

제7강

자바스크립트 (1)

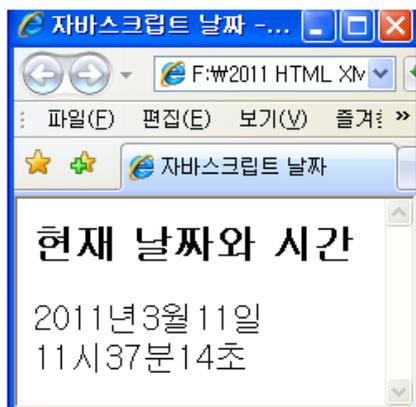
1. 자바스크립트의 개요**1) 자바스크립트란?**

- 자바스크립트는 Sun Microsystems사와 Netscape사가 공동으로 제작하여 보급한 언어이다.
- 자바스크립트 언어를 사용함으로써 사용자와 웹 페이지간의 상호작용이 가능하게 된다. 즉 동적인 웹 페이지 작성이 가능
- 자바스크립트의 예)

```

<HTML>
<HEAD>
  <TITLE> 자바스크립트 날짜 </TITLE>
  <SCRIPT LANGUAGE = "JavaScript">
    <!--
      document.write("<H3> 현재 날짜와 시간 </H3>");
      today = new Date();
      document.write(today.getFullYear(), "년",
        today.getMonth()+1, "월",
        today.getDate(), "일 <BR>");
      document.write(today.getHours(), "시",
        today.getMinutes()+1, "분",
        today.getSeconds(), "초");
    //-->
  </SCRIPT>
</HEAD>
<BODY></BODY>
</HTML>

```



->새로고침 버튼을 클릭할때마다 시간이 바뀌어 출력된다. (동적 웹페이지 제공)

- 자바와 자바스크립트는 같은 개발사에서 개발되어 비슷하면서도 다른 언어이다. 구문적으로는 두 언어 모두 C언어의 기본 구문에 바탕을 두고 있지만, 두 언어 간에 직접 관

련성은 없다. 자바는 범용 언어적인 성격을 가지지만, 자바스크립트는 웹 프로그래밍을 위한 스크립트 언어이다.

특성	자바	자바스크립트
작성 방법	별도의 파일 작성	HTML 내에 직접 삽입
실행 방식	서버에서 컴파일된 후 클라이언트에서 실행	클라이언트에서 직접 해석되고 실행
변수 선언	자료형을 반드시 선언	자료형 선언이 불필요
객체지향	객체지향 언어	완전한 객체지향 언어가 아님
보안	소스를 볼 수 없으므로 보안 유지 가능	메뉴를 통해 소스를 볼 수 있어 보안성이 없음

