

Equality / Comparison in typescript

AGENDA

1. Nhắc lại về == và === bên Javascript
2. So sánh bên typescript?
3. So sánh object
4. null vs undefined

1. Nhắc lại về == và === bên Javascript

- So sánh khác kiểu dữ liệu với `==`, thì sẽ **được chuyển về number** để so sánh. (**)
- Với 3 dấu `===` thì sẽ giữ nguyên giá trị để so sánh.

Cảm ơn a Tiệp Phan đã góp ý sau khi video được đăng tải, mình xin bổ sung để làm rõ ý chỗ này nhé mn. Ý mình nói ở trên chỉ định cho các primitive types, không tính object mn nhé. Tại thường mình sẽ không dùng `==` để so sánh object, trừ khi mình muốn check cái reference giữa 2 object. Về object và null/undefined, có một số notes ở 2 phần tiếp theo nè.

Tham khảo chi tiết tại đây: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Equality>

```
2 == '2'; // true
false == 0; // true

2 === '2'; // false
false === 0; // false
```

2. So sánh bên typescript?

```
2 == '2';
2 === '2';
// ts error: This condition will always return 'false' since the types
'number' and 'string' have no overlap. ts(2367)
```

Cảm ơn anh Trường Nguyễn và anh Tiệp Phan góp ý sau khi video được đăng tải. Chỗ này mn lưu ý có thể bypass cái error bằng việc thêm kiểu any vào nhé. Đồng thời, cái này là type error, mọi người vẫn có thể compile ra javascript được và vẫn chạy như javascript nhé 😊

3. So sánh object

- object, array và function là tham chiếu.
- Khi so sánh 2 tham chiếu, thì địa chỉ của tham chiếu được so sánh, chứ không phải giá trị thật sự.

```
const studentA = { id: 1, name: 'Alice' };  
const studentB = studentA;  
  
studentA === studentB; // true;  
studentA === { id: 1, name: 'Alice' }; // false
```

```
// usually, we just compare the id of two objects  
const studentA = { id: 1, name: 'Alice' };  
const studentB = { id: 1, name: 'Alice' };  
  
studentA.id === studentB.id; // true
```

```
let student; // undefined  
  
// fetch data  
student = {  
  id: 1,  
  name: 'Alice',  
}  
  
// check truthy  
if (student) {}
```

```
let student = {};  
  
// fetch data  
student = {  
  id: 1,  
  name: 'Alice',  
}  
  
if (student?.id) {}
```

4. null vs undefined

- Something hasn't been initialized : **undefined**.
- Something is currently unavailable: **null**
- **null** is equal to **undefined** if using **==**

```
null == undefined; // true;
null == 0; // false
null >= 0; // true

if (something == null) {} // either null or undefined
if (something) {} // any truthy value
if (!something) {} // any falsy value
```

```
// Check if a global variable existed
if (typeof window !== 'undefined') {
  // if window object is available or not
  // do something on client side
}
```

```
JSON.stringify({
  id: null,
  name: undefined,
});
// will return "{\"id\":null}"
// undefined is removed from the JSON 💡
```

Tham khảo

- <https://basarat.gitbook.io/typescript/recap/equality>
- <https://basarat.gitbook.io/typescript/recap/references>
- <https://basarat.gitbook.io/typescript/recap/null-undefined>

Series - Typescript cơ bản 🎨

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.

- [Đăng ký fan cứng](#) để xem series này đầy đủ và sớm nhất nhé.



Kết nối với mình

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