Are braces necessary in one-line statements in JavaScript?

Asked 8 years, 6 months ago Active 4 months ago Viewed 74k times

134

16

I once heard that leaving the curly braces in one-line statements could be harmful in JavaScript. I don't remember the reasoning anymore and a Google search did not help much.

Is there anything that makes it a good idea to surround all statements within curly braces in JavaScript?

I am asking, because everyone seems to do so.

javascript

edited Mar 13 at 22:07
João Pimentel Ferreira
4,713 2 26 37

asked Jan 25 '11 at 18:17
Tower
46.3k 101 306 475

3 Note: only the first statement is assuming the scope, even if you have several statements on one line, so it is not "one line statements" but rather single statement – Kris Ivanov Jan 25 '11 at 18:28

1 You may be thinking of the issue described in this answer – Blorgbeard Jan 25 '11 at 18:29

@Blorgbeard: no, I actually replied to that answer while ago. – Tower Jan 25 '11 at 20:14

Hah, so I see. Never mind then :) – Blorgbeard Jan 25 '11 at 21:45

18 Answers

No

But they are recommended. If you ever expand the statement you will need them.

173

```
if (cond)
    alert("Condition met!")
else
    alert("Condition not met!")
```

However it is highly recommended that you always use braces because if you (or someone else) ever expands the statement it will be required.

This same practice follows in all C syntax style languages with bracing. C, C++, Java, even PHP all support one line statement without braces. You have to realize that you are only saving **two characters** and with some people's bracing styles you aren't even saving a line. I prefer a full brace style (like follows) so it tends to be a bit longer. The tradeoff is met very well with the fact you have extremely clear code readability.

```
if (cond)
{
    alert("Condition met!")
}
else
{
    alert("Condition not met!")
}
```

answered Jan 25 '11 at 18:18
Josh K
20k 19 74 126

17 +1, informative answer. Personally though, I've never found it useful to do this "recommended" thing. I've never coded python, so I don't just insert things and expect the indentation to matter. If I add a statement, I also add braces. Always. Can't remember a single time it bit me. Not in C, not in C# not in JavaScript. – Jakob Jan 25 '11 at 18:21

15 @Kirk: Douglas Crockford recommends it. I agree that it is a subjective personal decision but when working in a group it is easier to simply type the braces. – Josh K Jan 25 '11 at 18:22

8 @Josh, oh, well Crockford said it. That must be the final word. ;) (just kidding) The issue is that the subjectivity of this point extends across all C-like languages, and strong opinions can be found throughout (for both positions). – Kirk Woll Jan 25 '11 at 18:27

10 My personal experience shows that not placing bracers can lead to big screw-ups when working on teams. – Sirs Oct 3 '12 at 11:50

3 It's a good practice to always use the braces {}. As @Arx said, there's a lot more room for error if you leave them out. Apple even had a bug in iOS's SSL/TLS because they didn't use braces. – Keenan Lidral-Porter Jan 6 '15 at 19:56

87

There's a readability aspect - in that when you have compound statements it can get very confusing. (Indenting helps, but doesn't mean anything to the compiler/interpreter)

```
var a;
var b;
var c;

//Indenting is clear
if (a===true)
  alert(a); //On on IF
alert(b); //Always

//Indenting is bad
if (a===true)
  alert(a); //On on IF
  alert(b); //Always but expected?

//Nested indenting is clear
if (a===true)
  if (b===true)
    alert(a); //Only on if-if
alert (b); //Always

//Nested indenting is misleading
if (a===true)
  if (b===true)
    alert(a); //Only on if-if
  alert (b); //Always but expected as part of first if?

//Compound line is misleading
//b will always alert, but suggests it's part of if
if (a===true) alert(a);alert(b);
else alert(c); //Error, else isn't attached
```

And then there's an extensibility aspect:

```
//Problematic
if (a===true)
  alert(a);
  alert(b); //We're assuming this will happen with the if but it'll happen always
else       //This else is not connected to an if anymore - error
  alert(c);
  alert(a); //on if
  alert(b); //on if
} else {
  alert(c); //on !if
}
```

(The thinking goes that if you always have the brackets then you know to insert other statements inside that block).

edited Jan 25 '11 at 18:32

answered Jan 25 '11 at 18:25
Rudu
13.6k 2 38 58

That's why we should always use it as one-liner: `if (a===true) alert(a);` . Now it's clear! – João Pimentel Ferreira Mar 13 at 22:16

50

The question asks about statements on one line. Yet, the many examples provided show reasons not to leave out braces based on multiple line statements. It is completely safe to not use brackets on one line, if that is the coding style you prefer.