2) 자바스크립트의 특징

① 스크립트 언어

- 어떤 프로그램에 의해 번역/수행되는 명령어
- 프로그램에 익숙하지 못한 사용자들에게 간단하게 필요한 기능을 제공

② 객체기반 언어

- 자바와는 달리 제한적으로 객체지향 개념을 도입

③ HTML 내에 삽입하여 사용

- <SCRIPT> 태그를 통해 HTML 내에 삽입해서 웹 브라우저에 의해 실행

④ 개발 및 결과 확인이 용이

- 간단한 편집기 사용 가능, 컴파일 단계가 없음

3) 자바스크립트의 장점

① 빠른 개발이 가능

- 규모가 작고 간단한 기능의 프로그램 작성에 적합
- 간단한 편집기 사용, 컴파일 과정의 불필요

② 배우기 쉽다

- 변수의 명시적 선언이 불필요
- 클래스/상속과 같은 복잡한 개념이 없음

③ 환경에 독립적

- HTML 내 삽입하므로 운영체제, 하드웨어에 독립

4) 자바스크립트의 단점

① 취약한 보안

- HTML 내 삽입하여 사용하므로 브라우저에 소스 보기가 가능

② 제공하는 메소드의 부족

- 자바스크립트는 브라우저의 기능과 밀접한 관계가 있어 자바스크립트 자체에서 제공하는 메소드는 브라우저에 제한적

5) 자바스크립트의 기본구조

- 자바스크립트는 <SCRIPT> 태그와 </SCRIPT> 태그 사이에 정의 된다.
- <SCRIPT> 태그는 HTML 문서의 <HEAD> 부분 또는 <BODY> 부분에 삽입하여 사용한다.
- <SCRIPT> 태그의 LANGUAGE 속성의 값으로 일반적으로 'JavaScript'를 입력한다.

```
<HTML>
<HEAD>
  <TITLE> 자바스크립트 기본 구조 </TITLE>
  <SCRIPT LANGUAGE = "JavaScript">
    <!--
      document.write("자바스크립트 기본구조");
    //-->
  </SCRIPT>
</HEAD>
<BODY>
  BODY 태그 내에서도 사용 가능
</BODY>
</HTML>
```

- 외부 파일(*.js)을 사용하는 경우

```
<HTML>
<HEAD>
  <TITLE> 자바스크립트 기본 구조 </TITLE>
  <SCRIPT SRC = "JSExample.js">
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

JSExample.js
document.write("자바스크립트 기본구조");
document.write("외부파일 사용 .js");
```

2. 자바스크립트의 기초

1) 자바스크립트의 변수

① 변수란?

- 컴퓨터 기억장소 셀에 대한 추상화
- 어떤 값을 저장하기 위한 기억장소의 상대적 주소에 대한 이름
- 프로그램에서 다루는 데이터의 저장 장소

② 변수 이름

- 영문자, 숫자, '_'의 조합으로 구성
- 한글, 특수문자는 사용 불가
- 첫 글자는 반드시 영문자로 시작
- 대소문자의 구별
- 예약어 사용 불가

③ 권장 사항

- 짧고 간단하며, 의미 있는 이름을 사용하면 판독력과 작성력 향상에 도움

④ 자바스크립트의 예약어

abstract	case	continue	extends	for
import	long	private	static	throw
var	boolean	catch	default	false
function	in	native	protected	super
throw	void	break	char	do
final	goto	instanceof	new	public
switch	transient	while	byte	class
double	finally	if	int	null
return	synchronized	true	with	case
const	else	float	implements	interface
package	short	this	try	

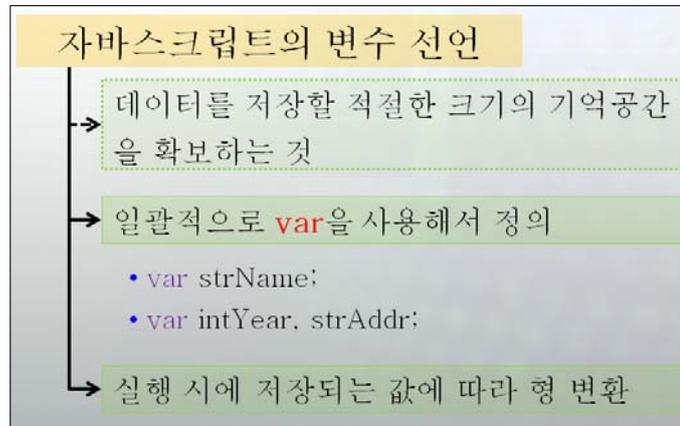
⑤ 식별자 지정의 예 (빨간 상자로 설명이 추가된 것은 잘못 지정된 식별자이다)

while	→ 예약어 사용
strName	
strName123	
학생이름	→ 한글 사용
str-Name	→ '-' 문자 사용
str_Name	
_strName	→ 첫 글자가 영문자가 아님
sName#123	→ '#' 문자 사용
123sName	→ 첫 글자가 숫자로 시작
Name.no	→ '.' 문자 사용

⑥ 변수 선언

- 변수 선언이란 데이터를 저장할 적절한 크기의 기억공간을 확보하는 것. 데이터의 종류 (자료형)에 따라 필요한 공간의 양이 달라진다.

- 특정 자료형을 지정하지 않고 일괄적으로 **var**을 사용해서 정의한다.

