

Comparison operators

1. So sánh **cùng** kiểu dữ liệu
2. So sánh **khác** kiểu dữ liệu
3. So sánh với **null/undefined**

Danh sách các phép so sánh trong Javascript

#	Operator	Desc
1	>, <	greater/less than
2	>=, <=	greater/less than or equal
3	==	equal (check value after type coercion)
4	!=	not equal
5	===	strict equality (check both type and value, type first and value after type coercion)
6	!==	strict non-equality

--> Kết quả của các phép so sánh là **boolean** (true/false)

A	opt	B	Result
1	<	2	true
1	>=	2	false
'2'	==	'1'	false
'2'	>	'1'	???
3	==	'3'	???
4	===	'4'	???
4	===	'4'	???
null	==	undefined	???
null	==	0	???
null	>=	0	???
"	==	0	???

So sánh cùng kiểu dữ liệu

1. So sánh số (number)
2. So sánh chuỗi (string)
3. So sánh boolean
4. So sánh object / array

So sánh số (number)

Cái này đơn giản, như toán học mình đã biết.

```
// Greater than / Less than
1 > 2; // false
1 < 2; // true

// Greater than or equal / Less than or equal
1 >= 2; // false
2 <= 2; // true

// Equal / Not equal
1 == 2; // false
1 != 2; // true
```

So sánh chuỗi (string)

- Chuỗi cũng có tất cả các operators so sánh như số, tuy nhiên cách so sánh chuỗi hơi khác biệt.

So sánh theo từng kí tự một, từ trái sang phải

- Nếu kí tự đầu tiên lớn hơn, thì string đó lớn hơn.
- Nếu kí tự đầu tiên nhỏ hơn, thì string đó nhỏ hơn.
- Nếu kí tự đầu tiên giống nhau thì đi kiểm tra tiếp kí tự tiếp theo.
- Tiếp tục cho đến khi hết chuỗi.

Làm sao biết kí tự nào lớn hơn? --> Tra trong bảng [Unicode UTF-16](#)

```
'a' > 'b' // false
'aa' <= 'b' // true (longer doesn't mean will be greater 😊)
'abc' > 'abd' // false (because c is less than d)

'a' > 'A' // true
'A' > '9' // true

'easy' == 'easy' // true
'easy frontend' != 'easy' // true
'easy frontend' > 'easy' // true
```

So sánh boolean

- Boolean values: true / false
- `true` is greater than `false` 😂

```
true > false // true
true == true // true
true == false // false
true != false // true
```

So sánh object / array

```
const student1 = { name: 'Hau' };
const student2 = { name: 'Hau' };

// student1 == student2?
```

- Không như kiểu dữ liệu primitive, object không thể so sánh lớn hơn, nhỏ hơn hay bằng được.
- Việc so sánh object thường được định nghĩa một cái hàm riêng đặc biệt để xử lý việc so sánh.
- Còn về so sánh bằng thì mình cần biết thêm về tham trị và tham chiếu, cái này sẽ nói sau trong chương object nhen.

--> Tối đây mình chỉ tạm thời quan tâm tới so sánh `number`, `string` và `boolean` 😊

So sánh khác kiểu dữ liệu

Khi so sánh khác kiểu dữ liệu, js sẽ **tự động convert** giá trị về dạng **number** để so sánh.

Nên để hiểu được phần này, bạn cần nắm được khi convert một giá trị nào đó về number thì nó sẽ thành giá trị bao nhiêu?

```
123 == '123' // true because '1' will be converted to 1
123 == '0123' // true because '01' will be converted to 1

false == 0 // true because false will be converted to 0
true == 1 // true because true will be converted to 1

'0' == false // true
'2' > true // ???
'123' >= 100 // ???
```

Tham khảo: <https://getify.github.io/coercions-grid/>

Strict equality (=== and !==)

Giữ nguyên giá trị (không tự động convert kiểu dữ liệu) và so sánh với nhau.

- Nếu khác kiểu dữ liệu, lập tức return **false**
- Nếu cùng kiểu dữ liệu, thì so sánh như cách thức đã đề cập ở phần so sánh cùng kiểu dữ liệu.

```
111 === '111' // false because they are different data types
0 === false // false
'' === 0 // false

1 === 1 // true
'abc' === 'abc' // true
```

NOTE: Nên sử dụng === thay vì == để hạn chế rủi ro không đáng có nhé. Nếu muốn dùng == thì phải hiểu thực sự mình đang làm gì nhé 😊

So sánh với null/undefined

```
// non-strict ==, null and undefined equal but not any other value.
null == undefined; // true

// strict check ===
null === undefined; // false because they are different type

// other comparisons
// null will converted to 0
// undefined will converted to NaN
null > 0 // false
null == 0 // false
null >= 0 // true WHAT??? 🤔
```

Tóm tắt:

- Các nhóm so sánh: lớn bé, bằng và không bằng
- Chỉ quan tâm tới kiểu primitive types, còn object/array thì sẽ có cách so sánh riêng.
- So sánh **cùng** kiểu dữ liệu, number thì như toán học, string thì so sánh từng ký tự dựa theo bảng UTF-16.
- So sánh **khác** kiểu dữ liệu, values sẽ được tự động convert về number để so sánh (type coercion)
- Nên dùng **===** cho đến khi gặp trường hợp đặc biệt cần dùng **==**
- Chỉ có cặp **null == undefined** cho **true**, còn so sánh với các giá trị khác đều cho **false**

🤔 HẠI NÃO CHƯA 🤔

Tham khảo:

- <https://javascript.info/comparison>
- <https://dorey.github.io/JavaScript-Equality-Table/unified/>
- <https://getify.github.io/coercions-grid/>

Khoá học Javascript cho người mới bắt đầu 2021 🌈

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>

