

# Part1-3(3장).

## 자바스크립트 연산자2

(논리연산자, 비트, 삼항 연산자, 기타연산자)

# I. 논리연산자

- 논리 and : &&
- 논리 or : ||

A	B	A and(&&) B	A or(  ) B
T	T	F	T
T	F	F	F
F	T	F	F
F	F	T	F

## Javascript3-1.html

```
<html>
<head><title> </title>
<script language="javascript">
<!--
a = 'A';
b = 'B';

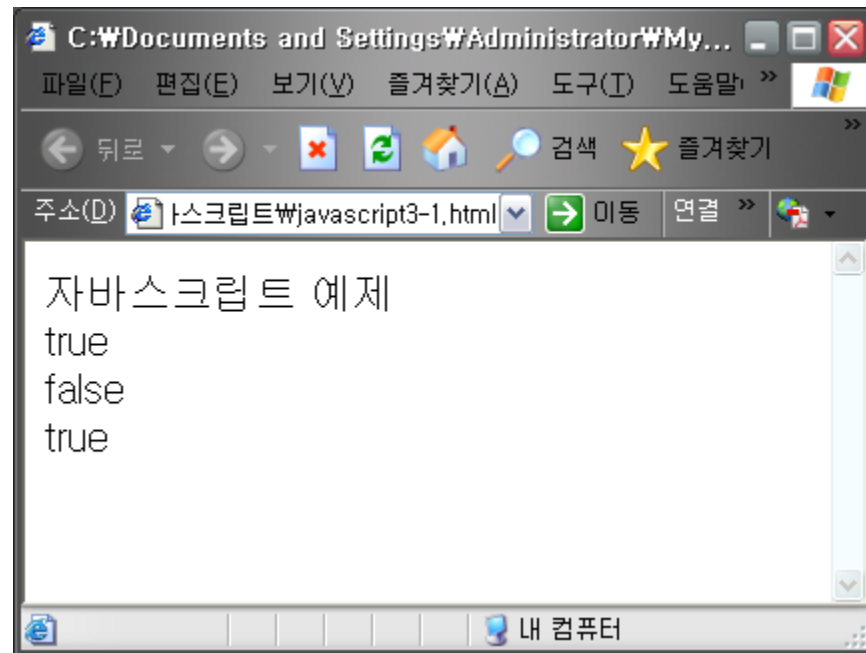
document.write("자바스크립트 예제"+"<br>");
c = !2 || 3 && !0
document.write(c +"<br>");

c = a < b && a == b
document.write(c +"<br>");

c = a < b || a == b
document.write(c +"<br>");

//-->
</script>
</head>
<body>
</body>
</html>
```

## 실행결과



## 2. 비트연산자

### (1) 쉬프트 연산자

- 비트를 이동시키는 연산자로 오른쪽으로의 이동과 왼쪽으로의 이동하는 두 가지 종류의 방식이 있다.

