



javascript 기초

안 화 수

JavaScript

❖ JavaScript 개요

- JavaScript는 1995년에 브렌던 아이크(Brendan Eich)에 의해 개발된 Script 언어이다.
- JavaScript는 웹 페이지를 동적으로 만들고 상호작용을 가능하게 하는데 사용된다.
- JavaScript의 공식 이름은 ECMAScript 이다.
- ECMAScript 버전은 ES1, ES2, ES3, ES5 및 ES6으로 축약되어 사용되었다.
- 2016년부터 새 버전의 이름은 연도별로 지정된다.
(ECMAScript 2016/2017/2018)
- ECMA(European Computer Manufacturers Association)는 정보와 통신 시스템을 위한 국제적 표준화 기구이다.

ECMAScript

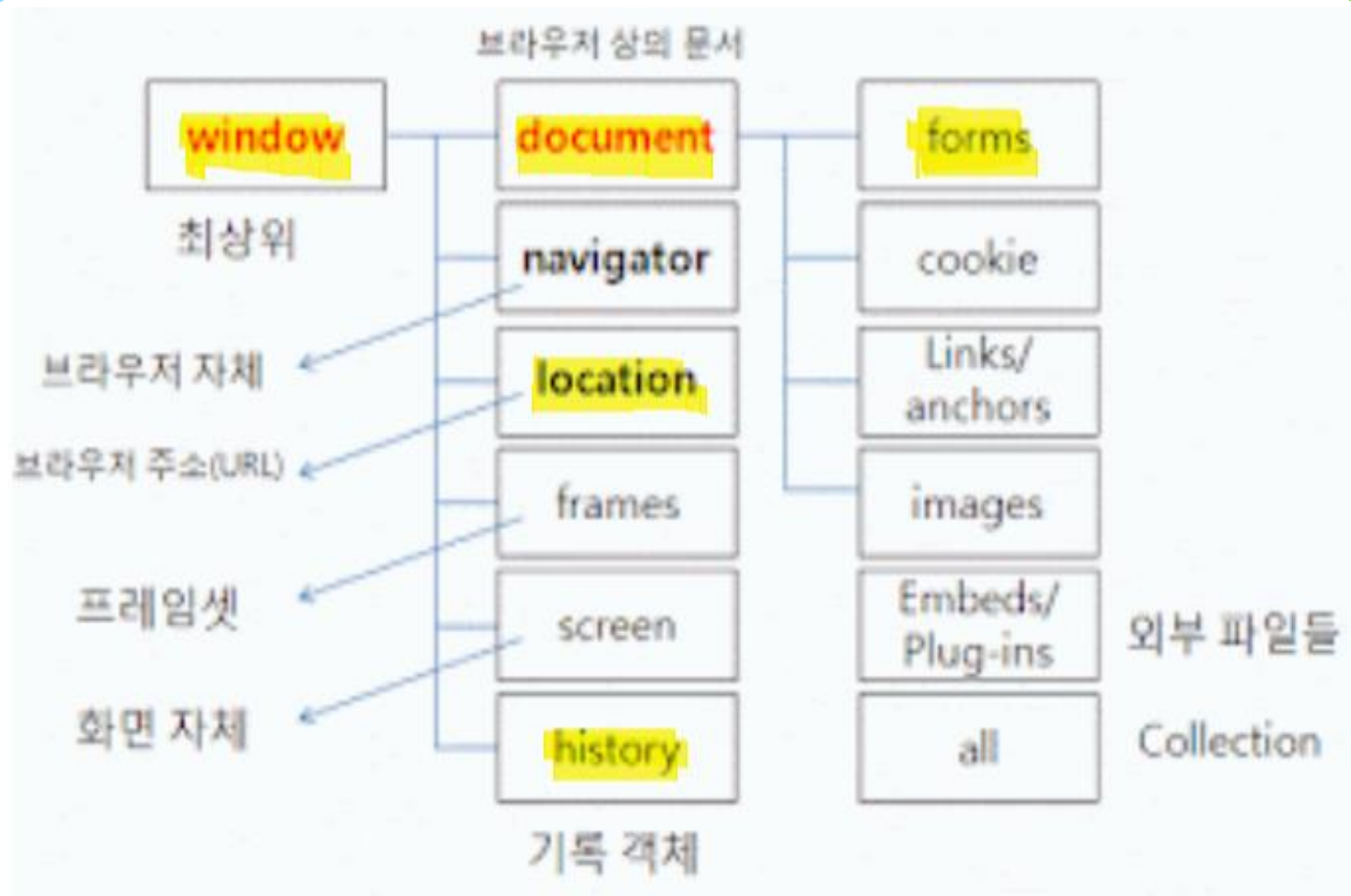
	공식 이름	설명
ES1	ECMAS크립트 1(1997)	초판
ES2	ECMAScript 2(1998)	편집 변경 사항
ES3	ECMAScript 3(1999)	정규식 추가 try/catch 추가 스위치 추가 do-while 추가
ES4	ECMAS크립트 4	출시되지 않음
ES5	ECMAScript 5(2009) 	"엄격 모드" 추가 JSON 지원 추가 String.trim() 추가 Array.isArray() 추가 배열 반복 메서드 추가 개체 리터럴에 후행 쉼표 허용
ES6	ECMAS크립트 2015	let 및 const 추가 기본 매개변수 값 추가 Array.find() 추가 Array.findIndex () 추가
	ECMA 스크립트 2016 	지수 연산자 추가(**) Array.includes() 추가
	ECMAS크립트 2017	문자열 패딩 추가된 Object.entries() 추가된 Object.values() 추가된 비동기 함수 추가된 공유 메모리 추가됨
	ECMAS크립트 2018 	나머지/확산 속성 추가 비동기 반복 추가 Promise.finally() RegExp에 추가 추가

