

자바스크립트

■ 자바스크립트란?

웹 페이지 문서에 프로그래밍 기능을 부여하여 방문자와의 상호 작용, 서버와의 데이터 전송과 같은 다양한 기능을 통한 역동적인 페이지 구성에 사용



■ 자바스크립트의 특징

- 인터프리터 방식의 스크립트 언어
- 웹 페이지 문서에 바로 코딩하여 삽입할 수 있어 작성과 편집이 용이
- 소스 코드에 대해 개방적이며 시스템 플랫폼에 독립적임
- 소스 코드를 쉽게 볼 수 있어 보안상 허점이 발생할 수 있음
- 웹 페이지와 상호 작용과 폼(form)을 제어하기 위해 주로 사용
- 객체 기반 언어로 클래스의 상속을 지원하지 않음
- 호환성이 좋아서 시스템에 구애받지 않고 사용

특징	자바	자바스크립트
언어종류	컴파일 언어	인터프리터 언어
실행방식	자바 가상 기계 위	브라우저 위
작성위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수선언	변수의 타입 반드시 선언	변수의 타입을 선언하지 않아도 사용 가능

■ 자바스크립트의 작성

자바 스크립트 기본 형식

```

<!DOCTYPE html>
<html>
  <head>
    <script>
      문장
    </script>
  </head>
  <body>

  </body>
</html>

```

- 문장

문장 끝에 ' ; '을 붙여서 문장이 종료됨을 알려 줌

- 주석

주석은 실행되지 않고 설명을 위한 코드

한 줄 주석은 ' // '를 사용 여러 줄 주석은 ' /* '와 ' */ '사이에 내용을 적어 사용한다.

- <script> ... </script>

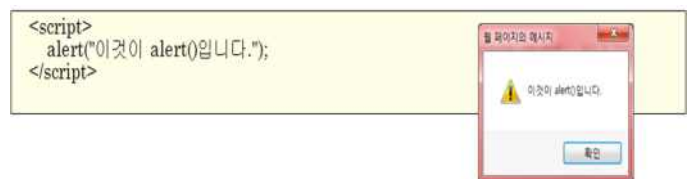
HTML 문서 내에 사용하기 위해서는 script 태그 내부에 적어 사용 가능하다. 하지만 여러 페이지에 사용을 위하여 외부 파일을 삽입하여 사용하는 경우가 많다.



■ 자바스크립트 코딩 시 확인 방법

- alert(변수);

alert() 함수는 특정 정보를 사용자에게 메시지 창으로 알려주기 위해 사용된다.



- document.write(변수);

document.write() 함수는 HTML 문서의 body 영역에 텍스트 및 변수를 출력할 때 사용하는 함수이다.



- console.log(변수);

console.log() 함수는 전문 디버깅 도구로 웹킨 엔진을 사용하는 크롬, 사파리 등에서 사용 가능하며 출력 값은 디버깅 기능을 가지고 있으며 콘솔 영역에 출력된다.

■ NOSCRIPT 태그

태그 사이에 웹 브라우저에서 스크립트 실행이 불가능할 경우 보여줄 내용을 입력 가능하다.

<NOSCRIPT> ... </NOSCRIPT>

■ 변수



■ 변수명 정의 규칙

- 첫 번째 문자는 영문자나 '_' 기호를 이용
- 영문/숫자/'_'의 조합
- 글자와 글자 사이에 공백 사용 불가
- 영문자의 경우 대소문자 구별
- for, in, if와 같은 예약어 사용 불가

자료형을 따로 선언하지 않으며 변수에 값을 대입하는 순간 변수의 자료형이 자동으로 정해진다. 변수를 선언하지 않는 경우에는 undefined 자료형으로 지정되었다가 값을 대입하는 순간 해당 자료형으로 변경된다.

: 문자열, 숫자, 논리, NULL, 객체, undefined 자료형

■ 변수의 선언

- 'var'를 사용하여 변수 선언
- 변수 선언과 동시에 값 초기화 가능
- 여러 개의 변수를 한번에 초기화 가능
- 변수 선언 후 저장되는 값에 따라 자료형 결정

변수의 선언

```
var 변수명1; // "변수명1" 선언
변수명1 = 값1; // "변수명1"에 "값1"을 대입
var 변수명2 = 값2; // "변수명2"를 선언하고 "값2"로 초기화
```

Source III-5 var.js

```
01 var kor, math, com, tot;
02 kor = 80; math = 70; com = 100;
03 tot = kor + math + com;
04 alert("총점" + tot + "점입니다.");
```

■ 실습예제

다음과 같이 작성하시오.

HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Javascript실습</title>
  <script src="js_var.js">
  </script>
</head>
<body>

</body>
</html>
```

Javascript :

```
var str="응용프로그래밍";
document.write("str의 내용 :"+str+"<br/>");
document.write("str의 자료형 :"+typeof(str)+"<hr/>");

var num=10;
document.write("num의 내용 :"+num+"<br/>");
document.write("num의 자료형 :"+typeof(num)+"<hr/>");

var n=10, bool;
bool=(n>0);
document.write("n의 내용 :"+n+"<br/>");
document.write("n의 자료형 :"+typeof(n)+"<hr/>");

var arr=["a","b","c"];
document.write("arr[2]의 내용 :"+arr[2]+"<br/>");
document.write("arr의 자료형 :"+typeof(arr)+"<hr/>");
```

※ 이때 a를 출력하기 위해서는 document.write 안의 내용을 어찌 바꿔야 할까요?

※ typeof()함수의 반환값은 무엇일까요?

■ 연산자

자바스크립트의 연산자는 다른 프로그래밍 언어와 마찬가지로 산술연산자, 관계(비교)연산자, 논리 연산자, 증감 연산자, 대입 연산자 등이 있다.

■ 산술연산자

연산자	설명	사용 예(A = 5, B = 2)	연산 결과(C의 값)
+	덧셈 연산	C = A + B	7
-	뺄셈 연산	C = A - B	3
*	곱셈 연산	C = A * B	10
/	나눗셈 연산	C = A / B	2
%	나머지 반환	C = A % B	1
-	음수로 변환	C = -A	-5

■ 관계연산자

연산자	설명	사용 예(A = 5, B = 2)	연산 결과(C의 값)
>	크다	C = (A > B)	true
>=	이상이다(크거나 같다)	C = (A >= B)	true
<	작다	C = (A < B)	false
<=	이하이다(작거나 같다)	C = (A <= B)	false
==	같다	C = (A == B)	false
!=	같지 않다	C = (A != B)	true

■ 논리연산자

연산자	설명	사용 예(A = 5, B = 2)	연산 결과(C의 값)
&&	논리 곱(AND)	C = (A < 3) && (B < 3)	false
	논리 합(OR)	C = (A < 4) (B < 3)	true
!	논리 부정(NOT)	C = ! (A < 3)	true

■ 증감연산자

연산자	설명
변수++	변숫값을 1 증가시킨다(후위 연산).
++변수	변숫값을 1 증가시킨다(전위 연산).
변수--	변숫값을 1 감소시킨다(후위 연산).
--변수	변숫값을 1 감소시킨다(전위 연산).

■ 실습예제

```
var a=10;
document.write("a="+a+"<br/>");
document.write("a="+(++a)+"<br/>");
document.write("a="+a+"<br/>");
document.write("a="+(++a)+"<br/>");
document.write("a="+a+"<br/>");
document.write("a="+(--a)+"<br/>");
document.write("a="+(--a)+"<br/>");
document.write("a="+a+"<br/>");
```

※ 다음의 결과가 어찌 나올지 예상해보세요.

■ 대입연산자

연산자	사용 예	결과
=	a = 5	a에 5를 대입
+=	a += 5	a = a + 5
-=	a -= 5	a = a - 5
*=	a *= 5	a = a * 5
/=	a /= 5	a = a / 5
%=	a %= 5	a = a % 5

■ 실습예제

```
var a=5;
document.write(a);
a+=3;
document.write(a);
a/=2;
document.write(a);
a%=2;
document.write(a);
```

※ 다음의 결과가 어찌 나올지 예상해보세요.

■ 제어문

조건에 따라 다른 문장을 수행하거나 특정 문장을 반복해서 수행할 경우 사용. 조건문과 반복문 있음.



- if 문
- switch 문
- for 문
- while 문
- do while 문

■ if문

단순 if 문 형식	단순 if 문 순서도
<pre>if (조건식){ 문장 }</pre>	

if ... else 문 형식	if ... else 문 순서도
<pre>if (조건식){ 문장 1 } else{ 문장 2 }</pre>	

다중 if 문 형식	다중 if 문 순서도
<pre>if (조건식 1){ 문장 1 } else if (조건식 2){ 문장 2 } else{ 문장 3 }</pre>	

■ switch문

switch 문 형식	switch 문 순서도
<pre>switch (조건식){ case 값1: 문장 1 break; case 값2: 문장 2 break; ... case 값 (n-1): 문장 (n-1) break; default: 문장 n break; }</pre>	

■ 실습예제

```
var score;
score=prompt("응포점수입력","응포점수를 입력하십시오");
if(score>=90){
  alert("A등급입니다.");
}
```

※ prompt()는 어떤 함수인가요?

※ 85, 90, 95를 입력했을 때 각각 어떻게 되나요?

※ 80이상 90미만은 "B등급입니다", 80미만은 "공부 하세요"가 나오도록 위의 실습예제를 수정해보세요.
(20XXX_if.html / 20XXX_if.js 로 저장 및 제출)

■ 실습예제

```
var score;
score=prompt("응포점수입력","응포점수를 입력하십시오");
if(score>=90)
  alert("A등급입니다.");
else if(score>=80)
  alert("B등급입니다.");
else if(score>=70)
  alert("C등급입니다.");
else
  alert("D등급입니다.");
```

다음은 switch 문으로 수정해보세요

(20XXX_switch.html/20XXX_switc.js로 저장 및 제출)

■ for문

for 문 형식	for 문 순서도
<pre>for (초기값; 조건문; 증감문){ 문장 }</pre>	

초기식 조건식 증감식
 for(i=0 ; i<10 ; i++)
 {
 document.write(i+"
");
 }
 반복되는 문장

■ 실습예제

for문을 활용하여 다음과 같이 출력하여보세요.

(20XXX_timetable.html / 20XXX_timetable.js)

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

■ while문

while 문 형식	while 문 순서도
<pre>while (조건식){ 문장 }</pre>	

■ 함수



생성	호출
function 함수명(){ 문장 }	함수명();

■ 실습예제

