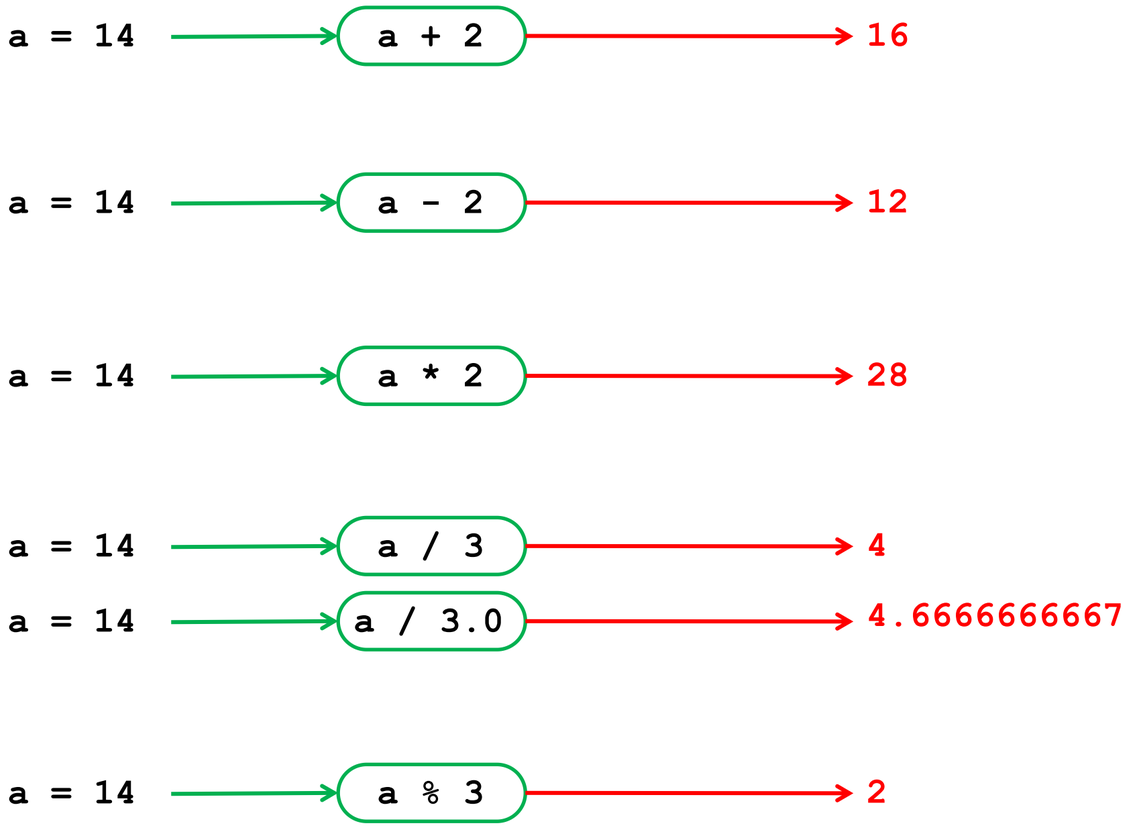
CORE PROGRAMMING

OPERATOR

- Toán tử arithmetic.
- Toán tử unary.
- Toán tử shift.
- Toán tử bitwise.
- Toán tử logical.
- Sự khác nhau giữa toán tử & và &&, | và ||.
- Toán tử assignment.
- Toán tử relational.
- Toán tử ternary.

Toán tử arithmetic.

- ***Arithmetic (toán tử số học) thực hiện các phép toán cơ bản trong biểu thức số học, gồm các toán tử sau:***
  - ***+: Phép cộng (có thể dùng để nối chuỗi).***
  - ***-: Phép trừ.***
  - ****: Phép nhân.***
  - ***/: Phép chia và chia lấy nguyên.***
  - ***%: Phép chia lấy dư.***