For example, the question asks if this is ok:

```
if (condition) statement;
```

It does not ask if this is ok:

```
if (condition)
  statement;
```

I think leaving brackets out is preferable because it makes the code more readable with less superfluous syntax.

My coding style is to never use brackets unless the code is a block. And to never use multiple statements on a single line (separated by semicolons). I find this easy to read and clear and never have scoping issues on 'if' statements. As a result, using brackets on a single if condition statement would require 3 lines. Like this:

```
if (condition) {
  statement;
```

I wouldn't force others to use this method, but it works for me and I could not disagree more with the examples provided on how leaving out brackets leads to coding/scoping errors.

edited Jan 6 '13 at 20:26

answered Jan 6 '13 at 12:41

peawormsworth
819 8 7

1 I always felt one should always include braces... but I'm rethinking it now. You have the airbnb style guide on your side! – senderle Sep 10 '15 at 18:35

Yet you forget that most code formatters change that to a 2-line format and you are back to problematic code. The vertical space argument is just silly. Readability always wins, and today's screens are huge. – Kim Nov 23 '17 at 1:11

The 2-lines added for each bracket, to surround your one-line statements are not a big cost compared to a potential damage that can be caused by a - even very careful developer - maintaining your code. You yourself can be a great developer with a mythical skills, but you can not assume that your colleagues are. KISS, wrap things with a context and make it as easy as possible for others or you will eventually get into trouble. – Maciej Tokarz Jan 24 '18 at 11:49

13

Technically no but otherwise absolutely Yes!!!

Forget about "It's personal preference","the code will run just fine","it has been working fine for me","it's more readable" yada yada BS. This could easily lead to very serious problems if you make a mistake and believe me it is very easy to make a mistake when you are coding(Don't belive?, check out the famous Apple go to fail bug).

Argument: "It's personal preference"

No it is not. Unless you are a one man team leaving on mars, no. Most of the time there will be other people reading/modifying your code. In any serious coding team this will be the recommended way, so it is not a 'personal preference'.

Argument: "the code will run just fine"

So does the spaghetti code! Does it mean it's ok to create it?

Argument: "it has been working fine for me"

In my career I have seen so many bugs created because of this problem. You probably don't remember how many times you commented out `'DoSomething()'` and baffled by why `'SomethingElse()'` is called:

```
if (condition)
    DoSomething();
SomethingElse();
```

Or added 'SomethingMore' and didn't notice it won't be called(even though the indentation implies otherwise):

```
if (condition)
  DoSomething();
  SomethingMore();
```

Here is a real life example I had. Someone wanted to turn of all the logging so they run find&replace `"console.log"` => `//"console.log"`:

```
if (condition)
   console.log("something");
SomethingElse();
```

See the problem?

Even if you think, "these are so trivial, I would never do that"; remember that there will always be a team member with inferior programming skills than you(hopefully you are not the worst in the team!)

Argument: "it's more readable"

If I've learned anything about programming, it is that the simple things become very complex very quickly. It is very common that this:

```
if (condition)
    DoSomething();
```

turns into the following after it has been tested with different browsers/environments/use cases or new features are added:

```
if (a != null)
   if (condition)
      DoSomething();
   else
      DoSomethingElse();
      DoSomethingMore();
else
```

```
    else
        alert("error a");
```

And compare it with this:

```
if (a != null) {
   if (condition) {
      DoSomething();
   }
   else {
      DoSomethingElse();
      DoSomethingMore();
   }
} else if (b == null) {
   alert("error b");
} else {
   alert("error a");
}
```

PS: Bonus points go to who noticed the bug in the example above.

edited Dec 19 '18 at 13:58

answered Feb 23 '17 at 13:03

Caner
35.9k 25 129 149

2 well the obvious bug is the DoSomethingMore(); But there is also another bug. if a is null and b is null, you only get "error b", you never get "error a". – rocketsarefast Jul 19 '18 at 15:36

11

There is no maintainability problem!

