

[Get started](#)[Open in app](#)**BetterProgramming**[Follow](#)

200K Followers



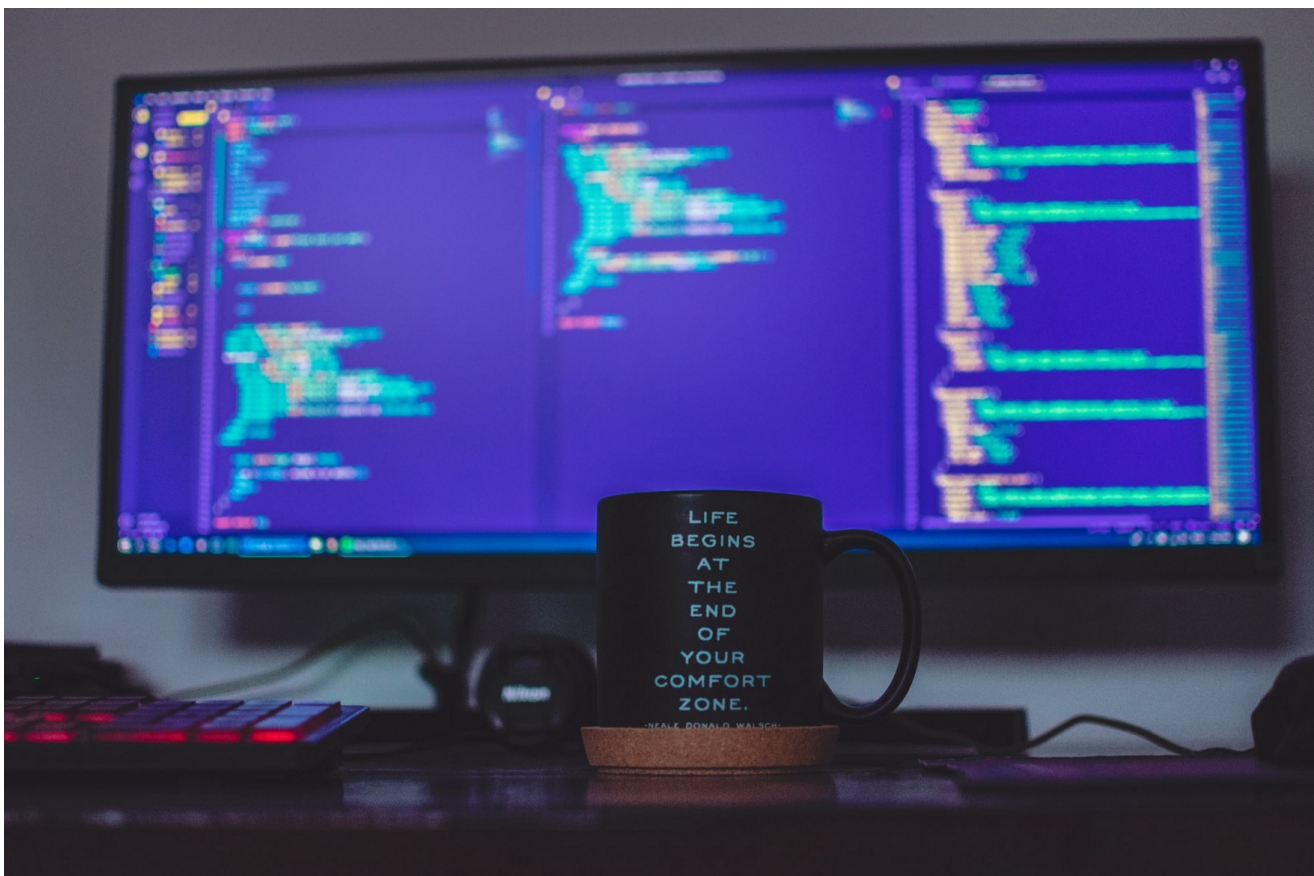
You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

15 Reasons Why I Use Emacs, With GIFs

VS Code has nothing on Emacs



Dominik Tarnowski Mar 10, 2020 · 5 min read ★



https://unsplash.com/photos/vc3iVL_znJ8

1. Org Mode

You've probably seen/used Markdown before. Org mode has a similar syntax and is also supported by GitHub READMEs, but unlike Markdown it has one specification and it doesn't have hundreds of flavors.

Keep reading to see what can actually be done, or skip to the following sections:

- 3. Export Org to presentations and documents.
- 5. In-editor LaTeX preview.
- 6. Org agenda
- 9. Executing code blocks.
- 11. Org tables.

