How To Write Pseudo-code

#webdev #beginners #advice #pseudocode

Milecia McG Apr 25, 2019 *Updated on Aug 26, 2019* · 3 min read

Have you ever had a really complex programming issue? An issue where you can probably write out the logic, but you

♡ 67 18 93 •••

What it is

Pseudo-code is "language" where you can write all of your coding logic without writing one line of language-specific code. You see this a lot in algorithm research, especially machine learning algorithms. That doesn't mean you can't use it for web development.

Why you would use it

There are projects that are so massive that if you don't take the time to write a little pseudo-code, you could end up lost in a sea of implemented code. When you write some pseudo-code, it gives you a chance to really think through potential issues. You're able to look at pure logic and program flow without worrying about how your code runs.

Writing pseudo-code before you start typing real code will also help you finish your projects faster. Think of it as a blueprint. You know where everything needs to go and how everything works together. So when you get to the actual building phase, you don't have as much to think about because you've already thought through what you need to do.

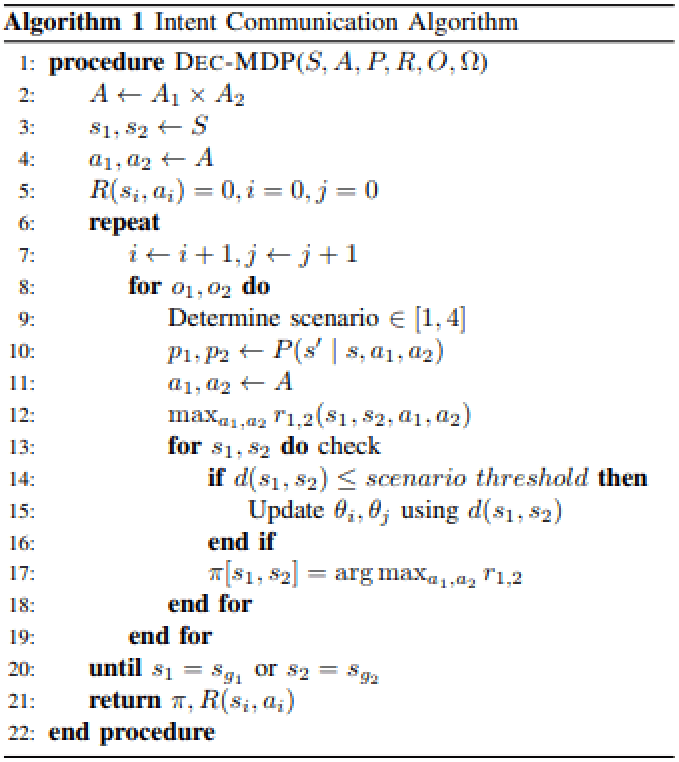
The best part is that pseudo-code doesn't depend on any programming language. That logic you just wrote out can be taken by anyone and translated into their language of choice. It gives you the freedom to reuse and improve the architecture of the application that you're building.

One of the more subtle uses of pseudo-code is to share it with other people. Sometimes you'll have a specific piece of logic that can be used across multiple projects, but they are all in different programming languages. When you have the pseudo-code available, you can give it to other programmers and they can write that logic in whatever language they need to.

Another great feature is that you can write pseudo-code in any format you like. You could use the academic format. It's incredibly structured and detailed, but it tends to have a lot of math involved. Or you can write out a simple outline of what you expect your code to do.

♡ 67 18 93 •••

Here's an example of some pseudo-code I wrote in one of my academic papers:



Algorithm 1 Intent Communication Algorithm

```
procedure Dec-MDP($S, A, P, R, O, \Omega$)
    $A \leftarrow A_1 \times A_2$
    $s_1, s_2 \leftarrow S$
    $a_1, a_2 \leftarrow A$
    $R(s_i, a_i) = 0, i = 0, j = 0$
    repeat
        $i \leftarrow i + 1, j \leftarrow j + 1$
        for $o_1, o_2$ do
            Determine scenario $\in [1, 4]$
            $p_1, p_2 \leftarrow P(s' \mid s, a_1, a_2)$
            $a_1, a_2 \leftarrow A$
            $\max_{a_1, a_2} r_{1,2}(s_1, s_2, a_1, a_2)$
            for $s_1, s_2$ do check
                if $d(s_1, s_2) \leq$ scenario threshold then
                    Update $\theta_i, \theta_j$ using $d(s_1, s_2)$
                end if
                $\pi[s_1, s_2] = \arg\max_{a_1, a_2} r_{1,2}$
            end for
        end for
    until $s_1 = s_{g_1}$ or $s_2 = s_{g_2}$
    return $\pi, R(s_i, a_i)$
end procedure
```

I'll be the first to admit that this is probably overkill for web development. If you find yourself using LaTex to write your pseudo-code, you might be making it more complicated than you need to. Odds are strong that a quick little write up in Word or even Notepad will be sufficient.

