

Should curly braces appear on their own line? [closed]

Asked 9 years, 3 months ago Active 1 month ago Viewed 260k times

278

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 8 years ago.

92

Should curly braces be on their own line or not? What do you think about it?

```
if (you.hasAnswer()) {  
    you.postAnswer();  
} else {  
    you.doSomething();  
}
```

or should it be

```
if (you.hasAnswer())  
{  
    you.postAnswer();  
}  
else  
{  
    you.doSomething();  
}
```

or even

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



```
else
    you.doSomething();
```

Please be constructive! Explain why, share experiences, back it up with facts and references.

coding-style

code-formatting

edited Sep 21 '11 at 22:34

community wiki

10 revs, 3 users 66%

Tom Wijsman

109 I find the "==" true" more distracting than the choice of brace placement. – [Dan Dyer](#) Sep 10 '10 at 22:36

11 @Dan: I think that always explicating the conditional expression greatly helps in clearness. – [Wizard79](#) Sep 10 '10 at 23:12

4 The only reason this would matter would be if your IDE/editor doesn't support matching curly bracket recognition. – [leeand00](#) Sep 26 '10 at 23:51

4 @leeand00: some of us still *print out* complex / unfamiliar code in order to study / annotate it. A good pretty-printer mitigates most of the problems though. – [Shog9](#) ♦ Sep 27 '10 at 0:20

2 sad the question is closed. After some time of indent based syntax usage I switched to (maybe weird) another braces structure. Like your first but closing brace in the last line of block. (after code line) – [cnd](#) Mar 16 '12 at 10:05

36 Answers

1 2 next



90



When I was a student I used to put curly braces on the same line, so that there are fewer lines, and the code gets printed on fewer pages. Looking at a single bracket character printed as the only thing in a line is annoying. (environment, paper wastage)

But when coding large applications, allowing some lines with only braces in them are affordable, considering the 'grouping' feeling it gives.

Whichever style you choose, be **consistent** so that it does not become an overhead for your own brain to process multiple styles in **related pieces of code**. In different scenarios (like above) i would say it is okay to use different styles, it's easier to 'switch context' at a high level

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



6 revs, 3 users 86%
dbza

- 6 On the other hand, the brace on the new line is an ANSI STANDARD, K&R is not. But the beauty about standards is, that there are so many different ones (see also uncyclopedia.wikia.com/wiki/AAAAAAAAA! on uncyclopedia). – [Quandary](#) Mar 31 '16 at 9:53

"there are fewer lines" I have Terabytes of Space and lots of Pixels. Why should i care about use more lines? – [12431234123412341234123](#) Sep 21 '16 at 6:27

- 1 @12431234123412341234123: I think he means because some people print the code out for code-review. And each not absolutely necessary newline is paper wasted, or a km² of forrest wasted at scale. However, if you don't print it out (i certainly don't) then ANSI is a lot better than K&R. Also, anybody who intends to print should probably use an automated code formatter - so this should be a question of tooling, not one of coding style. – [Quandary](#) Oct 11 '18 at 16:00

I agree that you should stay consistent, I'm been using curly braces on new lines for years, but I have to mix in the other way in my code, such as when calling functions, anonymous functions, object literals etc. Any expression-based languages / anonymous code make brackets on the same line much easier – [David Callanan](#) Dec 6 at 22:24

You should never do the 3rd method.

250

Skimping on braces might save you a few keystrokes the first time, but the next coder who comes along, adds something to your else clause without noticing the block is missing braces is going to be in for a lot of pain.

Write your code for other people.

edited Mar 26 '16 at 2:23

community wiki
2 revs, 2 users 89%
[rhettg](#)

- 115 I wish I knew where that little bit of wisdom originated. Because writing your code for people who won't bother to read it is about as pointless as you can get... – [Shog9](#) ♦ Sep 12 '10 at 3:19

- 70 The second programmer can add his own braces when he adds something. He's not stupid, and in a coding convention that encourages omitting braces for simple stuff like this, he'll know to look. – [Ken Bloom](#) Oct 22 '10 at 14:25

- 26 Optional braces are not optional. There are few worse design decisions that were made in C and carried over to its descendants. That it lives on in a language as recent as C# makes me *rage*. – [Adam Crossland](#) Oct 29 '10 at 15:58

11 There's one advantage to style #3 that you're all missing: You get more code on your screen at once. – [Loren Pechtel](#) Sep 21 '11 at 23:52



205



+50

For a long time I argued that they were of equal worth, or so **very close to equal** that the possible gain by making the right choice was far, far, below the **cost of arguing** about it.

Being **consistent is important**, though. So I said let's flip a coin and get on to writing code.

I've seen programmers resist change like this before. Get over it! I've switched many times in my career. I even use different styles in my C# than in my PowerShell.

