

변수란?

데이터(값)을 저장할 수 있는 메모리 공간
오직 한 개의 데이터만 저장
새로운 데이터가 들어가면 기존의 데이터는 갱신

변수선언

```
var 변수명;  
var 변수명 = 값;
```

변수를 만들 때 주의사항

1. 숫자로 시작하면 안된다.
2. 대소문자를 구분한다.
3. 소문자로 시작한다.
4. 단어의 조합에는 CamelCase로 작성한다.
5. 자바스크립트의 예약어는 안된다.
- document, window, location...

```
<script>  
    var box;  
    box = 100;  
    box = 30;  
  
    document.write(box);  
</script>
```

```
<script>  
    var 1num = 10;  
    var num100 = 10;  
    var document = "test";  
    var num = "hello";  
</script>
```

변수에 저장할 수 있는 자료형

1. 문자형(string) - 큰따옴표/작은따옴표 사용, 문자로 인식

2. 숫자형(number)

3. 논리형(boolean) - true / false

4. undefined

- 변수를 만든 후 초기화하지 않은 상태

- 변수에 값이 등록되기 전의 기본값

5. null

- 객체를 담을 변수를 초기화할 때

- 빈 값(변수에 저장된 데이터를 비우고자 할 때)

```
<script>
    var s = true;
    var t = 10>=100;
    var k = Boolean("hello");
</script>
```

```
<script>
    var a = true;
    var b = false;
    var c = 10>5;
    var d = Boolean(null);

    document.write(a, "<br>");
    document.write(b, "<br>");
    document.write(c, "<br>");
    document.write(d, "<br>");

</script>
```

Boolean()메서드에 데이터를 입력하면

0, null, undefined --> false

나머지는 모두 true

변수

- * 객체처럼 데이터명과 값으로 이루어진 데이터 덩어리를 저장할 수는 없다.
- * 숫자나 문자열 같은 값뿐만 아니라 객체나 배열도 저장할 수 있다.

배열(Array)

변수 하나에 여러 개의 데이터를 담을 수 있다.

```
<script>
|   var userName = ["yssmi", "moodang-teacher", "smack"];
|   document.write(userName[1]);
</script>
```

변수의 종류

1. 전역변수

- * 전역에서 사용하는 데이터를 담는 변수
- * 어디서든 사용 가능

3. 매개변수(파라미터)

- * 함수 외부에서 함수 내부로 데이터를 전달하기 위한 용도로 사용하는 변수

2. 지역변수

- * 특정 영역에서만 사용할 수 있는 변수
- * 주로 함수내부에서 만들어지는 변수

4. 멤버변수(프로퍼티)

- * 클래스 내부에 만들어지며 주로 객체에서 사용하는 정보를 담는 변수

연산자

다양한 계산 작업을 위한

1. 산술
2. 문자 결합
3. 복합대입
4. 증감
5. 비교
6. 논리
7. 삼항 조건

1. 산술 연산자

| 종류 | 기본형 | 설명 |
|----|-------|-----|
| + | A + B | 더하기 |
| - | A - B | 빼기 |
| * | A * B | 곱하기 |
| / | A / B | 나누기 |
| % | A % B | 나머지 |

```
<script>
    var num1 = 15;
    var num2 = 2;
    var result;

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

2. 문자결합 연산자

- 연산대상이 문자형 데이터
- 여러개의 문자를 하나로 결합할 때 사용
- 문자형 데이터가 하나라도 포함되어 있으면 다른 연산대상도 모두 문자형으로 변환

| 종류 | 기본형 | 설명 |
|----|-------|---------------|
| + | A + B | A, B를 결합 = AB |

```

<script>
    // 문자형 + 문자형 = 하나의 문자형 데이터
    var str1 = "부채살";
    var str2 = "스테이크";
    var result1 = str1 + str2;

    document.write(result1);

    // 문자형 + 숫자형 = 하나의 문자형 데이터
    var str3 = "100";
    var num1 = 200;
    var result2 = str3 + num1;

    document.write(result2);
</script>

```

```

<script>
    var t1 = "학교종이";
    var t2 = "땡땡땡";
    var t3 = 8282;
    var t4 = "어서 모이자!";

    var result;

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

```

3. 대입 연산자

| 종류 | 설명 | |
|--------|---------|----------------------|
| A = B | A = B | B를 A에 대입 |
| A += B | A = A+B | A와 B를 더해서 그 값을 A에 대입 |
| A *= B | A = A*B | |
| A /= B | A = A/B | |
| A %= B | A = A%B | |

```
<script>
    var num1 = 10;
    var num2 = 3;

