체의 생성 - new 연산자

- new 연산자는 새로운 객체를 생성할 때 사용
- 앞에 new를 붙이고 내장 객체의 함수명을 써서 작성

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

- a. 날짜 정보를 제공해주는 Date 객체를 new 연산자를 이용하여 생성
- b. 앞으로 사용할 객체에 today라는 인스턴트 네임(객체 이름표)를 부여
- c. 날짜와 관련된 정보가 궁금할 때는 today에게 물어보는 형식
 ex) today야, 현재 연도에 대한 정보를 보여줘!
 today.getFullYear();

1) Date 객체

- 현재 또는 특정 날짜와 시간에 대한 정보를 제공.
- 객체 생성 시 특정 날짜를 지정하지 않았다면 현재 컴퓨터 시스템에 입력된 날짜에 대한 정보를 제공.

```
var 변수명 = new Date();  
var 변수명 = new Date(년, 월, 일, 시, 분, 초)
```

가. 날짜 정보 제공 메서드

메서드 종류	설명
getFullYear()	날짜에서 연도에 대한 정보를 제공
getMonth()	날짜에서 월에 대한 정보를 제공. 0~11 반환 0=1월, 11=12월
getDate()	날짜에서 일에 대한 정보를 제공
getDay()	날짜에서 요일에 대한 정보를 제공 0=일요일, 6=토요일
getHours()	시간에서 시(0~23)에 대한 정보를 제공
getMinutes()	시간에서 분(0~59)에 대한 정보를 제공
getSeconds()	시간에서 초(0~59)에 대한 정보를 제공
getMilliseconds()	시간에서 밀리(1/1000)초에 대한 정보를 제공
getTime()	1970년 1월 1일 이후 경과 시간을 밀리초로 제공

