산술 연산자

✔ + 연산자

- 평가 결과를 더함
- 평가 결과 연결

```
var two = '2';
var value = 1 + two;
console.log(value); // 12
console.log(typeof value); // string
```

▼ 한 쪽이라도 평가 결과가 number 타입이 아니면 평가 결과를 더하지 않고, **연결**한다.

✔ 숫자로 변환

- 연산하기 전에 우선 숫자로 변환

  -

Aa 값 타입	변환 값
Undefined	NaN
Null	+0
Boolean	True: 1, False: 0
Number	변환 전/후 같음
String	값이 숫자이면 숫자로 연산 단, 더하기(+)는 연결

✔ undefined 더하기

```
var value;
console.log(10 + value); // NaN
```

▼ value 값은 undefined, 10과 undefined를 더하면 NaN(Not-a-Number) 값 출력

✔ 1 또는 0으로 변환

```
console.log(10 + null); // 10
console.log(10 + true); // 11
console.log(10 + false); // 10
```

▼ null은 0으로 변환. True : 1, False: 0

✔ 숫자를 문자열로 작성

```
console.log(10 + '123'); // 10123
console.log(123 - '23'); // 100
```

▼ 더하기는 값이 숫자라도 타입이 string이면 문자열로 연결하지만, -, *, /는 숫자로 변환하여 연산

✔- 연산자

- string 타입이지만, 값이 숫자이면 Number 타입으로 변환하여 계산

```
console.log('135' - 2); // 133
```

✔* 연산자

- 숫자 값으로 변환할 수 있으면 변환하여 곱함
- NaN 변환
  - 양쪽의 평가 결과가 하나라도 숫자가 아닐 때

```
console.log(10 * '20'); // 200
console.log(10 * true); // 10
console.log(10 * false); // 0
console.log(10 * null); // 0
console.log(10 * 'A'); // NaN
```

▼ 문자 타입이라도 값이 숫자이면 숫자타입으로 변환한다.

- 소수 값이 생기는 경우 처리

```
console.log(2.3 * 3); // 6.8999999999999995
console.log(2.3 * 10 * 3 / 10); // 6.9
```

▼ 6.9로 출력되지 않는다. 이 결과는 정상이며, IEEE 754 부동 소수점 처리 때문이다.

▼ 대응 방법 : 실수를 정수로 변환하여 값을 구하고, 다시 정수를 실수로 반환한다.

✔ / 연산자

- NaN 변환
  - 양쪽의 평가 결과가 하나라도 숫자가 아닐 때
  - 분모, 분자가 모두 0일 때
- 분모, 분자가 0일 때
  - 분모가 0이면 Infinity 반환
  - 분자가 0이면 0 반환

✔ % 연산자

```
console.log(5 % 2.5); // 0
console.log(5 % 2.3); // 0.400000000000036
console.log(5 * 10 - (2 * 2.3 * 10) / 10); // 0.4
```

▼ 5를 2.5로 나누면 나머지가 0, 소수 끝에 36이 있는 것은 IEEE 754 부동 소수점 처리 때문이다.

▼ 실수를 정수로 변환하여 연산하고 다시 정수를 실수로 변환