Here's an example of some simple pseudo-code I wrote for one of my web development projects:

```
IF userlogin = true
    API call to get user data
    Assign data to variables
    Re-route user to dashboard
ELSEIF userlogin failed more than 3 times
```

67 18 93

```
    Re-route user to home page
ELSE

    Log bad login attempt
    Show error message
    Clear login form
```

You don't have to be super technical with your pseudo-code, but typically the more detail you can pack into it the easier it is to write the actual code. Think about it like you're writing an outline for your program. It gives you a chance to really think through what you are trying to accomplish and you can see exactly how all of the code comes together.

The main things you need to focus on with pseudo-code are:

- The logical flow of your program
- The details of the complex parts of your program
- A consistent format

It really doesn't take much to write pseudo-code besides some hardcore thinking. As you write it, you'll start to see places where you could add more detail or places where you can take away some detail. Remember, this is mainly for your use so write it in a way that makes sense to you.

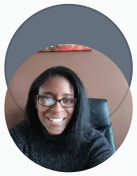
Personally, I love pseudo-code. It helps me keep my train of thought clear when I start typing the real code. When you have all of your logic already planned out, you get so much more time to experiment with performance and optimization. On top of that, you don't have to think as hard when you're deep into the code writing (yay for planned laziness).

What do you think? Do you think pseudo-code is worth the time or would you rather just jump into the code?

Hey! You should follow me on Twitter because reasons: https://twitter.com/FlippedCoding

♡ 67 18 93 •••

Posted on Apr 25, 2019 by:



Milecia McG

@flippedcoding

Starting classes soon! | Software/Hardware Engineer | International tech speaker | Random inventor and slightly mad scientist with extra sauce

Follow

Discussion

Subscribe

DEV

Add to the discussion

▼

mlaj

Apr 25 '19 •••

I usually write pseudo code first, then I code around it. The pseudo code becomes basic comments for my actual code.

I also make a simplified version with checkboxes that I put on our task manager for my superiors to see. They can then understand where I'm going with this task.

Great article !

♡ 67 18 93 •••

s.shahlan Nov 24 '19 •••

Can share some example?

♡ 1 Reply

▼

crashley1992 Apr 25 '19 •••

I love reading your posts. I'm newer to programming and I appreciate how you cover a lot of things that aren't brought up in tutorials. I think pseudo-code is something I need to get in the habit of doing more. Especially if I step away from a project for awhile and come back to it.

♡ 6 Reply

▼

Patrick Kilter Apr 25 '19 •••

As mentioned in another comment, I write my pseudo code as comments in my script / program and fill it in Afterwards.

So you know what you thought at this moment even a few years later.

And another people will understand what you did too :)

♡ 5 Reply

▼

Milecia McG Apr 25 '19 •••

♡ 67 18 93 •••

♡ 5 Reply

Ariane

Apr 25 '19 •••

I'm just new to coding and I love this approach. Thank you for this post!

♡ 6 Reply

▼

Ariane

Apr 25 '19 •••

I also think the academic pseudo-code is interesting

♡ 3 Reply

▼

Milecia McG

Apr 25 '19 •••

Thank you! I'm glad it's helpful.

♡ 2 Reply

▼

Manuel Fernandez

Apr 25 '19 •••

I've been coding for about a year and a half, and today I bumped into a really complex task at work, which involved (as usual) reading another coworker's code. Luckily, I had read this post on my lunch time, and it really helped me to better understand the existing codebase (since I was able to grasp the logic behind the problem by using pseudo code). Thank

♡ 67 18 93 •••

♡ 5 Reply

▼

John 'BBQ' Wollner

May 12 '19 •••

This a really great post Milecia! I use pseudo code all the time and have for years, particularly on a greenfield project where I'm staring at a blank page/editor just to get mu thoughts on the page in a logical format w/o the burden of syntax, frameworks, OS, and all the other details that one must consider in coding.

Even if I have the general design in my head, there are often some complex transformation or formulae that I need to literal walk through on paper to validate my thinking and firm up the approach. It gives me the freedom to express my approach in "brain language" rather than computer language which I find very freeing.

When I learned to program the second (I learned in junior high first, then forgot most of it lol), we used a tool called an IPO chart which stood for INPUT-PROCESS-OUTPUT which had 3 columns listing basically what you'd expect, but the process was often pseudo code (sometimes flow charts but only for the handed in homework lol). I often take the same approach today and I've been coding for 37y - One is never too experienced to not need a helping hand.

Thanks again - keep posting!

