A Drip of JavaScript

# Checking Date Equality in JavaScript

*Originally published in the A Drip of JavaScript newsletter.*

When working with dates, one of a programmer's most common needs is to check whether one date matches another. And if you're like most programmers, the first way that you thought of doing it is something like this:

```
function isChristmas (dateToTest) {
    var christmas = new Date("12/25/2014");
    return (dateToTest === christmas);
}
```

Unfortunately, this function will never return `true`.

```
console.log(isChristmas(new Date("12/25/2014")));
// => false
```

The reason it will never return `true` is because of the way object equality works in JavaScript. Object equality isn't tested by the internal value of the object, but by identity. In other words, if it isn't the exact same copy of the `Date` object, it isn't considered equal.

To make our `isChristmas` function work, we need to check equality in a different way.

```
function isChristmas (dateToTest) {
    var christmas = new Date("12/25/2014");
    return (dateToTest.getTime() === christmas.getTime());
}
```

Here we compare the return values of `getTime`. The `getTime` method returns an integer representing the number of milliseconds since midnight of January 1, 1970 (the beginning of the Unix epoch).

And we can see that we now get the correct result.

```
console.log(isChristmas(new Date("12/25/2014")));
// => true
```

But if we happen to compare against a `Date` object that is the same day, but a different hour, we'll have trouble again.

```
console.log(isChristmas(new Date("12/25/2014 12:00")));
// => false
```

In order to take different times on the same day into account we might try checking only the year, month, and date of the month.

```
function isChristmas (dateToTest) {
    return (dateToTest.getFullYear() === 2014) &&
           // getMonth is 0-indexed
           (dateToTest.getMonth() === 11) &&
           (dateToTest.getDate() == 25);
}

console.log(isChristmas(new Date("12/25/2014 12:00")));
// => true
```

Despite the "gotcha" of `getMonth` returning a 0-indexed number for months, this function now works. However, it doesn't take into account the complexities of timezones and working with local time versus UTC/GMT time.

Because of these complexities, it is generally better to lean on a robust and well-tested library like Moment.js to do things like date comparisons.

But most importantly, now you know not to count on JavaScript's equality operators when comparing dates.

Thanks for reading!

Josh Clanton

---