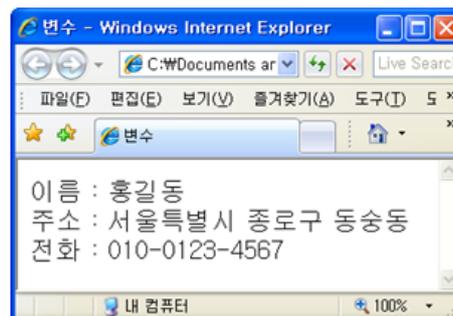


⑦ 변수 사용의 예

```

<HTML>
<HEAD>
  <TITLE> 변수 </TITLE>
  <SCRIPT LANGUAGE = "JavaScript">
    <!--
      var strName, strAddress, strPhone;
      strName = "홍길동";
      strAddress = "서울특별시 종로구 동숭동";
      strPhone = "010-0123-4567";
      document.write("이름 : " + strName + "<BR>");
      document.write("주소 : " + strAddress + "<BR>");
      document.write("전화 : " + strPhone + "<BR>");
    //-->
  </SCRIPT>
</HEAD>
<BODY></BODY>
</HTML>

```



2) 자바스크립트의 자료형

- 자바스크립트는 특별한 자료형의 선언이 필요 없이 **var** 자료형을 사용한다.
- 저장되는 값에 따라 자동으로 형변환된다.

① 기본적 자료형

기본적인 자료형	
정수	<ul style="list-style-type: none"> • 87 • 0127, 0x57
부동소수점	<ul style="list-style-type: none"> • 12.34 -0.1234 • 2e6
논리형(Boolean)	<ul style="list-style-type: none"> • true, false
Null	<ul style="list-style-type: none"> • 값 없음
문자열	<ul style="list-style-type: none"> • “ ” • “17” vs 17

② 특수문자

특수 문자	설명
\n	다음 줄로 이동
\t	Tab 키 입력
\b	BackSpace 키 입력
\r	Return 키 입력
\\	\ 기호 삽입
'\'	' 기호 삽입
\"	“ 기호 삽입

③ 자료형 변환

```

var myVar;

document.write("출력값 : " + myVar + "<BR>");

myVar = "홍길동";

document.write("문자열 : " + "\"" + myVar + "\"" + "<BR>");

myVar = 25;

document.write("정수 : " + "\"" + myVar + "\"" + "<BR>");

```

```

출력값 : undefined
문자열 : "홍길동"
정수 : '25'

```

- 처음에는 myVar 변수에 문자열 '홍길동'을 저장한다. 똑같은 변수에 정수 25를 저장하면 자동으로 정수로 형변환되어 저장된다.

3) 자바스크립트 연산자

- 연산자란 수학적, 논리적 연산을 수행하는 기호를 나타낸다.

① 산술 연산자

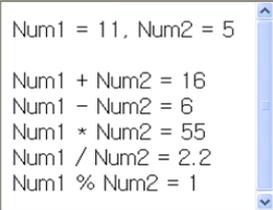
- 산술 연산을 위한 연산자이다.

- 산술 연산자의 종류

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
%	나머지

-

```
var intNum1, intNum2;
intNum1 = 11; intNum2 = 5;
document.write("Num1 = " + intNum1
  + ", Num2 = " + intNum2 + "<BR><BR>");
document.write("Num1 + Num2 = "
  + (intNum1 + intNum2) + "<BR>");
document.write("Num1 - Num2 = "
  + (intNum1 - intNum2) + "<BR>");
document.write("Num1 * Num2 = "
  + (intNum1 * intNum2) + "<BR>");
document.write("Num1 / Num2 = "
  + (intNum1 / intNum2) + "<BR>");
document.write("Num1 % Num2 = "
  + (intNum1 % intNum2) + "<BR>");
```



② 대입 연산자

- '=' 즉 우변의 내용을 좌변의 변수에 대입할 경우 사용하는 연산자이다.
- 연산 후 대입 연산자 : 좌변 변수의 원래 값에 일반적인 산술 연산 값을 저장할 경우 사용하는 연산자이다. 예) 'a = a + 10' 과 'a += 10'은 같은 결과다.
- 대입 연산자의 종류

연산자	설명
+=	덧셈 후 대입
-=	뺄셈 후 대입
*=	곱셈 후 대입
/=	나눗셈 후 대입
%=	나머지 연산 후 대입

- 예)

```

var a, b, c, d, e;

a = b = c = d = e = 14;

a += 5;
b -= 5;
c *= 5;
d /= 5;
e %= 5;

document.write("a += " + a + "<BR>");
document.write("b -= " + b + "<BR>");
document.write("c *= " + c + "<BR>");
document.write("d /= " + d + "<BR>");
document.write("e %= " + e + "<BR>");