A few years ago I was working on a team (~20 developers) that decided to ask for input, and then make a decision, and then enforce that across all the code base. We'd have 1 week to decide.

Lots of groans & eye-rolling. Lots of "I like my way, because it's better" but no substance.

As we were studying the finer points of the question, someone asked how to deal with this issue in brace-on-the-same-line style:

```
void MyFunction(  
    int parameterOne,  
    int parameterTwo) {  
    int localOne,  
    int localTwo  
}
```

Note that it's not immediately obvious where the parameter list ends, and the body begins. Compare to:

```
void MyFunction(  
    int parameterOne,  
    int parameterTwo)  
{  
    int localOne,  
    int localTwo  
}
```

We did some reading on how folks around the world had dealt with this problem, and found the pattern of adding a blank line after the open brace:

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



```
int parameterTwo) {  
  
    int localOne,  
    int localTwo  
}
```

If you're going to make a visual break, you may as well do it with a brace. Then your visual breaks become consistent, too.

Edit: Two alternatives to the 'extra blank line' solution when using K&R:

1/ Indent the function arguments differently from the function body

2/ Put the first argument on the same line as the function name and align further arguments on new lines to that first argument

Examples:

1/

```
void MyFunction(  
    int parameterOne,  
    int parameterTwo) {  
    int localOne,  
    int localTwo  
}
```

2/

```
void MyFunction(int parameterOne,  
                int parameterTwo) {  
    int localOne,  
    int localTwo  
}
```

/Edit

I still argue that consistency is more important than other considerations, but if we don't have an **established precedent**, then brace-on-next-line is the way to go.

edited Feb 5 '16 at 8:50

community wiki

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



- 34 FYI, I may sound like a reasonable person, but I'm actually a nut. For simple, single-line blocks, I will use neither braces nor newlines, making 'if (foo) bar()' all one line. I strive to make my code simple enough that it's not a problem. – [Jay Bazuzi](#) Sep 11 '10 at 4:42
- 40 Came here to post exactly this. Tons of people that keep the opening brace on the same line follow it up with a *blank line* (especially at the start of classes and methods) because otherwise, it's hard to separate the class/method header from the body. Well, if you're going to use an extra line anyway, you may as well put the brace there and get the added benefit of indentation being easier to see. – [Yevgeniy Brikman](#) Oct 29 '10 at 20:14
- 27 I've not seen the blank line - I'm more familiar with double-indent of the parameters for MyFunction() when they stray onto another line. – [Armand](#) Dec 1 '10 at 18:07
- 35 Breaking out the parameters to multiple lines like that is maddening. – [Fosco](#) Nov 10 '11 at 17:59
- 10 The "function parameter" argument is a red herring. Obviously the arguments should be double intended. No problem whatsoever to distinguish it from the following code. – [David Ongaro](#) Mar 22 '15 at 21:24

101

The cardinal rules are:

1. Follow the project's existing coding standard.
2. If there is no coding standard and you are editing an existing code-base owned by someone else - be consistent with the style of the existing code, no matter how much you like / dislike it.
3. If you are working on a green-field project - discuss with other team members, and come to a consensus on a formal or informal coding standard.
4. If you are working on a green-field project as the sole developer - make up your own mind, and then be **ruthlessly consistent**.

Even if you have no external constraints on you, it is (IMO) best to look for an existing (widely used) coding standard or style guideline, and try and follow that. If you roll your own style, there's a good chance that you will come to regret it in a few years.

Finally, a style that is implemented / implementable using existing style checkers and code formatters is better than one that needs to be "enforced" manually.

edited Nov 13 at 9:43

community wiki
2 revs
[Stephen C](#)

- 10 This answer deserves more votes. – [AShelly](#) Nov 1 '10 at 23:04

72

The benefit of the first method is that it is more vertically compact, so you can fit more code on your screen, and that is why I prefer it. The only argument I heard in favor of the second method is that it makes it easier to pair opening and closing brackets, but most IDE's have a keyboard shortcut for that, and it's actually a false statement- instead of pairing an opening bracket to a closing bracket you can pair a closing bracket to the "start of block" expression (if, else, for, while) on the same indentation level, so it's just as easy to determine where the start of the block is.

I see no reason to waste an entire line just for a bracket when the preceding for/while/if construct already visually indicates the start of a block.

