What is the prefered style for single decision and action statements? [closed]

Asked 16 years, 4 months ago   Modified 6 years, 10 months ago

Viewed 990 times

▲

1

▼

🔖

🕘

> **Closed**. This question is opinion-based. It is not currently accepting answers.
>
> ---
>
> 💡 **Want to improve this question?** Update the question so it can be answered with facts and citations by editing this post.
>
> Closed 8 days ago.
>
> Improve this question

In the case of languages that support single decision and action without brackets, such as the following example:

```
if (var == true)
    doSomething();
```

What is the preferred way of writing this? Should brackets always be used, or should their usage be left as a preference of the individual developer? Additionally, does

this practice depend on the size of the code block, such as in the following example:

```
if (var == 1)
    doSomething(1);
else if (var > 1 && var < 10)
    doSomething(2);
else
{
    validate(var);
    doSomething(var);
}
```

language-agnostic   language-features   readability

Share

Improve this question

Follow

20 Answers

Sorted by: Highest score (default)

There isn't really a right answer. This is what coding standards within the company are for. If you can keep it consistent across the whole company then it will be easy to read. I personally like

10

```
if ( a == b)    {
    doSomething();
}
else {
    doSomething();
}
```

but this is a holy war.

I recommend

```
if(a==b)
{
    doSomething();
}
```

10

because I find it far easier to do it up-front than to try to remember to add the braces when I add a second statement to to success condition...

```
if(a==b)
    doSomething();
    doSomethingElse();
```

is very different to

```
if(a==b)
{
    doSomething();
```

```
        doSomethingElse();
    }
```

see [Joel's article](#) for further details

You should ALWAYS run with braces in an if/else. Always always always. – immutabl Dec 1, 2010 at 15:11

8

I tend to use braces at all times. You can get some subtle bugs where you started off with something like:

```
if(something)
  DoOneThing();
else
    DoItDifferently();
```

and then decide to add another operation to the `else` clause and forget to wrap it in braces:

```
if(something)
  DoOneThing();
else
    DoItDifferently();
    AlwaysGetsCalled();
```

`AlwaysGetsCalled()` will always get called, and if you're sitting there at 3am wondering why your code is behaving

all strange, something like that could elude you for quite some time. For this reason alone, I always use braces.

My preference is to be consistent, e.g., if you use brackets on one block, use brackets all throughout even with just one statement:

```
if (cond1)
{
   SomeOperation();
   Another();
}
elseif (cond2)
{
   DoSomething();
}
else
{
   DoNothing();
   DoAnother();
}
```

But if you have just a bunch of one liners:

```
if (cond1)
    DoFirst();
elseif (cond2)
```

```
    DoSecond();
else
    DoElse();
```

Looks cleaner (if you don't mind the dummy method names ;) that way, but that's just me.

This also applies to loop constructs and the like:

```
foreach (var s as Something)
    if (s == someCondition)
        yield return SomeMethod(s);
```

You should also consider that this is a convention that might be more suited to .NET (notice that Java peepz like to have their first curly brace in the same line as the if).

Share Improve this answer

Follow

edited Aug 14, 2008 at 14:30

answered Aug 14, 2008 at 14:27

Jon Limjap
95.3k ● 15 ● 103 ● 153

3

Chalk this one to lack of experience, but during my seven-year stint as a code monkey I've *never* actually seen anyone make the mistake of not adding braces when adding code to a block that doesn't have braces. That's precisely *zero* times.

And before the wisecrackers get to it, no, the reason wasn't "everyone always uses braces".

So, an honest question -- I really would like to get actual replies instead of just downvotes: does that ever actually happen?

(Edit: I've heard enough outsourcing horror stories to clarify a bit: does it ever actually happen to **competent programmers**?)

Share   Improve this answer

Follow

answered Aug 14, 2008 at 19:19

Rytmis
32k ●8 ●61 ●69

> I've made this mistake, and I've found & fixed it when someone else made this mistake. Whether we were competent is open for discussion. – Jay Bazuzi Sep 20, 2008 at 17:02

It doesn't really matter, as long as you're consistent with it.

There does seem to be a tendency to demand sameness within a single statement, i.e. if there's brackets in one branch, there's brackets everywhere. The Linux kernel coding standards, for one, mandate that.

Share Improve this answer

Follow

I would strongly advocate *always* using braces, even when they're optional. Why? Take this chunk of C++ code:

```
if (var == 1)
   doSomething();
doSomethingElse();
```

Now, someone comes along who isn't really paying enough attention and decides that something extra needs to happen if (var == 1), so they do this:

```
if (var == 1)
   doSomething();
   doSomethingExtra();
doSomethingElse();
```

It's all still beautifully indented but it won't do what was intended.

By always using braces, you're more likely to avoid this sort of bug.

I personnally side with McConnell's explanation from Code Complete.

Use them whenever you can. They enhance your code's readability and remove the few and scarce confusions that might occur.

There is one thing that's more important though....Consistency. Which ever style you use,make sure you always do it the same way.

Start writing stuff like:

```
If A == true
   FunctA();

If B == "Test"
{
   FunctB();
}
```

You are bound to end up looking for an odd bug where the compiler won't understand what you were trying to do

2

and that will be hard to find.

Basically find the one you are comfortable writing everytime and stick to it. I do believe in using the block delimeters('{', '}') as much as possible is the way to go.