나. 날짜 지정 메서드

메서드 종류	설명
setFullYear()	날짜에서 연도만 새로 지정할 때
setMonth()	날짜에서 월만 새로 지정할 때 1월=0, 12월=11
setDate()	날짜에서 일만 새로 지정할 때
setHours()	시간만 새로 지정할 때
setMinutes()	분만 새로 지정할 때
setSeconds()	초만 새로 지정할 때
setMilliseconds()	밀리초만 새로 지정할 때
setTime()	현재까지 경과된 밀리초를 새로 지정할 때
goGMTString()	날짜를 그리니치 천문대 표준시로 표시
toString()	날짜를 문자 형식으로 표시

```

<script type="text/javascript">
//
    document.write("&lt;h1&gt;현재날짜/시간 정보&lt;/h1&gt;")
    var today=new Date();
    var nowMonth=today.getMonth()+1;    //현재 월 (0~11)
    var nowDate=today.getDate();        //현재 일
    var nowDay=today.getDay();          //현재 요일 (0:일요일~6:토요일)
    var nowHours=today.getHours();      //현재 시간
    var nowMinutes=today.getMinutes();  //현재 분
    var nowSeconds=today.getSeconds();  //현재 초
    var nowTime=today.getTime();        //1970년1월1일부터 밀리초 경과된 시간

    document.write("월 : "+nowMonth+"&lt;br /&gt;");
    document.write("일 : "+nowDate+"&lt;br /&gt;");
    document.write("요일 : "+nowDay+"&lt;br /&gt;");
    document.write("시 : "+nowHours+"&lt;br /&gt;");
    document.write("분 : "+nowMinutes+"&lt;br /&gt;");
    document.write("초 : "+nowSeconds+"&lt;br /&gt;");
    document.write("경과시간 : "+nowTime+"&lt;br /&gt;");
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="115 544 743 561" data-label="Text">
<p>ex1) 이미지를 이용하여 요일별로 다른 이미지가 출력되도록 만들어 보세요.</p>
</div>
<div data-bbox="465 957 529 973" data-label="Page-Footer">
<p>- 18 -</p>
</div>
```

```

<script type="text/javascript">
//
    var today=new Date();
    var nowDay=today.getDay(); // 0~6(일요일~토요일)
    if(nowDay==1){
        document.write('&lt;img src="images/img1.jpg" alt="월요일" /&gt;');
    }else if(nowDay==2){
        document.write('&lt;img src="images/img2.jpg" alt="화요일" /&gt;');
    }else if(nowDay==3){
        document.write('&lt;img src="images/img3.jpg" alt="수요일" /&gt;');
    }else if(nowDay==4){
        document.write('&lt;img src="images/img4.jpg" alt="목요일" /&gt;');
    }else if(nowDay==5){
        document.write('&lt;img src="images/img5.jpg" alt="금요일" /&gt;');
    }else if(nowDay==6){
        document.write('&lt;img src="images/img6.jpg" alt="토요일" /&gt;');
    }else{
        document.write('&lt;img src="images/img7.jpg" alt="일요일" /&gt;');
    }
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="115 544 240 561" data-label="Section-Header">
<h2>2) Math 객체</h2>
</div>
<div data-bbox="115 564 884 623" data-label="List-Group">
<ul>
<li>- 단순히 수를 더하거나 뺄 경우 ==&gt; 산술 연산자 사용</li>
<li>- 소수점 반환, 최대/최소값 등 특정 공식을 사용해야 할 경우 ==&gt; 수학 객체 메서드를 사용</li>
<li>- 수학 객체는 new(생성자 키워드)를 사용하지 않는 정적인 객체</li>
</ul>
</div>
<div data-bbox="121 630 902 843" data-label="Table">
<table>
<tr>
<th>메서드 종류</th><th>설명</th></tr>
<tr>
<td>max(숫자1, 숫자2, 숫자3)</td><td>숫자 중 최대값을 반환</td></tr>
<tr>
<td>min(숫자1, 숫자2, 숫자3)</td><td>숫자 중 최소값을 반환</td></tr>
<tr>
<td>round(숫자)</td><td>소수점 첫째 자리를 반올림</td></tr>
<tr>
<td>ceil(숫자)</td><td>소수점 자리를 무조건 올림</td></tr>
<tr>
<td>floor(숫자)</td><td>소수점 자리를 무조건 절삭</td></tr>
<tr>
<td>abs(숫자)</td><td>수의 절대값을 반환</td></tr>
<tr>
<td>random()</td><td>0과 1 사이의 난수를 발생</td></tr>
</table>
</div>
<div data-bbox="465 957 529 973" data-label="Page-Footer">
<p>- 19 -</p>
</div>
```

```

<script type="text/javascript">
//
    var maxNum=Math.max(30,70,5);
    document.write("최대값 : "+maxNum+"&lt;br /&gt;");

    var minNum=Math.min(8,10,15);
    document.write("최소값 : "+minNum+"&lt;br /&gt;");

    var roundNum=Math.round(3.5);
    document.write("반올림 : "+roundNum+"&lt;br /&gt;");

    var ceilNum=Math.ceil(5.1);
    document.write("소숫점 올림 : "+ceilNum+"&lt;br /&gt;");

    var floorNum=Math.floor(10.5);
    document.write("소숫점 내림 : "+floorNum+"&lt;br /&gt;");

    var absNum=Math.abs(-10);
    document.write("절대값 : "+absNum+"&lt;br /&gt;");

    var ranNum=Math.random();
    document.write("0~1 난수 : "+ranNum+"&lt;br /&gt;");

//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="115 627 664 644" data-label="Text">
<p><b>ex1) 접속할 때마다 랜덤으로 이미지가 출력되도록 만들어 보세요.</b></p>
</div>
<div data-bbox="130 659 871 780" data-label="Text">
<pre>
&lt;script type="text/javascript"&gt;
//<![CDATA[
    var imgNum=Math.floor(Math.random()*3)+1;
    document.write("&lt;img src=' images/img"+imgNum+".jpg' alt='랜덤이미지' /&gt;");
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="115 793 636 831" data-label="List-Group">
<ul>
<li>- 파일명이 img는 공통적이고 뒤에 부은 숫자만 1~3으로 구분</li>
<li>- 1~3까지의 난수를 활용</li>
</ul>
</div>
<div data-bbox="117 855 859 873" data-label="Text">
<p>※ random 메서드를 이용해서 원하는 숫자(작은수~큰수) 사이의 난수값을 도출하는 공식</p>
</div>
<div data-bbox="130 886 581 904" data-label="Text">
<pre>Math.floor(Math.random()*(큰수-작은수+1))+작은수</pre>
</div>
<div data-bbox="465 957 529 973" data-label="Page-Footer">
<p>- 20 -</p>
</div>
```

3) String 객체

- 내장 객체 중 활용빈도가 가장 높음
- 문자형 데이터 = 문자형 변수 또한 String 객체
- 변수에 숫자를 저장한 후 String 객체에 해당하는 메서드를 사용하면 에러 발생
- new 연산자를 사용해서 생성 가능.

