

# Javascript

## 변수와 네이밍 규칙 & 출력문

- 연산자 종류와 3항 연산자 -

```
<script>
//1. 자바스크립트 공부 처음 출력문////////
////////////////////////
/*
    alert("경고 창에 출력!");
    alert("경고 창에 출력!\n자바스크립트 경고창 줄바꿈\n제이쿼리는 라이브러리 프로그램이다.");
    console.log("콘솔 창에 출력!\n내용출력 확인!\n줄바꿈 메타문자");
*/

//변수 규칙
//네이밍 규칙 카멜케이스 기법 사용 예] navMain, goTop, mainSlide ...
//카멜케이스 기법이란? 중간에 연결되는 단어는 대문자를 사용합니다.
//첫 글자는 영문 소문자로 사용
//공백은 사용할 수 없습니다.
//중간은 숫자사용가능 예] a123, abc9, babo2
//변수와 연산자 = 사이는 되도록 공백을 띄운다. 권장사항
//변수 선언 끝은 반드시 ; (세미콜론 semi colon)으로 종결 합니다.
//변수에 _(언더바) 특수문자 사용가능 합니다.
//변수에 $(달러기호) 특수문자 사용가능 합니다.

//변수선언 예1]
//한 줄로 선언 가능합니다.
a = 20; b = 30;

//변수선언 예2]
//권장사항 : 변수는 줄을 바꾸어서 선언합니다.
a = 20;
b = 30;

//연산자
// = 연산자(산술) (등호) 대입연산자
// 예] a = 20;
// 해설] 변수 a = 정수 20을 대입(기억==저장)합니다.

// == 연산자(단순비교) : 참(true) 거짓(false) 결과
// b(결과 true) = 20 == '20';
// 인용부호는 안쪽이 작은따옴표이면 바깥쪽은 큰따옴표를 사용합니다.
// 인용부호는 안쪽이 큰따옴표이면 바깥쪽은 작은따옴표를 사용합니다.
b = 20 == '20';
```

```

    console.log( b + " = 20 == '20'" ); //+ 연결연산 변수 문자 연결
    console.log( b + ' = 20 == "20"' ); //+ 연결연산 문자 변수 연결

//  해설]
//  정수 20 과 '20'을 비교해서 같으면 논리값 true 를 변수 b에 저장합니다.
//  다르면 false를 변수 b에 저장합니다.

// ===연산자(논리값 까지 비교) : 참(true) 거짓(false) 결과
//  b(결과 false) = 20 === '20' ;
//  b = 20 === '20';
//  console.log( b + " = 20 === '20'" ); //+ 연결연산 변수 문자 연결
//  console.log( b + ' = 20 === "20"' ); //+ 연결연산 문자 변수 연결

//  해설]
//  === 3개로 표시된 비교문은 논리값 까지 비교하기 때문에
//  20은 정수이고 integer 으로 인식 함
//  '20'은 문자열 string 으로 인식 함
//  정수 20 과 30을 비교해서 같으면 논리값 true 를 변수 b에 저장합니다.
//  다르면 false를 변수 b에 저장합니다.


// + 연산자(산술) : 덧셈
//  예] a = 30 + 20;
//  해설] 변수 a = 30 + 20 덧셈 연산 결과를 대입(기억==저장)합니다.


// ++ 연산자(증가) : 1씩 증가
//  a = 0;
//  예] a++;
//  a = 0;
//  a++;
//  console.log(a + " = a++" );

//  해설]
//  a++; //반드시 1씩만 증가 1,2,3,4....
//  a+=1;
//  a=a+1;

//  a+=1; 1,2,3,4...
//  a+=2; 2,4,6,8...
//  a+=3; 3,6,9,12...

//  a=a+1; 1,2,3,4...
//  a=a+2; 2,4,6,8...
//  a=a+3; 3,6,9,12...

//  증감연산을 사용할 때에는 반드시 변수 a 에 초기 값은 선언 되어 있어야 합니다.
//  증감연산을 사용할 때에 변수 a 에 초기 값이 없으면 오류 메시지가 출력됩니다.
//  초기 값 변수 선언 예시] a = 0, a = 5, a = -1; 음수, 양수 모두 사용 가능합니다.

//  후위 연산 a++ 연산을 뒤(후)에 합니다.

