Basic git commands everyone should know

`git init | git init [folder]`

Git init is used to initialize an empty repository from the folder you are currently using thi
path, both ways are correct. It's used while starting a new project or if you want to initializ
project.

`git clone [repo URL] [folder]`

Git clone is used to copy the existing repository to the specified folder on your computer. C
repo URL as a parameter, then it will copy the repository to the folder from where you use
copy the repository to a different location on your computer, add a folder path as a second

`git add [directory | file]`

Git add stage all changes in the directory or in the file, and it depends on what you add as
followed by git commit and git push commands.

`git commit -m "[message]"`

This command is used to commit all staged changes with the custom message passed as a
parameter to -am it's possible to add and commit changes at once.

`git push`

This is the command, which pushes changes to the origin branch.

`git status`

Git status is used to check the status of the modified files and it shows which files are stage

`git log`

Git log is used to display the history of the commit in the default format.

`git diff`

Git diff shows all unstaged differences between the index and the current directory. This c
staged to display differences between staging files and the most recent versions. And anotl
command with the file name to display differences between the file and in the last commit

`git pull`

Git pull is used to get changes from the original branch, and it merges the changes into the

`git fetch`

This command retrieves the most recent changes from the origin branch but doesn't merge

Git branch commands

`git branch`

This command displays the list of all branches in the repository. It can also create a non-ex branch name as a parameter.

```
git branch -d [branchname]
```

Using -d flag will delete the branch with the specified branch name.

```
git checkout [branchname]
```

This command switch to the branch named [branchnamed]. If you add -b flag before branc new branch, which will be created automatically.

```
git merge [branchname]
```

It merges the branch with the specified branch name to the current branch.

Git undoing changes commands

```
git revert [commit]
```

This command creates a new commit that undoes changes made in the specified commit a branch.

```
git reset [filename]
```

It remotes specifies a file from the staging and leaves the working directory unchanged.

Git config commands

```
git config -global user.email [user_email]
```

```
git config -global user.name [user_name]
```

The commands above are used to set current user email and name configuration.