That said, I do believe that the closing bracket should be in its own line because we need something to indicate the end of a block and its indentation structure in a visible way.

edited Sep 11 '10 at 13:57

community wiki

3 revs

EpsilonVector

12 No... I'm saying why reduce the amount of code that can fit on your screen by doing something that doesn't add to the code's clarity? – EpsilonVector Sep 11 '10 at 9:54

7 When I was beginning coding I liked each brace on its own line, now I prefer the first method – NimChimpsky Sep 21 '10 at 9:01

58 There is a huge body of research, going all the way back to the early Steam Age (Weinberg, "Psychology of Computer Programming"), that shows that programmer comprehension falls off DRAMATICALLY when the amount of code that must be viewed is more than can be seen at one time (i.e., one screenful, one printer page). This phenomenon argues STRONGLY for viewing vertical space as a valuable resource, not to be wasted gratuitously, and thus the first method is preferred. – John R. Strohm Nov 1 '10 at 21:34

10 LOL @ "wasting an ENTIRE line". OMG! Not that!! =P – Nick Spreitzer Dec 12 '10 at 19:47

7 @Julio In college I favored method 1 strongly, and couldn't stand to read method 2. After going to work at a company that uses C#, where the standard is method 2, I've come to like that just as well. I can now read or use either; neither one bothers me. People who have a strongly aversive reaction to one or another are generally overreacting to something that they are unfamiliar with. – KChaloux Jan 17 '13 at 15:54

I prefer

48

```
if (you.hasAnswer())
{
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



```
{
    you.doSomething();
}
```

over

```
if (you.hasAnswer()) {
    you.postAnswer();
} else {
    you.doSomething();
}
```

because the line `you.postAnswer();` is much easier to read and find at first glance. In the second way, it gets blended in with the line above it (`you.hasAnswer()`) making my eyes have to focus more to read it.

answered Nov 1 '10 at 20:50

community wiki
JD Isaacks

7 This is true until your program exceeds the height of your screen. ;) – [weberc2](#) Apr 4 '13 at 18:09

13 @weberc2 I think that when your program exceeds the height of the screen, two lines less won't change a lot. – [Mageek](#) Jun 29 '13 at 10:41

13 10 years ago, I would have agreed about screen space. Today, I use a 1920*1200 screen. It fit a LOT of code, more than my brain can process at once. The first method allows me to pull back and see the different scope opening/closing without having to read it. – [LightStriker](#) Apr 6 '14 at 13:34

3 I could never fathom why I preferred this method but it's for exactly this. – [Declan McKenna](#) Jul 30 '15 at 17:00

2 @Mageek This is belated, but it's not 2 lines, it's 2 lines for every scope. That's $O(N)$, not $O(1)$. I don't actually feel that strongly about it; it's more important that you pick a style that makes long parameter lists readable. – [weberc2](#) Dec 18 '15 at 21:56

I prefer the first method. Braces are totally not worth separate line.

39

The thing is that braces are not important. They're just **syntactical trash**, which is absolutely unnecessary to understanding of what code is for, of it's purpose and the way it's implemented. They're just a tribute to old-style C-like languages where visual grouping of operators was impossible due to low screen space available.


```

if (you.hasAnswer()){
    you.postAnswer();
}else{
    you.doSomething();
}

```

answered Sep 11 '10 at 6:59

community wiki
P Shved

- 7 I don't know about Haskell or Ruby, but Python is whitespace sensitive, which is why it doesn't require braces or other delimiters to denote blocks. The braces are not just syntactical noise; they serve an actual purpose. – Robert Harvey Sep 16 '10 at 16:11
- 14 @Robert, In C you have to do *both* whitespace and braces. In Python you should do only whitespace. Which is better? – P Shved Sep 16 '10 at 17:34
- 5 @Pavel, in C' you don't have to do whitespace. – Ken Bloom Oct 22 '10 at 14:26
- 7 @KenBloom C programs without whitespace are impossible to read. So you have to do them anyway. – P Shved Oct 23 '10 at 11:21
- 6 Regardless of if braces are a good idea or not, the mere existence of languages that don't use them doesn't seem like an argument for or against them. It only suggests that it is possible to have a language without them, not that it is a good or poor language design. – Jason Apr 6 '11 at 14:52

▲
37
▼
Use Python and sidestep the argument completely.

answered Oct 30 '10 at 2:47

community wiki
Mark Ransom

- 17 +1 SyntaxError: not a chance – Seth Nov 3 '10 at 0:56
- 6 This is simply not an option for the vast, vast majority of projects. Plus, indentation-for-grouping has it's share of problems. – Bryan Oakley Jul 18 '11 at 14:02
- @Bryan, I realize that this isn't very practical. I just thought it was a point of view that needed to be out there, stronger than just a comment. And I've never run into the problems caused by indentation that you imply, probably because I don't mix tabs and spaces. – Mark Ransom Jul 18 '11 at 15:46

Use C++ and sidestep the argument completely (plus static typing, speed, and a compiler).

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



Feb 2 '15 at 5:00



The position of curly braces should be

28

meta data



configurable in the IDE by the programmer. That way, those pesky braces in all code, regardless of author, look the same.

answered Nov 2 '10 at 12:43

community wiki
Jonathan

7 Totally agree. It's presentation and not data. – [Petruza](#) Nov 9 '12 at 14:57

The issue is that if you let everyone set their own, things get messy very quickly as commits are done. – [Andy](#) Jun 4 '16 at 1:22

1 @Andy: That's exactly the point, the IDE will change how they look, but only in the IDE! The actual source will not be touched. For version control, you can add hooks that translate whatever the setting for curly braces was to a common situation, so that everyone checks code out the same way. – [klaar](#) Aug 1 '16 at 7:44

@klaar Every modern IDE i've used will change tabs to spaces and move braces to their own line or the end of the "opening" line; I'm not sure why you think the source isn't touched in these cases, and that is the reason for my comment. It IS typically changed by the IDEs depending on the developers settings, which means during a commit I'll see lots of changes which are just noise as the braces got moved to thier own line, thus hiding the ACTUAL change someone did. – [Andy](#) Aug 1 '16 at 22:26

@Andy: Isn't there the possibility to use hooks that convert those discrepancies regarding whitespace and braces to a uniform standard uppon commit, to circumvent the noise problem you described? Either way, a proper versioning system *should* transcend petty things like whitespace or other nonsensical things. – [klaar](#) Aug 2 '16 at 7:17



It depends.

19

If I am coding in Javascript or jQuery, I use the first form:



```
jQuery(function($) {  
    if ($ instanceof jQuery) {  
        alert("$ is the iQuerv object!");  
    }  
});
```

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



But if I am coding in C#, I use the second form, because that is the canonical way to do it in C#.

```
public int CalculateAge(DateTime birthDate, DateTime now)
{
    int age = now.Year - birthDate.Year;
    if (now.Month < birthDate.Month
        || (now.Month == birthDate.Month && now.Day < birthDate.Day))
        age--;
    return age;
}
```



Note that your example can be written

```
if (you.hasAnswer())
    you.postAnswer();
else
    you.doSomething();
```

in C#.

answered Sep 10 '10 at 23:09

community wiki
[Robert Harvey](#)

- 1 It can be written in a lot of languages like that, because a block-statement is a statement. Adding! :-)- [Tamara Wijsman](#) Sep 11 '10 at 0:34
 - 2 According to the "Framework Design Guidelines" the "canonical way" is to place the opening brace on the same line (i.e. the first form). Just sayin' ... - [Uwe Honekamp](#) Sep 11 '10 at 14:48 
 - 3 @Uwe: Perhaps. But Microsoft adopted the "aligned braces" approach for all of its MSDN C# examples, and it's baked into Visual Studio, so... - [Robert Harvey](#) Sep 11 '10 at 15:05
- @Uwe: That's Cwalina's book and it's terribly named as it is much more than that. The FDG on MSDN has nothing to say about that. Also I wonder, why would the *Framework Design* Guidelines say anything about *C# coding* practice? - [R. Martinho Fernandes](#) Sep 15 '10 at 3:15
- 3 You should, in fact, put curly braces on the same line in Javascript. You can cause errors if curly braces are on their own line. For example, see [encosia.com/...](#) - [Joseph Hansen](#) May 6 '15 at 21:51 

19

```
if (value > maximum);
{
    dosomething();
}
```

than it is in this example

```
if (value > maximum); {
    dosomething();
}
```

The `; {` just looks more wrong to me than a line ending with `;` so I'm more likely to notice it.


answered Oct 30 '10 at 0:54

community wiki
bmb

11 You make a good argument, but personally, this has only ever happened to me once in my 5 years programming. I couldn't figure out why it wasn't executing, posted it on SO and someone quickly pointed out the semi-colon to me. However, every time it is condensed to use that 1 less line, I find it harder to read. – [JD Isaacks](#) Nov 2 '10 at 14:53

6 The `;"` looks like a kind of winking grumpy face or maybe a person with a moustache. – [glenatron](#) Nov 2 '10 at 23:28

1 +1 Great example in answer: very subtle mistake, easily overlooked. Thought provoking too on layout showing this up. – [therobyounknow](#) Jan 17 '11 at 13:27

10 Of course any decent IDE will flag the empty control statement and any decent compiler will issue a warning. – [Dunk](#) Feb 28 '13 at 15:16 

@Dunk The only flaw in your argument (which I vigorously agree with) is that so many people are using interpreted languages these days (JavaScript, PHP, et al) that a lot of "programmers" wouldn't know a compiler from a double latte. – [Craig](#) Dec 1 '17 at 18:11

I prefer a slight variant of 1)

15