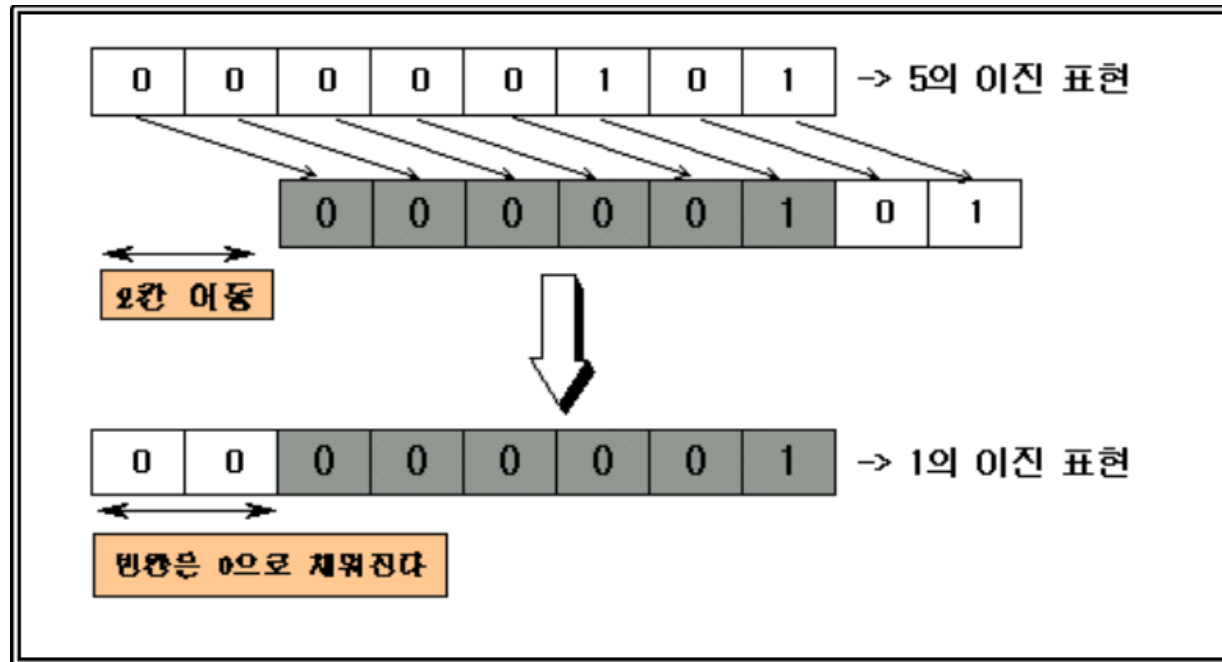
연산자	뜻	예	의미
>>	오른쪽으로 이동	5 >> 2	5의 이진수표현에서 오른쪽으로 2칸 쉬프트
<<	왼쪽으로 이동	5 << 2	5의 이진수표현에서 왼쪽으로 2칸 쉬프트

## Javascript3-2.html

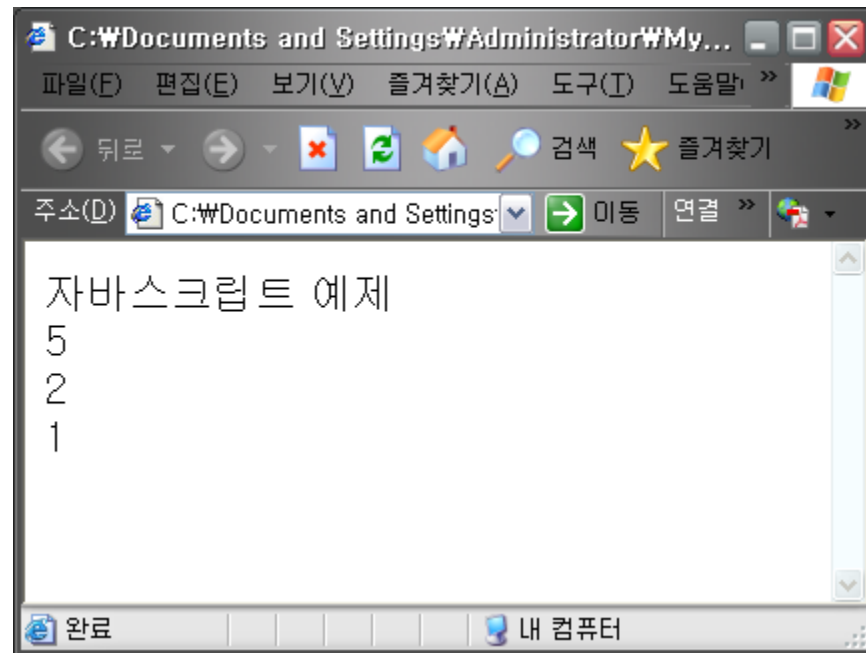
```
<html>
<head><title> </title>
<script language="javascript">
<!--
a = 5;
b = 2;
c = a >> b;

document.write("자바스크립트 예제"+"<br>");
document.write(a + "<br>");
document.write(b + "<br>");
document.write(c + "<br>");
//-->
</script>
</head>
<body>
</body>
</html>
```

## 쉬프트연산자 이동원리



## 실행결과





## 2. 비트연산자

### (2) 비트 논리연산자

- **&(비트 AND), |(비트 OR), ^(비트 XOR), ~(비트 NOT)** 등이 있다.