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

</script>
```

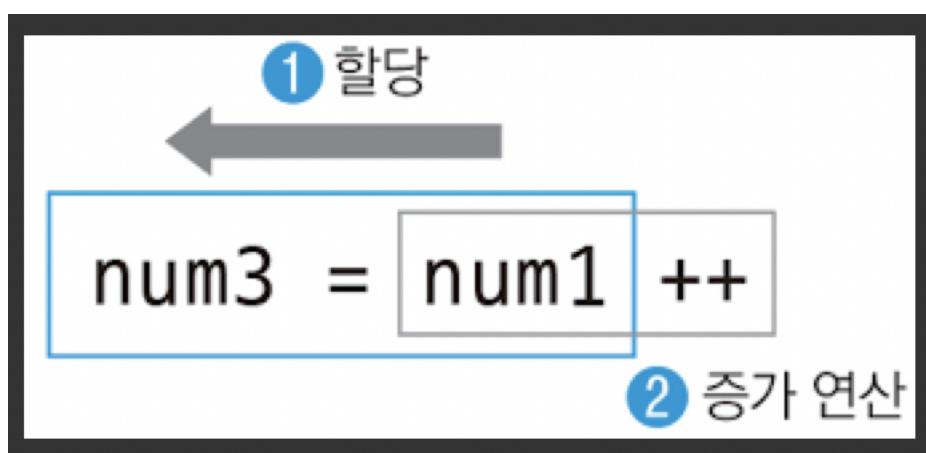
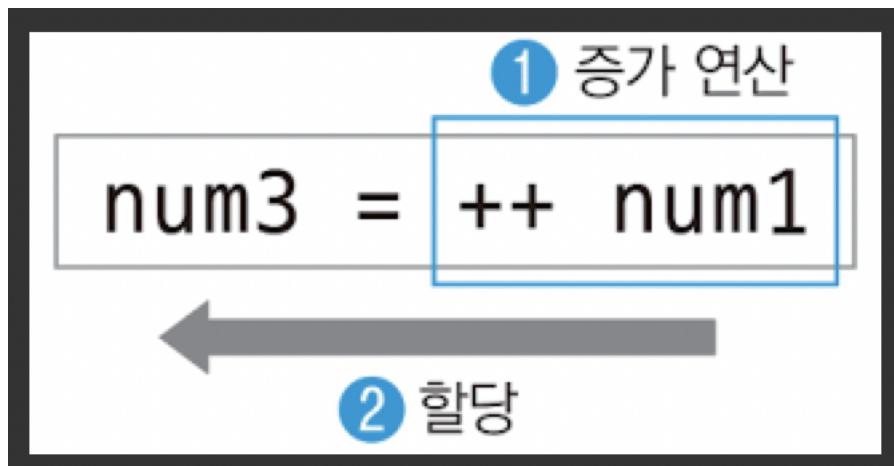
```
<script>
    var txt = "방탄소년단";
    txt += "BTS";
    txt += "대박났어!";

    document.write(txt);
</script>
```

```
<script>
    var str = "<table border='1'>";
    str += "<tr>";
    str += "<td>1</td><td>2</td><td>3</td>";
    str += "</tr>";
    str += "</table>";

    document.write(str);
</script>
```

4. 증감 연산자



```
<script>
    var num1 = 10;
    var num2 = 20;
    var result;

    num1--;
    document.write(num1, "<br>");

    num1++;
    document.write(num1, "<br>");

    result = num2++;
    document.write(result, "<br>");

    result = ++num2;
    document.write(result);
</script>
```

5. 비교 연산자 (true / false)

| 종류 | 설명 |
|---------|-----------------------------|
| A > B | A가 B보다 크다. |
| A < B | A가 B보다 작다. |
| A >= B | A가 B보다 크거나 같다. |
| A <= B | A가 B보다 작거나 같다. |
| A == B | A와 B는 같다. (데이터형까지 비교하지는 않음) |
| A != B | A와 B는 같지 않다. |
| A === B | A와 B는 완전히 같다. |

```
<script>
    var a = 10;
    var b = 20;
    var c = 10;
    var d = "20";
    var result;

    result = a > b;
    document.write(result, "<br>");

</script>
```

6. 논리 연산자 (true / false)

| 종류 | 설명 |
|----|-----------------------------------|
| | or연산자 : 피연산자 중 하나라도 true이면 true |
| && | and연산자 : 피연산자 중 하나라도false이면 false |
| ! | not연산자 : 청개구리, 항상 반대로 값을 반환 |

연산자 우선순위

1. 단항 연산자 (--, ++, !)
2. 산술 연산자 (*, /, %, +, -)
3. 비교 연산자 (>, >=)
4. 논리 연산자
5. 대입 연산자 (=, +=, -=...)

괄호를 사용해라!

$2 + 1 * 3 = ?$

$(2+1) * 3 = ?$