```
if (you.hasAnswer()) {
    you.postAnswer();
} // note the break here
else {
```

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



Why?

- I think *a/ways* putting braces on their own line decreases readability. I can only fit a certain amount of source code on my screen. Bracket style 2) makes heave algorithms with a lot of nested loops and conditionals painfully long.
- However, I want `else` to start on a new line because `if` and `else` belong together, visually. If there's a bracket in front of the `else`, it's much more difficult to spot what belongs to what.
- 3) disqualifies itself. We all know what bad things can happen if you leave out the brackets and forget about it.

edited Apr 6 '11 at 14:57

community wiki

2 revs

Alexander Gessler

1 I have seen this one around where I work. It's interesting. – [Almo](#) Jul 20 '12 at 15:16

1 I also like this style better, as it allows me to put comment above the `else` line when needed, and/or put a blank line between the if-block and the else-block to make things look less crammed. Bracket style #2 does nothing other than distancing the actions from the conditions. With that said, my favorite is definitely python's no bracket style :) – [sayap](#) Oct 28 '12 at 3:26

4 If maximizing the number of code lines on the screen is important then just do away with newlines altogether. You'll be able to get a lot of lines on one screen. I prefer to not have anything cause me to pause and think while reading, ie. my definition of more readable. With the braces my mind ignores them. Without the braces my mind has to pause and align the control blocks. Not a long pause, but a pause none-the-less. – [Dunk](#) Feb 28 '13 at 15:21

1 Yes, if and else belong together, BUT so do `{` and `}` and as `}` is on a separate line, `{` should be on a separate line, too. "I can only fit a certain amount of source code on my screen" And that's exactly why saying the 3) would be "disqualifying itself" is no option at all. After a decade of working with 3) I have not forgotten adding brackets when adding a new line of code, ever, nor do I know anyone who ever had. If I have to adjust code to people, who can't read properly, where does it end? Stopping using certain language features, because some of the codes readers may not understand them? – [Kaiserludi](#) May 5 '14 at 18:25

I did read somewhere that the authors of some book wanted their code formatted like this:

10

```
if (you.hasAnswer())
{
    you.postAnswer();
}
else
{
    you.doSomething();
}
```

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



But space constraints from their publisher meant that they had to use this:

```
if (you.hasAnswer()) {  
    you.postAnswer();  
} else {  
    you.doSomething();  
}
```

Now I don't know whether that's true (as I can't find it any more), but the latter style is very prevalent in books.

On a personal level I prefer the brackets on a separate line as:

- a) they indicate a new scope
- b) it's easier to spot when you've got a mismatch (though this is less of an issue in an IDE that highlights errors for you).

answered Sep 11 '10 at 14:06

community wiki
ChrisF

... The second option also facilitates both of your points (with indentation alone serving the purpose of the brace/indentation combo). :) – weberc2 Apr 3 '13 at 20:24

▲ Ah, the [One True Brace Style](#).

10

It has everything needed for a Holy Way - even a prophet (Richard "my way or the highway" Stallman).

▼ The guy was so wrong about so many things, but GNU is spot-on when it comes to braces.

[Update] I have seen the light, and now worship [Allman](#)

edited Jul 5 '17 at 7:24

community wiki
2 revs
Mawg

You can't do any worse than two indentation levels per block, except for lisp style, of course, that goes without saying. – [ergosys](#) Jan 28 '12 at 5:57

4 +1 for the link on brace styles. It shows that whatever your style, many great people disagree with you. – [Florian F](#) Sep 2 '14 at 14:28

@RobertHarvey There is no extra work, if it is, you don't use the right tool to write code or dint configure it right. The benefit is much more readable code, you see every error in the bracket very fast and you can easy read only the code from while ignore subblocks. – [12431234123412341234123](#) Sep 21 '16 at 7:13

Second example, I'm very big on readability. I can't stand looking at if blocks any other way =(

9

answered [Sep 21 '10 at 3:57](#)community wiki
[Bryan Harrington](#)

2 Research indicates it's easier to read compact code once a code base exceeds the height of the screen. – [weberc2](#) Apr 3 '13 at 20:25

5 @weberc2, could you provide DOIs to those research paper? – [Grzegorz Adam Kowalski](#) Mar 9 '16 at 9:12

Simple answer: what is easier to debug ?

9

```
// Case 1:
void dummyFunction() {
    for (i = 0; i != 10; ++i) {
        if (i <= 10)
            std::cout << "i is: " << i << "\n";
            std::cout << 10 - i << " steps remaining\n";

            // Some hard work here
            // which is really hard
            // and does take some screen estate
        }
        else
            std::cout << "We'll never get there";
    }
} // COMPILER ERROR HERE
```

```

{
    for (i = 0; i != 10; ++i)

        if (i <= 10)
        {
            std::cout << "i is: " << i << "\n";
            std::cout << 10 - i << " steps remaining\n";