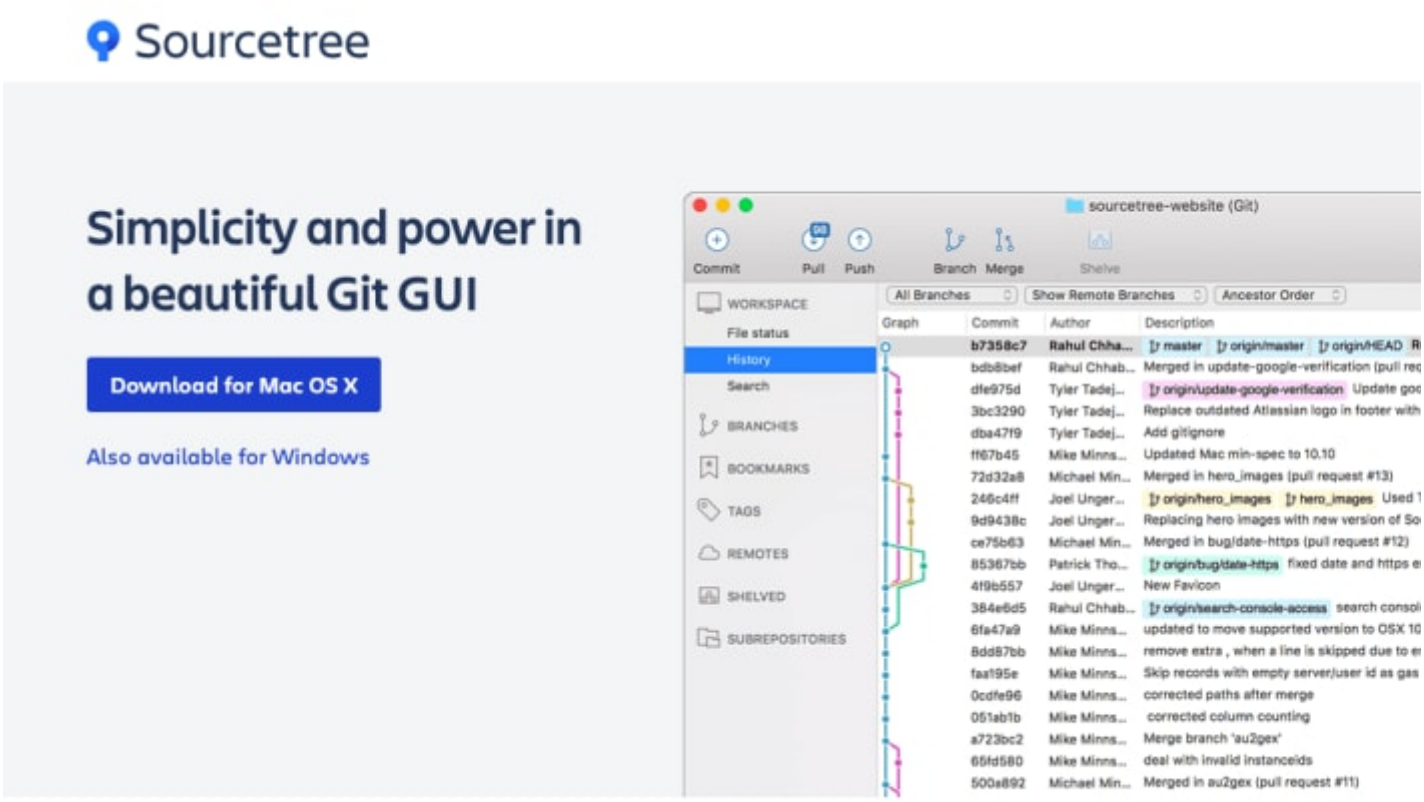
```
git config --global --edit
```

And this command is very useful, as it allows for editing user configuration in a text edito

Git GUIs

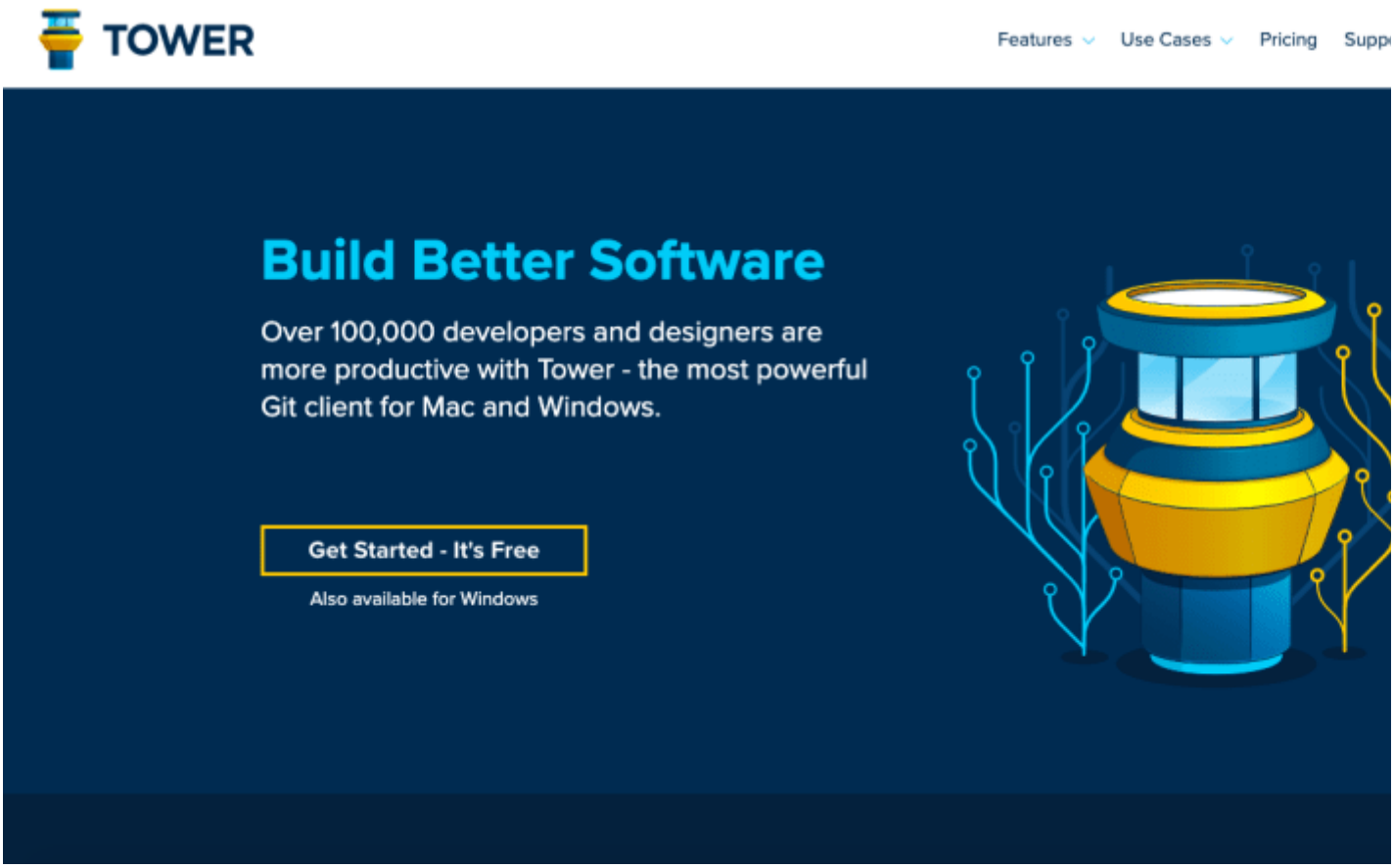
Not everyone likes to use git in the command line. It's very easy to make a mistake there, v revert it. That's why GUIs for git become very popular. Let's check a few of them.