비트논리 and(&)

x	y	x & y	의미
0	0	0	거짓
0	1	0	거짓
1	0	0	거짓
1	1	1	참

## 비트논리 or(|)

x	y	$x \mid y$	의미
0	0	0	거짓
0	1	1	참
1	0	1	참
1	1	1	참

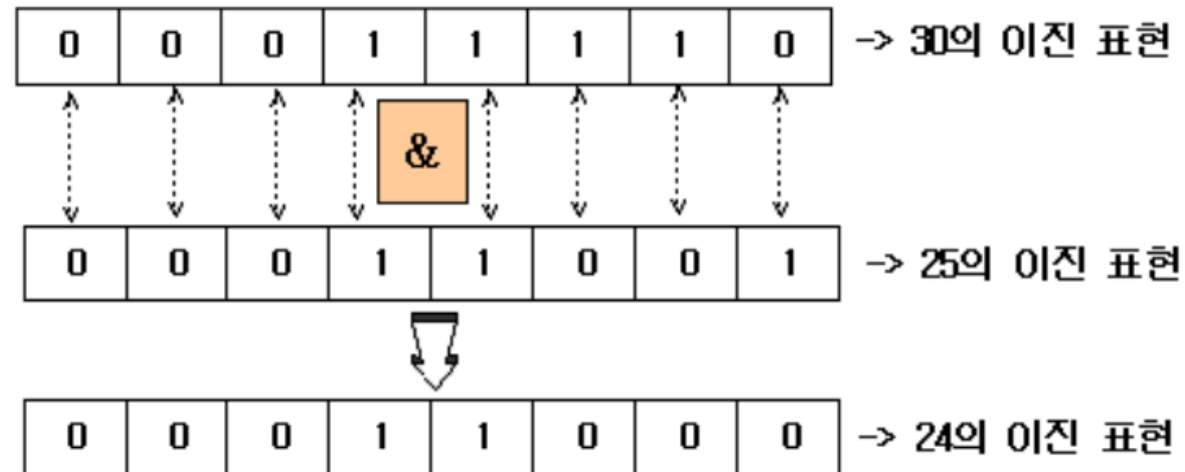
## 비트논리 xor(^)

x	y	$x \wedge y$	의미
0	0	0	거짓
0	1	1	참
1	0	1	참
1	1	0	거짓

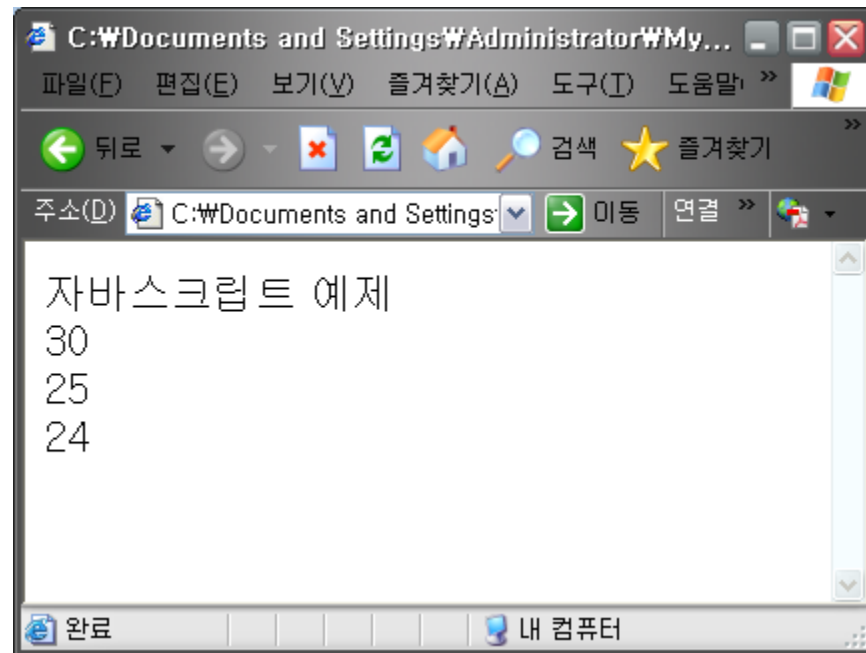
```
<html>
<head><title> </title>
<script language="javascript">
<!--
a = 30;
b = 25;
c = a & b;

document.write("자바스크립트 예제"+"<br>");
document.write(a + "<br>");
document.write(b + "<br>");
document.write(c + "<br>");
//-->
</script>
</head>
<body>
</body>
</html>
```