I don't want to start a question inside another, but there is something related to this that I want to mention to get your mental juices going. One the decision of using the brackets has been made. Where do you put the opening bracket? On the same line as the statement or underneath. Indented brackets or not?

```
If A == false {
   //calls and whatnot
}
//or
If B == "BlaBla"
{
   //calls and whatnot
}
//or
If C == B
   {
   //calls and whatnot
   }
```

Please don't answer to this since this would be a new question. If I see an interest in this I will open a new question your input.

I've always used brackets at all times except for the case where I'm checking a variable for NULL before freeing it, like is necessary in C

1

In that case, I make sure it's clear that it's a single statement by keeping everything on one line, like this:

```
if (aString) free(aString);
```

Share Improve this answer

Follow

answered Aug 14, 2008 at 14:28

Ryan Ahearn
7,934 ●7 ●53 ●57

There is no right or wrong way to write the above statement. There are plenty of accepted coding styles. However, for me, I prefer keeping the coding style consist throughout the entire project. ie. If the project is using K&R style, you should use K&R.

1

Share Improve this answer

Follow

answered Aug 14, 2008 at 14:32

Eldila
15.7k ●23 ●59 ●62

Ruby nicely obviates one issue in the discussion. The standard for a one-liner is:

1

```
do_something if (a == b)
```

and for a multi-line:

```
if (a == b)
  do_something
  do_something_else
end
```

This allows concise one-line statements, but it forces you to reorganize the statement if you go from single- to multi-line.

This is not (yet) available in Java, nor in many other languages, AFAIK.

Share Improve this answer

Follow

answered Aug 14, 2008 at 16:29

James A. Rosen

65.2k ● 62 ● 184 ● 263

1

As others have mentioned, doing an if statement in two lines without braces can lead to confusion:

```
if (a == b)
    DoSomething();
    DoSomethingElse(); <-- outside if statement
```

so I place it on a single line if I can do so without hurting readability:

```
if (a == b) DoSomething();
```

and at all other times I use braces.

Ternary operators are a little different. Most of the time I do them on one line:

```
var c = (a == b) ? DoSomething() :
DoSomethingElse();
```

but sometimes the statements have nested function calls, or lambda expressions which make a one-line statement difficult to parse visually, so I prefer something like this:

```
var c = (a == b)
    ? AReallyReallyLongFunctionName()
    :
AnotherReallyReallyLongFunctionOrStatement();
```

Still more concise than an if/else block but easy to see what's going on.

Share   Improve this answer

Follow

answered Aug 14, 2008 at 19:14

Adam Lassek
35.5k ● 14 ● 93 ● 109

Sun's Code Conventions for the Java programming Language has this to say:

1

The if-else class of statements should have the following form:

```
if (condition) {
    statements;
}

if (condition) {
    statements;
} else {
    statements;
}

if (condition) {
    statements;
} else if (condition) {
    statements;
} else {
    statements;
}
```

Share Improve this answer

Follow

answered Sep 16, 2008 at 22:42

Eek

Our boss makes us put { } after a decision statement no matter what, even if it's a single statement. It's really annoying to add two extra lines. The only exception is ternary operators.

I guess it's a good thing I have my code monitor in portrait orientation at 1200x1600.

Share Improve this answer

answered Aug 14, 2008 at 14:27

I prefer

```
if (cond)
    {
    //statement
    }
```

even with only a single statement. If you were going to write something once, had no doubts that it worked, and never planned on another coder ever looking at that code, go ahead and use whatever format you want. But, what does the extra bracketing really cost you? Less time in the course of a year than it takes to type up this post.

Yes, I like to indent my brackets to the level of the block, too.

Python is nice in that the indentation defines the block. The question is moot in a language like that.

0

Share Improve this answer

Follow

answered Aug 14, 2008 at 14:42

Baltimark

9,272 ● 12 ● 38 ● 35

I tend to agree with Joel Spolsky on that one with that article ([Making Wrong Code Look Wrong](#)) with the following code example :

0

```
if (i != 0)
bar(i);
foo(i);
```

Foo is now unconditionnal. Wich is real bad!

I always use brackets for decision statements. It helps code maintainability and it makes the code less bug prone.

Share Improve this answer

Follow

edited Aug 14, 2008 at 14:46

answered Aug 14, 2008 at 14:33

Pascal Paradis
4,295 ● 5 ● 38 ● 50

I use curly braces around every statement if and only if at least one of them requires it.

Share Improve this answer

Follow

answered Sep 16, 2008 at 21:50

spitzak

In Perl if you are doing a simple test, sometime you will write it in this form:

```
do_something if condition;

do_something unless condition;
```

Which can be really useful to check the arguments at the start of a subroutine.

```
sub test{
  my($self,@args) = @_;

  return undef unless defined $self;

  # rest of code goes here

}
```

Share Improve this answer   answered Sep 17, 2008 at 18:02

Follow   Brad Gilbert

                                     34.1k ● 11 ● 79 ● 130

The golden rule is that, when working in an existing project, follow those coding standards.

When I'm at home, I have two forms.

The first is the single line:

```
if (condition) doThis();
```

and the second is for multiple lines:

```
if (condition) {
    doThis();
}
```

Share Improve this answer

Follow

I used to follow the "use curly braces always" line like an apparatchik. However, I've modified my style to allow for omitting them on single line conditional expressions:

0

```
if(!ok)return;
```
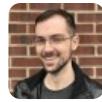
For any multistatement scenario though I'm still of the opinion that braces should be mandatory:

```
if(!ok){

    do();

    that();

    thing();
}
```

Share Improve this answer

Follow

answered Aug 14, 2008 at 15:08

t3rse
10.1k ● 11 ● 60 ● 85

I'm not sure although I don't doubt there are apparatchiks on StackOverflow. – t3rse Mar 13, 2009 at 17:09