Sourcetree



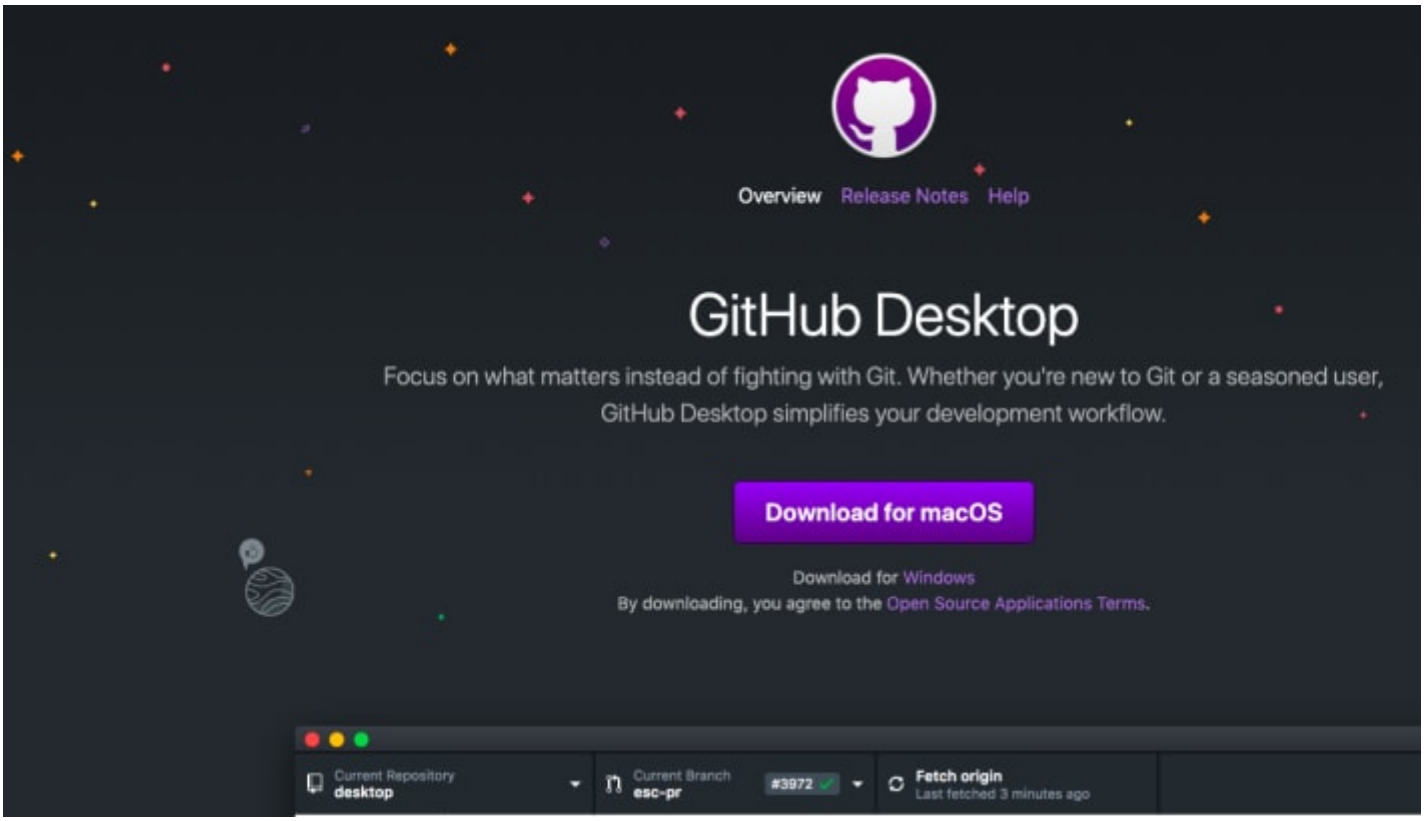
Sourcetree is a git GUI which is available for Mac and Windows, and it's free. I really like i much easier for me, as I can see all the changes very clearly.

