What is the proper way to format code?

Asked 15 years, 9 months ago Modified 13 years, 3 months ago

Viewed 12k times

19

When I write code, I try to group lines of similar code together, then leave a blank line and write another block.

I believe this contributes to the neatness and readability of the code.

I'm not a big fan of bunching stuff together without any line spacing. It looks like hell, it's hard to read and it's difficult to follow.

One of the teachers I had, downgraded one of my assignments because I had spaced my code logically. He said, 'When you have to read code all day in the real world, you won't put this line spacing in and you'll be thanking me." Of course, I never did and never will thank him.

Now that I'm in the real world, most of the code files I see that have absolutely no line spacing are poorly written and poorly thought out.

This is probably more prevelant in VB type languages than in C type languages, but the same concept applies.

Two questions come to mind:

- Where do you leave a blank line in your code?
- How much line spacing is too much?

formatting

Share

Improve this question

Follow

edited Mar 12, 2009 at 14:23

GEOCHET

21.3k ● 15 ● 77 ● 99

asked Mar 12, 2009 at 13:12

user72491

3,665 ● 6 ● 25 ● 18

1 "He was an ass (WHICH IS WHY he's teaching)" ?? wtf? – Romain Linsolas Mar 12, 2009 at 13:20

1 Hehe. Typo, but I'm gonna leave it, just because. You know the saying, 'Those who can't do teach.' yada yada yada – user72491 Mar 12, 2009 at 13:24

".. and poorly thought out" .. that's actually a good point - dividing into logical groupings mean you're considering each portion of the code individually, instead of as some behemoth method – nailitdown Mar 12, 2009 at 13:52

Edited: removed the completely irrelevant comment about the teacher in question. Please stick to the facts when you ask a question. – Carl Seleborg Mar 12, 2009 at 14:00

Shouldn't this be community wiki? – David Thornley Mar 12, 2009 at 17:18

22 Answers

Sorted by: Highest score (default)

19

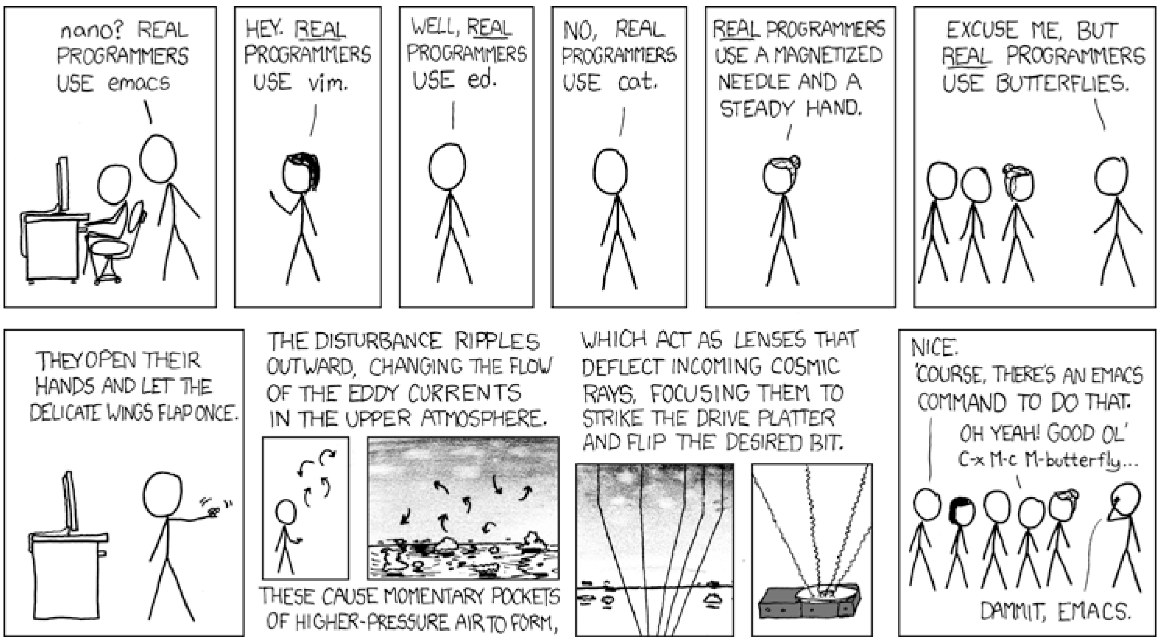
I follow Microsoft's Guidelines for C#.