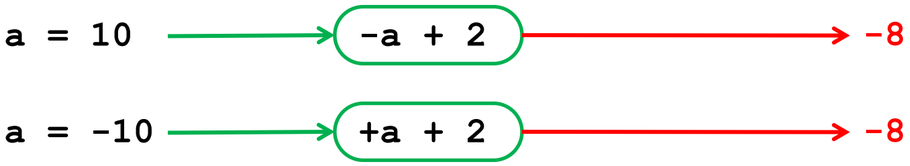


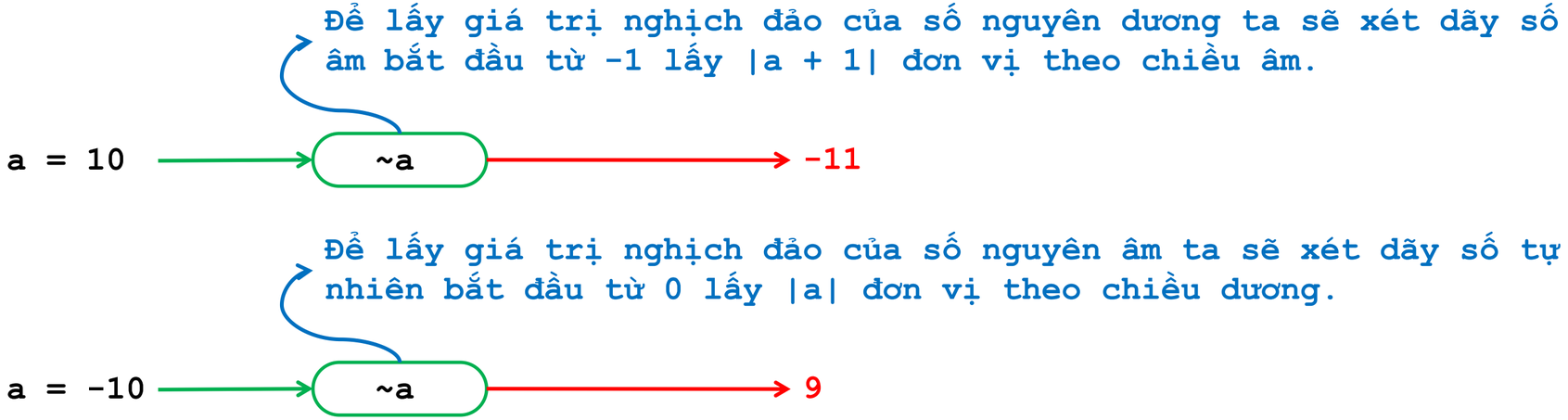
Toán tử unary.

- ***Unary (toán tử một ngôi) có hai dạng là postfix và prefix:***
  - ***Prefix (tiền tố): Được thực hiện trước khi phép toán giữa hai số diễn ra, gồm các toán tử sau:***
    - ***+expression: Lấy dương biểu thức số.***
    - ***-expression: Lấy âm biểu thức số.***
    - ***~expression: Lấy nghịch đảo biểu thức số nguyên.***
    - ***!expression: Lấy phủ định biểu thức logic.***
    - ***++expression: Tăng một đơn vị số.***
    - ***--expression: Giảm một đơn vị số.***
  - ***Postfix (hậu tố): Được thực hiện sau khi phép toán giữa hai số diễn ra, gồm các toán tử sau:***
    - ***expression++: Tăng một đơn vị số.***
    - ***expression--: Giảm một đơn vị số.***

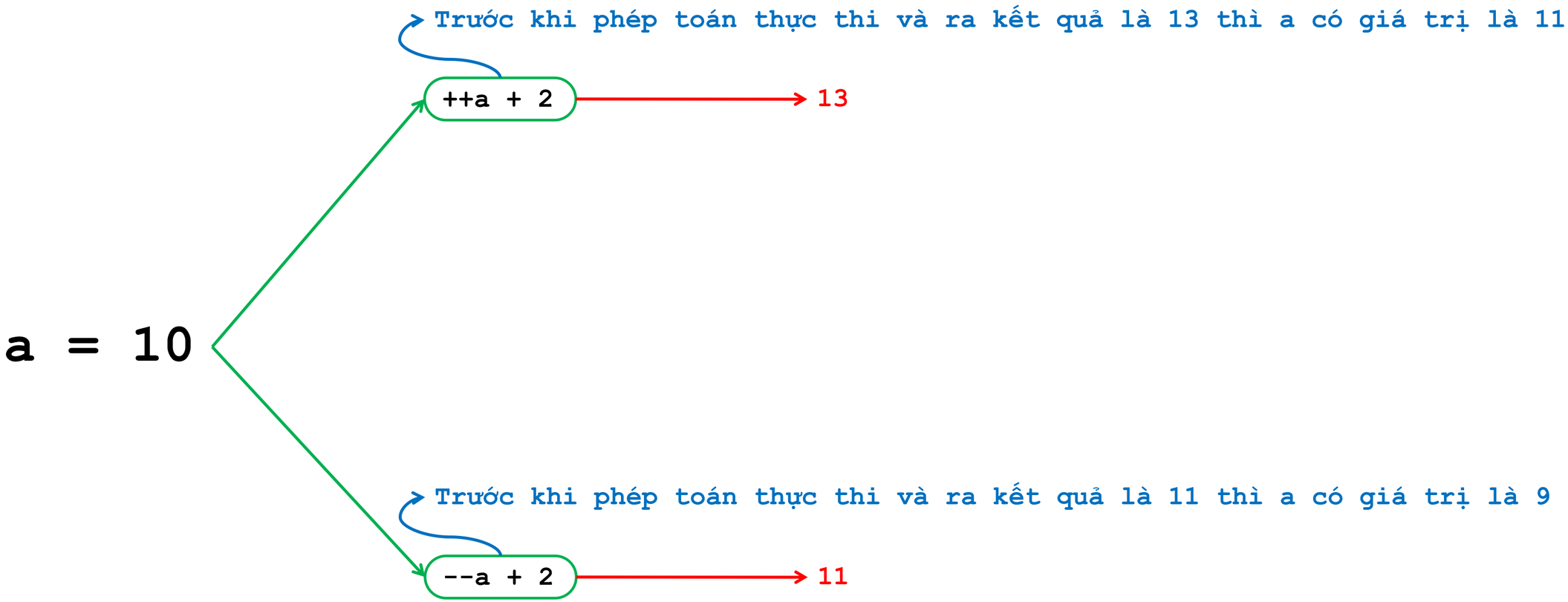
Toán tử unary.