Tower



Tower is another great tool that makes using git nice and easy. It's also available for both M
is not free. I had a chance to use it, they have a trial option, so everyone can try this softwa

Github Desktop



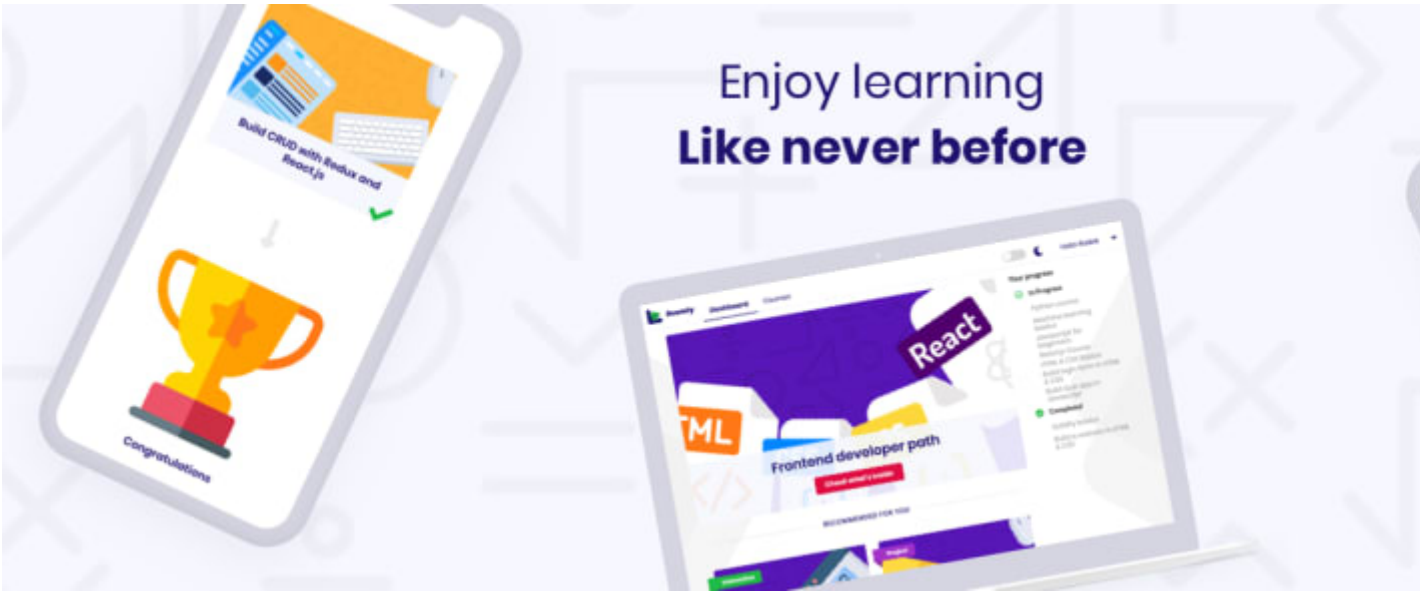
Github Desktop is another competitive tool, allowing us to use git in a nice and user-friend
both Mac and Windows. Also, it's an open-source app, and it's free to use.

Conclusion

In this article, I went through the most basic git commands to keep it in one place and mak
use it. Also, I listed a few terms which understanding may be very helpful to understand v
of the commands. In the end, I've presented three GUIs that I had an opportunity to use. Ir
be very helpful, especially for junior programmers, who don't feel comfortable width comı
like to have an easier and more visible way of working with git.

If you would like to learn how to create your first git repository in Github, check out this Y
share some information about git GUIs.

Have a nice coding!



Thank you for reading,
Anna from Duomly

Posted on Jan 8 by:

Duomly
@duomly

We believe everyone can learn how to code, so we are making learning fun and easy!

Discussion

Add to the discussion

Kristina Gocheva

I would like to disagree that writing git commands in the command line is hard. On the contrary, us
ABC especially when you have the "cheatsheet". If you want to understand git, use the command li
process in your head, not with UI.