2. Vim Bindings

Doom Emacs has great Vim bindings, and so, so many great leader features out of the box.

Unlike VS Code, the Vim bindings plugin isn't terribly slow, it feels as fast as native, not to mention that they work everywhere; the terminal, opening files, search and replace, and much more, but more on that later.

3. Exporting Org to Presentations and Documents

Thanks to many Org mode extensions, you can export your documents to:

- LaTeX PDF
- [Beamer/Reveal.js](#) presentation
- HTML
- [iCalendar](#)
- Markdown

And the conversion process is almost instantaneous.

1 BINARY SEARCH

- requires ordered list

```
# find index of value in the array
def f(arr, v):
    print(arr)
    # middle item, for even size round up
    # e.g. arr = [7,8] → mid = 2nd item
    mid = len(arr) // 2
    # found it
    if arr[mid] == v:
        return mid
    # its on the left
    if arr[mid] > v:
        return f(arr[:mid], v)
    # its on the right
    return f(arr[mid:], v) + mid
```

reveal.js preview

4. Searching Inside of Emacs

[Swiper](#) can be used to fuzzy search inside of files, with highlighting of the selected lines in real-time.

```
10 package main
9
8 import (
7     "context"
6     "fmt"
5     "time"
4
3     "github.com/shurcool/githubv4"
2     "golang.org/x/oauth2"
1 )
11
1 const token = "....."
2
3 type User struct {
4     ID          githubv4.ID
5     Login       githubv4.String
6     Email       githubv4.String
7     CreatedAt   githubv4.DateTime
8     Following   struct {
9         TotalCount githubv4.Int
10    }
11    Followers struct {
12        TotalCount githubv4.Int
13    }
14    Repositories struct {
15        TotalCount githubv4.Int
16        Nodes      []struct {
17            PushedAt githubv4.DateTime
18        }
19    }
20 }
21
22 func main() {
23     client := githubv4.NewClient(oauth2.NewClient(context.Background(), token))
24     user := client.GetUser(context.Background(), "shurcool")
25     fmt.Println(user)
26 }
27
28 // graphQL: "repositories(last:1 orderBy: {field: PUSHED_AT direction: ASC})"
```

2.3k gitbot/main.go 11:0 Top LF UTF-8 Go 1.14 ✓

SPC s s

5. In-Editor LaTeX Preview

You can press `C-c-x-l` or `Enter` to preview your LaTeX code in the editor itself (provided you have TeXLive installed):

```

40 #+TITLE: Algorithmics
39 #+REVEAL_THEME: black
38
37 ◉ Binary Search...
4  ◉ Central Limit Theorem...
2  ◉ Number of Combinations
1  To get number of sample of combinations  $r$  from  $n$  objects (order doesn't
41 matter):
1
2  $$C(n,r) = \frac{n!}{r!(n-r)!}$$

```

925 org/comp1201.org 41:7 All LF UTF-8 Org

Emacs let you add new org-mode tasks with `Alt+Shift+Enter` along with the following:
 You no longer have to preview your notes in `EdTeX`

- `C-c-d` to add a deadline.

