



제1강 자바스크립트 기초

Javascript & JQuery Programming

Contents

1. 자바스크립트 소개

2. 자바스크립트 예제

3. Javascript 기본

4. 변수 자료형

5. 연산자 및 제어문

6. 내장함수

7. 사용자함수

8. 지역변수 & 전역변수

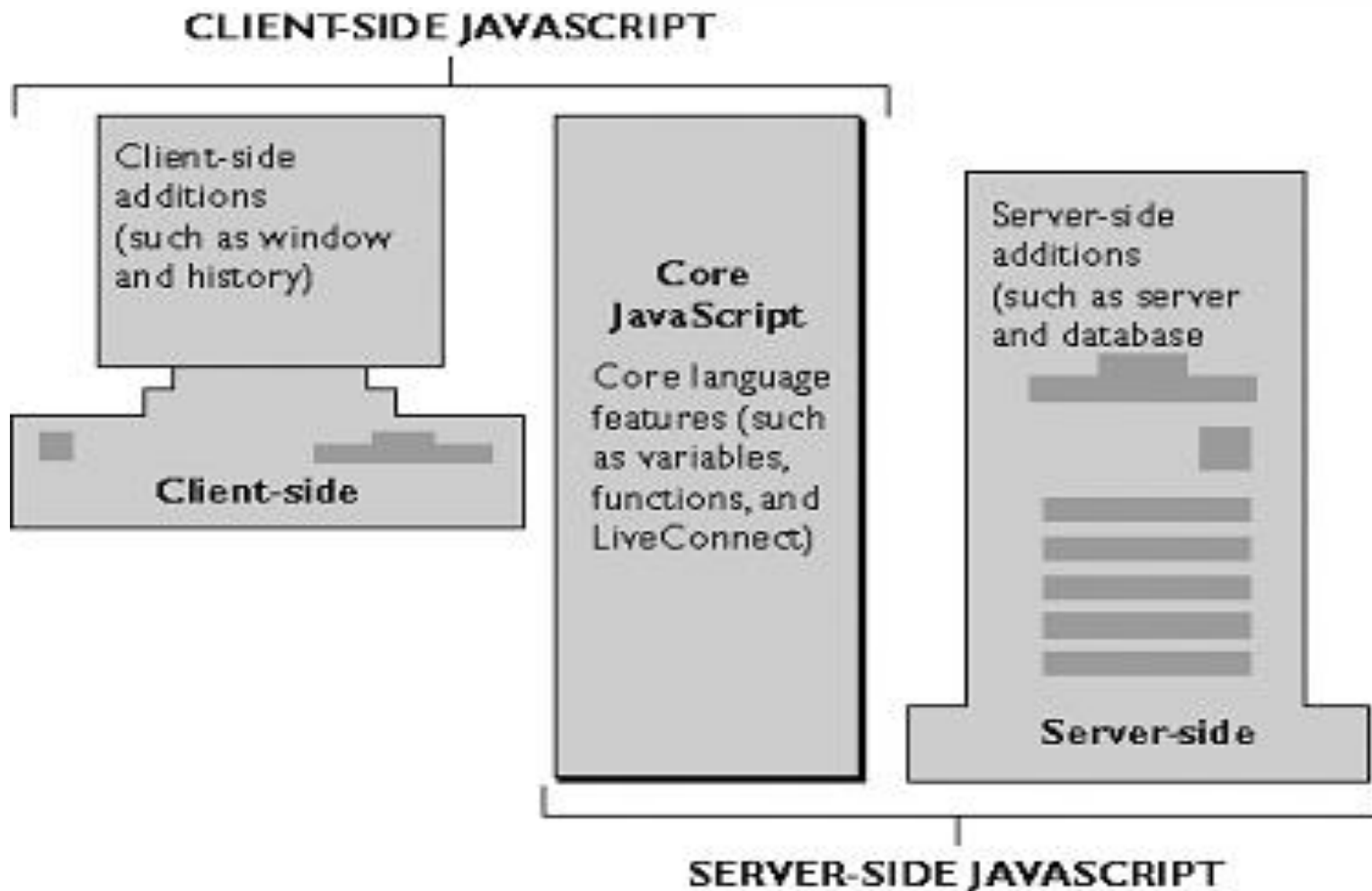
자바스크립트 소개

❖ 자바스크립트란 무엇인가?

- 자바스크립트(JavaScript)는 기존 HTML의 정적인 특성을 보완하여
- 사용자의 요구에 따른 상호 작용이 가능한 웹 페이지를 작성하게 하는 대표적인 스크립트 언어이다.

자바스크립트 소개

❖ 클라이언트 측 자바스크립트와 서버 측 자바스크립트



자바스크립트 소개

❖ 자바스크립트 작성 방법: <SCRIPT>태그 사용

- 자바스크립트(JavaScript) 실행 구조
- * alertjs.html

```
<script type="text/javascript">  
  
</script>
```

```
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
<style type="text/css">  
  
</style>  
  
<script type="text/javascript">  
|   alert("Hello Javatpoint");  
</script>  
  
</head>
```

자바스크립트 소개

❖ 자바스크립트 소스 파일을 삽입하는 방법

❖ *alertjs2.html

- src(=source) 옵션에 미리 작성된 소스 파일명을 기술
`<script src="소스파일명.js">`

`</script >`

- 이 때 주의해야 할 점은 자바스크립트 소스를 담은 파일의 확장자는 일반적으로 ".js"이어야 한다.