The problem with all of you is that you Put semicolons everywhere. You don't need curly braces for multiple statements. If you want to add a statement, just use commas.

```
if (a > 1)
 alert("foo"),
 alert("bar"),
 alert("lorem"),
```

This is valid code that will run like you expect!

edited May 22 '12 at 18:36

answered Mar 11 '12 at 15:02

william malo
1,916 1 14 18

2 Don't you mean `if`, `else` and `alert` and not `If`, `Else` and `Alert`? – Anish Gupta May 19 '12 at 10:34

3 I have never seen this before, using commas for multiple statements!!!!!! – xrDDDD Feb 15 '14 at 1:50

10 While this works in JavaScript, it's beyond me why you would ever want to do that. I am venturing a guess that the majority of developers are not aware of this (myself included prior to reading this), which I suspect would then quickly become a maintainability issue among developers. Sometimes, the most clever way is not the best. – Simon Oct 7 '14 at 15:22

12 This is horrible. If someone adds a statement and forgets to turn the semicolon into a comma on the now second to last statement in the block, you have a bug that can be *really* hard to spot because the comma and semicolon at the end of the line look way too similar. – Ingo Bürk Jan 25 '15 at 8:53

1 I prefer a "Hemingwayian" approach: very clean. And without space between `if` and `(`, like `if(true) doSomething();` – victorf Aug 17 '15 at 14:28

7

In addition to the reason mentioned by @Josh K (which also applies to Java, C etc.), one special problem in JavaScript is automatic semicolon insertion. From the Wikipedia example:

```
return
a + b;

// Returns undefined. Treated as:
//   return;
//   a + b;
```

So, this may also yield unexpected results, if used like this:

```
if (x)
   return
   a + b;
```

It's not really much better to write

```
if (x) {
   return
   a + b;
}
```

but maybe here the error is a little bit easier to detect (?)

answered Jan 25 '11 at 18:33
Chris Lercher
31.3k 16 91 122

All of those examples look horrible to me, unless the writers are temporary and paid by line or until it works. – Cees Timmerman Apr 29 '17 at 21:43

6

It's a matter of style, but curly braces are good for preventing possible dangling else's.

edited May 31 '14 at 18:15
Peter Mortensen
14.3k 19 88 116

answered Jan 25 '11 at 18:26
pault
61 3

3

There is no programming reason to use the curly braces on one line statements.

This only comes down to coders preferences and readability.

Your code won't break because of it.

answered Jan 25 '11 at 18:19
Yahel
7,903 2 18 29

Here is why it's recommended

```
if(someVal)
    alert("True");
```

Then the next developer comes and says "Oh, I need to do something else", so they write

```
if(someVal)
    alert("True");
    alert("AlsoTrue");
```

Now as you can see "AlsoTrue" will always be true, because the first developer didn't use braces.

edited May 31 '14 at 18:13
Peter Mortensen
14.3k 19 88 116

answered Jan 25 '11 at 18:22
Amir Raminfar
29.8k 6 74 110

That's not correct, you are missing the 'else': if(someVal) alert("True"); else alert("AlsoTrue"); would be correct. I would not use it because I like the {} for it is far better readable. – Gerrit B Aug 13 '15 at 21:21

1 Huh? I didn't have an else statement. I was saying that without curly braces, it can lead to bugs if someone adds a new line. I think you didn't understand my point. – Amir Raminfar Aug 14 '15 at 22:00

I think what he's saying is that, the 2nd line will get executed no matter what. An If statement without braces can only execute 1 line. – PostCodeism Jun 17 '16 at 16:35

2

I'm currently working on a minifier. Even now I check it on two huge scripts. Experimentally I found out: You may remove the curly braces behind for,if,else,while,function* if the curly braces don't include ';','return','for','if','else','while','do','function'. Irrespective line breaks.

```
function a(b){if(c){d}else{e}} //ok
function a(b){if(c)d;else e}   //ok
```

Of course you need to replace the closing brace with a semicolon if it's not followed by on other closing brace.

A function must not end in a comma.