- ***Prefix.***

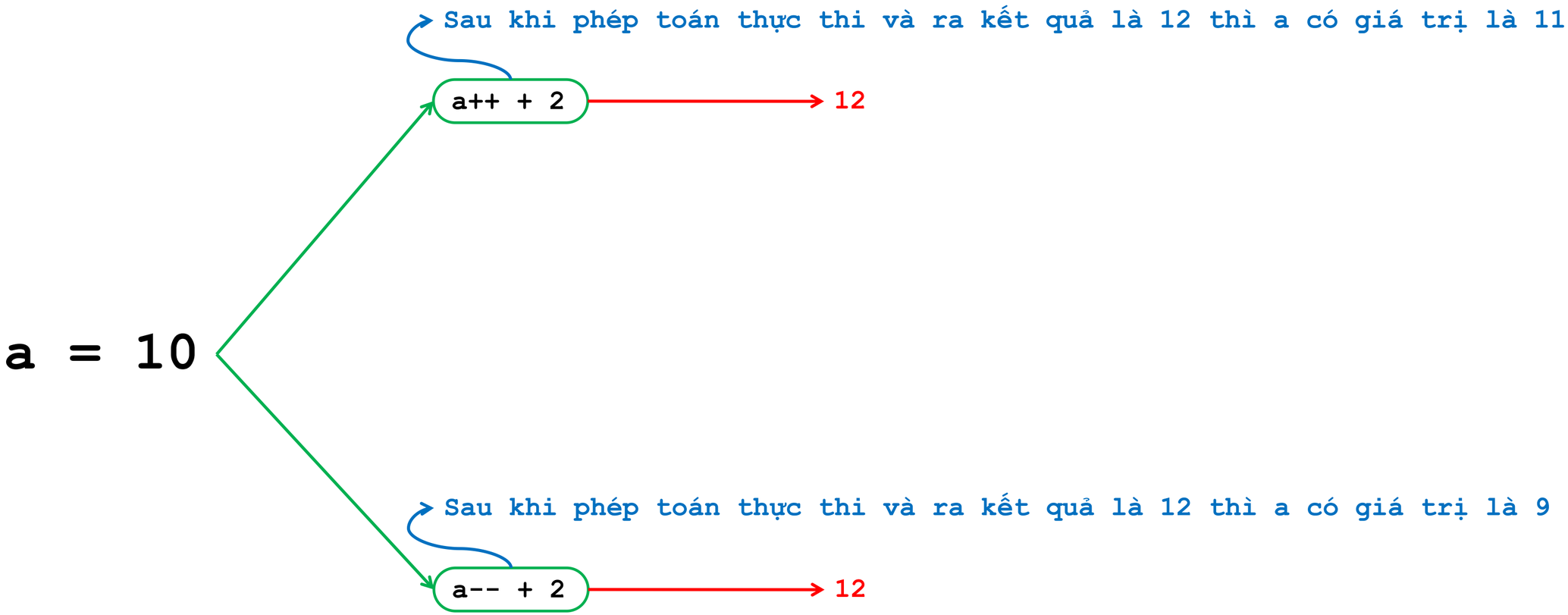




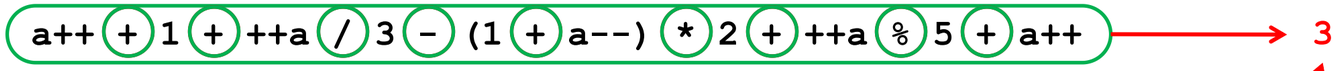
- ***Prefix.***


Trước khi phép toán thực thi và ra kết quả là 13 thì a có giá trị là 11
++a + 2
13
a = 10
Trước khi phép toán thực thi và ra kết quả là 11 thì a có giá trị là 9
--a + 2
11

- ***Postfix.***


Sau khi phép toán thực thi và ra kết quả là 12 thì a có giá trị là 11
a++ + 2
12
a = 10
Sau khi phép toán thực thi và ra kết quả là 12 thì a có giá trị là 9
a-- + 2
12