```

```

a += 19
b -= 9
c *= 70
d /= 2.8
e %= 4

```

③ 증감 연산자

- 증가, 감소 연산자 : 변수의 앞이나 뒤에 붙여 사용되며 변수의 값을 1씩 증가시키거나 감소시킬 때 사용한다.
- 증가, 감소 연산자의 종류

연산자	설명
++	변수의 값 1씩 증가
--	변수의 값 1씩 감소

- '++변수' 와 '변수++'의 차이점

```

Y = ++X; // X에 1을 먼저 더한 후 Y에 대입;
        연산 후 최종값 : X = 11, Y = 11
Y = X++; // X의 값을 Y에 먼저 대입하고, X의 값을 1증가;
        연산 후 최종값 : X = 11, Y = 10

```

- 예)

```

var a, b;
a=b=9;
document.write("++ a : " + ++a + "<BR>");
document.write("a++ : " + a++ + "<BR>");
document.write("a : " + a + "<BR>");
document.write("b-- : " + b-- + "<BR>");
document.write("--b : " + --b + "<BR>");
document.write("b : " + b + "<BR>");

```

```

++a : 10
a++ : 10
a : 11
b-- : 9
--b : 7
b : 7

```

④ 비교 연산자

- 비교 연산자는 두 변수 사이의 관계를 비교하기 위하여 사용되는 연산자이다.
- 연산 결과의 참과 거짓에 따라 Boolean 값인 true, false를 반환한다.
- 비교 연산자의 종류

연산자	설명
==	같다
<	보다 작다
<=	보다 작거나 같다
>	보다 크다
>=	보다 크거나 같다
!=	같지 않다

- 예)

```
var a, b;
a=11; b=5;
document.write("a = " +a+ ", b = "+b+ "<BR><BR>");
document.write("a == b : " + (a == b) + "<BR>");
document.write("a < b : " + (a < b) + "<BR>");
document.write("a <= b : " + (a <= b) + "<BR>");
document.write("a > b : " + (a > b) + "<BR>");
document.write("a >= b : " + (a >= b) + "<BR>");
document.write("a != b : " + (a != b) + "<BR>");
```

```
a = 11, b = 5
a == b : false
a < b : false
a <= b : false
a > b : true
a >= b : true
a != b : true
```

⑤ 논리 연산자와 비교 연산자

- 논리 연산자란 조건의 참과 거짓을 판단하기 위한 연산자이다.
- 두 값을 비교하여 해당 연산자에 따라 참과 거짓을 판별하여 값으로는 true나 false를 반환한다.
- 논리 연산자의 종류

연산자	설명
&&	두 조건이 모두 참이면 참이 됨 (AND)
	두 조건중 하나만 참이면 참이 됨 (OR)
!	조건 값을 반대로 만듦 (NOT)

- 비트 연산자는 각각의 비트 별로 연산을 하는 것으로 2진수 연산에 사용한다.
- 비트 연산자는 참과 거짓에 따라 1, 0 값을 반환한다.
- 비트 연산자의 종류

연산자	설명
&	두 조건이 모두 참이면 참이 됨 (AND)
	두 조건중 하나만 참이면 참이 됨 (OR)
^	두 조건의 값이 같으면 참이 됨 (XOR)

- 예)

```

var a;
a=5;
document.write("a > 1 && a < 5 : "+(a>1 && a<5)+"<BR>");
document.write("a > 1 || a < 5 : " +(a>1 || a<5)+"<BR>");
document.write("! (a > 1) : " +!(a > 1)+"<BR>");

document.write("a > 1 & a < 5 : " +(a>1 & a<5)+"<BR>");
document.write("a > 1 | a < 5 : " +(a>1 | a<5)+"<BR>");
document.write("a > 1 ^ a < 5 : " +(a>1 ^ a<5)+"<BR>");

```

```

a > 1 && a < 5 : false
a > 1 || a < 5 : true
! a > 1 : false
a > 1 & a < 5 : 0
a > 1 | a < 5 : 1
a > 1 ^ a < 5 : 1