※ 보편적인 생성법

```
var 변수명 = new String("자바스크립트");
```

※ 간편한 생성법

```
var 변수명 = "자바스크립트"
```

ex) var theText = "web js hi ciy"

0	1	2	3	4	5	6	7	8	9	10	11	12
w	e	b		j	s		h	i		c	i	y

- theText라는 변수에 13개의 인덱스로 구분되어 저장
- 문자의 공백도 각각 하나의 문자로 인식
- 각 낱글자별 인덱스 번호는 0부터 부여 ==> 총 문자개수 - 1 = 마지막 인덱스 번호

메서드 종류	설명	결과
theText.bold()	텍스트를 진하게 표현하여 반환	
theText.link("URL")	텍스트에 링크를 걸어 반환	
theText.length	텍스트의 개수를 반환	13(글자)
theText.toLowerCase()	영문자를 소문자로 바꿔 반환	
theText.toUpperCase()	영문자를 대문자로 바꿔 반환	
theText.indexOf("i")	처음부터 "i"를 찾아 최초로 일치하는 인덱스 번호를 반환. 없을 경우 -1을 반환	8
theText.lastIndexOf("i")	끝에서부터 "i"를 찾아 최초로 일치하는 인덱스 번호를 반환. 없을 경우 -1을 반환	11
theText.charAt(5)	5번 인덱스에 저장된 문자 데이터를 반환	s
theText.substring(4,6)	4번부터 6번 인덱스까지의 문자 데이터를 반환	js
theText.substr(10,3)	10번 인덱스부터의 3글자를 반환	ciy
theText.replace("web","bye")	텍스트 중 "web"을 "bye"로 치환해서 반환	bye
theText.slice(7,9)	7번부터 9번 인덱스 전까지의 일부만 잘라서 반환	hi
theText.concat("good")	끝에 "good"이라는 문자를 결합하여 반환	good
theText.split(" ")	" "(공백)을 기준으로 데이터를 분리(coma)	

```
<script type="text/javascript">
//
    var theText="web js hi ciy";
    document.write(theText.bold()+"&lt;br /&gt;");
    document.write(theText.link("http://www.naver.com")+"&lt;br /&gt;");
    document.write(theText.length+"&lt;br /&gt;");
    document.write(theText.toLowerCase()+"&lt;br /&gt;");
    document.write(theText.toUpperCase()+"&lt;br /&gt;");
    document.write(theText.indexOf("i")+"&lt;br /&gt;");
    document.write(theText.lastIndexOf("i")+"&lt;br /&gt;");
    document.write(theText.charAt(5)+"&lt;br /&gt;");
    document.write(theText.substring(4,6)+"&lt;br /&gt;");
    document.write(theText.substr(10,3)+"&lt;br /&gt;");
    document.write(theText.replace("web","bye")+"&lt;br /&gt;");
    document.write(theText.slice(7,9)+"&lt;br /&gt;");
    document.write(theText.concat(" good")+"&lt;br /&gt;");
    document.write(theText.split(" ")+"&lt;br /&gt;");
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="465 957 529 973" data-label="Page-Footer"><p>- 22 -</p></div>
```

```

<script type="text/javascript">
//
    document.write("&lt;h3&gt;문자치환&lt;/h3&gt;");
    var theText1="img2_out.jpg";
    theText1=theText1.replace("out.jpg","over.jpg");
    document.write(theText1+"&lt;br /&gt;"); //out을 over로 치환

    document.write("&lt;h3&gt;문자추출&lt;/h3&gt;");
    var theText2="images/img10.jpg";
    theText2=theText2.split("/")[1];
    theText2=theText2.substring(3); //문자열의 3번 인덱스부터 끝까지의 문자만 남김
    theText2=parseInt(theText2); //정수 10만 남김
    document.write(theText2+"&lt;br /&gt;");

    document.write("&lt;h3&gt;마지막 문자 추출&lt;/h3&gt;");
    var theText3="hello javascript";
    var lastIndex=theText3.length-1;
    theText3=theText3.charAt(lastIndex);
    document.write(theText3+"&lt;br /&gt;")

    document.write("&lt;h3&gt;특정문자 포함 여부확인하기&lt;/h3&gt;");
    var theText4="hello javascript tel phone";
    var findIndex=theText4.indexOf("phone");
    if(findIndex&lt;0){
        document.write("텍스트에 찾는 문자가 포함되지 않았습니다.")
    } else{
        document.write("텍스트에 찾는 문자가 포함되어 있습니다.")
    }
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="116 710 913 790" data-label="List-Group">
<ul>
<li>- 문자치환 ==&gt; 웹 페이지 내의 버튼 이미지를 바꿀 때 많이 사용.</li>
<li>- 문자추출 ==&gt; 요소의 속성값을 가져와 그 숫자를 활용할 경우에 많이 사용</li>
<li>- indexOf 메서드 ==&gt; 찾는 문자의 인덱스 번호 반환. 찾는 문자가 포함되어 있는지 여부를 알아볼 때 많이 사용.</li>
</ul>
</div>
<div data-bbox="466 957 528 973" data-label="Page-Footer">
<p>- 23 -</p>
</div>
```