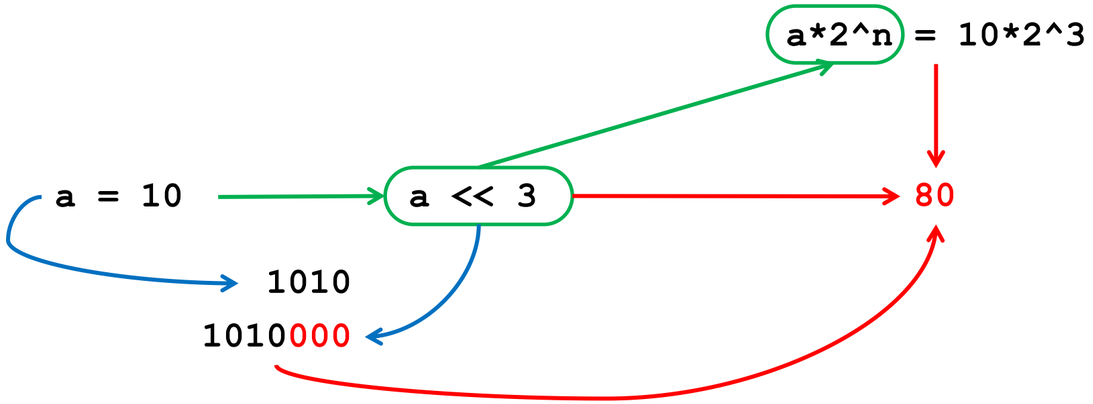
- ***Biểu thức trộn lẫn postfix và prefix.***

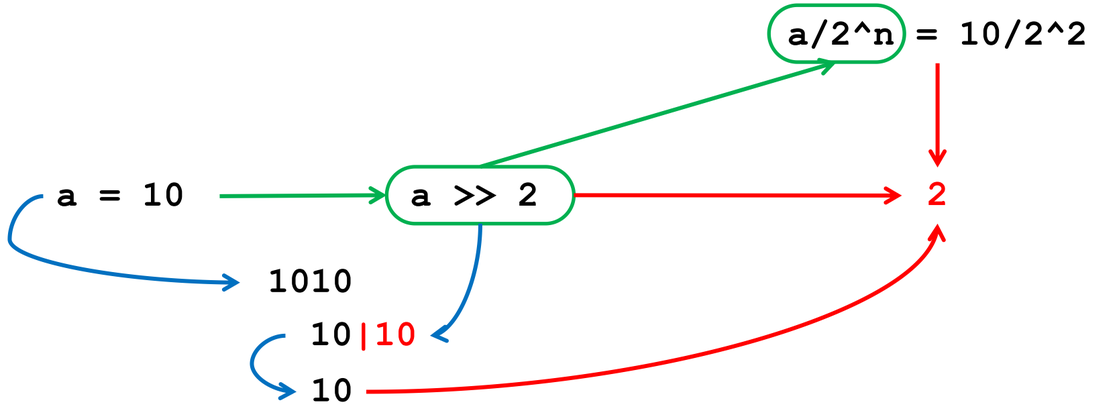



```
a = 10
a++ + 1 + ++a / 3 - (1 + a--) * 2 + ++a % 5 + a++
10  + 1 + ++a / 3 - (1 + a--) * 2 + ++a % 5 + a++
a = 11
a = 12
10  + 1 + 12  / 3 - (1 + a--) * 2 + ++a % 5 + a++
10  + 1 + 12  / 3 - (1 + 12 ) * 2 + ++a % 5 + a++
a = 11
a = 12
10  + 1 + 12  / 3 - (1 + 12 ) * 2 + 12  % 5 + a++
10  + 1 + 12  / 3 - (1 + 12 ) * 2 + 12  % 5 + 12
a = 13
```

Toán tử shift.