```

- 논리/비교연산자 진리표

A	B	A && B A & B	A B A B	A ^ B
true 1	true 1	true 1	true 1	0
true 1	false 0	false 0	true 1	1
false 0	true 1	false 0	true 1	1
false 0	false 0	false 0	false 0	0
A		!A		
true		false		
false		true		

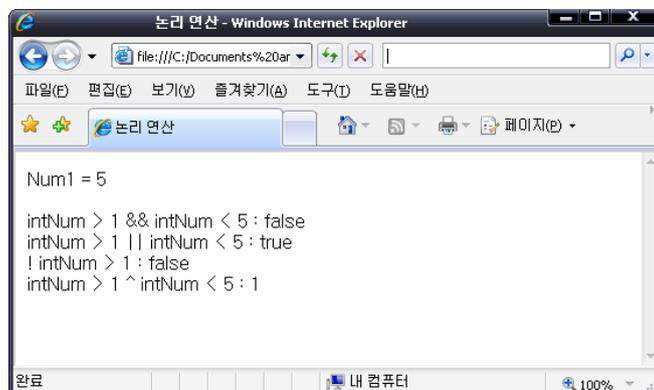
- 예)

```

<SCRIPT LANGUAGE="JavaScript">
<!--
  var intNum;
  intNum = 5;

  document.write("Num1 = " + intNum + "<BR><BR>");
  document.write("intNum > 1 && intNum < 5 : " +
    (intNum > 1 && intNum < 5 ) + "<BR>");
  document.write("intNum > 1 || intNum < 5 : " +
    (intNum > 1 || intNum < 5) + "<BR>");
  document.write("! intNum > 1 : " +
    (! (intNum > 1)) + "<BR>");
  document.write("intNum > 1 & intNum < 5 : " +
    (intNum > 1 & intNum < 5 ) + "<BR>");
  document.write("intNum > 1 | intNum < 5 : " +
    (intNum > 1 | intNum < 5) + "<BR>");
  document.write("intNum > 1 ^ intNum < 5 : " +
    (intNum > 1 ^ intNum < 5) + "<BR>");
//-->
</SCRIPT>

```



⑥ 이항 연산자

- 이항 연산자 : 어떠한 조건을 판단하여 두 개의 값 중 하나의 값을 선택하는 것이다.

- 형식 **(조건) ? 수식1 : 수식2**

- 예) a가 5보다 크면 앞에 a-5(=7)가 출력되고, 작으면 a+5(=17)를 출력한다.

```

var a;
a = 12;
document.write( (a > 5) ? a-5 : a+5 );

```

⑦ 연결 연산자

- 연결 연산자 : 문자열을 합칠 때 필요한 연산자로 '+'를 사용한다.
- 숫자와 숫자에 '+' 연산자를 사용하면 산술 연산자로 계산이 되지만 문자열과 문자열, 문자열과 숫자 사이에 '+' 연산자를 사용하면 연결연산자가 된다.

```
intX = 10;
strX = "연결";
strY = "연산자";

intX + strX; // 값 : 10연결

strX + strY; // 값 : 연결연산자
```

⑧ 연산자 우선순위

- 연산자의 우선순위는 아래 표의 위가 우선순위가 가장 높고 아래로 내려오는 순서 이다. (C언어와 같다.)

괄호	()
증감, 부정	++, --, !
산술	*, /, %
	+, -
비교	<, <=, >, >=
	==, !=
비트	&
	^
논리	&&
이항	?:
대입	+=, -=, *=, /=, %=

연습 문제

1. 자바와 자바스크립트의 차이점에 대해 설명하시오.

특성	자바	자바스크립트
작성 방법	별도의 파일 작성	HTML 내에 직접 삽입
실행 방식	서버에서 컴파일 된 후 클라이언트에서 수행됨	클라이언트에서 직접 해석되고 실행됨
변수 선언	변수의 자료형을 반드시 선언해야 함	변수의 자료형을 선언할 필요가 없음
객체지향	모든 객체를 정의하여 사용할 수 있는 객체지향 언어	클래스 선언이나 상속등의 개념이 없으므로 완전한 객체지향 언어가 아님
보안	소스를 볼 수 없으므로 보안을 유지할 수 있음	소스보기 메뉴를 통하여 소스를 볼 수 있으므로 보안성을 가지지 못함

2. 자바스크립트의 기본구조에 대해 설명하시오.