Edit: The standard for C# is **Don't fight the IDE**. If you hit `CTRL K+D`, the IDE will automatically put blank lines in between code sections.

To follow that up, if you look at C# sample code on MSDN or anywhere else, there's normally a blank line in between each logically placed group. So There will be a blank line after all your member variables, a blank line after each method, etc.

In response to the comments expressing shock and horror that I use an IDE for C# programming:

REAL PROGRAMMERS


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

Share Improve this answer Follow

edited Feb 8, 2017 at 14:07
Community Bot
1 ● 1

answered Mar 12, 2009 at 13:15
George Stocker
57.9k ● 29 ● 180 ● 238

Also, if you go with what you see, you will need to use their wack HTML indentation rules... – leppie Mar 12, 2009 at 13:25

Any spec that defines where you can and can't put spaces is a big negative for me (except for marking up API documentation where consistency is key). Spaces, or lack of, don't increase or decrease the understandability of the code. Guidelines should be about understandability of code. – Skizz Mar 12, 2009 at 13:27

@Skizz, FAIL! Sure it does decrease usabillity if you have less readabillity. – Filip Ekberg Mar 12, 2009 at 13:37

I dooon't think Ctrl+K+D adds spacing between logical blocks of code within a method, which imo is what the question is about. – Peter Lillevold Mar 12, 2009 at 13:58

@Peter Lillevold: The author isn't clear. So I have to interpret. My answer is that the IDE tells us where we ought to place blank lines, and where it says not to, we shouldn't. – George Stocker Mar 12, 2009 at 13:59

11

I think I do something similar, but there are no hard rules. I like code to spaced in 'paragraphs' of grouped/related logic.

Code with no extra line spaces is terrible to read.

Share Improve this answer

Follow

answered Mar 12, 2009 at 13:15

leppie
117k ● 18 ● 200 ● 300

10

It sounds like I space lines of code similar to you.

But this is an irrelevant, personal preference and everybody is going to have thier own 'right way' to do it. The most important thing, IMHO, is to **adapt the style of the environment** you are going into.

Also, you will find code like this in the 'real world' ... but it sounds like you have higher aspirations. ;-)

EDIT: ... not higher aspirations than the 'real world', but higher than the mediocre crap common in the 'real world'. ... if you do happen to have higher aspirations than the 'real world' however, you might want to see a professional. ;-)

Share Improve this answer

Follow

answered Mar 12, 2009 at 13:23

John MacIntyre
13k ● 13 ● 69 ● 107

8

My rule of thumb is:

> Put a single blank line between blocks of code that can be described with one comment.

But generally I agree that many large functions with lots of white space should be broken into smaller functions.

Share Improve this answer

Follow

answered Mar 12, 2009 at 13:21

Joel

19.4k ● 3 ● 66 ● 84

6

Since we don't have an example of your personal idea of "logical" line spacing, we can't really say whether or not you deserved to be downgraded.

I tend to group similar statements or steps in a sequence together before spacing (such as variable declarations, looping, etc.)

Share Improve this answer

Follow

edited Mar 12, 2009 at 14:23

answered Mar 12, 2009 at 13:15

TheTXI

37.9k ● 10 ● 88 ● 110

4

Basically the same thing as everyone else is saying. COBOL had very distinct rules about what a sentence and a paragraph was. I guess, in the back of my mind, I follow those. If you have a large IF statement, you didn't put a period until the very end of the nest. Similarly, I put a blank line after my last } // end if

And yes, I put the // end if, // end for, // end method stuff in there. Some of the more vocal people I know don't like it, but I like it. They say you shouldn't make your if-statements huge and that you're probably coding wrong if you **NEED** the // end if stuff, and I don't **NEED** it, but I do find it makes it easier to read. Call me old fashioned, OCD, whatever.