JavaScript

❖ JavaScript 특징

- 동적으로 이벤트를 처리하며 데이터 타입을 명시할 필요가 없는 인터프리터 언어 이다.
- 객체지향 프로그래밍과 함수형 프로그래밍을 모두 표현할 수 있다.
- HTML의 내용, 속성, 스타일 등을 변경 할 수 있다.
- 이벤트를 처리하고 사용자와의 상호작용을 가능하게 한다.
- AJAX 기술을 이용해 서버와 실시간 통신 기능을 제공한다.

브라우저 내장객체



JavaScript

❖ JavaScript 주석

프로그램의 실행에 전혀 영향을 주지 않는 코드로, 프로그램을 설명하는데 주로 사용된다.

- 주석 단축키 : Ctrl + Shift + /
- 주석해제 단축키 : Ctrl + Shift + ⌘

방법	형태
❶ 한 행 주석 처리	// 주석문
❷ 여러 행 주석 처리	/* 주석문 주석문 */

```
<script>  
    // 주석은 코드 실행에 영향을 주지 않습니다.  
    /*  
    alert('Hello JavaScript .. !');  
    alert('Hello JavaScript .. !');  
    alert('Hello JavaScript .. !');  
    */  
</script>
```

JavaScript

❖ JavaScript 시작

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>first.html</title>
</head>
<body>
<script>
    window.document.writeln("처음 작성하는 자바스크립트1<br>");
    document.write('처음 작성하는 자바스크립트2'+ '<br>');
    document.write("처음 작성하는 자바스크립트3"+"<br>");
</script>

<script type="text/javascript">
// 단일행 주석
// document.write("처음 작성하는 자바스크립트4<br>");
</script>

<script language="javascript">
// 다중행 주석 : Ctrl + Shift + /
/*    document.write("처음 작성하는 자바스크립트5<br>");
    console.log('콘솔에 출력됨');    */
</script>
</body>
</html>
```

JavaScript

❖ JavaScript 소스코드 위치

1. 내부 JavaScript

자바스크립트 코드는 `<head> </head>` 혹은 `<body> </body>` 태그안에 모두 사용할 수 있다.