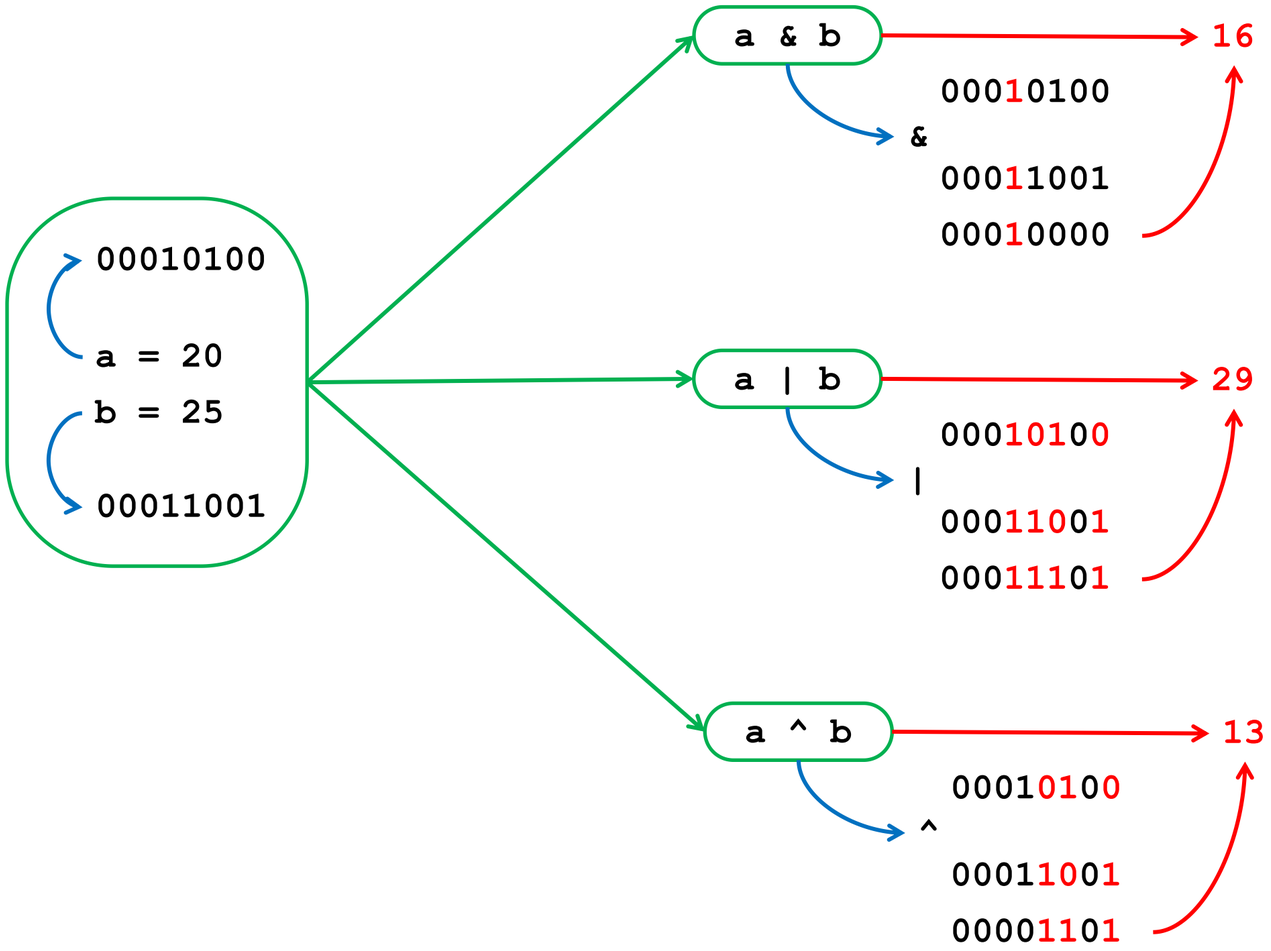
- ***Shift (toán tử dịch bit) thực hiện các phép toán dịch trái hay phải ở dạng bit của số nguyên, gồm các toán tử sau:***
  - ***<<(n): Dịch trái n bit.***
  - ***>>(n): Dịch phải n bit.***





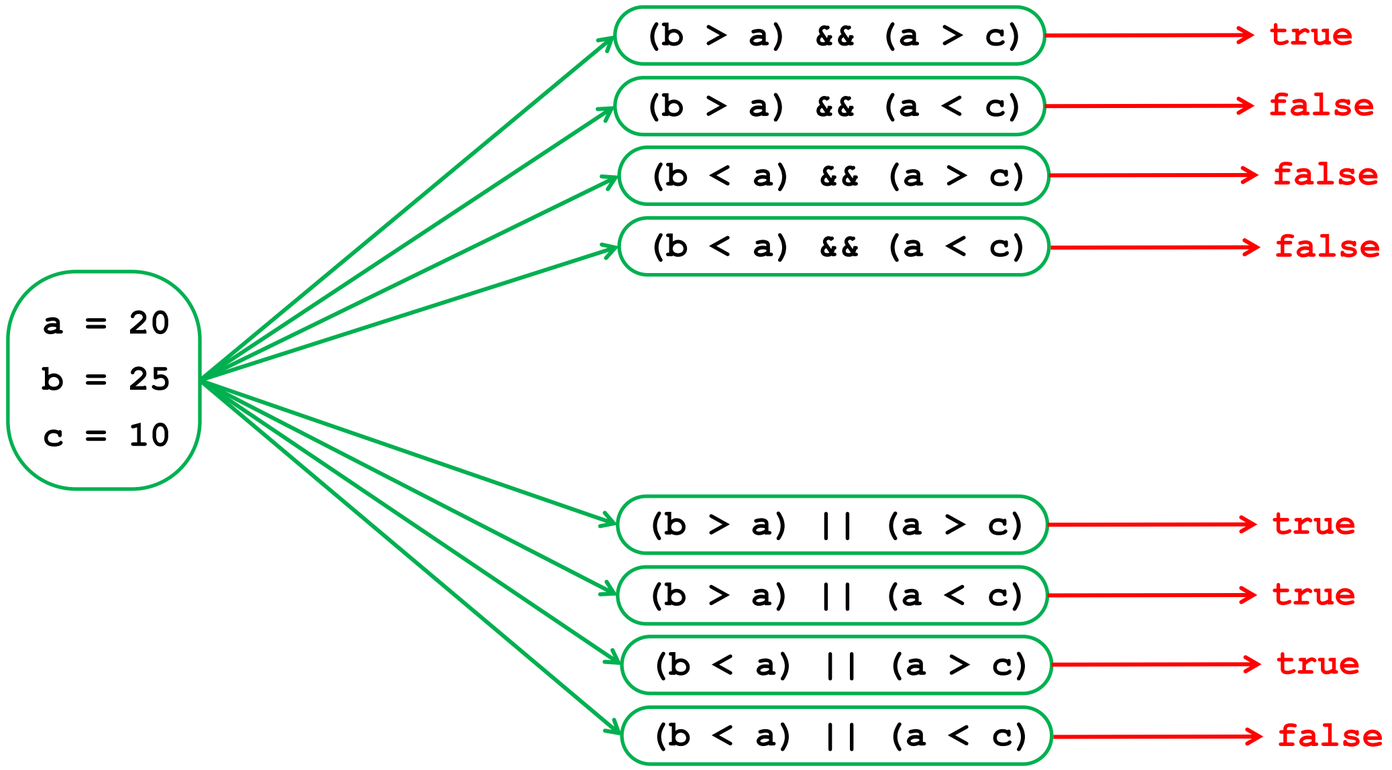
Toán tử bitwise.

