Trong JavaScript, tất cả các tham số của hàm mặc định được truyền như là giá trị. Điều này có nghĩa là JavaScript sao chép giá trị của các biến mà bạn truyền tới hàm vào biến cục bộ. Bất cứ sự thay đổi nào mà bạn thực hiện trên biến cục bộ trong hàm không làm ảnh hướng tới tham số mà bạn truyền vào. Một cách khác, sự thay đổi của các tham số không ảnh hưởng ra bên ngoài của hàm.

Nếu tham số truyền vào hàm theo cách truyền tham chiến, điều này sẽ làm thay đổi các biến mà bạn truyền vào hàm, tức là có sự ảnh hưởng ở bên ngoài hàm.

Truyền tham trị (Passing by value of primitives values)

Primitives values là các kiểu dữ liệu có sẵn ví dụ: kiểu số nguyên, kiểu số thực, kiểu ký tự.

Hãy xem ví dụ sau.

function square(x) {

    x = x \* x;

}

let y = 10;

square(y);

console.log(y); // -- no change

Cách làm việc của đoạn mã lệnh như sau. Đầu tiên, định nghĩa hàm square() có một tham số truyền vào là x. Hàm làm nhiệm vụ thay đổi giá trị của tham số x.

Tiếp theo, khai báo và khởi tạo biến y và gán cho y giá trị 10.



Sau đó, truyền y vào hàm square(). Khi truyền biến y vào hàm square(), JavaScript sao chép giá trị của y sang biến x.

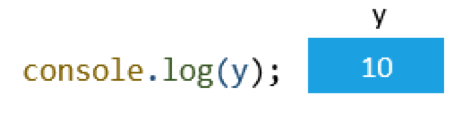


Tiếp theo, hàm square thay đổi biến x. Tuy nhiên,

Nó sẽ không làm thay đổi giá trị của biến y bởi vì x và y là hai biến khác nhau.



Cuối cùng, giá trị của y không bị thay đổi sau khi sau khi hàm square() được gọi.



Nếu sử dụng cách truyền tham chiếu, giá trị của biến y sẽ được thay đổi thành 100.

Truyền tham chiếu (Passing by value of references)

Hãy xem ví dụ sau:

function turnOn(machine) {

    machine.isOn = true;

}

let computer = {

    isOn: false

};

turnOn(computer);

console.log(computer.isOn); // true;

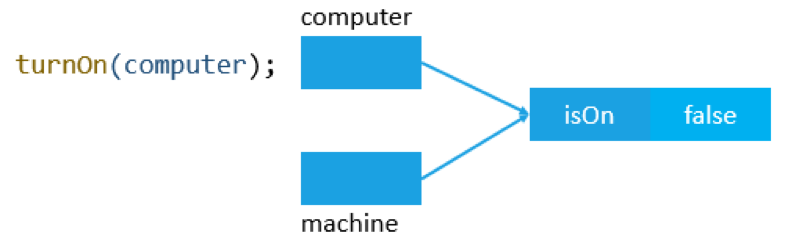
Cách làm việc của đoạn mã lệnh trên

Đầu tiên, định nghĩa hàm turnOn() nhận tham số đầu vào là một đối tượng machine. Hàm này sẽ thiết đặt thuộc tính isOn của đối tượng machine là true.

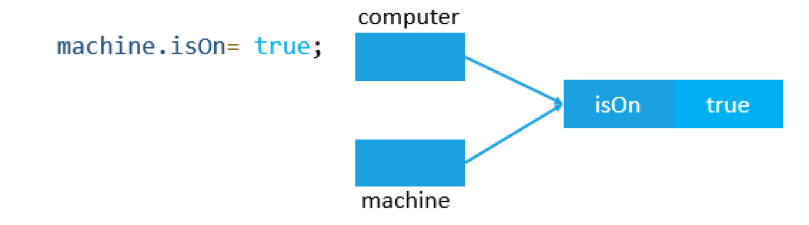
Tiếp theo, khai báo biến computer và gán cho nó một đối tượng với thuộc tính isOn được thiết đặt là false. Vậy biến computer chứa một giá trị tham chiếu như một con trỏ mà trỏ tới đối tượng như sau:



Sau đó, truyền biến computer vào hàm turnOn(). JavaScript sao chếp giá trị của biến computer vào biến machine. Kết quả là cả biến computer và machine đều trỏ tới cùng một đối tượng như sau:



Tiếp sau đó, trong hàm turnOn(), thuộc tính isOn của đối tượng machine được thiết đặt là false qua biến tham chiếu machine.



Cuối cùng, truy cập vào thuộc tính isOne của đối tượng computer sẽ thấy trả về là true.



Vậy có một giá trị tham chiếu được truyền tới hàm bằng cách truyền tham chiếu.

Trong thực tế, khi bạn truyền giá trị tham chiếu tới hàm, tức là bạn đang truyền tham chiếu tới đối tượng, chứ không phải truyền đối tượng. Do đó, hàm có thể sửa đổi các thuộc tính của đối tượng.

Ngoài ra, khi truyền tham chiếu thới hàm, hàm đó không làm thay đổi tham chiếu để trỏ tới một đối tượng khác.

Hãy xem ví dụ sau:

function turnOn(machine) {

    machine = {

        isOn: true

    };

}

let computer = {

    isOn: false

};

console.log(computer.isOn); // false;

Trong ví dụ này, hàm turnOn() thay đổi đối tượng machine mà đang trỏ tới một đối tượng khác.

Sau khi truyền đối tượng computer vào hàm turnOn(), thuộc tính isOn của đối tượng computer là false.

Nếu biến computer được truyền theo cách truyền tham chiếu, biến computer sẽ thay đổi hướng trỏ tới tới đối tượng mới mà thuộc tính isOn là true.

Tuy nhiên, khi chúng ta truy cập lại vào thuộc tính isOn, giá trị của nó vẫn là false chỉ ra rằng giá trị gốc của biến được truyền không thay đổi thậm trí giá trị của tham số đã bị thay đổi ở bên ngoài hàm.

Vậy, bạn cần hiểu rằng, tất cả các tham số truyền vào hàm trong JavaScript mặc định là truyền bằng giá trí chứ không phải truyền bằng tham chiếu.