            // Some hard work here
            // which is really hard
            // and does take some screen estate
        }
        else
            std::cout << "We'll never get there\n";
    }
} // COMPILER ERROR HERE

```

In which case did you diagnose the issue first ?

I don't care much for personal preferences (there are many other styles, including whitesmith and al.) and I don't care much... as long as it doesn't hamper my ability to read the code and **debug** it.

As to the "waste space" argument, I don't buy it: I tend to add blank lines between logical groups anyway to make the program clearer...

answered Nov 5 '10 at 18:49

community wiki
Matthieu M.

1 They are both as easy to debug, mainly since it's a short block of code. The indentation is consistent making it easy to visualize the actual code blocks.
– Htbaa Apr 6 '11 at 14:44


@Htbaa: indeed :) So why bother ? – Matthieu M. Apr 6 '11 at 15:22

@MatthieuM. The first block makes more sense to me, because the newlines (in the second block) between the function signature, the for-statement and the if-statement have me believe they are unrelated, but clearly they aren't. Blank lines are to separate unrelated bits of code; code that's close to other lines of code means that they are in fact related. This is all 'imo' of course, but I wondered what your point was. EDIT: also any proper IDE will notice any brace missing and give you a smattering of errors upon interpreting your code. – klaar May 3 '16 at 11:48

▲ Not that anyone will notice, but this is why braces belong on the **same line** as the conditional (except for very long conditionals, but

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.





```
while(true);
{
    char c;
    getchar(); //Wait for input
}
```

Quick! What does this code do? If you answered "infinite loop asking for input", you are wrong! It doesn't even get to the input. It gets caught at `while(true)`. Notice that semicolon at the end. This pattern is actually more common than it seems like it should be; C requires you to declare your variables at the beginning of a block, which is why a new one was started.

A line of code is a thought. Braces are a part of the thought containing the conditional or loop. Therefore, they belong on the **same line**.

answered Nov 6 '10 at 7:48

community wiki
[Christian Mann](#)

-
- 1 This is by far the best argument for K&R style I have seen, the rest are laughable with today's IDE systems with code folding support. This does only apply to C style languages which support `;` block ends. This is also why I despise this block ending system which IMHO is outdated and the Go language proves it. I have seen this issue many times although not in this scenario. It usually happens where they intend to add something to the statement and forget to. – [Jeremy](#) Sep 22 '16 at 16:19
-



I like the first method. It seems neater IMO, and it's more compact, which I like.

5

EDIT: Ah, a third. I like that one the best when possible, as it's even smaller/neater.

edited Sep 11 '10 at 1:51

community wiki
[2 revs](#)
[Ullallulloo](#)



You could write it:

5

```
you.hasAnswer() ? you.postAnswer() : you.doSomething();
```

(or configuring) the default brace style, so I went with Egyptian in that case too. Now I'm fine with both.

answered Sep 11 '10 at 4:14

community wiki
FeatureCreep

to be used like that, `postAnswer()` and `doSomething()` should return value for ternary operator, which is often not the case: they can very well return void (no value). and also (at least in C#) result of `?:` should be assigned to some variable – ASh Aug 17 '18 at 12:01

Nearly all the responses here are saying some variation on "Whatever you do, stick with either one or two".

4

So I thought about it for a moment, and had to admit that I just don't see it as that important. Can anyone honestly tell me that the following is hard to follow?

```
int foo(int a, Bar b) {
    int c = 0;
    while(a != c)
    {
        if(b.value[a] == c) {
            c = CONST_A;
        }
        c++;
    }
    return c;
}
```

I'm not sure about anyone else... but I have absolutely zero problems mentally switching back and forth between styles. It did take me a few moments to figure out what the code did, but that's the result of me just randomly typing C-like syntax. :)

In my not-so-humble opinion, opening braces are almost completely irrelevant to code readability. There are a few corner cases listed above where one style or the other makes a difference, but for the most part, judicious use of blank lines cleans that up.

FWIW, our coding styles at work use a slightly more structured form 1 and a modified form 3. (C++)

```
if (x) {
    // blank line is required here
    ...
}
```

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