[소스파일명.js]

`<!-- 자바스크립트 명령들 -->`

- 자바스크립트 명령으로 미리 작성된 소스(source) 파일을 사용하여 자바스크립트를 실행하는 장점은 미리 작성된 소스(source) 파일을 서로 다른 여러 HTML 문서에서 삽입하여 사용할 수 있다

자바스크립트 소개

❖ 자바스크립트 소스 파일을 삽입하는 방법

- 함수가 있는 소스를 function.js 파일로 생성하여 적용

```
<!DOCTYPE>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <style type="text/css">

    </style>

    <script type="text/javascript">
      function msg(){
        alert("Hello Javatpoint");
      }
    </script>
  </head>

  <body>
    <p>Welcome to JavaScript</p>
    <form>
      <input type="button" value="click" onclick="msg()"/>
    </form>
  </body>
</html>
```

자바스크립트 예제

❖ 배경색 바꾸는 예제

■ * bgcolor.html

```
<script>
function change() {
    document.bgColor='red'; }
</script>
<body>
<center>
<form>
<input type = "button" value = "배경색 변경"
onclick="change();">
</form>
</center>
</body>
```


자바스크립트 예제

❖ 암호체크 *passwd.html

[illegible]

❖ 단락글자 지우기 *phide.html

- <!DOCTYPE html>
- <html>
- <head>
- <style>
- #demo {
- width: 320px; height: 175px;
- background-image: url(images/text-box-png-16.png);
- background-repeat: no-repeat;
- text-align: center; line-height: 175px;
- font-size: 300%; font-weight: bold;
- }
- </style>
- <body>
- <h2>버튼 클릭으로 글자 숨기기 예제</h2>

❖ 단락글자 지우기 ***phide.html**

- `<p id="demo">환영합니다</p>`
- `<button type="button" onclick="hide();">Click Me!</button>`
- `<script>`
- `function hide() {`
- `document.getElementById('demo').style.display='none';`
- `}`
- `</script>`
- `</body>`
- `</html>`

자바스크립트 예제 * exam1.html

1- 다음 덧셈 예제를 동작하시오(코드작성 후)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script>
    // 1 동작 함수 정의 - 확인
    // 2 기능
    // showSum = function() { } // 특수한 방법
    function showSum() {
        // 2-1 첫번째 숫자를 가져옴 (변수a 임시 저장)
        num1 = document.getElementById("first").value;
        // 2-2 위에 것 반복
        num2 = document.getElementById("second").value;
        // 2-3 연산
        // (문자) -> 정수 변경
        result = parseInt(num1) + parseInt(num2);
        // 2-4 세번째 결과값에 보여줌
        document.getElementById("sum").value = result;
    }
</script>
</head>
<body>
    <input type="text" id="first">
    +
    <input type="text" id="second">
    <button onclick="showSum()">=</button>
    <input type="text" id="sum">
</body>
</html>
```

자바스크립트 기본

❖ 코딩시 지켜야 할 점

- 영문자, 숫자, 특수 문자들을 사용하여 자바스크립트 프로그램은 내용에 따라 여러 라인의 문장으로 구성된다.
- ① 각 라인은 한 문장이상을 자유롭게 코딩할 수 있다.
- ② 한 라인에 하나 이상의 문장을 기술하고자 할 때는 문장과 문장 사이에 반드시 세미콜론(;)을 표시하여 구분 한다.
- ③ 만일 한 라인에 하나의 문장만을 기술하고자 할 때는 문장 끝에 세미콜론(;)을 생략해도 상관없다.
- ④ 프로그램 실행 순서는 특별한 제어가 없는 한 라인 순으로 처리된다.

자바스크립트 기본

❖ 주석(comment) 달기

- 한 줄의 주석을 달아줄 때는 "//"를 사용하고,
- 여러 줄의 주석을 달아줄 경우에는 "/*" 와 "*/"를 사용한다.

[예]

/* 여러 줄의 주석을 달아줄 경우에 사용하는 주석문입니다. */
// 한 줄의 주석을 달아줄 때 사용하는 주석문입니다.

자바스크립트 기본

❖ 변수 이름을 만들 때 주의할 점

- ① 자바스크립트에서 변수 이름을 만들 때 첫 자는 반드시 영문자나 "_"로 시작해야 하고, 두 번째 문자부터는 영문자, 숫자 등을 조합하여 의미있게 구성할 수 있다.
- [올바르게 만든 예] : `sum`, `irum`, `_javaScript`, `lhk140` 등
- [잘못 만든 예] : `7up`, `*irum`, `140script` 등
- ② 대문자와 소문자를 구별해서 변수 이름을 만들어야 한다.
- 대소문자로 서로 다르게 구성되었기 때문에 서로 다른 변수로 취급됨을 주의해야 한다.
- `result`, `Result`, `RESULT`, `REsult` 등
- `basic`, `BASIC`, `Basic` `baSiC` 등
- ③ 자바스크립트에서 특수한 용도를 가지고 미리 정의된 단어들이 `if`, `var` 등과 같은 예약어(reserved word)는 변수 이름으로 사용할 수 없다.

자바스크립트 기본

❖ 변수명에서 대소문자를 구별하지 않아 오류가 발생한 예 * **error_var.html**

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=utf-8" />
  <script type="text/script">
    a = 10;    b = 5;
    RESULT = 100;
    result = a + b;
    document.write(" <h3> 변수명에 대소문자가 구별되지 않아<p>");
    document.write(" 10 + 5의 결과가 100이라고 잘못 나온 예제입니다.");
    alert( a + " + " + b + " = " + RESULT);
  </script>
</head>
</html>
```


변수 자료형

❖ 자바스크립트의 4가지 자료형(data type)

- 수치 자료형
- 문자열 자료형
- boolean 자료형
- null

❖ 수치 자료형 - 정수

- ① 10진수 정수형
 - 0에서 9 사이의 숫자로 나타낸다.
- ② 8진수 정수형
 - 8진수(octal) 정수 표현은 숫자 0으로 시작되어 0과 7사이의 숫자로 나타내며, 8진수 정수로 표현된 자료는 출력될 때 10진수로 변환되어 출력된다.
 - Ex: 015, 067 등
- ③ 16진수 정수형
 - 16진수(hexadecimal) 정수 표현은 0x(숫자 0과 알파벳 x) 혹은 0X로 시작되어 0에서 f사이의 숫자로 나타내며, 16진수 정수로 표현된 자료는 출력될 때 10진수로 변환되어 출력된다.
 - Ex: 0xa1, 0x6b 등