Tested on Chrome and FF.

edited Jan 25 '15 at 8:26

answered Jan 22 '15 at 18:37
B.F.
389 5 8

1

There are many problems in javascript. Take a look at JavaScript architect Douglas Crockford talking about it The **if** statement seems to be fine but the **return** statement may introduce a problem.

```
return
{
    ok:false;
}
//silent error (return undefined)

return{
    ok:true;
}
//works well in javascript
```

answered Sep 9 '15 at 13:15
kamayd
350 5 9

1

I found this answer searching about a similar experience so I decided to answer it with my experience.

Bracketless statements do work in most browsers, however, I tested that bracketless methods in fact do not work in some browser.

As of February 26th 2018, this statement works in Pale Moon, but not Google Chrome.

```
function foo()
   return bar;
```

answered Feb 26 '18 at 18:49

1

Not responding the question directly, but below is a short syntax about if condition on one line

Ex:

```
var i=true;
if(i){
  dosomething();
}
```

Can be written like this:

```
var i=true;
i && dosomething();
```

answered Jun 25 '18 at 7:44
Alee
417 5 14

0

Sometimes they seem to be needed! I couldn't believe it myself, but yesterday it occurred to me in a Firebug session (recent Firefox 22.0) that

```
if (! my.condition.key)
    do something;
```

executed *do something* despite *my.condition.key* was **true**. Adding braces:

```
if (! my.condition.var) {
    do something;
}
```

fixed that matter. There are myriards of examples where it apparently works without the braces, but in this case it definitely didn't.

```
if (condition)
    do something; do something else;
```

are difficult to find.

edited Aug 14 '13 at 6:58

answered Aug 13 '13 at 15:36
Tobias
1,522 3 16 36

I am curious in what scenario the lack of braces made the if condition true, can you recall or give a real example of it? – gitsitgo Jul 10 '14 at 15:53

Not easily, I'm afraid. It was nearly a year ago ... – Tobias Jul 11 '14 at 15:23

The conditional expression is always evaluated before the statement. I'm also very curious to see a real example of this because it would represent a bug in the interpreter. – Semicolon Apr 12 '15 at 20:13

0

I would just like to note that you can also leave the curly braces off of just the else. As seen in this article by John Resig's.

```
if(2 == 1){
    if(1 == 2){
        console.log("We will never get here")
    }
} else
    console.log("We will get here")
```

answered Mar 5 '15 at 14:47
Johnston
9,272 9 49 93

The [Qt braces style][1] requires blocks on both sides of an `else` to match brace usage -- this example would require the braces on the `else` block in order to pass a code review. [1]: wiki.qt.io/Qt_Coding_Style#Braces – pixelgrease Jun 23 '15 at 17:50

There is a way to achieve multiple line non curly braces if statements (Wow what english) but it is kinda tedius:

```
if(true)
   funcName();
else
   return null;


function funcName(){
  //Do Stuff Here...
}
```

answered Apr 24 '16 at 23:56
amanuel2
3,194 20 54

You don't have to have so many new lines when omitting curly braces. The above can also be written on two lines as: `if (true) funcName()` and `else return null` – phobos Sep 8 '16 at 12:30

0

The beginning indentation level of a statement should be equal to the number of open braces above it. (excluding quoted or commented braces or ones in preprocessor directives)

Otherwise K&R would be good indentation style. To fix their style, I recommend placing short simple if statements on one line.

```
if (foo) bar();    // I like this. It's also consistent with Python FWIW
```

instead of

```
if (foo)
   bar();   // not so good
```

If I were writing an editor, I'd make its auto format button suck the bar up to the same line as foo, and I'd make it insert braces around bar if you press return before it like this:

```
if (foo) {
  bar();    // better
```

```
if (foo) {
  bar();    // consistent
  baz();    // easy to read and maintain
}
```

answered Mar 10 '18 at 18:17

Toku

1

0

Always found that

```
if(valid) return;
```

is easier on my eye than

```
if(valid) {
  return;
}
```

also conditional such as

```
(valid) ? ifTrue() : ifFalse();
```

are easier to read (my personal opinion) rather than

```
if(valid) {
  ifTrue();
} else {
  ifFalse();
}
```

but i guess it comes down to coding style

answered Apr 3 '18 at 6:52