- ***Bitwise (toán tử bit) thực hiện các phép toán bit giữa các số nguyên, gồm các toán tử sau:***
  - ***&: Nhân bit. Trong đó, 1 & 1 = 1, còn lại đều là 0.***
  - ***|: Cộng bit. Trong đó, 0 | 0 = 0, còn lại đều là 1.***
  - ***^: Loại trừ bit giống. Trong đó, 1 ^ 0 = 1 hoặc 0 ^ 1 = 1, còn lại đều là 0.***



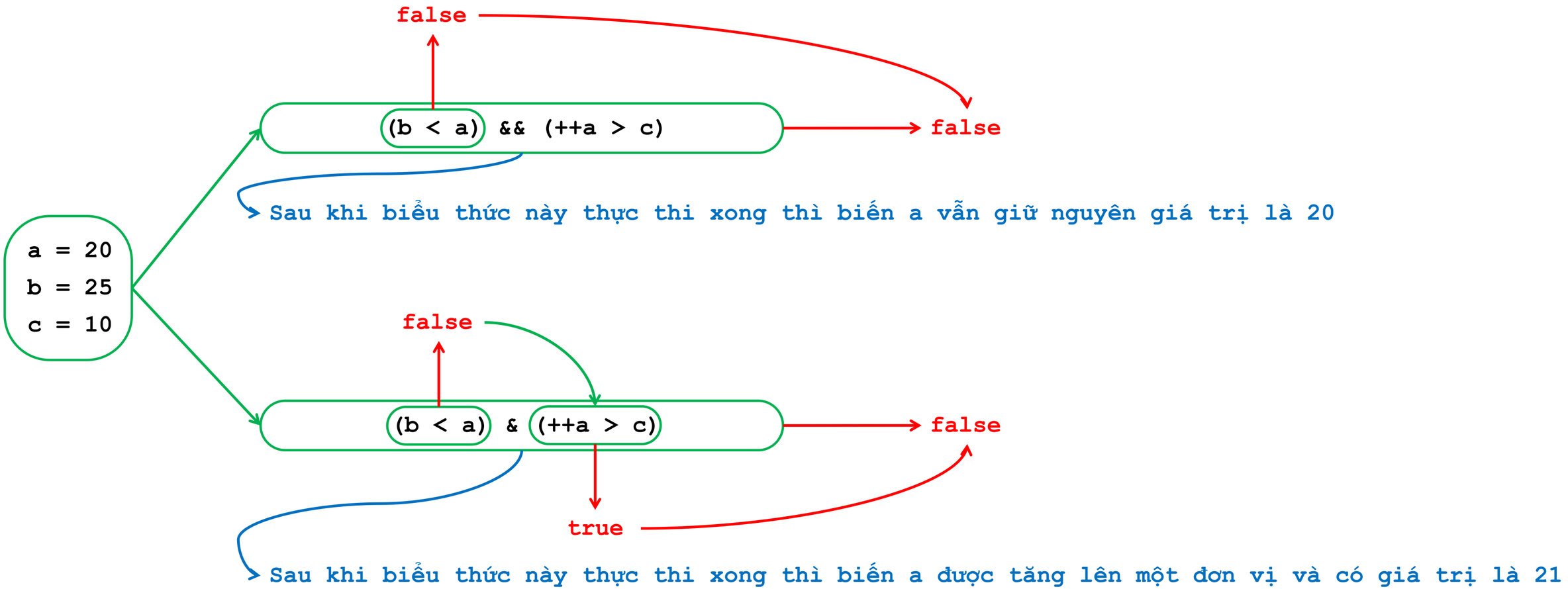
Toán tử logical.