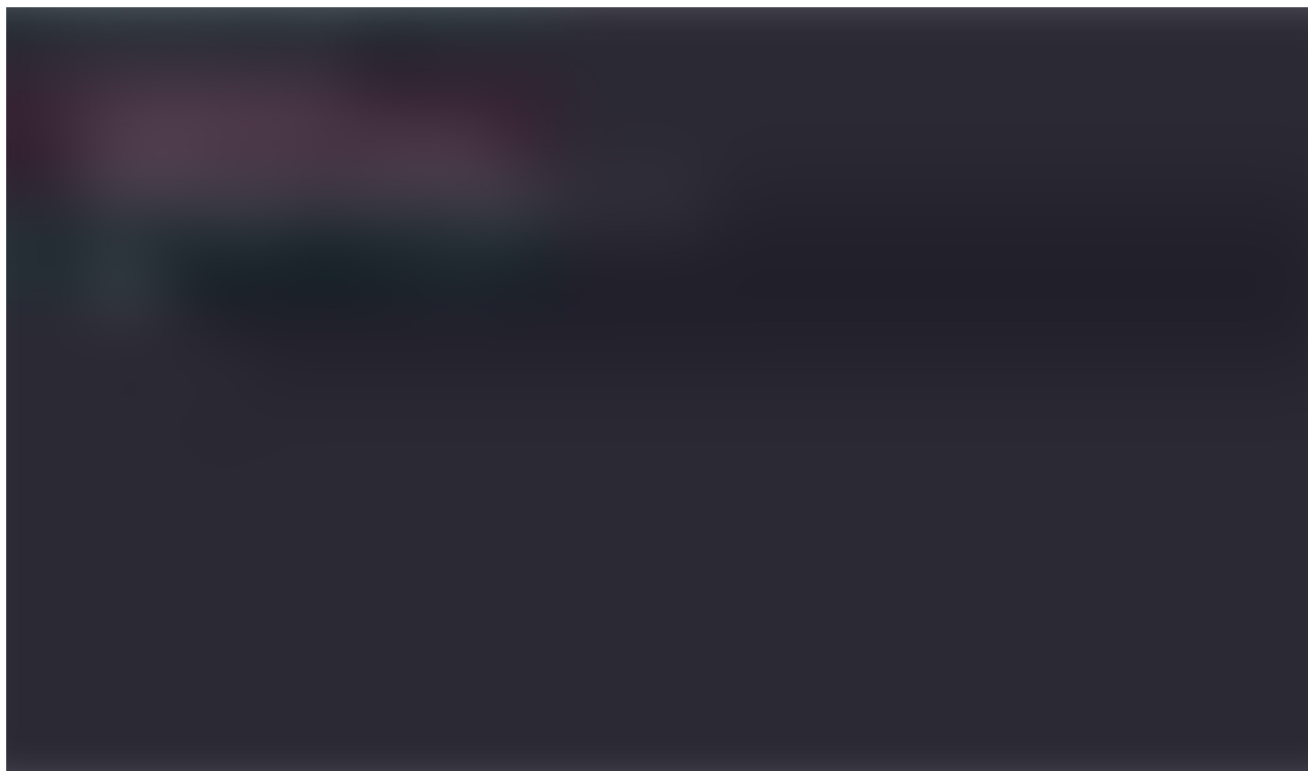
- `C-c-s` to add a scheduled date.

- `C-c-t` to change/toggle the state of the tasks (e.g. `TODO` `DONE`).

```

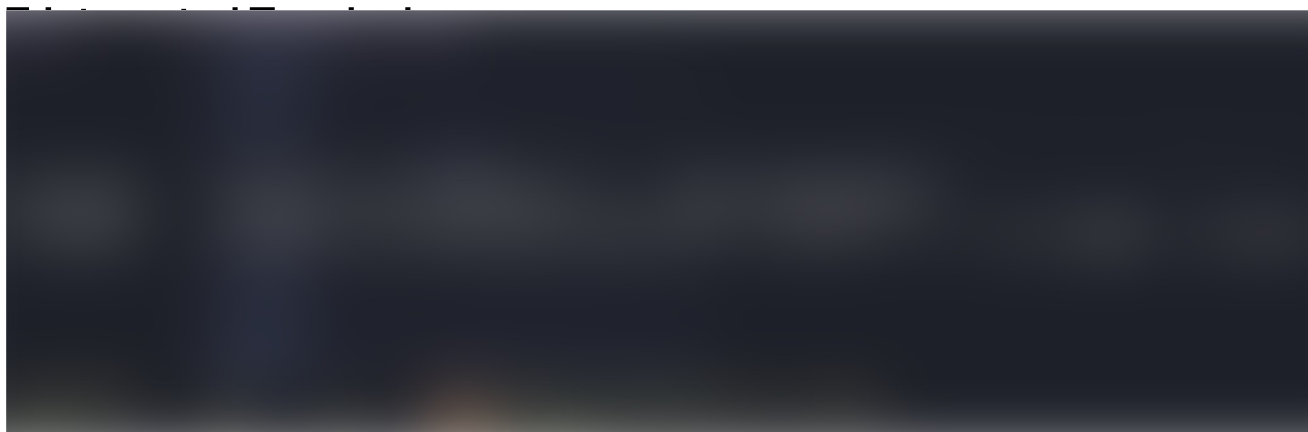
10  ◉ Time
11  | *Best* | Θ(1) |
12  | *Average* | Θ(log(n)) |
13  | *Worst* | Θ(log(n)) |

```



Viewing your agenda

You can set this up to include your iCalendar file for your university timetable, current to-do tasks, or even repeated habit tasks.