Share Improve this answer Follow

answered Mar 12, 2009 at 13:23
belgariontheking
1,349 ● 1 ● 10 ● 15

3

Vertical space is usually at a premium relative to horizontal space, so **excessive spacing** usually isn't a good idea. I do think it's a good idea to logically separate blocks of code with a **single blank line**.

Sounds like your teacher is mostly a jerk. As the saying goes, "those who can, do; those who can't are ~~on StackOverflow~~ teach." ;)

Share Improve this answer Follow

answered Mar 12, 2009 at 13:15
John Feminella
311k ● 48 ● 346 ● 361

I think you hit the spot there: a single blank line, no more.
– leppie Mar 12, 2009 at 13:16

I use line spacing very similar to you. The code I've found in the real world often tend to be laid out similarly, or at

3

least not so bunched together that it can't be read.

Share Improve this answer
Follow

answered Mar 12, 2009 at 13:16
Andy
3,854 ● 25 ● 28

3

Unless you go nuts with the vertical spacing, I see no problem with separating out your code. I know people like to toss around the idea of using little methods, but something even if your method does one single thing, that single thing could take a lot of code to accomplish. Not everything can be done in a screenful of code.

Share Improve this answer
Follow

answered Mar 12, 2009 at 13:23
Tundey
2,965 ● 1 ● 24 ● 28

3

Your teacher was probably half right in that in the real World you won't have the line spacing. Certainly on the Big Ball of Mud code bases I come across you're lucky if you get a line space let alone comment of explanation.

As an aside the 73 year old programmer that wrote most of this Big Ball of Mud is still working there and his explanation is that the binaries need to be kept as small as possible, I didn't bother to check whether the compilers of 20 to 30 years ago were that inefficient they couldn't strip whitespace but I'm somewhat skeptical.

I use single blank lines to break up logical sections in my own code as I find it greatly enhance readability.

As always the best test with these type of readability concerns is to grab some tricky piece of code you wrote and haven't looked at for over a year and see if you can get a grasp on it quickly. If you can then your code will be better than most of what you'll see in the real World!

Share Improve this answer Follow

answered Mar 12, 2009 at 13:36

sipsorcery
30.7k ● 25 ● 106 ● 158

Sounds like that 73-year-old a) doesn't know how compilers work, and b) does not heed the "premature optimization is the root of all evil" principle. – Juliet Mar 12, 2009 at 14:30

---

1 Back when I wrote BASIC on my TRS-80, the interpreter did essentially no preprocessing, so whitespace was a performance hit. Literally every system I've used since has had some sort of preprocessing, so whitespace does not slow execution down or increase executable size. – David Thornley Mar 12, 2009 at 17:21

3

I consider code as an article. Have you ever tried to read 2 pages of article which has no paragraph or line spacing in it?

I agree with you, not having line spaces between logical groups is just insane.

Share Improve this answer Follow

edited Mar 12, 2009 at 14:39

answered Mar 12, 2009 at 14:02

dr. evil
27.2k ● 37 ● 134 ● 202

3

> One of the teachers I had, downgraded one of my assignments because I had spaced my code logically. He said, 'When you have to read code all day in the real world, you won't have this line spacing and you'll be thanking me."

Unless you're seperating blocks with 5 or 10 lines of whitespace (which would drive anyone nuts), you're instructor is just being an ass.

Coding standards are not etched in stone, and they are certainly not the same for all software shops. All companies have different coding standards. For what its worth, some of the coding standards at my company state explicitly "visually seperate logically related blocks of code using a single blank line".

Although we should strive not to write 200-line long methods, its still very common for all of our short methods to contain more than one control flow element, and we should understand that vertical whitespace is just as important as horizontal whitespace in readability. You can satisfy the "single method, single purpose" principle even if you put a blank line between a for-loop and an if-statement in the same method.

