

What is the "double tilde" (~~) operator in JavaScript? [duplicate]

Asked 8 years, 5 months ago Active 5 years, 6 months ago Viewed 117k times



511

This question already has an answer here:

[What does ~~ \("double tilde"\) do in Javascript?](#) 9 answers



I'm seeing this in some code, and I have no idea what it does:



153

```
var jdn = function(y, m, d) {  
  var tmp = (m <= 2 ? -1 : 0);  
  return ~~((1461 * (y + 4800 + tmp)) / 4) +  
    ~~((367 * (m - 2 - 12 * tmp)) / 12) -  
    ~~((3 * ((y + 4900 + tmp) / 100)) / 4) +  
    d - 2483620;  
};
```

What's the ~~ operator do?

javascript

edited Sep 15 '12 at 9:05



Drew Noakes

201k

128

548

646

asked May 11 '11 at 23:18



jismo

2,561

3

11

4

marked as duplicate by [Factor Mystic](#), [Tikhon Jelvis](#), [tkanzakic](#), [Rubens](#), [Mario Sannum](#) Apr 28 '13 at 15:59

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

3 You've probably figured it out by now but it returns the number of days between the millennium and a given date – [Awalias](#) Apr 11 '13 at 10:52

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

~~'9'//number – [STEEL](#) Mar 3 '14 at 6:55 

- 1 Not having comments is one thing but what I hate most is that someone named this jdn . I don't even know what that stands for. – [Andy](#) Feb 16 '15 at 6:47
- 1 Math.trunc(), not Math.floor() – [Xitalogy](#) Jun 8 '17 at 22:17

4 Answers



654



That `~~` is a double NOT bitwise operator.

It is used as a [faster substitute for `Math.floor\(\)`](#) .



edited Aug 14 '12 at 3:09

answered May 11 '11 at 23:21



[ghoppe](#)

18.1k 3 24 19

- 286 @ghoppe: Worth noting that it differs from `.floor()` in that it actually just removes anything to the right of the decimal. This makes a difference when used against a negative number. Also, it will always return a number, and will never give you `NaN` . If it can't be converted to a number, you'll get `0` . – [RightSaidFred](#) May 11 '11 at 23:27
- 16 @ghoppe: Yes, the two not operations are actually faster than the single floor method. They run in about 0.2 microseconds instead of 0.5 microseconds when I test it in Firefox on my computer. That means that you need to use it a lot before it's noticable. In a functon like the one in the OP it's just micro optimisation, and only makes the code harder to follow. – [Guffa](#) May 11 '11 at 23:51
- 23 I ran into an integer overflow issue using this technique with very large numbers (the result of dividing numbers from the Navigation Timing API by 62 during base-62 encoding). For instance, in Firefox, Chrome and IE, `~~(2419354838.709677) == -1875612458`, whereas `Math.floor(2419354838.709677) == 2419354838`. – [Jacob Wan](#) Jul 12 '12 at 22:08
- 7 DON'T USE `~~`! In IE10, it's 8 times slower than doing other bitwise operations for the same thing! Better use zero right shift (`>>0`), it's the fastest one, and looks more similar to signed-to-unsigned conversion (`>>>0`). – [Triang3l](#) Dec 21 '12 at 14:20 
- 3 Similar to `Math.trunc()`, but not `Math.floor()` – [Xitalogy](#) Jun 8 '17 at 22:21 

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

162

It's two single tilde operators, so it does a bitwise complement (bitwise not) twice. The operations take out each other, so the only remaining effect is the conversion that is done before the first operator is applied, i.e. converting the value to an integer number.

Some use it as a faster alternative to `Math.floor`, but the speed difference is not that dramatic, and in most cases it's just micro optimisation. Unless you have a piece of code that really needs to be optimised, you should use code that describes what it does instead of code that uses a side effect of a non-operation.

Update 2011-08:

With optimisation of the JavaScript engine in browsers, the performance for operators and functions change. With current browsers, using `~~` instead of `Math.floor` is somewhat faster in some browsers, and not faster at all in some browsers. If you really need that extra bit of performance, you would need to write different optimised code for each browser.

See: [tilde vs floor](#)

edited Mar 20 '14 at 22:06



Kara

4,242

10

46

53

answered May 11 '11 at 23:38



Guffa

583k

82

605

900

92 +1 for "it hides the intention of the code", i wasted 10 minutes to know what `~~` does. Anyway I also have to admit it's already strong in me the dark side that's already tempting me to use `~~` in place of `Math.floor` forever in my new code now on. :))) – [Marco Demaio](#) Feb 4 '12 at 12:16

2 Note that micro tests like JSPerf (necessarily) run the test code enough times that on-the-fly runtime optimizations (such as in V8) kick in. That test shows that (if used very heavily) `Math.floor()` can be as fast as `~~` on Chrome, but not that it *is always* the same speed. These days it's just quite hard to say for sure whether or not one bit of code is "faster" than another (accounting for different browsers and invocation scenarios). – [Phrogz](#) May 31 '12 at 21:47

1 Why on earth is Chrome 22 so much slower than Chrome 8?? – [Matt Sach](#) Jan 8 '13 at 14:23

@MattSach: The figures are only comparable if tested on the same computer, or if plenty enough people have tested it. Chrome has so many versions that there is rarely more than a handful of persons that has tested the code with each version. – [Guffa](#) Jan 8 '13 at 14:31

4 Just remember that `Math.floor()` exists for a reason. Don't go off using `~~` because it's 2 microseconds faster than `Math.floor` if you don't understand where it might cause overflows or other unexpected results. – [dudewad](#) Jan 23 '14 at 18:10

```
~(5.5) // => -6
~(-6)  // => 5
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

For more info, see:

- <http://dreaminginjavascript.wordpress.com/2008/07/04/28/>

edited Aug 25 '11 at 15:39

answered May 11 '11 at 23:25



[bowersenior](#)

11.5k 2 43 49

10 ~(-5.5) => 4, ~(4) => -5, ~~(-5.5) => -5. Therefor, not the same as Math.floor – [zzzzBov](#) Aug 22 '11 at 18:51

@zzzzBov, I updated the post to clarify that ~~ is not the same as Math.floor() for negative numbers. – [bowersenior](#) Aug 25 '11 at 15:42

Math.floor(-5.5), which would give -6. Bcos Math.floor will return largest integer less than or equal to a given number. Math.floor(-5.00000001) also give -6. – [SridharKritha](#) Jun 6 '18 at 11:38

The diffrence is very simple:

19

Long version

If you want to have better readability, use Math.floor. But if you want to minimize it, use tilde ~~.

There are a lot of sources on the internet saying Math.floor is faster, but sometimes ~~. I would not recommend you think about speed because it is not going to be noticed when running the code. Maybe in tests etc, but no human can see a difference here. What would be faster is to use ~~ for a faster load time.

Short version

~~ is shorter/takes less space. Math.floor improves the readability. Sometimes tilde is faster, sometimes Math.floor is faster, but it is not noticeable.

answered Jun 8 '12 at 8:16



[Jason Stackhouse](#)

1,129 2 11 18

4 Right. It's primarily a stylistic choice, like the choice between Boolean(foo), (foo ? true : false), or !!foo when you want to cast a variable to a boolean. – [Patrick Fisher](#) Mar 17 '13 at 8:35

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