```
<head>
```

```
  <script>
```

```
    alert('Hello World~!!')
```

```
  </script>
```

```
</head>
```


JavaScript

2. 외부 JavaScript

- **외부 JavaScript 파일 생성**

external.js

```
function printDate(){  
    alert('함수 호출 성공');  
}
```

- **외부 JavaScript 파일 불러오기**

<head>

<script src="external.js"> </script>

</head>

JavaScript 변수

❖ JavaScript 변수

메모리상에 데이터를 저장하는 기억공간의 이름이다.

■ 변수 만드는 형식

```
var 변수명 = 값(데이터);
```

```
let 변수명 = 값(데이터);
```

```
const 변수명 = 값(데이터);
```

변수명 = 값(데이터); 선언자가 없으면 **전역 변수**가 된다.

JavaScript 변수

❖ 변수 선언자 : var

- var를 사용해서 변수를 선언하거나 또는 변수 선언과 동시에 값을 저장할 수 있다.
- var를 사용해서 변수 선언하면 한 번에 여러 개의 값도 할당할 수 있다.
- var 선언자로 선언된 변수에 값을 재할당 할 수 있다.

```
<script>
    var abc;
    var abc=10;
    var abc=100;           // 같은변수에 재할당 가능함
    console.log(abc);      // 100

    var a=10, b=20, c=30;  // 동시에 여러개 변수에 값 할당 가능함
</script>
```

JavaScript 변수

❖ 변수 선언자 : let

- ECMAScript 6(ES6)부터 var 말고도 let과 const로 변수를 선언할 수 있다.
- let를 사용해서 변수를 선언하거나 또는 변수 선언과 동시에 값을 저장할 수 있다.
- let를 사용해서 변수 선언하면 한 번에 여러 개의 값도 할당할 수 있다.
- let 선언자로 선언된 변수에 값을 재할당 할 수 없다.

```
<script>
```

```
let t;
```

```
let abc=10;
```

```
let abc=100;
```

```
// error:같은변수에 재할당 안됨
```

```
let a=10, b=20, c=30; // 동시에 여러개 변수에 값 할당 가능함
```

```
</script>
```

JavaScript 변수

❖ 변수 선언자 : const

- ECMAScript 6(ES6)부터 var 말고도 let과 const로 변수를 선언할 수 있다.
- let를 사용해서 변수를 선언하거나 또는 변수 선언과 동시에 값을 저장할 수 있다.
- let를 사용해서 변수 선언하면 한 번에 여러 개의 값도 할당할 수 있다.
- **const 선언자 변수는 반드시 초기화를 해야한다. (상수)**
- **const 선언자로 선언된 변수에 값을 재할당 할 수 없다. (상수)**

```
<script>
```

```
const con;
```

```
// error:변수 초기화 해야함
```

```
const abc=10;
```

```
const abc=100;
```

```
// error:같은변수에 재할당 안됨
```

```
const a=10, b=20, c=30; // 동시에 여러개 변수에 값 할당 가능함
```

```
</script>
```

JavaScript 변수

❖ 변수명 작성시 주의 사항

1. 변수명은 대.소문자를 구분 한다.

```
var apple  
var APPLE
```

2. 변수명은 영문자, 숫자, \$, _ 로 만들 수 있다.

```
var apple3, var apple$, var apple_
```

3. 변수명의 첫 글자는 숫자로 시작할 수 없다.

```
var 3apple (x)  
var $apple (0)  
var _apple (0)
```

4. 예약어로 설정된 단어들은 변수명으로 사용할 수 없다.

```
ex) true, false, for, while, if.....
```

JavaScript 변수

```
<script>
  // 1. 숫자 데이터
  var num = 10;
  var pi = 3.14;
  document.write("num=" + num + "<br>");
  document.write("pi=", pi + "<br>");
  document.write(typeof(num)+"<br>");           // number
  document.write(typeof(pi)+"<br>");           // number

  // 2. 문자 데이터
  var str1 = '자바스크립트';
  var str2 = "javascript";
  document.write("str1=", str1 + "<br>");
  document.write("str2=", str2 + "<br>");
  document.write(typeof(str1)+"<br>");         // string
  document.write(typeof(str2)+"<br>");         // string

  // 3. 논리 데이터
  var b1 = true;
  var b2 = false;
  document.write("b1=", b1 + "<br>");
  document.write("b2=", b2 + "<br>");
  document.write(typeof(b1)+"<br>");           // boolean
  document.write(typeof(b2)+"<br>");           // boolean