[Edit to add] And just a few more comments:

1) Its very presumptuous that several people in this thread are assuming that the OP is writing 200-line methods, or that there is a necessary correlation between adding blank lines and writing sloppy methods.

2) For what its worth, while the OP's instructor is utterly wrong in assuming that *his* coding standards are the the same everywhere. However, you should treat the programming course as its own little software shop with its own standards, so your code should be written in a way which follows those standards.

If your instructor is grading you based on how well your code conforms to coding standards, then insist on getting a list of standards. I know if *my* grade was docked because it didn't conform to standards that the instructor had never given to me (or if his standards say "prefix variables with their datatype"), heads would be rolling.

Share Improve this answer Follow

edited Mar 12, 2009 at 14:40

answered Mar 12, 2009 at 14:14

Juliet
81.4k ● 46 ● 199 ● 229

3

I've just been working on some code that goes in the opposite direction; each statement is separated from the next by a blank line. The authors also liked to use four lines of comment right aligned at column 60 instead of a one-line comment indented at code level. It is hard to read, and tedious to fix. Another feature of the code (C code), the break from a previous case is 'attached' to the case of the next, but there's a blank line after the case, separating it from its code. Ick!

Blank lines around blocks of code are good. Not having too many blocks of code in a single function is good. Blank lines around every line of code is unpleasant. Too much, or too little, of a good thing is a bad thing.

Share Improve this answer Follow

answered Mar 12, 2009 at 17:04

Jonathan Leffler
752k ● 145 ● 946 ● 1.3k

1 My thoughts exactly. We have code written by a (former) programmer that fits your description to the letter, and the

"blank line between every line of code" is indeed annoying and doesn't help anyone. – Mike Spross Apr 27, 2009 at 0:49

2

We have a coding standart that states that we should not put more than one blank line in a row. All the other is up to developer. In practice we do as you say - try to group logically connected lines into groups and isolate them with blank lines.

Share Improve this answer Follow

answered Mar 12, 2009 at 13:15

sharptooth
170k ● 108 ● 535 ● 1k

2

Use little methods, in order to have a low [cyclomatic complexity](). High cyclomatic complexity in methods generally means that your method must be divided into several others sub methods, which will be more easier to read, and more easier to test!

I think that your code is facing this kind of problem.

Generally, a method with 100+ lines is too big and too complex, and then must be refactored.

To summarize, instead of breaking code with line spaces, break it into separate methods...

Share Improve this answer Follow

answered Mar 12, 2009 at 13:19

Romain Linsolas
81.5k ● 51 ● 204 ● 276

Cyclomatic complexity doesn't take into account the big dumb case-type statements that are sometimes useful. They have high C.C., but are easy to understand. – David Thornley
Mar 12, 2009 at 17:23

2

Spaces help to make your code more readable and I think the consensus here is that they are not for the most part a bad idea. But nobody seems to have pointed out that along with the blank line a comment may not be amiss.

```
/* This section preps the widgets for display */
block
of some
code

/* This section passes the widgets to the display
handler */
and
so
on
```

Remember most code will be read many times over it's life so anything you can do to make life easier for future maintainers is a great plus.

Share Improve this answer Follow

answered Mar 12, 2009 at 13:47
Jeremy French
12.1k ● 6 ● 47 ● 72

1) I agree with you about line spacing

2

2) In my office there's a lot of lack of line spacing because "you get to see more of the code on one page that way." They also put multiple statements on one line, and over (IMHO) use ?:... I hate it.

3) I do disagree with the phrase "That's why he is a teacher". as an (ex) teacher, I have to say that I would downgrade people for NOT putting space between sections before I took off points for putting space. I don't think I did either. My point is that him being an ass is orthogonal to his being a teacher. (EDIT: that section got edited from the original, but I'm leaving it here to maintain the rule of 3...)

Don't malign teachers!