자료형

❖ 수치 자료형 - 실수

- ① 110.23, 3.14159 등과 같이 소수 부분이 있는 숫자.
- ② 지수 형식은 실수 중에서 아주 작은 숫자나 아주 큰 숫자를 짧게 표현하기 위해 사용한다. E(exponent)는 10의 몇승을 의미한다.
- $120000000 = 1.2 * 10^8 \rightarrow 1.2E+8$

```
<script type="text/script">
```

```
a = 10;
```

```
b = 5.444
```

```
result = a + b;
```

```
document.write("결과는 : " + result);
```

```
</script>
```

[실행 결과]

결과는 : 15.443999999999999

❖ 문자 자료형

- ① 자바스크립트에서는 예와 같이 특수 문자 이중 따옴표(" ")나 단일 따옴표(' ') 사이에 들어가는 자료를 모두 문자열로 처리한다.
- "대화 상자", "10 + 5",
■ '두수', '123'
- ② 자바스크립트는 문자열 자료임을 나타내주는 이중 따옴표(")나 단일 따옴표(')를 원래의 문자 자료로 출력하고 싶으면, 특수 문자 "\\"(backslash)를 앞에 적어 준다.

자료형

❖ 문자 자료형 예제 * `backslash.html`

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=utf-8" />
  <script type="text/javascript">
    // 출력 결과와 같이 이중 따옴표 (") 가 출력된다.
    document.write("\\"이중 따옴표\"의 출력을 보여주는 예제입니다.<BR>");

    // 출력 결과와 같이 단일 따옴표 (') 가 출력된다.
    document.write("\'단일 따옴표\'의 출력을 보여주는 예제입니다.");
  </script>
</head>
</html>
```

❖ boolean 자료형 * ifcond.html

- ① 조건에서 참(true)과 거짓(false) 값으로 결과를 나타내는 불리언(Boolean) 자료형이 요구된다.
- ② 불리언 자료형은 true와 false라는 값만 사용된다.
- a = true, b = false

```
<html>
<head>
  <meta http-equiv="Content-Type"
  |      content="text/html; charset=utf-8" />
  |
  |
  <script type="text/script">
a = false; // 비교식등의 결과
if(a)
|   document.write("true");
else
|   document.write("false");
  </script>
</head>

</html>
```

❖ null 형

- ① 자바스크립트는 특별하게 null이라는 자료 값이 사용되고 있다. 이 null이라는 값은 말 그대로 아무 값도 없다는 것을 의미한다.
- ② 자바스크립트에서 변수를 선언할 때 var라는 예약어 (reserved word)를 사용하여 변수를 선언해주는데, 만일 어떤 자료형의 값을 갖게될지 모르는 변수의 경우에는 null 값으로 초기화해 주면 안전하다.

❖ 변수를 선언하지 않고 사용하는 예

- 자!! 자바스크립트는 다른 프로그래밍 언어와 다르게 변수에 자료형을 선언하지 않고, 변수에 자료를 할당하기만 하면 해당하는 4가지 자료형(data type) 중의 하나인 변수로 자동 선언되어 사용된다.

- | | |
|--------------------------------|---|
| ■ <code>javaScript = 80</code> | // <code>javaScript</code> 는 정수형 변수이다. |
| ■ <code>pi = 3.14159</code> | // <code>pi</code> 는 실수형 변수로 이다. |
| ■ <code>name = "임 홍준"</code> | // <code>name</code> 은 문자열 변수이다. |
| ■ <code>bool = false</code> | // <code>bool</code> 은 불리언 변수이다. |
| ■ <code>empty = null</code> | // <code>empty</code> 는 <code>null</code> 값을 갖는 변수이다. |

❖ var를 이용한 변수 선언 방법

- `var javascript` // javascript 정수형 변수이다.
- `var pi = 3.14159` // pi는 실수형 변수이다.
- `var name = "임 성준"` // name은 문자열 변수이다.
- `var bool = false` // bool은 불리언 변수이다.
- `var empty = null` // empty는 null 값을 갖는 변수이다.

- 자료형은 결정되지 않고 선언만 된 경우이다.

`var area;`

- radius는 초기 값 10을 갖는 정수형, pi는 초기 값을 3.14159로 갖는 실수형 변수로서 사용된다.

`var radius=10, pi = 3.14159;`

- area 변수는 실수와 정수들의 곱한 결과가 저장되기 때문에 실수형 변수가 된다.

`area = pi * radius * radius`

❖ 자동으로 변수의 자료형 변환 : 묵시적인 형 변환

- `var test = 10` `// test는 처음에 정수형 변수이다.`
- `test = "문자열로 변환"` `// 이때는 문자열 변수로 변환된다.`
- `test = true` `// 이때는 불리언 변수로 변환된다.`

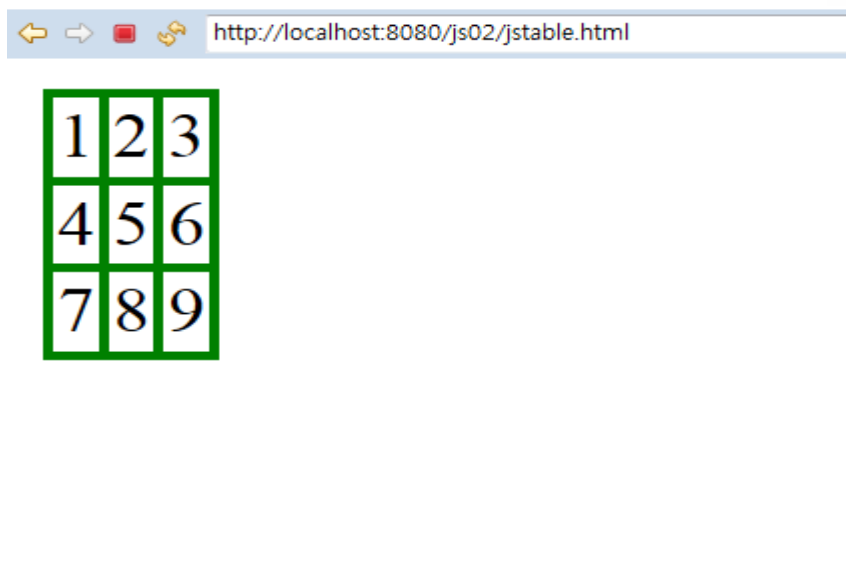
❖ document.write() 줄바꿈 출력하기

- ⑧ document.write() 명령으로 출력할 때 "("와 ")" 사이에 `<h3>`, `<hr>`, `<p>`, `
` 등 HTML 태그를 문자열 상수처럼 취급하여 HTML 태그의 기능을 활용하여 다양한 출력을 할 수 있다.
- * triangle.html



❖ document.write() 테이블 그리기

- ⑧ document.write() 명령으로 출력할 때 "("와 ")" 사이에 `<h3>`, `<hr>`, `<p>`, `
` 등 HTML 태그를 문자열 상수처럼 취급하여 HTML 태그의 기능을 활용하여 다양한 출력을 할 수 있다.
- table.html



❖ 1. 문자결합 출력

- txt1="아름다운"
- txt2="우리강산" 대입하고 두 문자열을 합쳐서 출력하시오
- txt3="은하철도"
- txt4=444
- txt5=555 대입하고 문자열과 숫자를 합쳐서 "은하철도 999"를 출력하시오

❖ 2. boolean 자료형 비교

- num1=30
- num2=20 대입하고
- num1>num2 이면 “참입니다” 아니면 “거짓입니다” 라고
- Alert 창을 띄우시오

❖ 3. boolean 자료형 비교

- 다음 코드를 설명하시오