  // 4. null 데이터
  var str = null;
  document.write("str=", str + "<br>");         // null
  document.write(typeof(str)+"<br>");         // object

  var result = 30;
  var result = "자바스크립트";
  document.write("result=" + result);          // result=자바스크립트
</script>
```

내장함수

❖ 내장함수

1. alert()

메시지 박스 출력할때 사용하는 함수.

2. confirm()

사용자로 부터 중요한 결정을 물어볼때 주로 사용하는 함수.

3. prompt()

사용자로 부터 키보드로 입력을 받을때 사용하는 함수.

내장함수

❖ 내장함수 : alert()

- 메시지 박스 출력하는 함수

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>alert함수</title>
</head>
<body>

<script>
    window.alert("경고창 출력");    // 메세지 박스
    alert("이름을 입력하세요.");
</script>

</body>
</html>
```

내장함수

❖ 내장함수 : confirm()

- 사용자로 부터 중요한 결정을 물어볼때 주로 사용하는 함수
- 확인 버튼을 클릭하면 true 값을 리턴하고, 취소 버튼을 클릭하면 false 값을 리턴한다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>confirm함수</title>
</head>
<body>

<script>
    var result = confirm("정말로 삭제 하시겠습니까?");
    alert(result);
    if(result){
        alert("삭제 성공");
    }else{
        alert("삭제 실패");
    }
</script>

</body>
</html>
```

내장함수

❖ 내장함수 : prompt()

- 사용자로부터 키보드로 입력을 받을때 사용하는 함수
- 형식 : prompt(' 메시지?' , ' 초기값')

```
<script>
    var name = prompt("이름을 입력하세요?", "");
    document.write("당신의 이름은 " + name + "<br>");
    document.write(typeof(name)+ "<br>");    // string
</script>
```

```
<script>
    var age = prompt("나이를 입력 하세요?", "");    // age = "25";
    alert(typeof(age));    // string
    if(age >= 20){
        alert("당신은 성인 입니다.");
    }else{
        alert("당신은 미성년자 입니다.");
    }
</script>
```

연산자

❖ 연산자 종류

1. 산술 연산자
2. 확장 대입 연산자
3. 증감 연산자
4. 조건 연산자
5. 비교 연산자
6. 논리 연산자

연산자

❖ 산술 연산자

```
<script>
    // 산술 연산자 : +, -, *, /, % (나머지)

    var num1 = 15;
    var num2 = 2;
    var result;

    result = num1 + num2;           // 더하기
    document.write("num1+num2=" + result + "<br>");

    result = num1 - num2;           // 빼기
    document.write("num1-num2=" + result + "<br>");

    result = num1 * num2;           // 곱하기
    document.write("num1*num2=" + result + "<br>");

    result = num1 / num2;           // 나누기
    document.write("num1/num2=" + result + "<br>"); // 7.5

    result = num1 % num2;           // 나머지
    document.write("num1%num2=" + result + "<br>");
</script>
```

연산자

❖ 산술 연산자

```
<script>
    // 산술 연산자
    var t1 = "학교종이";
    var t2 = "땡땡땡";
    var t3 = 8282;
    var t4 = "어서 모이자";
    var result;

    result = t1 + t2 + t3 + t4;
    document.write("result=" + result);
</script>
```

연산자

❖ 확장 대입 연산자

```
<script>
    // 확장 대입 연산자 : +=, -=, *=, /=, %=
    var num1 = 10;
    var num2 = 3;

    num1 += num2;        // num1 = num1 + num2
    document.write("num1="+num1+"<br>");    // 13

    num1 -= num2;        // num1 = num1 - num2
    document.write("num1="+num1+"<br>");    // 10

    num1 *= num2;        // num1 = num1 * num2
    document.write("num1="+num1+"<br>");    // 30

    num1 /= num2;        // num1 = num1 / num2
    document.write("num1="+num1+"<br>");    // 10

    num1 %= num2;        // num1 = num1 % num2
    document.write("num1="+num1+"<br>");    // 1
</script>
```

연산자

❖ 증감 연산자