SPC o A a to show your org agenda

```
11 package main
10
9 import (
8     "context"
7     "fmt"
6     "time"
5
4     "github.com/shurcool/githubv4"
3     "golang.org/x/oauth2"
2 )
1
12 const token = "....."
1
2 type User struct {
3     ID          githubv4.ID
4     Login       githubv4.String
5     Email       githubv4.String
6     CreatedAt   githubv4.DateTime
7     Following   struct {
8         TotalCount githubv4.Int
9     }
10    Followers struct {
11        TotalCount githubv4.Int
12    }
13    Repositories struct {
14        TotalCount githubv4.Int
15    }
16 }
17
18 func main() {
19     ctx := context.Background()
20     client := githubv4.NewClient(&http.Client{
21         Transport: &oauth2.StaticTokenSource{
22             Tokens: []string{token},
23         },
24     })
25     user := client.GetUser(ctx, "shurcool")
26     fmt.Println(user)
27 }
```

gitbot/main.go 12:24 Top 1L LF UTF-8 Go 1.14 ✓
-- VISUAL LINE --

“SPC o e” to toggle the terminal

Resize the terminal with your mouse

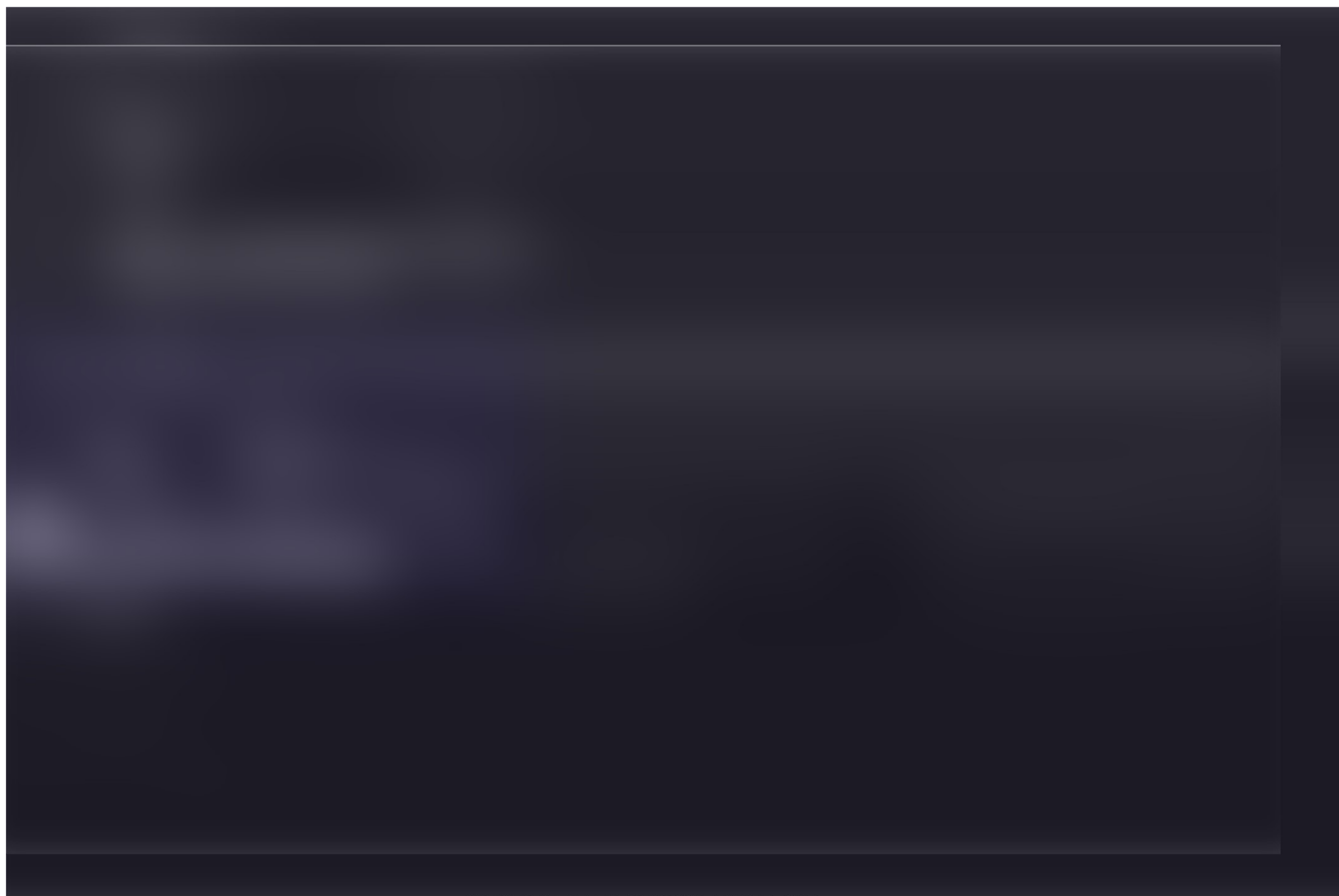
8. LSP Support

I know, these features are covered by most other decent editors, even Vim (contrary to the popular belief that Vim is only good for editing basic text/scripts), all thanks to Microsoft introducing the LSP standard.

Jump definitions

- Local definitions

- Globally defined definitions



Jumping to definitions with `gd`

But even if you can, it doesn't mean that you should

Folding/unfolding code

You can toggle `za`, open `zo`, or close `zc` a section of the code.



Fold/Unfold code with “za”

Autocomplete

Emacs also has exceptionally fast autocomplete support thanks to its polished LSP extension.

9, 10. Executing Code Blocks and Code Symbols

Org-mode allows you to execute code blocks of languages such as Lisp, Python, Bash, JavaScript, and even allows you to save session variables between code blocks.

Also, notice the symbols that replace common keywords. This is what pretty code does in Doom Emacs.

```
52 #+TITLE: Algorithmics
51 #+REVEAL_THEME: white
50
49 ◎ Binary Search...
10 ◎ Central Limit Theorem...
 8 ◎ Number of Combinations...
 2 ◎ Executing code blocks
 1
53 » python
 1 «
 2

python: :results replace :exports code :tangle no :hlines no :noweb no :cache no :session none
```

Autocomplete (using company-lsp)

11. Org Tables

Tables are automatically formatted for you, so you don't have to deal with those ugly, unaligned Markdown tables.

```

40
39 ② Binary Search...