```

```

// 전위 연산 ++a 연산을 앞(전)에 합니다.
// a = 5;
a = 5;
console.log(a++ + ' : 후위연산'); //연산결과 5 출력
console.log(a + ' : 연산 후에 증가 값 출력 '); //연산결과 6 출력

// 후위 연산 예문
// 예] console.log(a++); //초기 값 5만 출력 그리고 1증가합니다.
// 예] console.log(a); //증가된 후에 출력하면 6이 출력됩니다.

// a++; //5 줄이 바뀌면(증가 연산 실행 후)
// 예] console.log(a); //증가된 후에 출력하면 6이 출력됩니다.

// 전위 연산 예문
// 예] console.log(++a); //초기 값 5+1연산을 먼저하고 6을 출력합니다.

// ++a //초기 값 5+1연산을 먼저하고 6
// 예] console.log(a); //을 6출력합니다.
a=5;
console.log(++a + ' : 전위연산');

// - 연산자 : 뺄셈
// 예] a = 30 - 20;
// 해설] 변수 a = 30 - 20 뺄셈 연산 결과를 대입(기억==저장)합니다.

// -- 연산자(감산)
// a = 10;
// 예] a--;
// 해설]
// a--; //반드시 1씩만 감소 10, 9, 8...
// a-=1;
// a=a-1;

// a-=1; 10, 9, 8...
// a-=2; 10, 8, 6...
// a-=3; 10, 7, 4...

// a=a-1; 10, 9, 8...
// a=a-2; 10, 8, 6...
// a=a-3; 10, 7, 4...

a = 10;
console.log(a-- + ' a--'); //10
console.log(a-- + ' a--'); //9
console.log(a-- + ' a--'); //8
console.log(a-- + ' a--'); //7
console.log(a-- + ' a--'); //6

```

```
// * 연산자 : 곱셈
// 예] a = 30 * 20;
// 해설] 변수 a = 30 * 20 곱셈 연산 결과를 대입(기억==저장)합니다.
```

```
// ** 연산자(거듭제곱) 2의 2승
// 예] a (4) = 2**2 , 8 = 2**3 ,16 = 2**4
// 예] a = 3**2 , 27 = 3**3 ,81 = 3**4
```

```
// / 연산자 : 나눗셈
// 예] a = 30 / 20;
// 해설] 변수 a = 30 / 20 나눗셈 연산 결과를 대입(기억==저장)합니다.
```

```
// % 연산자 : 나머지
// 예] a = 30 % 20;
// 해설] 변수 a = 30 % 20 나머지 연산 결과를 대입(기억==저장)합니다.
```

```
a = 11 % 2;
console.log('나머지 연산 %');
console.log(a + ' = 11 % 2');
```

```
// 3항 연산자
// 비교문(1항) ? 참(2항) : 거짓(3항)
// 예문] 정수 5를 2로 나누어서 나머지가 0이면 짝수 긍정문
// 아니면 홀수 출력
// 3항연산자 사용 출력 프로그래밍
```

```
a = 5 % 2;
b = (a === 0 ? '짝수' : '홀수');
console.log(b + " = a == 0 ? '짝수' : '홀수' ");
```

```
//연산의 우선 순위
a = 3 + 5 * 7 - 4 / 2**2;
//1. 4=2**2
//2. 35=5*7
//3. 1=4/4
//4. 38=3+35(5*7)
//5. 37=38-1(4/4)
console.log(a + ' = 3 + 5 * 7 - 4 / 2**2;' );
```

```
</script>
```