```
<script>
    // 증감 연산자 : ++, --
    var num1 = 10;
    var num2 = 20;

    num1--;                // 후행 연산
    document.write("num1="+ num1+"<br>");           // 9

    num1++;                // 후행 연산
    document.write("num1="+ num1+"<br>");           // 10

    result = num2++;        // 후행 연산
    document.write("num2="+ num2+"<br>");           // 21
    document.write("result="+ result+"<br>");       // 20

    result = ++num2;        // 선행 연산
    document.write("num2="+ num2+"<br>");           // 22
    document.write("result="+ result+"<br>");       // 22
</script>
```


연산자

❖ 조건 연산자

```
<script>
    // 조건 연산자
    // 변수 = (조건식) ? 값1 : 값2;
    // 조건식 참이면 값1을 변수에 리턴하고
    // 조건식이 거짓이면 값2를 변수에 리턴한다.

    // 키보드로 정수 2개를 입력 받아서 최대값과 최소값을 구하는 프로그램을
    // 조건 연산자를 이용해서 작성하세요?

    var n1 = prompt("정수1을 입력 하세요?", "");
    var n2 = prompt("정수2를 입력 하세요?", "");
    alert(typeof(n1));    // string
    alert(typeof(n2));    // string

    var max = (n1 > n2) ? n1 : n2;
    var min = (n1 < n2) ? n1 : n2;

    document.write("max="+ max + "<br>");
    document.write("min="+ min + "<br>");
</script>
```

연산자

❖ 비교 연산자

```
<script>
    // 비교 연산자 : >, >=, <, <=, ==, !=

    var a = 10;
    var b = 20;
    var c = 10;
    var f = "20";
    var result;

    result = a > b;
    document.write("a>b="+ result + "<br>");    //false

    result = a < b;
    document.write("a < b="+ result + "<br>");    // true

    result = b == f;
    document.write("b == f="+ result + "<br>");    // true

    result = a != b;
    document.write("a!=b="+ result + "<br>");    // true

    result = b != f;
    document.write("b != f="+ result + "<br>");    //false

    document.write('k' > 'z');                    //false
</script>
```

연산자

❖ 비교 연산자

```
<script>
    // 키보드로 정수 2개를 입력 받아서 최대값과 최소값을 구하는 프로그램을 작성하세요?

    var n1 = prompt("정수1을 입력하세요?", "");
    var n2 = prompt("정수2를 입력하세요?", "");
    alert(typeof(n1));           // string
    alert(typeof(n2));           // string

    var max, min;

    if(n1 > n2){                  // 문자형 변수끼리 비교 연산을 할 수 있다.
        max = n1;
        min = n2;
    }else{
        max = n2;
        min = n1;
    }
    document.write("max="+ max + "<br>");
    document.write("min="+ min + "<br>");
</script>
```

연산자

❖ 논리 연산자 : &&, ||, !

5과목의 점수를 입력 받아서 합격, 불합격을 판별하는 프로그램을 작성하세요?
단, 과목당 과락은 40점이고, 평균 60점 이상을 받아야 합격한다.

■ 형변환 함수

Number() : 문자를 숫자로 변환해주는 함수

Number("30") ---> 30

Number("3.14") ---> 3.14

연산자

❖ 논리 연산자

```
<script>
    var n1 = prompt("과목1 점수를 입력하세요?", "");    // string
    var n2 = prompt("과목2 점수를 입력하세요?", "");
    var n3 = prompt("과목3 점수를 입력하세요?", "");
    var n4 = prompt("과목4 점수를 입력하세요?", "");
    var n5 = prompt("과목5 점수를 입력하세요?", "");

    // var total = n1 + n2 + n3 + n4 + n5;    합이 구해지지 않는다.
    // console.log(total);

    // Number() : 문자를 숫자로 변환해주는 함수
    // Number("30")    --->    30
    // Number("3.14")    --->    3.14

    var total = Number(n1)+Number(n2)+Number(n3)+Number(n4)+Number(n5);
    console.log('total:', total);
    var avg = total / 5;
    console.log('avg:', avg);

    if(n1>=40 && n2>=40 && n3>=40 && n4>=40 && n5>=40 && avg>=60) {
        alert("합격");
    }else{
        alert("불합격");
    }
}
</script>
```