```
var kor=80, com=96, sum=0, average=0;

function total(){
    sum=kor+com;
    alert(sum);
}

function avg(){
    average=sum/2;
    alert(average);
}

total();
avg();
```

※ 다음의 소스 코드의 결과를 확인하시오

※ 매개 변수란 무엇인가?

※ 리턴값(반환값)이란 무엇인가?

■ 내장 함수

자바 스크립트에서 미리 만들어져 제공되는 함수

■ 실습예제

HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Javascript실습</title>
    <script src="js_var.js"></script>
</head>
<body>
    
    <input type="button" onclick="changeImage()" value="눌러보세요">
</body>
</html>
```

JS :

```
function changeImage(){
    document.getElementById(Image).src="b.jpg";
}
```

※ 실행결과를 확인해보세요

※ getElementById() 함수의 어떤 함수일까요?

■ 실습예제

This is a paragraph. This is a paragraph.

색바꾸기 → 색바꾸기

※ 버튼 클릭시 색이 빨간색으로 바뀌도록 만드시오.

■ 객체

객체 지향 프로그래밍 (OOP, Objective-oriented programming)에서 데이터(속성)와 데이터에 관련된 동작(메소드)를 포함한 개념.

Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

※사용자 지정 객체

HTML :

```
<body>
  <p>JavaScript Objects</p>
  <p id="test"></p>
</body>
```

JS :

```
var car = {
  name : "Fiat",
  model : "500",
  weight : 850,
  drive : function(){
    return this.name + "is driving";
  }
}
document.getElementById("test").innerHTML = car.fullName();
```

※내장객체 : Math

미리 저장되어 있는 객체

- 난수 생성

0(포함) ~ 1(미포함) 사이의 난수를 생성

```
document.write(Math.random());
```

- 1~10 사이의 정수 생성

```
document.write(Math.floor(Math.random()*11));
```

■ 실습예제

5~12 사이의 난수를 생성하여라.

- Math.round()

반올림 값을 반환하는 함수

```
document.write(Math.round(4.2));
document.write(Math.round(6.5));
document.write(Math.round(8.7));
```

- Math.abs()

절대값을 반환하는 함수

```
document.write(Math.abs(-4.2));
```

- Math.floor()

버림값을 반환하는 함수(<->ceil)

```
document.write(Math.floor(3.2));
document.write(Math.floor(7.9));
```

※내장객체 : Date

- Date 객체 생성

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

- 임의로 값 지정

```
var da = new Date(2018,8,20,8,10,20,0);
document.write(da);
```

※내장객체 : Array

- Array(배열) 생성

```
var arr = ["one","two","three"];
var arr2 = new Array("four", "five", "six");
```

이때 arr[1] 의 값은?

- 배열의 길이

```
document.write(arr.length);
```

- 배열의 추가

```
arr[arr.length]="add";
```

■ 이벤트

사용자가 키를 누를 때와 같이 어떠한 조건을 만족시킬 경우 일어난다. 이벤트에 따라 또 다른 사건이 발생하면 이 사건을 액션이라고 한다.

마우스이벤트

이벤트	발생 시점
blur	마우스의 포커스를 다른 곳으로 이동했을 때
click	하이퍼링크나 버튼을 클릭했을 때
change	폼 입력 양식의 선택 값을 변경했을 때
focus	폼 입력 양식에 포커스가 위치했을 때
mouseover	마우스가 위에 있을 때
mouseout	마우스가 떠났을 때
mousedown	마우스 버튼을 눌렀을 때
mousemove	마우스를 움직였을 때
mouseup	마우스 버튼을 눌렀다 놓았을 때

기타이벤트

이벤트	발생 시점
select	폼 입력 양식을 선택했을 때
submit	폼 입력 양식의 전송 버튼을 클릭했을 때
load	웹 문서가 브라우저로 읽혀졌을 때
unload	다른 웹 문서로 이동할 때
error	웹 문서나 이미지 로딩 중 오류가 발생했을 때
reset	폼 입력 양식의 리셋 버튼을 클릭했을 때
dblclick	더블 클릭을 했을 때
dragdrop	마우스를 드래그 앤드 드롭(drag & drop) 했을 때
keydown	키를 입력했을 때
keypress	키를 눌렀을 때(문자 키만 인식)
keyup	키를 눌렀다 놓았을 때
move	윈도나 프레임을 움직였을 때
resize	윈도나 프레임 크기를 변경했을 때

■ 이벤트 핸들러

이벤트가 발생했을 때 이를 처리하는 함수로 이벤트 이름 앞에 'on'을 붙여 작성한다.