```
var result=confirm("Do you like javascript?");  
//var theText=result ? "good" : "No-good";  
if(result) {  
    theText = "GOOD";  
} else {  
    theText = "NOT GOOD";  
}  
document.write(theText);
```

❖ 4. 출력하기

- 자신의 나이를 나타내는 변수를 선언하고 출력
- 자신의 키를 나타내는 변수를 선언하고 출력
- 자신의 이름을 나타내는 변수를 선언하고 출력
(`<table>` 또는 `<p>` 태그로 출력)

❖ 5. 다이아몬드 출력하기

- (writeln 또는 alert 함수 사용)
- *webdev_day09-ex2.html
- (\n을 활용하세요)

```
★
★ ★
★ ★ ★
★ ★ ★ ★
★ ★ ★
★ ★
★
```

Section 2-1.

변수 보충 자료

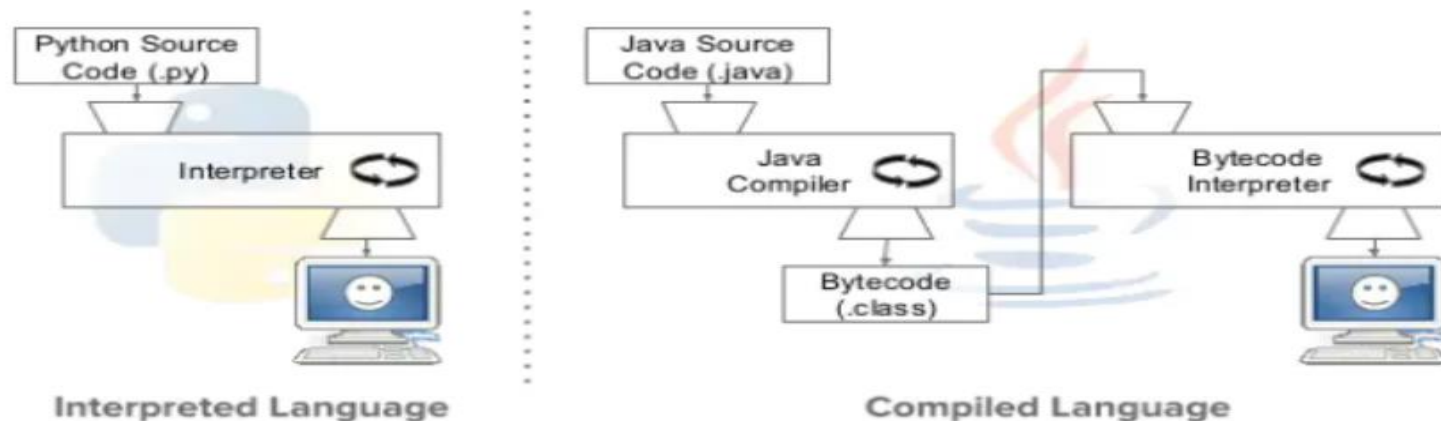
❖ HTML5 기술의 핵심



변수 보충 자료

❖ 자바스크립트 & 자바 비교

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



❖ 자바스크립트 역사

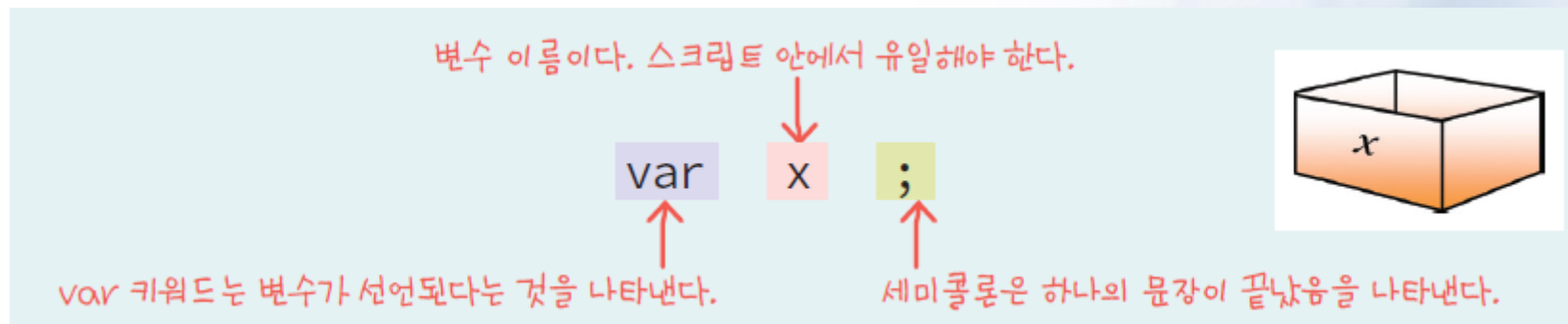
- 넷스케이프의 브렌던 아이크(Brendan Eich)가 개발
- 처음에는 라이브스크립트(LiveScript)
- 최신 버전은 자바스크립트 1.8.5
- ECMA(European Computer Manufacturer's Association)이 ECMAScript라는 이름으로 표준을 제정-> ECMA-262



변수 보충 자료

❖ 변수

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



변수 보충 자료

❖ 자바스크립트 문장 실행

- 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 보충 자료

❖ 변수 예제