- ***Logical (toán tử logic) thực hiện các phép toán điều kiện trong biểu thức điều kiện, gồm các toán tử sau:***
  - ***&&: Giao hai biểu thức điều kiện. Trong đó true && true = true, còn lại đều là false.***
  - ***||: Hợp hai biểu thức điều kiện. Trong đó false || false = false, còn lại đều là true.***



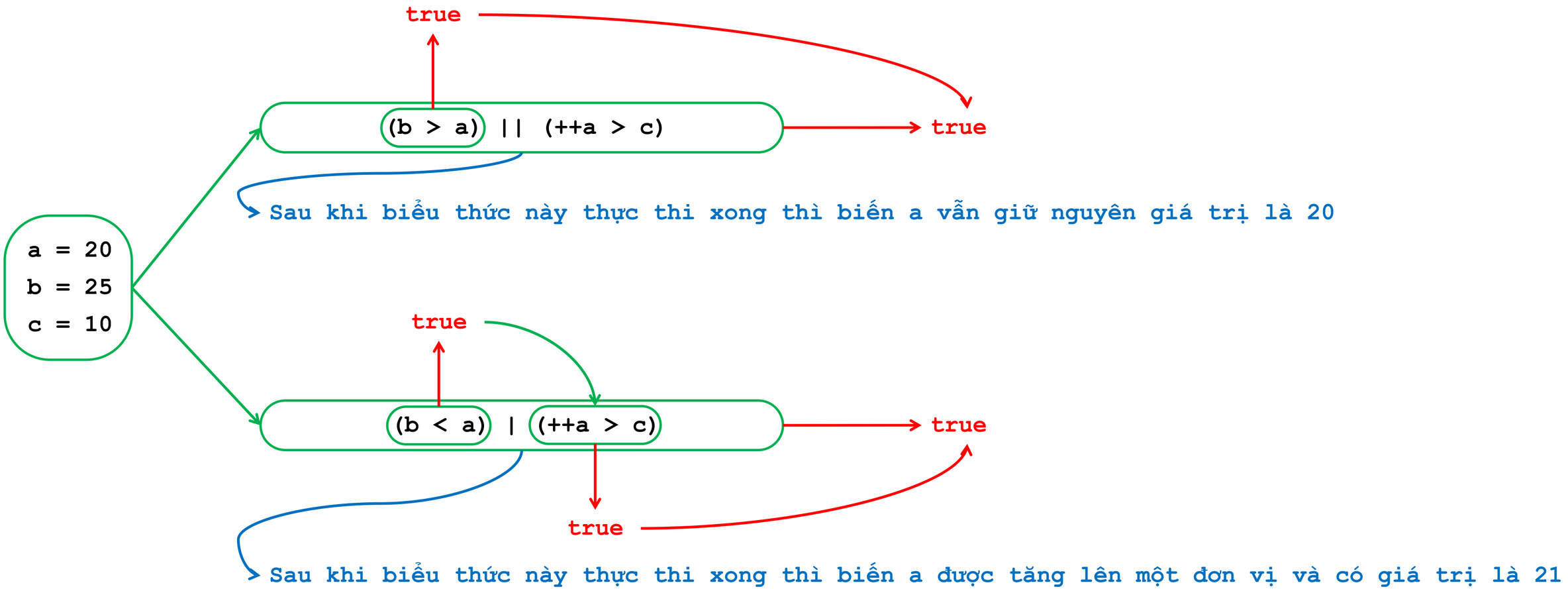
Sự khác nhau giữa toán tử & và &&, | và ||.

- ***Trong biểu thức điều kiện ngoài việc sử dụng các toán tử && để giao hai điều kiện thì ta cũng có thể sử dụng toán tử & để thay thế.***
- ***Toán tử && sẽ bỏ qua điều kiện thứ hai nếu điều kiện đầu tiên sai.***
- ***Toán tử & vẫn sẽ thực thi qua điều kiện thứ hai nếu điều kiện đầu tiên sai.***



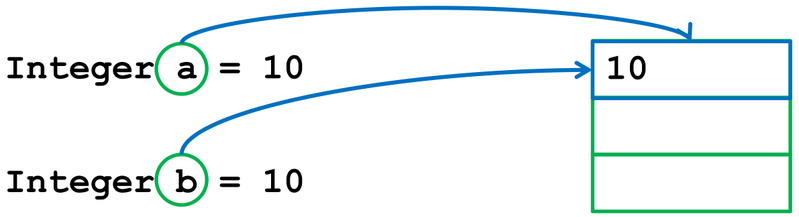
Sự khác nhau giữa toán tử & và &&, | và ||.