```
// blank line is required here too, unless this line is only another '}'

if (x) y = z; //allowed

if (x)
    y = z; // forbidden
```

I'm curious if those who strongly prefer form 2 would find this version of form 1 better, just because the blank line gives a stronger visual separation.

answered Sep 26 '10 at 23:43

community wiki
jkerian

4 As your example shows, indentation is so much important than braces for readable code. In fact, some languages make indentation the *only* way to nest statements! – Roger Pate Sep 27 '10 at 12:24 ✎

1 Ok, I honestly find your inconsistent example hard to read. Not REALLY hard, but harder than if it were consistent. – Almo Jul 20 '12 at 15:14

I agree with Almo. It is not a case of "is it really hard". It is a case of "it is definitely harder", even if not hard. So why make things harder? In the "toy" examples people give of course there is little difference. In my experience, When I inherit nasty code from someone else and they used method 1, quite frequently it becomes necessary to go ahead and turn it into method 2 just to be able to follow the logic. Because of the fact that it becomes frequently necessary; it automatically answers the question of which method is better and easier to understand. – Dunk Feb 28 '13 at 15:12 ✎

@Dunk: I cannot fathom code that would be noticeably improved by swapping such irrelevant details around. – jkerian Feb 28 '13 at 15:28

@jkerian-Apparently you haven't inherited much code from others who have long left either the project or company. I can't fathom not running into that situation by anyone with some years of experience. But then again, everybody's work situation is different. Also, if you have to do "formal" code reviews, formatting makes quite a difference. Being able to read the code naturally is very important. Sure I can pause and think to match up braces, but that slows the process down. One way does not require pausing, the others do. That's why I don't see why any other choice could be recommended. – Dunk Mar 6 '13 at 21:05

I'm surprised this hasn't been raised yet. I prefer the second approach because it allows you to select the block more easily.

4

When the braces begin and end on the same column and on their own line, You can select from the margin or with the cursor on column 0. This generally amounts to a more generous area with mouse selection or fewer keystrokes with keyboard selection.

I originally worked with braces on the same line as the conditional, but when I switched I found it accelerated the rate at which I worked. It's not right and slow of course, but it's something that will slow you down slightly working with braces next to your conditionals.

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



1 Old timers like me use three keystrokes to select the block no matter where the damn braces are. – [ergosys](#) Jan 28 '12 at 6:10

I personally like the second way.

2

However, the way I'm going to demonstrate is in my opinion best because it results in greatest job security! A fellow student from my university asked me for help with his homework and this is how his code looked like. Whole program looked like one single block. The interesting thing is that 95% of the bugs in the program he made came from mismatched braces. The other 5% were obvious once the braces were matched.

```
while(1){
i=0;
printf("Enter coded text:\n");
while((s=getchar())!='\n'){
    if(i%1==0){
        start=(char*)realloc(input,(i+1)*sizeof(char));
    }
    if(start==NULL){
        printf("Memory allocation failed!");
        exit(1);}
    input=start;
    input[i++]=s;}
start=(char*)realloc(input,(i+1)*sizeof(char));
if(start==NULL){
    printf("Memory allocation failed!!!");
    exit(1);}
input=start;
input[i]='\0';
    puts(input);
```

answered [Sep 11 '10 at 14:27](#)

community wiki
[AndrejaKo](#)

9 Bad, bad, I mean terrible, terrible example. The problem is not the braces! It's the crazy indentation! – [R. Martinho Fernandes](#) Sep 15 '10 at 3:20

② Martinho Fernandes I thought that closing braces and indentation go together – [AndrejaKo](#) Sep 15 '10 at 3:24

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



23:27

In fact, thinking about this motivated my own answer to this question. – [jkerian](#) Sep 26 '10 at 23:47

"95% of the bugs in the program he made came from mismatched braces" - only in interpreted languages, not compiled. – [Mawg](#) Sep 3 '14 at 7:50



My personal preference is for the first method, probably because that's the way I first learned PHP.

2

For single-line `if` statements, I'll use



```
if (you.hasAnswer()) you.postAnswer();
```

If it's not `you.postAnswer();` but something a lot longer, such as `you.postAnswer(this.AnswerId, this.AnswerText, this.AnswerType);` I'll probably revert to the first type:

```
if (you.hasAnswer) {  
    you.postAnswer(this.AnswerId, this.AnswerText, this.AnswerType);  
}
```

I will never use a line-break, and I'll never use this method if there's also an `else` statement.

```
if (you.hasAnswer()) you.postAnswer();  
else you.doSomething()
```

is a theoretical possibility, but not one I'd ever use. This would have to be turned into

```
if (you.hasAnswer()) {  
    you.postAnswer();  
} else {  
    you.doSomething();  
}
```

answered [Oct 29 '10 at 17:09](#)

community wiki
[TRiG](#)



They should not; first method for me.

2

When I look at the second one, because of the unused lines (those only having braces on it, other than the very last closing brace), it feels like it breaks the continuity of the code. I can't read it as fast because I need to take special attention to empty lines which usually mean a separation in code purpose or something like this, but in no case "this line belongs to a curly brace" (which only repeats the meaning of indentation).

Anyway, just like when you write text... adding an indentation at the beginning of a paragraph is superfluous if there is a blank line before it (double sign of paragraph change), there is no need to waste lines for braces when we are properly indenting.

Plus, as already stated, it allows to fit more code in the screen, which otherwise is a little bit counterproductive.

answered Oct 30 '10 at 5:54

community wiki
[Joanis](#)

It depends on the platform/language/conventions

2

In Java:

