TSLint double vs triple equality

Asked 5 years, 5 months ago   Active 4 years, 7 months ago   Viewed 24k times

▲

21

▼

★

2

I know that a single equality sign means assignment; double means equality; and triple means equality and the same type.

What I don't understand why the typescript linter would want me to use triple equality signs in this case:

```
function gcf(a: number, b: number): number
{
    return (b == 0) ? (a) : (gcf(b, a % b));
}
```

> TsLint: == should be ===

I know that 0 is a number and I also know that b is a number (or else I'll get a compilation error). So why would I want to use triple equality signs in this case?

`typescript`   `equality`

asked Mar 29 '14 at 15:07

user886079
511   1   7   17

While it's not as likely in TypeScript to cause an issue, it's a strict comparison -- I'm not sure why you wouldn't want to use it? stackoverflow.com/a/359509/95190 It's generally good practice to use it in JavaScript. – WiredPrairie Mar 29 '14 at 22:47

this.password1 == this.password2 is getting failed and return false. What is the reason for this? – Lahiru Gamage Oct 14 '18 at 14:42

2 Answers

22

```
var foo:number = null;

if (foo == undefined) {
    console.log('is undefined'); // actually null
}
```

For more info on why these are equal https://stackoverflow.com/a/359509/95190

Personally : I have had this rule disabled and never had any issues. I don't compare with `true/false/null/undefined` , just `if` them. And typescript prevents comparing `strings` and `numbers` so that is not an error I need to deal with.

edited May 23 '17 at 12:33	answered Mar 30 '14 at 0:14
Community ♦	basarat
1 1	**153k** 29 289 393

6

Using the triple equality operator also saves you in cases when the resultant Javascript may be called from an outside file (i.e. outside of the TypeScript environment). Pure JS files aren't processed by tslint, and by 'requiring' the triple equality, tslint makes the resultant Javascript file that little more resiliant.

answered Feb 5 '15 at 3:31

Darren Oster
4,969 9 41 58