```
<script>  
  var x;  
  x = "Hello World!";  
  alert(x);  
</script>
```


❖ 변수 예제

```
<script>
  var s;

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

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

❖ 연산자의 종류

- 단항연산자
- 이항연산자
 - 산술연산자
 - 비교연산자
 - 논리연산자
- 대입연산자
- 삼항연산자



❖ 연산자의 우선순위

자바스크립트 연산자 우선순위(비트 연산자 제외)

순위	기능	연산자					
1	괄호	()					
2	증감/논리 연산자 not	++	--	!			
3	산술 연산자 곱셈	*	/	%			
4	산술 연산자 덧셈	+	-				
5	비교 연산자 대소	<	<=	>	>=		
6	비교 연산자 같음	==	===	!=	!==		
7	논리 연산자 and	&&					
8	논리 연산자 or						
9	대입 연산자	=	+=	-=	*=	/=	%=

연산자와 제어문

❖ 단순 if 문

- if (조건) {
 명령문(들);
}
- ① 조건이 참인 경우 실행되는 명령문이 하나 이상 이
기 때문에 중괄호 {}가 사용된 경우
- if (number > 0) {
 sum = sum + number;
 document.write("양수이다"); }
- ② 조건이 참인 경우 실행되는 명령문이 하나인 경우
는 중괄호 {}을 생략할 수 있다.
- if (score >= 90) document.write("excellent");

연산자와 제어문

❖ 단순 if 문 예제

```
<script type="text/javascript">  
  
var bird_img = '<img src = "bird.gif" >';  
var tiger_img = '<img src = "tiger.gif" >';  
var select = prompt( "1과 2중 하나를 입력하라", "1");  
if (select == 1) document.write(bird_img);  
if ( select == 2) document.write(tiger_img);  
  
</script>
```

연산자와 제어문

❖ if-else 문

- if (조건) {
- 명령문(들)1;}
- else {
- 명령문(들)2;}

```
if ( x > 0 )  
    { x = x + 1;  
      z = x - 1; }  
else  
    { x = x - 1;  
      z = x - 1; }
```

연산자와 제어문

❖ else if 구를 사용한 다중 if 문

- 여러 개의 조건을 기술할 때 사용하는 if문으로서 조건을 모두 만족하지 못했을 경우에는 else 다음의 중괄호 {}안의 명령문(들)n을 실행시킨다.

■

```
if (조건1) {  
    명령문(들)1; }  
else if (조건2) {  
    명령문(들)2; }  
...  
else {  
    명령문(들)n; }
```

연산자와 제어문

❖ 중첩 if 문 * **mulif.html**

- if문 안에 다른 if 문이 들어간다

```
gender = prompt( "남자인 경우는 'm'을 여자인 경우는 f를 입력, "m");  
age = prompt( "나이를 입력하라", "");  
// 남/녀를 구분하는 외부 if문  
  
if (gender == "m")  
{  
    if ( age >= 18)  
        alert( "성인 남자");  
    else  
        alert( "남자 미성년자");  
}  
else if (gender == "f")  
{  
    if ( age >= 18)  
        alert( "성인 여자");  
    else  
        alert( "여자 미성년자");  
}  
else  
{  
    alert( "남녀 구분이 잘못 입력되었다");  
}
```


연산자와 제어문

❖ 연습 과제

❖ * **ex2.html**

- var score 에 prompt로 입력받아서
- 학점을 출력하되
- A+ B+ C+ D+ 도 학점을 출력하도록 추가하시오
- 예를 들어 90~94점은 A이고 95~100은 A+ 이다

❖ switch case 문

- 다중 조건문 으로서 변수 값에 의해 여러 가지 경우로 분기하는데 유용하게 사용될 수 있다
- switch (정수식 또는 문자열 식){
 case label1:
 명령문(들)1;
 break;
 case label2:
 명령문(들)2;
 break;
 default:
 명령문(들)n;
}

연산자와 제어문

❖ 반복의 원리



```
for(i=1; i<=10; i++) {  
    sum = sum + i  
}  
document.write(sum);
```

i

sum

i	sum
1	
2	
3	
4	
...	
10	

❖ for 문

- for문은 초기 값을 토대로 조건을 판단하여 증가 또는 감소하면서 반복 실행하는 명령문으로서 특히 배열 처리에서 자주 이용되는 유용한 반복문이다.

```
for (초기식; 조건식; 증가식; )  
{  
    반복 처리될 명령문(들);  
}
```