```
void someMethod() {  
    if (you.hasAnswer()) {  
        you.postAnswer();  
    } else {  
        you.doSomething();  
    }  
}
```

In C#

```
void someMethod()  
{  
    if (you.hasAnswer())  
    {  
        you.postAnswer();  
    }  
    else  
    {  
        you.doSomething();  
    }  
}
```

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



```
}  
}
```

In C:

```
void someMethod()  
{  
    if (you_hasAnswer()) {  
        you.postAnswer();  
    } else {  
        you.doSomething();  
    }  
}
```

I hate when Java guys use their style in C# code and vice versa.

edited Nov 6 '10 at 16:04

community wiki
2 revs
OscarRyz

3 The C style always annoyed me. Be consistent! – [Christian Mann](#) Nov 6 '10 at 7:51

▲ All I can say is that if you're a fan of method #3, you are going to be persecuted by every IDE code-formatter on earth.

1

answered Sep 11 '10 at 5:06

community wiki
[Phil Cohen](#)

▲ I use the first method simply because it is more compact and allows more code on the screen. I myself have never had a problem with pairing up braces (I always write them out, together with the `if` statement before adding the condition, and most environments allow you to jump to the matching brace).

1

▼ If you *did* need to pair up braces visually, then I would prefer the second method. However that allows less code at one time which requires you to scroll more. And that, for me at least, has a greater impact on reading code than having neatly aligned braces. I hate

But; the most important thing of all is consistency. Use one or the other - never both!

answered Sep 22 '10 at 22:03

community wiki
[gablin](#)



When I was first learning programming at 12, I put the braces on the next line because the Microsoft coding tutorials are like that. I also indented with 4-space TABS that time.

0



After a few years, I learned Java and JavaScript, and saw more braces-on-same-line code, so I changed. I also began to indent with 2-spaces SPACES.

answered Sep 21 '10 at 3:47

community wiki
[Ming-Tang](#)

5 +1, -1. Why would you NOT indent with tabs as any editor can adjust the tab length to your arbitrary length? Otherwise, you lead a lot of us who like true indents at 8 to curse your code. – [Jé Queue](#) Nov 5 '10 at 18:05



There is a 4th option that keeps the braces aligned, but does not waste space:

0



```
if (you.hasAnswer())
{
    you.postAnswer();
    i.readAnswer();
}
else
{
    you.doSomething();
}
```

The only problem is that most IDE's autoformatters choke on this.

answered Sep 28 '10 at 2:10

community wiki
[AShelly](#)

- 4 That seems horrendous. Think of the extra effort you have to go through if you want to insert a line at the top, or remove the top line. You can't just delete the line and move on, you must remember to re-insert the curly brace. – [Bryan Oakley](#) Jul 18 '11 at 14:06

lol this is awesome! :) better than first style! – [nawfal](#) Jan 14 '13 at 8:50

Apparently it even has a name. [The Horstman Syyle](#) is mentioned in [wikipedia](#). I've worked with a codebase like this, it's really not bad to use. – [AShelly](#) Jan 14 '13 at 14:29



-1



It all depends on you as long as you are not working on a project where some coding constraints or some standards have been set by the project manager that all the programmers who are working on that project have to follow while coding.

I personally would prefer the 1st method.

Also I didn't get what you wanna show by the 3rd method?

Isn't that a wrong way? For example consider a situation as..

```
if (you.hasAnswer())
    you.postAnswer();
else
    you.doSomething();
```

Now what if someone wants to add some more statements in the **if** block?

In that case if you use the 3rd method the compiler will throw the syntax error.

```
if (you.hasAnswer())
    you.postAnswer1();
    you.postAnswer2();
else
    you.doSomething();
```

answered [Sep 12 '10 at 6:05](#)

community wiki
[Chankey Pathak](#)

@FinnNk : Exactly ! – [Chankey Pathak](#) Sep 12 '10 at 15:18

- 2 If someone wants to add another statement, they can put in the braces themselves. Any programmer worth his salt really should be able to figure that out. – [Robert Harvey](#) Sep 16 '10 at 16:18

I wanted to say that his 3rd method is wrong. – [Chankey Pathak](#) Sep 16 '10 at 16:47

- 3 @Robert Harvey, I've seen very experienced coders miss adding the braces when modifying existing code. I think the problem is that indentation is a much stronger clue to meaning than braces (especially since there are multiple brace styles,) so it's quite easy to overlook the missing brace if the indentation looks like what you expect. – [AShelly](#) Sep 28 '10 at 2:13

1

2

next