Share Improve this answer Follow

edited Mar 12, 2009 at 16:46
Jonathan Leffler
752k ● 145 ● 946 ● 1.3k

answered Mar 12, 2009 at 14:02
Brian Postow
12.1k ● 21 ● 85 ● 133

2

Not to sound pretentious, but I think of blank lines in code as having a similar function to phrase marks in music notation, or line breaks in poetry: they communicate intent to the reader without changing the semantics of the content.

Here's a method I just copied and pasted from a project:

```
public static void Startup(bool startupUser)
{
    URI = "";
    AutoComplete = new AutoCompleteWrapper();
    SessionToken = Guid.Empty;
    ExternalDataSetLock = new object();

    ConfigDS.Init();
    CalendarDS.Init();
    CalendarDSBuilder.Init();

    if (startupUser && UserName != null)
    {
        string autoCompleteFilename =
Path.Combine(UserFolder, "autocomplete.xml");
        AutoComplete.Load(autoCompleteFilename);
    }
}
```

What do the blank lines do here? They clarify that there are basically three different kinds of initialization taking place in this method. If I add a new property that needs to be initialized on startup, I know where I'm going to put the line that initializes it. The blank lines also hint at how I intend to refactor this function if it doubles in size.

The risk of using blank lines is the same as the danger of any other kind of implied meaning: it's implied. The implication you're making when you write the code may not be the implication that the person reading the code understands.

But mercy, that's no reason not to use them.

answered Mar 12, 2009 at 17:34

Share Improve this answer Follow

Robert Rossney
96.6k ● 24 ● 148 ● 218

2

I know this is an old question, but I tripped over it doing a search for something else, and had an observation: look at the existing answers. They're generally short, yet they contain blank lines.

That's what paragraphs are. As in writing, lines are a quick, visual way to separate concepts. It doesn't mean each paragraph above should be broken into a separate answer; it doesn't mean each code "paragraph" should be broken into a separate method.

It's a silly complaint. If it were still possible to talk to that professor, I'd explain what "wall of text" means.

Share Improve this answer Follow

answered Sep 15, 2011 at 14:37

spinn
324 ● 1 ● 6

1

In my opinion, adding a blank space to your code is a hint that you should split your function into smaller parts. Cleaning up your code by adding spaces is preferable to a mile long list of unseparated lines. Cleaning up your code by separating out smaller functions is better.

Share Improve this answer Follow

edited Mar 12, 2009 at 13:51

answered Mar 12, 2009 at 13:15

Magnar
28.8k ● 8 ● 61 ● 65

Hrm... So adding a blank line before and after declaring the variables at the start of a function indicates that they should be in a new function? I don't think so. – user72491 Mar 12, 2009 at 13:17

I disagree with this. If a function/method gets too long *including* blank lines, that's a smell. Omiting blank lines to cram more into vertical space is also a smell. – slim Mar 12, 2009 at 13:19

Single purpose is a better guide than blank lines but they are a damn good hint when you've added one to logically separate functionality. :) – Lazarus Mar 12, 2009 at 13:21

Putting this back to 0, although I don't entirely agree with the comment, I think there's at least some sense of truth and don't think it deserves -1. – Daniel Sloof Mar 12, 2009 at 13:45

@treant: If you declare so many variables at the start of your function that you feel the need to separate them out, that's another hint that your functions are too large. – Magnar Mar 12, 2009 at 13:47

1

I'm not going to begin to suggest any specifics. There are plenty of good suggestions here and elsewhere on the interweb.

But I would say be consistent. At least within an application or module - since I know I refine my approach from time to time, but I try to keep all code in the same places looking the same.

If you are consistent, it's easy enough for any other developer to pick your tempo and style.

Share Improve this answer

Follow

answered Mar 12, 2009 at 17:26

CJM

12k ● 20 ● 79 ● 115

0

I've started following the Microsoft Design Guidelines found here:

[Design Guidelines for Class Library Developers]()

Share Improve this answer

Follow

answered Mar 12, 2009 at 13:16

Lazarus

43k ● 4 ● 45 ● 54