자바스크립트는 <SCRIPT> 태그와 </SCRIPT> 태그 사이에 정의 된다. <SCRIPT> 태그는 HTML 문서의 <HEAD> 부분 또는 <BODY> 부분에 삽입하여 사용한다. 그리고 자바스크립트를 삽입할 때에는 일반적으로 <SCRIPT> 태그의 LANGUAGE 속성의 값으로 'JavaScript'로 값을 입력한다.

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    자바스크립트 코드
//-->
</SCRIPT>
```

3. 자바스크립트의 변수를 선언할 때 식별자의 조건에 대해 설명하시오.

- 영문자와 숫자, '_' 로 이루어져 있다.
- 특수 문자나 메타 문자, 한글은 사용할 수 없다.
- 첫 자로는 영문자만 사용할 수 있다.
- 대소문자의 구별이 있다.
- 예약어(자바스크립트에서 미리 정의해 놓은 용어)는 식별자로 사용할 수 없다.

확인 문제

※ 다음의 빈 칸에 알맞은 단어를 작성하시오.

- ()란 어떤 프로그램에 의해 번역 되거나 수행되는 명령어들을 말한다. 즉 웹 문서가 요청 될 때 프로그램 명령어 형식으로 코드를 만들어 이것을 합쳐 보여주는 것을 의미한다.
- 자바스크립트를 외부 파일로 따로 만들어 사용할 경우에는 확장자를 ()로 지정해야 한다.
- ()이란 12.34 또는 -12.34와 같이 소수가 포함되어 있는 10진수를 말한다. 부동소수점이란 의미는 소수점이 움직인다는 뜻이다.
- 자바스크립트에 “”나 ‘’를 직접 입력하고 싶을 때는 해당 문자 앞에 ()를 붙이면 입력이 가능하다.
- 논리 연산자로 두 조건중 하나만 참이 되면 참이되는 연산자 기호는 ()이다.

※ 다음 문장의 참 거짓 여부를 판별하시오.

- 자바스크립트는 보안에 취약하다. ()
- 자바스크립트는 특별한 자료형의 선언이 필요 없이 저장되는 값에 따라 자동으로 형변환된다. ()
- 두 조건중 하나만 참이면 참이 되는 비트 연산자의 기호는 & 이다. ()
- 문자열을 합칠 때는 연결 연산자로 () 문자를 사용한다. ()
- 연산자 우선순위에서 최 우선순위에 해당하는 연산자는 ‘()’와 ‘[]’이다.()

※ 다음의 문제에 해당되는 보기를 고르시오.

- 다음 중 자바와 자바스크립트의 차이점으로 잘못 설명한 것은? ()

	특성	자바	자바스크립트
1)	작성 방법	별도의 파일 작성	HTML 내에 직접 삽입
2)	실행 방식	클라이언트에서 직접 해석되고 실행됨	서버에서 컴파일 된 후 클라이언트에서 수행됨
3)	변수 선언	변수의 자료형을 반드시 선언해야 함	변수의 자료형을 선언할 필요가 없음
4)	보안	소스를 볼 수 없으므로 보안을 유지할 수 있음	소스보기 메뉴를 통하여 소스를 볼 수 있으므로 보안성을 가지지 못함

-
12. 다음 중 자바의 장점에 해당하지 않는 것은? ()
- 1) 빠른 개발 가능
 - 2) 배우기 쉬움
 - 3) 환경에 독립적
 - 4) 풍부한 메소드 제공
13. 다음 중 사용 불가능한 식별자는? ()
- 1) catch
 - 2) varName
 - 3) str_this
 - 4) try123
14. A에 1의 값이 저장되어 있고 B에 1의 값이 저장되어 있을 경우 A^B 에 대한 출력되는 연산의 결과로 맞는 것은? ()
- 1) 1
 - 2) 0
 - 3) true
 - 4) false
15. 다음 중 우선순위로 최상위에 위치해 있는 연산자는? ()
- 1) =
 - 2) +
 - 3) &&
 - 4) !