♡ 4 Reply

▼

MarkAurit

Apr 26 '19 •••

Pseudo-code is a great tool for working with large code bases that you didnt write in the first place, especially if they are complex. Or your own, 6 months later after you've moved on! As a senior I'll often tell younger devs to pseudo-code a task and bring it back to me before writing code; if their pseudo-code is correct I know they understand the problem and have much higher confidence they will write correct code.

♡ 67 18 93 •••

Frank Puffer

Apr 26 '19 •••

Yeah, but why not let them write real code and do a code review soon. It might cause a bit more work for you but it is safer and the junior will probably learn more. Coding is not like building a house. Code can easily be modified. Version management tools exist. Refactoring exists.

♡ 1

Reply

Andrei Andreev

Apr 27 '19 •••

Different level of abstraction which allows for ultra efficient communication and review of logic.

♡ 1

Reply

Frank Puffer

Apr 26 '19 •••

I completely disagree.

Pseudocode might have made sense in the age of assembler, FORTRAN and C programming but nowadays the two main reasons for using it are probably:

1. Not mastering your programming language.

2. Not knowing about refactoring - nobody gets a non trivial task right at the first try. Code is not something you write

♡ 67 18 93 •••

when you are not yet ready to code: Draw diagrams write specification documents, talk to people, build a prototype, write unit tests (TDD).

What sucks most about pseudocode is that it will not provide any feedback.

Why does this matter to me after all? Because I think that it is much more helpful in the long run to put your energy into the real code.

♡ 2 Reply

▼

codelitically_incorrect Apr 25 '19 •••

Why did you bother posting academic pseudo code? It ruins the opportunity to share the article to someone who is learning

♡ 2 Reply

▼

Milecia McG Apr 25 '19 •••

It's just there as another example of what pseudo-code can look like and just how complex it can be if you need it. ☺

♡ 3 Reply

▼

Claire McCrea Sep 11 '19 •••

I see both sides here.

♡ 67 18 93 •••

Maybe it would help if the examples were used in a different way? Like, “pseudo-code can be simple [example] or complex [example], but it’s overall goal is to...”

Thanks for the article! It answered a few questions I had 😊

♡ 1 Reply

▼

KerushaG

Apr 27 '19 •••

Hi, nice article but I'm a beginner and I know I should nail this habit in now. I'm doing a bootcamp and the exercises are getting a bit more challenging, when I try to write pseudo code, I get trapped by 2 things. 1) I come from a more academic background, lots of essay writing, and so my pseudo code never comes out in simple digestible chunks and 2) sometimes I am not even sure how I will need to do what I need to do and so I end up just jumping right in and playing around, I never go back to writing pseudo code and if I made any attempts if offers me no guidance, because I'm like learning on the go about what works and doesn't. Any advice on how to improve on this?

♡ 1 Reply

▼

Alits1988

Feb 25 •••

If I ever face any problem in writing something then I always take the expert writing tips which I can easily read by look at here now and I think it is the best writing service I have ever used in my life. I will suggest every student try this.

♡ 1 Reply

♡ 67 18 93 •••

Alits1988 Jul 21 •••

Well, you're thinking well because everybody needs it. According to your post, I can give you more information. But for that, you will study, and

♡ 1 Reply

▼

EDDISON HAYDEN LEWIS Jun 5 '19 •••

Yes, Pseudo-codes are exactly necessary initially as it would allow the team(s) to brainstorm broadly as opposed to jumping straight into code.....

♡ 2 Reply

▼

yriiarutiunian Apr 25 '19 •••

I can barely remember the last time I had to write pseudo-code. Usually I did, when I had to write something really difficult to just imagine in my mind. Nowadays, the stuff I code are pretty simple

♡ 2 Reply

▼

Olimpio Apr 26 '19 •••

Love it. Clearly is the best way to approach an algorithmic problem

♡ 67 18 93 •••

▼

Frank Puffer

Apr 26 '19 •••

What makes you think so? Does real code really suck that much? If yes, shouldn't we better find out why that is so and try to do something against it?

♡ 1

Reply

▼

Tersarset

Aug 19 •••

Thanks for providing good information, I like to do coding so your post obviously helpful for me. After my schooling I decided to do coding work but my friends said that I can;t do that just because I hate writing. But they didn't know I used edu birdy to took writing help and rest my free time I spent in to learning coding. Now they are free and i'm working just because I did my 100%.

♡ 1

Reply

Code of Conduct • Report abuse

More from Milecia McG

How to Measure DevOps Metrics

#devops #webdev #cicd

♡ 67 18 93 •••

#devops #webdev

Learn From The Pros! DevOps Round-Up

#devops #webdev

♡ 67 🦄 18 🔖 93 •••