## 비트논리연산자 원리



## 실행결과



## 2. 비트연산자

### (1) 비트 not 연산자

- ~ (틸더)
- 이진표현의 비트를 만대로 만든다

```
<html>
<head><title> </title>
<script language="javascript">
<!--
a = -1;
b = -4;
c = ~a ;
document.write("자바스크립트 예제"+"<br>");
document.write(a + "<br>");
document.write(b + "<br>");
document.write(c + "<br>");
c = ~b ;
document.write(c + "<br>");
//-->
</script>
</head>
<body>
</body>
</html>
```

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

→ -1의 이진 표현



0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

→ 0의 이진 표현



1	1	1	1	1	1	0	0
---	---	---	---	---	---	---	---

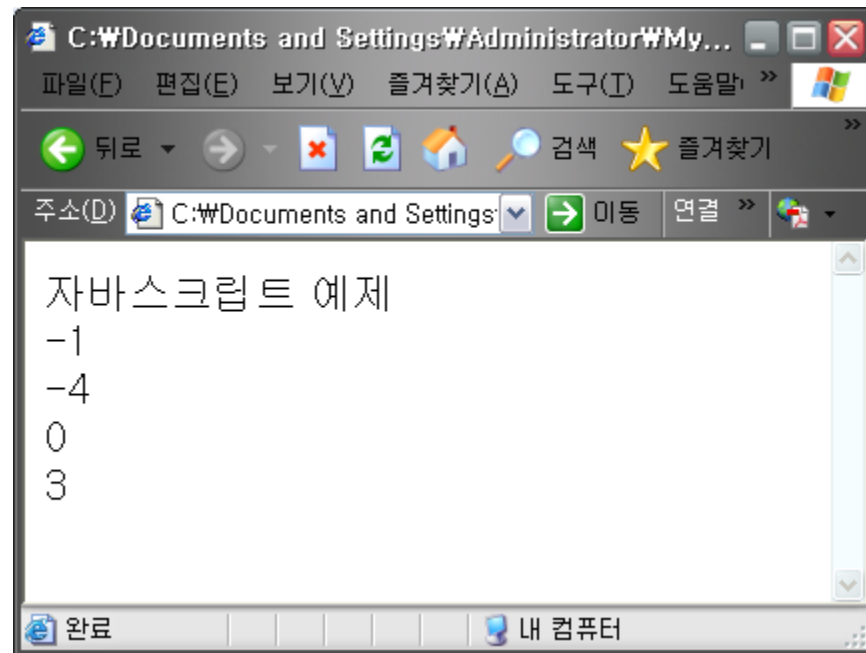
→ -4의 이진 표현



0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

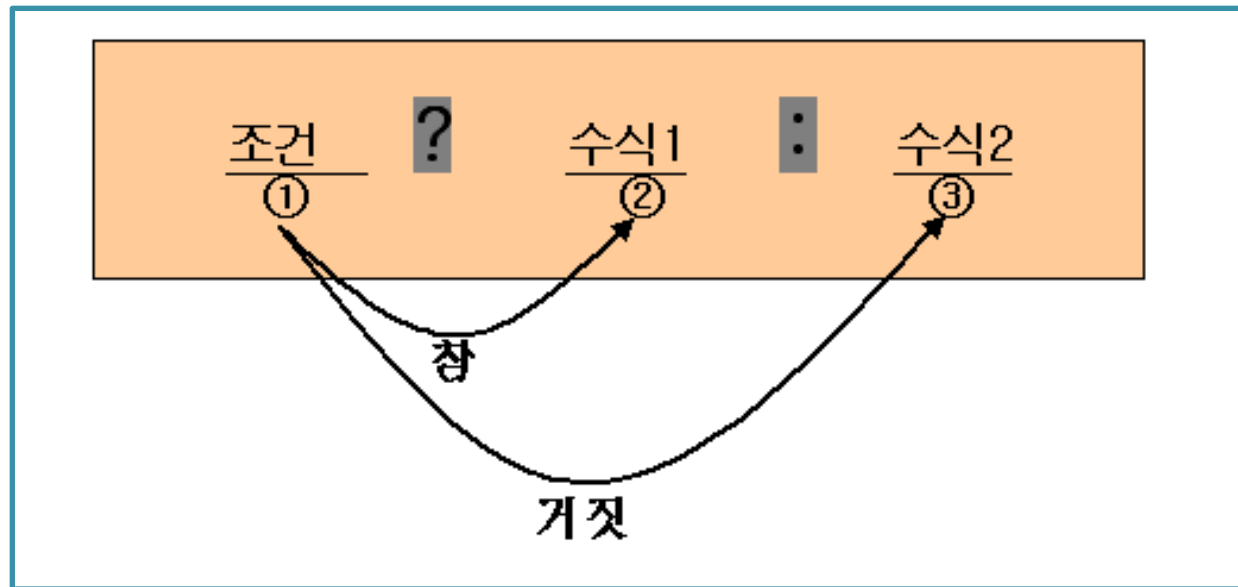
→ 3의 이진 표현