4) Array 객체

- 일반변수 ==> 한 번에 한 개의 데이터만 저장
- 배열 ==> 하나의 변수에 여러 개의 데이터를 저장
- 하나의 변수에 여러 개의 공간으로 이뤄진 배열객체를 대입하고 저장. 각 공간마다의 인덱스(index) 번호가 존재.

※ new 연산자를 사용한 배열 객체 생성법

```
var 변수명 = new Array();
```

```
변수[0] = "값1";
```

```
변수[1] = "값2";
```

```
변수[2] = "값3";
```

```
변수[3] = "값4";
```

```
변수[4] = "값5";
```

※ 저장 공간을 지정한 배열 객체 생성법

```
var 변수명 = new Array(5);
```

- 전체 공간을 5개로 나눠서 총 5개의 데이터만 저장

※ 편리한 배열 객체 생성법

```
var 변수명 = new Array[값1, 값2 값3, 값4];
```

- new 연산자를 사용하지 않고 변수에 여러 개의 값을 지정하는 방법


```

<script type="text/javascript">
//
    //특정 인덱스에 저장된 데이터 출력
    var name=new Array();
    name[0]="싸이";
    name[1]="조용필";
    name[2]="악동뮤지션";
    name[3]="로이킴";
    name[4]="장재인";
    document.write(name[3]+"&lt;br /&gt;"); //3번 인덱스에 저장된 데이터를 출력

    //오늘의 요일 알아보기
    var days=new Array("일","월","화","수","목","금","토");
    var today=new Date();
    var yoi=days[today.getDay()]; //오늘 요일에 해당하는 값을 반환 일:0 ~ 토:6
    document.write(yoi+"요일 &lt;br /&gt;"); //오늘의 요일 반환

    // 전체 합 구하기
    var money=[100,100,200,50];
    var theLen=money.length;
    var total=0;
    for(var i=0; i&lt;theLen; i++){
        total=total+money[i];
    }
    document.write(total);
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="465 957 529 973" data-label="Page-Footer">
<p>- 25 -</p>
</div>
```

```

<script type="text/javascript">
//
    //reverse()-배열 데이터 역순 정렬
    var theFood=["짜장면","냉면","갈비탕","돈까스","삼겹살"];
    theFood.reverse();
    console.log(theFood)

    //sort()-배열 데이터 이름 또는 숫자 정렬
    var theSubway=["2호선","9호선","7호선","4호선","3호선"];
    theSubway.sort();
    console.log(theSubway);

    //concat()-배열들에 데이터를 결합
    var Mountain1=["남산","도봉산","관악산","청계산","북한산"];
    var Mountain2=["설악산","지리산","한라산","속리산"];
    var joinResult=Mountain1.concat(Mountain2);
    console.log(joinResult);

    //slice(시작인덱스,마지막인덱스)-원하는 데이터만 선택
    var theName=["싸이","조용필","악동뮤지션","로이킴","장재인"];
    var result=theName.slice(2,4);
    console.log(result);

    //join("구분자")-구분자 기준으로 데이터를 합칩니다.
    var theProduct=["냉장고","컴퓨터","선풍기","밥솥","핸드폰"];
    var joinResult=theProduct.join("-");
    console.log(joinResult);
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="465 957 529 973" data-label="Page-Footer">
<p>- 26 -</p>
</div>
```

```
<script type="text/javascript">
//
    var theSubway=["2호선","9호선","7호선","4호선","3호선"];
    var newLine1="1호선";
    var newLine2="분당선";

    theSubway.shift();
    console.log(theSubway);
    theSubway.unshift(newLine1);
    console.log(theSubway);

    theSubway.pop();
    console.log(theSubway);
    theSubway.push(newLine2);
    console.log(theSubway);
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="90 460 243 478" data-label="Text"><p>6. 함수(function)</p></div><div data-bbox="90 480 237 500" data-label="Text"><p>7. 이벤트(Event)</p></div><div data-bbox="90 501 450 520" data-label="Text"><p>8. 브라우저 객체(Browser Object Model)</p></div><div data-bbox="465 957 531 974" data-label="Page-Footer"><p>- 27 -</p></div>
```