43 ② Central Limit Theorem...
2 ② Number of Combinations...

2 #+TITLE: Algorithmics
1 #+REVEAL_THEME: white
3 |
1 ② Binary Search
2 - requires ordered list
3 » python :results output
4 # find index of value in the array
5 f f(arr, v):
6     print(arr)
7     # middle item, for even size round up
8     # e.g. arr = [7,8] -> mid = 2nd item
9     mid = len(arr) // 2
10    # found it
11    if arr[mid] == v:
12        ↳ mid
13    # its on the left
14    if arr[mid] > v:
15        ↳ f(arr[:mid], v)
16    # its on the right
17    ↳ f(arr[mid:], v) + mid
18
19 print(f([1,2,4,6,7,8,9,10], 8))
20 «
21
22 ○ Example...

```

14. Magit — Best Git CLI I've Ever Used

It can do so much that instead of trying to show everything in this article, you should definitely just watch the above video instead.

TL/DR; stage, unstage, branch, push, pull, commit, and more, all from your editor, just a few keystrokes away.

15. Commenting Out Multiple Lines

In Doom Emacs, Magit can be launched with `SPC g g`.

You can simply use `M-;` to toggle comments on the selected lines.

```
18     Repository struct {
17         Stargazers struct {
16             TotalCount githubv4.Int
15             Nodes      []User
14         } `graphql:"stargazers(last:100)"`
13     } `graphql:"repository(name:$name,owner:$owner)"`
12 }
11
10 func getUserDetails(client *githubv4.Client, owner string) UserQuery {
9     query := UserQuery{}
8     variables := map[string]interface{}{
7         "owner": githubv4.String(owner),
6     }
5     err := client.Query(context.Background(), &query, variables)
4     if err != nil {
3         fmt.Println(err)
2     }
1     return query
56 }

1 func follow(client *githubv4.Client, user string) {
2     var m struct {
3         FollowUser struct {
4             User struct {
5                 ID githubv4.ID
6             }
7         }
8     }
9     err := client.Query(context.Background(), &m, variables)
10    if err != nil {
11        fmt.Println(err)
12    }
13}
```

However, if you're not that into Vim bindings but you still want a great out of the box experience, try Centaur Emacs.

Sign up for programming bytes

By Better Programming

A monthly newsletter covering the best programming articles published across Medium. Code tutorials, advice, career opportunities, and more! [Take a look.](#)

 Get this newsletter

[Vim](#) [Emacs](#) [Vscod](#) [Programming](#) [Software Development](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