연산자와 제어문

❖ 예제 for

- 1에서 100까지의 합을 구하기

```
var sum =0;
for (number =1; number <= 100; number++) {}
    sum = sum + number;
}
document.write( " 1 + 2 + 3 + ... + 99 + 100 = " + sum);
```

연산자와 제어문

❖ 예제 **for * loop1.html**

- 1에서 100까지의 3의 배수의 합을 구하기

연산자와 제어문

❖ 중첩 for 문

- 외부 반복문과 내부 반복문간의 관계

[프로그램 ex4-16.html] 외부 반복문과 내부 반복문간의 관계

```
외부 FOR loop {
  내부 FOR loop {
    for (i= 1; i <= 3; i++)
      { document.write( " 외부 반복 제어 변수 i 값 -> " + i + "<br>");
    }
  }
}
```

연산자와 제어문

❖ while 문

- 조건이 참인 동안(while) 명령문들을 반복 실행하고, 그렇지 않으면 반복 구간을 빠져 나와 다음 문장으로 실행을 옮긴다.

```
초기값;  
while ( 조건식 )  
{  
    반복 처리될 명령문(들);  
    증감식;  
}
```


연산자와 제어문

❖ 예제 while

- 1에서 100까지 정수 중에 짝수인 값들의 합을 구하시오

```
var sum =0;
i = 1;
while(i<=100) {
    if(i%2==0){
        sum = sum + i;
    }
}
document.write( " 2 + 4 + ... + 98 + 100 = " + sum);
```

❖ 예제 **while * loop2.html**

- 1에서 100까지 정수중에 2의 배수이거나 3의 배수인 수의 합을 구하기

❖ do while문

- while문과 동일하나 차이점은 먼저 명령문을 수행한 후 반복 구간 마지막에서 조건을 검사하여 반복 여부를 결정한다. 즉, 반복 구간은 최소한 한번 실행된다.

초기값;

Do

{

반복 처리될 명령문(들);

증감값;

}

while (조건);

연산자와 제어문

❖ 예제 **do while**

- 1에서 10까지 정수들의 곱을 구하시오

```
var mux = 1;
i = 1;
do {
    mux = mux * i;
}
(i<=100) {
document.write( " 1 * 2 * ... + 9 * 10 = " + mux);
```

❖ 예제 **do while * loop3.html**

- Math.random() 함수를 사용하여 1-100사이의 랜덤한 수를 10번 출력하시오
(단 실수를 정수로 바꾸는 함수는 parseInt(n) 사용)

연산자와 제어문

❖ break문

- 자신이 포함된 하나의 반복문 또는 switch문을 빠져 나온다.
- 주로 if문과 함께 사용해서 특정 조건을 만족하면 반복문을 벗어나게 한다

```
var sum = 0;
var i = 1;

while(true) {

    if(sum>=100) {
        break;
    }

    sum = sum + i;

    document.write( i + "일때 : " + sum);
    document.write("<br>");
}
```

i	sum
0	0
1	1
2	3
3	6
...	...
13	91
14	105

연산자와 제어문

❖ continue 문

- 자신이 포함된 반복문의 끝으로 이동한다.(다음 반복으로 넘어간다.)
- continue문 이후의 문장들은 수행되지 않는다.

```
var i = 1;

for(i=1; i<=10; i++) {

    if(i%3==0) {
        continue;
    }

    document.write(i);
    document.write("<br>");
}
```

[실행결과]

1
2
4
5
7
8
10

❖ 연습3-1 *webdev_day09-ex31.html

- 구구단 프로그램: 단 수를 입력하면 (2~9단)을 입력하면 해당 구구단을 출력해준다

❖ 함수의 개념과 장점

- 함수: 해결해야 할 프로그램을 기능 중심의 여러 개의 작은 부분 문제들로 분해할 때 각 부분 문제와 관계된 여러 문장들의 집단을 함수(function)라 부른다.
- 함수를 사용할 때의 장점
 - ① 복잡한 프로그램을 함수별로 나누어 여러 사람이 공동 작성할 수 있다.
 - ② 반복 기술하지 않고 필요할 때마다 공동 이용할 수 있다.
 - ③ 함수별로 손쉽게 오류를 찾을 수 있고 수정할 수 있다.
 - ④ 프로그램의 설계가 단순해서 프로그램을 이해하기 쉽고, 유지보수하기가 쉽다.

❖ 내장 함수 : alert() 함수

- ① alert() 함수는 내장 함수로서 경고나 인사말 등과 같은 정보가 있는 메시지를 출력하는 경우에 사용.



형 식	alert (메시지)
용 도	사용자의 요구를 받을 필요가 없는 정보가 담긴 메지만을 출력시켜 주는 대화상자이다.
인 수	메시지 : 대화상자에 나타나는 메시지이다.
	확인(ok) 버튼을 누를 경우 alert() 함수 명령이 종료한다.

❖ 입력 대화상자 : **prompt()** 함수

- ① **prompt()** 내장 함수는 사용자로부터 자료를 입력 받는데 사용하는 텍스트 박스로 구성되는 대화상자로서 사용자에게 문자열 자료를 입력받는데 사용.



❖ confirm() 함수

- ① confirm() 내장 함수는 사용자가에게 질문 메시지를 보여 주고 사용자가 선택한 확인 또는 취소 버튼에 따라 프로그램을 진행해 나갈 때 사용.
- ② 사용자가 확인(ok) 버튼을 누를 경우에 ok_no 변수에 true 값이 할당되고, 만일 취소(cancel) 버튼을 누를 경우에는 false 값이 할당된다.
- `ok_no = confirm("번호가 잘못되었다.!! 계속하려면 '확인' 종료하려면 '취소'");`



❖ eval() 함수

- 문자열로 입력된 수식을 계산하여 주는 편리한 함수.
- ① `document.write(eval(10 * 5));` // 50이 출력된다
- ② `eval()` 함수는 `prompt()` 함수와 같은 입력 대화상자를 통해 입력받은 수식을 처리할 때 아주 편리하다.