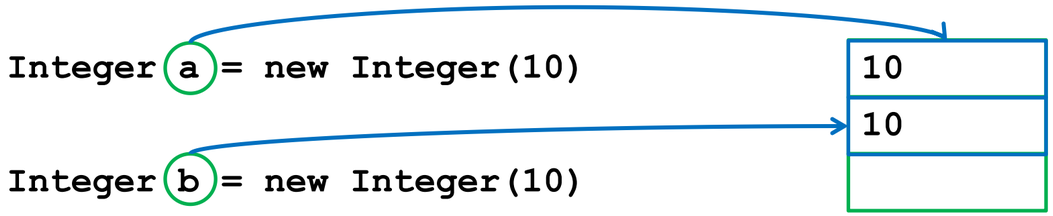
- ***Trong biểu thức điều kiện ngoài việc sử dụng các toán tử || để hợp hai điều kiện thì ta cũng có thể sử dụng toán tử | để thay thế.***
- ***Toán tử || sẽ bỏ qua điều kiện thứ hai nếu điều kiện đầu tiên đúng.***
- ***Toán tử | vẫn sẽ thực thi qua điều kiện thứ hai nếu điều kiện đầu tiên đúng.***



Toán tử assignment.

- ***Assignment (toán tử gán) thực hiện các phép gán cho biến nói chung, gồm các toán tử sau:***
  - ***=: Tham chiếu biến đến địa chỉ ô nhớ lưu trữ giá trị.***
  - ***+=(n): Ví dụ a = a + 10 => a += 10 (có thể dùng để nối chuỗi).***
  - ***-=(n): Ví dụ a = a - 10 => a -= 10.***
  - ****=(n): Ví dụ a = a * 10 => a *= 10.***
  - ***/=(n): Ví dụ a = a / 10 => a /= 10.***
  - ***%=(n): Ví dụ a = a % 10 => a %= 10.***
  - ***&=(n): Ví dụ a = a & 10 => a &= 10.***
  - ***|=(n): Ví dụ a = a | 10 => a |= 10.***
  - ***^=(n): Ví dụ a = a ^ 10 => a ^= 10.***
  - ***<<=(n): Ví dụ a = a << 10 => a <<= 10.***
  - ***>>=(n): Ví dụ a = a >> 10 => a >>= 10.***





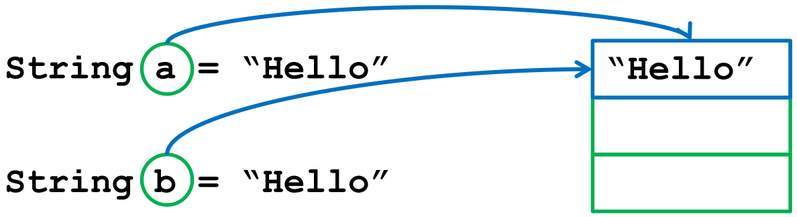
Toán tử relational.

- ***Relational (toán tử quan hệ) thực hiện các phép toán so sánh, gồm các toán tử sau:***
  - ***<: So sánh bé hơn về giá trị số.***
  - ***>: So sánh lớn hơn về giá trị số.***
  - ***<=: So sánh bé hơn hoặc bằng về giá trị số.***
  - ***>=: So sánh lớn hơn hoặc bằng về giá trị số.***
  - ***Từ khóa instanceof: So sánh quan hệ kế thừa giữa các đối tượng.***
  - ***==: So sánh bằng về địa chỉ.***
  - ***!=: So sánh khác về địa chỉ.***

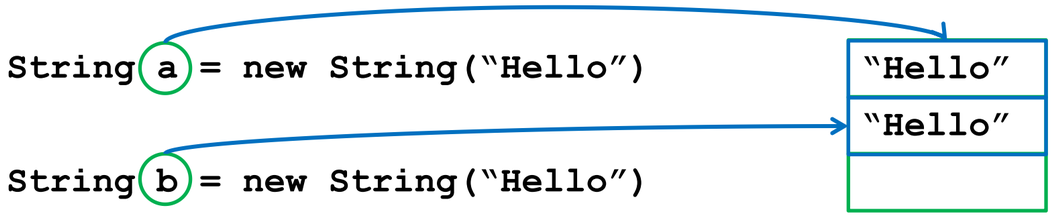
```
10 > 14   ──→ false
10 >= 14  ──→ false
10 < 14   ──→ true
10 <= 14  ──→ true
```

```
Object obj = new Integer(10)
```

```
obj instanceof String  ──→ false
obj instanceof Integer ──→ true
```




```
a == b  ──→ true
a != b  ──→ false
```




```
a == b  ──→ false
a != b  ──→ true
```

Toán tử ternary.

- ***Ternary (toán tử ba ngôi) thực hiện câu lệnh điều kiện đơn giản và trả về giá trị ứng với diều diện đúng, toán tử có dạng: điều kiện ? giá trị thứ nhất : giá trị thứ hai.***
- ***Khi điều kiện đúng thì giá trị thứ nhất sẽ được trả về, điều kiện sai thì giá trị thứ hai sẽ được trả về.***