## 실행결과



### 3. 3항연산자

- 항이 3개가 필요하기 때문에 3항연산자라고 함.
- 연산 원리는 그림과 같음

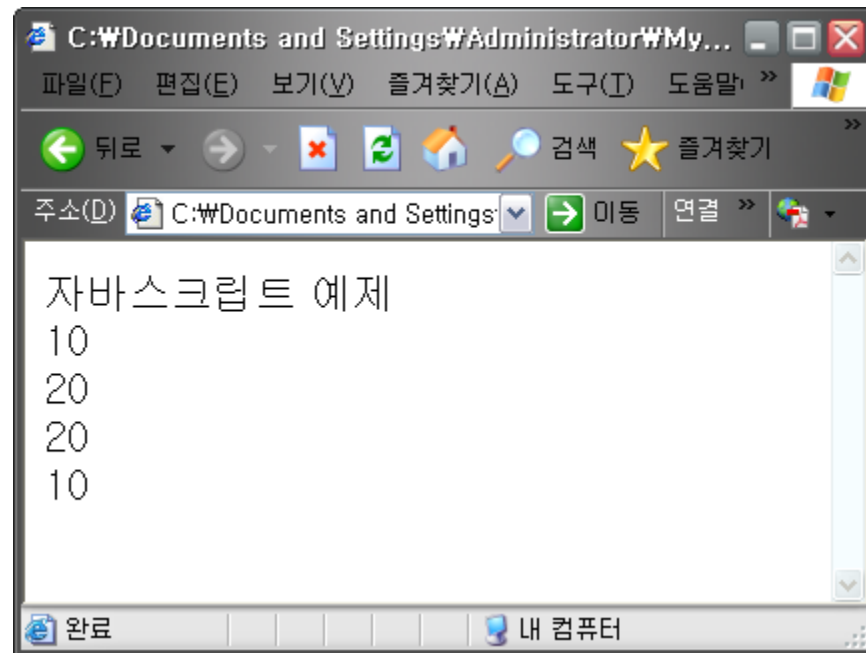


```
<html>
<head><title> </title>
<script language="javascript">
<!--
x = 10;
y = 20;
z = ( x > y ? x : y);

document.write("자바스크립트 예제"+"<br>");
document.write(x + "<br>");
document.write(y + "<br>");
document.write(z + "<br>");

z = ( y > x ? x : y);
document.write(z + "<br>");
//-->
</script>
</head>
<body>
</body>
</html>
```

## 실행결과



## 4. 기타 연산자(문자연결연산자)

- + 기호를 사용함
- 예) A + "res"(A에 10 이 저장된 경우)
- 10res가 출력됨

## Javascript3-6.html

```
<html>
<head><title> </title>
<script language="javascript">
<!--
A = 10;

document.write("자바스크립트 예제"+"<br>");
document.write( A + "res");

//-->
</script>
</head>
<body>
</body>
</html>
```