```

<script>
  var a = 10;
  var b = 20;
  var m = 30;
  var n = 40;
  var result;

  result = a > b || b >= m || m > n;
  document.write(result, "<br>");

  result = a <= b && b >= m && m <= n;
  document.write(result, "<br>");

  result = !(a > b);
  document.write(result);
</script>

```

조건문 if

조건식의 값이 참인지 거짓인지에 따라 자바스크립트 코드를 제어

```
if ( 조건식 ) {
    자바스크립트 코드
}
```

```
<script>
|     var num = 10;
|     if (num>500) {
|         |     alert("Hi~");
|     }
</script>
```

```
<script>
|     var walkAmount = prompt("당신의 하루 걷는 양은 몇 보입니까?", 0);

|     if (walkAmount >= 10000) {
|         |     document.write("살 좀 빠지겠는걸?", "<br>");
|     }
|     document.write("스크립트 종료");
</script>
```

조건문 if

조건식에 논리형 데이터가 아닌 다른 형이 오는 경우

0, null, ""(빈문자), undefined ---> false

위 상황을 제외하고 모두 true

```
<script>
    var userName = prompt("넌 누구니?", "");
    if (userName) {
        document.write(userName+"님 반가워요!", "<br>");
    }
</script>
```

조건문 else

조건식을 만족할 경우와 그렇지 않은 경우에 따라 실행되는 코드를 다르게 구성

```
if ( 조건식 ) {
    자바스크립트 코드 1;
} else {
    자바스크립트 코드 2;
}
```

```
<script>
    var num = prompt("당신이 좋아하는 숫자는?", 0);

    if (num % 2 == 0) {
        document.write("짝수를 좋아하는군");
    } else {
        document.write("홀수를 좋아하는군");
    }
</script>
```

조건문 else if

두가지 이상의 조건식과 정해 놓은 조건을 만족하지 않는 경우를 구분

```
if ( 조건식 ) {
    자바스크립트 코드 1;
} else if {
    자바스크립트 코드 2;
} else if {
    자바스크립트 코드 3;
} else{
    자바스크립트 코드 4;
}
```

```
<script>
    var month = prompt("지금은 몇 월입니까?", "0");

    if (month >= 3 && month <= 5) {
        alert("봄이구나!");
    } else if (month >= 6 && month <= 8) {
        alert("여름이네!");
    } else if (month >= 9 && month <= 11) {
        alert("가을이 왔군!");
    } else {
        alert("겨울이군!");
    }
</script>
```

중첩 if문

조건문 안에 또 조건문

```
if ( 조건식1 ) {
    if ( 조건식2 ) {
        자바스크립트 코드;
    }
}
```

```
<script>
    // 아이디와 비밀번호 초기설정
    var id = "yssmi";
    var pwd = "1234";

    // 사용자 아이디와 비밀번호를 입력 받음
    var userId = prompt("아이디를 입력하세요", "");
    var userPwd = prompt("비밀번호를 입력하세요", "");

    if (userId == id) {

        if (userPwd == pwd) {
            document.write(userId + "님 환영합니다!");
        } else {
            alert("비밀번호를 확인해 주세요.");
            location.reload();
        }

    } else {
        alert("아이디가 일치하지 않습니다.");
        location.reload();
    }

</script>
```

switch

변수에 저장된 값을 switch문의 경우의 값을 비교, 코드를 실행

```
var 변수 = 초기값;  
  
switch (변수) {  
    case 값1 : 코드 1;  
    break;  
  
    case 값2 : 코드 2;  
    break;  
  
    default: 코드 3;  
}
```

```
<script>  
    var site = prompt("다음, 네이버, 구글 중에서 즐겨 사용하는 검색 사이트는?", "");  
    var url;  
  
    switch (site) {  
        case "다음": url = "www.daum.net";  
        break;  
  
        case "네이버": url = "www.naver.com";  
        break;  
  
        case "구글": url = "www.google.com";  
        break;  
  
        default: alert("이상한거 쓰네?");  
    }  
  
    if (url) location.href = "http://" + url;  
</script>
```

while

코드를 원하는 횟수만큼 반복

```
var 변수 = 초기값;  
while (조건식) {  
    자바스크립트 코드;  
    증감식;  
}
```

```
<script>  
    var i = 1;  
    while (i <= 10) {  
        document.write("안녕하세요", "<br>");  
        i++;  
    }  
    document.write("----- END -----");  
</script>
```