형식	<code>eval(문자열 수식)</code>
용도	문자열 수식을 인수로 기술하면 문자열을 수식으로 변환한 후에 수식 계산을 수행하여 준다.
인수	문자열 수식: 수식 계산에 사용할 문자열을 상수 또는 변수로 기술한다.

❖ 문자열을 정수로 변환하는 함수 : **parseInt()**

- ① 문자열을 정수로 변환하는 내장 함수로서 입력된 문자열을 10진수, 2진수, 8진수, 16진수의 정수로 변환한다.
- 주의 할 점은 어떤 진수로 변환이 되어도 결과는 10진수 정수로 출력된다

❖ 문자열을 정수로 변환하는 함수 : **parseInt()**

- 문자열 "120"이 10진수로 변환되어 10진수 120으로 출력된다.
- `document.write(parseInt("120", 10));`
- 114가 출력된다.
- `document.write(parseInt("114tel03ab"), 10);`
- 3으로 출력된다.
- `document.write(parseInt("3.14159");`

형식	<code>parseInt(문자열 [, 진수])</code>
용도	문자열을 정수로 변환하는 내장 함수로 입력된 문자열을 10진수, 2진수, 8진수, 16진수 정수로 변환한다.
인수	문자열 : 정수로 변환할 문자열을 적는다. 진수 : 변환할 진수(10, 2, 8, 16진수)로서 10진수로 변환할 때는 생략해도 된다.

❖ parseFloat()

- ① 문자열을 실수로 변환하는 내장 함수.
- // 23.44가 출력된다.
- document.write(parseFloat("23.44"));
- // 3.14159가 출력된다.
- document.write(parseFloat("3.14159pi3ab"));

형식	parseFloat(문자열)
용도	문자열을 실수로 변환하는 내장 함수이다.
인수	문자열 : 실수로 변환할 문자열을 적는다.

❖ parseInt()/parseFloat() 함수를 사용 이유

- prompt() 함수에 입력된 자료는 문자열형 자료이기 때문에 prompt() 대화상자를 사용하여 입력받은 자료를 수치로 사용할 때는 문자열을 정수/실수로 변환하는 parseInt()/parseFloat() 함수를 사용해야 한다
- [예] `compute_string = prompt("계산할 수식을 입력하십시오", "");`
- prompt() 함수에서 입력받은 문자열 자료를 정수로 변환한다.
- `int_num = parseInt(compute_string);`
- prompt() 함수에서 입력받은 문자열 자료를 실수로 변환한다.
- `float_num = parseFloat(compute_string);`

❖ 섭씨온도를 화씨로 변환하기

❖ * **transform.html**

- prompt() 함수를 사용하여 섭씨온도 입력
- 변환식 $F = C * 9.0 / 5.0 + 32$
- 섭씨와 화씨를 모두 출력

❖ isNaN()

- isNaN(= is Not a Number) 함수는 문자인지 숫자인지를 구별하는 함수로서 테스트 값이 문자이면 "true"를 숫자이면 "false"로 결과를 나타낸다.
- "100"은 false로 출력되어 숫자임을 알 수 있다.
- num = isNaN("100");
- document.write (num); // false로 출력된다.
- "string"은 true가 출력되어 문자임을 알 수 있다.
- document.write (isNaN("string"));
- "10kkk"와 같이 숫자와 문자가 혼합되어 있으면 true가 출력되어 문자로 판단한다.
- document.write (isNaN("10kkk"));

❖ setTimeout()

- ① 일정 시간을 설정한 다음에 설정한 시간이 경과하고 나면 자동으로 특정 명령을 실행시키는 함수이다.

형식	setTimeout(명령, 시간)
용도	타이머(timmer) 기능처럼 일정 시간을 설정한 다음에 설정한 시간이 경과하고 나면 특정 명령을 실행시키는 함수이다.
인수	<ul style="list-style-type: none">· 명령 : 일정 시간이 경과한 후에 실행할 명령으로서 보통 기능을 갖는 함수를 적는다.주의할 점은 명령 인수는 따옴표(")로 양쪽을 막아서 기술해야 한다.· 시간 : 일정한 대기 시간을 기술하는데 시간 단위는 micro second이다. (ms, 1/1000초) <p>예 : <code>setTimeout("location.href = 'http://www.hanmir.com' ", 5000);</code></p>
반환값	setTimeout() 함수에서 설정한 시간을 해제할 때 사용하는 식별자로서 clearTimeout() 함수에서 사용된다.

❖ setTimeout() 예제

❖ * timeout.html

- 5초후 네이버 페이지로 이동하도록 구현

```
<script type="text/javascript">  
    setTimeout("location.href = 'http://www.naver.com' ", 5000);  
</SCRIPT>
```

❖ setInterval()

- ① setInterval() 함수는 설정된 시간 간격마다 setInterval() 함수에서 정의한 명령이 반복 실행되는 함수이다.

형식	setInterval(명령, 시간)
용도	한번 시간을 설정하여 두면 설정된 시간 간격마다 setInterval() 함수에서 정의한 명령이 반복 실행된다.
인수	<ul style="list-style-type: none">· 명령 : 일정 시간이 경과한 후에 반복 실행할 명령으로서 보통 기능을 갖는 함수를 적는다. 주의할 점은 명령 인수는 따옴표(")로 양쪽을 막아서 기술해야 한다.· 시간 : 반복 실행할 시간 간격을 기술하는데 시간 단위는 micro second이다. (ms, 1/1000초) <p>예 : setInterval("window.status=new Date()", 1000);</p>
반환값	setInterval(명령, 시간) 함수에서 설정한 시간을 해제할 때 사용하는 식별자로서 clearTimeout() 함수에서 사용된다. 사용 예는 3)절의 clearTimeout() 함수에 기술되어 있다.