```
<script>  
    var i = 1;  
  
    while (i <= 30) {  
        if (i%2 == 0 && i%6 == 0) {  
            document.write(i, "<br>");  
        }  
        i++;  
    }  
</script>
```

for

조건식을 만족할 때까지 특정 코드를 반복

```
for (초기값; 조건식; 증감식) {  
    자바스크립트 코드;  
}
```

```
<script>  
    for (var i = 1; i <= 10; i++) {  
        document.write(i, "<br>");  
    }  
</script>
```

객체

자신의 정보를 가지고 있는 독립적인 주체이다.

우리 주위에 보이는 컴퓨터, 책상, 의자, 마우스, 키보드 등이 모두 자신만의 특성을 가진 객체인 셈이다.

Object = Property + Method

TV (객체-Object)

특징 (Property)

- 가로크기
- 세로크기
- 색상
- 무게
- ...



기능 (Method)

- 켜다
- 끄다
- 볼륨을 높이다
- 볼륨을 줄이다
- ...

TV.가로크기; TV.켜다();

객체의 종류

| 내장객체 | 브라우저 객체 모델(BOM) | 문서 객체 모델(DOM) |
|-------------------------------------------------|------------------------------------------------------|--------------------------------------------|
| 자바스크립트 엔진에 내장 | 브라우저에 계층구조로 내장 | HTML문서 구조 |
| 문자(String) 날짜(Date) 배열(Array) 수학(Math) | window screen location history navigator | <html> <head> <body> <h1> |

자바스크립트는 객체 기반 언어

다양한 객체들로 구성, 각 객체마다 다양한 메서드를 가지고 있음.

객체

참조 변수 = **new** 생성 함수();

```
var tv = new Object();
```

```
<script>
  var tv = new Object();

  tv.color = "white";
  tv.price = 3000000;

  tv.info = function() {
    document.write("TV색상 : "+tv.color+"  
");
    document.write("TV가격 : "+tv.price);
  }

  tv.info();
</script>
```

```
<script>
  var car = {
    color: "black",
    price: 50000000,
    info: function() {
      document.write("자동차 색상 : "+car.color+"  
");
      document.write("자동차 가격 : "+car.price);
    }
  }
  car.info();
</script>
```

객체는 여러개의 자료형을 한꺼번에 저장하는 자료형

배열은 객체를 기반으로 만들어짐 → 배열과 객체는 상당히 유사

차이점은,

배열은 요소에 인덱스로 접근하고 객체는 키를 사용 (아래에서 name, age, eat 등이 키)

```
<script>
  var person = {
    name: "무당선생",
    age: 47,
    eat: function(eatItem) {
      alert(this.age+"세 되신 "+this.name+"께서 "+eatItem+"를 드십니다.");
    }
  };
</script>
```

배열 대부분에 값 하나 하나를 요소라 한다.

객체 내부의 값 하나 하나를 속성(Property)이라고 한다.

객체의 속성 중에서 자료형이 함수인 속성을 메서드(Method)라고 한다.

함수

- 어떤 일련의 작업을 수행하는 여러 명령어의 집합
- 즉시 실행되지 않고 필요할 때마다 호출하여 사용

```
function 함수명() {  
    자바스크립트 코드;  
}
```

```
<script>  
    function myFnc() {  
        document.write("안녕하세요!<br>");  
        document.write("좋은 아침입니다.");  
    }  
</script>
```

익명함수

변수에 참조해서 사용

```
var myFnc = function() {  
    자바스크립트 코드;  
}
```

익명함수 방식은 호이스팅(hoisting)을 지원하지 않음

```
<script>  
    var count = 0;  
  
    function sayHello() {  
        count++;  
        document.write("hello"+count, "<br>");  
    }  
  
    var sayBye = function() {  
        count++;  
        document.write("bye"+count, "<br>");  
    }  
</script>
```

매개변수

함수 내부에 값을 전달할 때 사용하는 변수

```
<script>
    function myFnc(name, area) {
        document.write("안녕하세요. "+name+"입니다.", "<br>");
        document.write("제가 사는 곳은 "+area+"입니다.");
    }

    myFnc("무당선생", "일산");
</script>
```

arguments

매개변수 없이 함수에 전달된 값을 받는 방법

함수에서 return문의 역할

1. 함수의 결과값을 반환
2. 코드의 강제 종료

1. 함수의 결과값 반환

```
<script>
    function sum(num1, num2) {
        var result = num1+num2;
        return result;
    }

    var test = sum(10, 30);
    document.write(test);
</script>
```

2. 코드의 강제 종료