❖ `clearTimeout()`, `clearInterval()`

- 반대로 멈추는 함수

함수 이름	설명
<code>setTimeout(함수, 시간)</code>	일정 시간 후 함수 실행
<code>setInterval(함수, 시간)</code>	일정 시간 간격으로 함수 반복 실행
<code>clearTimeout(id)</code>	실행되고 있는 timeout 을 중지
<code>clearInterval(id)</code>	실행되고 있는 interval 을 중지

❖ setInterval() * interval.html

```
10  <body>
11  <p>페이지 랜덤숫자를 2초 간격으로 발생한다</p>
12  <p><input type="button" value="정지" onclick="stop();"></p>
13  <div id="demo"></div>
14
15  <script>
16      var idnum = setInterval(RandomNumber ,2000);
17
18      function RandomNumber() {
19          document.getElementById("demo").innerHTML+="<br>" +Math.random();
20      }
21
22      function stop() {
23          clearInterval(idnum);
24      }
25  </script>
26  </body>
```


❖ 사용자 함수를 정의하는 방법 : **function** 문

- 사용자의 용도와 목적에 맞게 독립된 기능을 수행하는 함수는 다음과 같은 형식으로 정의한다.

```
function 함수명([전달받는 인수 리스트]) {  
    명령문(들);  
    [return 식;]  
}
```

❖ 사용자 함수를 정의하는 방법 : function 문

- 전달받는 인수 리스트(argument list)는 함수가 특별한 기능을 수행하는데 필요한 초기 자료로서 여러 개의 인수가 필요할 때는 콤마(,)를 통해 구분하면 되고, 인수가 필요 없으면 생략하면 된다
- 중괄호 { } 내의 모든 명령이 실행되어 "}"를 만나면 함수는 종료되면서 호출한 곳으로 복귀하게 된다.
- 만일 함수를 호출한 곳으로 함수의 실행 결과를 반환(return)하고 싶으면 이 경우에 [return 식]을 사용한다.
- return 명령은 함수의 처리를 종료할 때도 사용한다

❖ 함수를 호출하는 방법

- [변수=] 함수명([전달할 인수 리스트]);
- sample();
- result = sample();
- intNum = parseFloat("23.44")
- [변수=]는 호출한 함수의 실행 결과를 특정 변수에 반환(return)받고 싶을 때 사용한다. 실행 결과를 변수에 반환(return)받고 싶지 않으면 생략한다.

❖ 반환 값이 없고 일만 하는 함수

- 기억할 변수 없이 함수명만 호출
- printText();

```
<script>
  function printText() {
    for(i=1; i<=5; i++) {
      document.write("hello"+"<br>");
    }
  }
  function printText2(input) {
    for(i=1; i<=5; i++) {
      document.write(input+"<br>");
    }
  }

  printText();
  printText2("hi javascript");
</script>
```

함수

- ❖ 반환 인수 전달이 필요 없는 함수 예
 - [변수=] 함수명([전달할 인수 리스트]);

```
function compute() {  
    var compute_string;  
    compute_string = prompt("계산할 수식을 입력하시요", "");  
  
    result = eval(compute_string);  
    document.write("<h4>" + compute_string + " = " + result + "<br>");  
}
```

❖ 반환 값이 있는 함수

- 기억할 변수에 반환값을 대입함
- `y=sum();`

```
function sum() {  
    rst=0;  
    for(i=1; i<=10; i++)    {  
        rst = rst+i;  
    }  
    return rst;  
}  
function sum2(num) {  
    rst=0;  
    for(i=1; i<=num; i++)    {  
        rst = rst+i;  
    }  
    return rst;  
}  
y=sum(); document.write("y="+y+"<br>");  
y2=sum2(100); document.write("y2="+y2);
```

❖ 지역변수

- ① 함수 내에 var를 사용하여 변수를 선언하면 그 변수는 함수 내에서만 유효한 지역 변수가 된다.

```
function input_name() {  
  // var로 선언한 name 변수는 함수 input_name()에  
  // 서만 유효한 지역 변수이다.  
  var name;  
  name = prompt("당신의 이름은 ?", "");  
}
```

❖ 전역변수

- ① 함수 내에서 var로 선언한 변수를 제외한 모든 변수는 프로그램 어디에서나 유효한 전역 변수가 된다.
- ② input_name() 함수에서 사용하는 name 변수는 var에 의해 선언하지 않았기 때문에 전역 변수이며, var를 사용하여 선언한 tel 변수 역시 함수 밖에서 선언된 변수이기 때문에 전역 변수이다.

```
function input_name() {  
    name = "임성준";  
}  
var tel;
```


❖ return 명령을 사용하는 경우

- ① 함수를 호출한 곳으로 함수의 실행 결과를 반환 (return)하고 싶을 때 [return 식] 형태로 사용한다.
- ② 함수의 처리를 종료할 때도 사용한다. 즉, 함수 내에 return 명령을 만나면 무조건 함수는 종료되면서 호출한 곳으로 복귀하게 된다.

연습문제

❖ 연습1_1 * exam1_1.html

- 주사위의 눈의 값을 출력하는 dice() 함수를 작성하고 1~6 사이의 랜덤값을 출력하시오

❖ 연습1_2 * exam1_2.html

- 자바스크립트 함수를 활용하여 가위 바위 보 게임을 작성하시오
- 컴퓨터가 랜덤하게 가위 바위 보를 생성함
- 사용자는 가위 바위 보 중 하나를 입력할 수 있음
- 컴퓨터와 사용자의 입력값을 비교하여 승패를 출력함