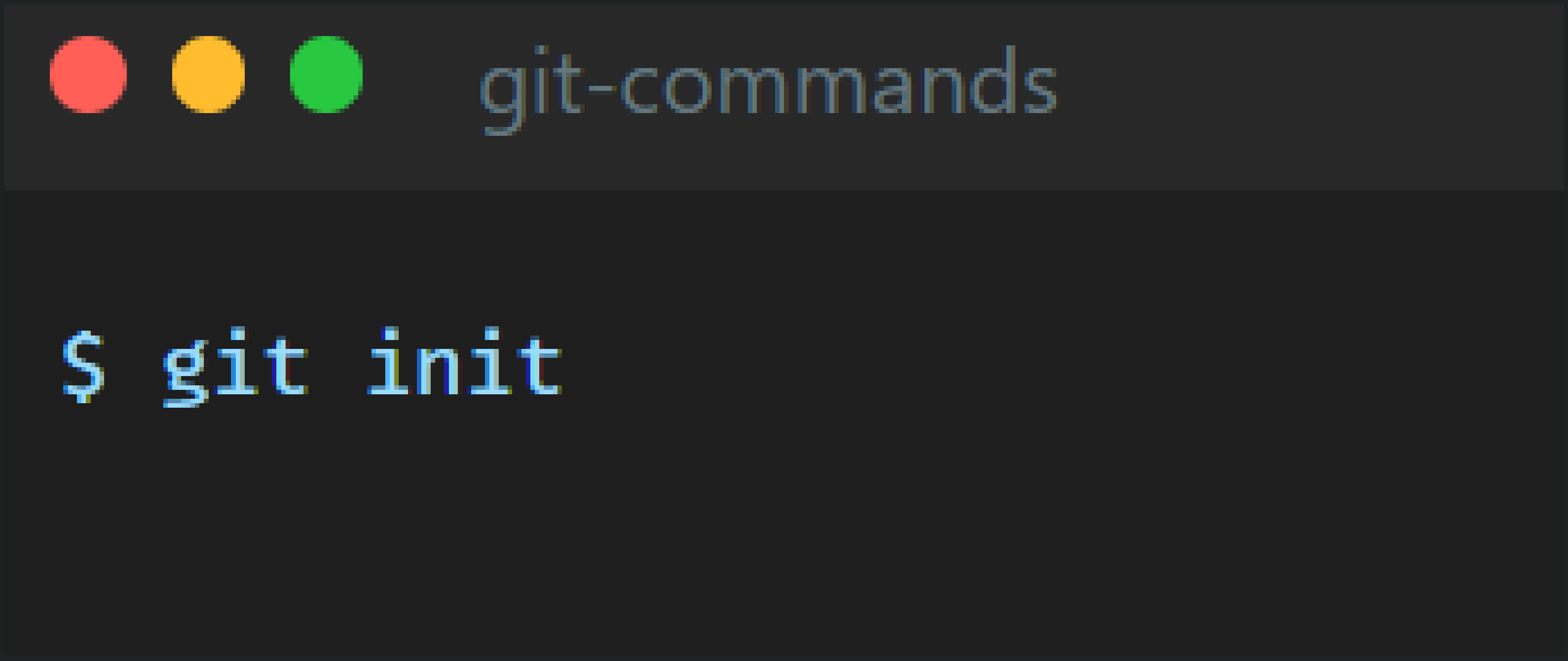
Common GIT Commands

Ibrahem Magdy

1- git init

Everything starts from here. The first step is to initialize a new Git repo locally in your project root. You can do so with the command below:

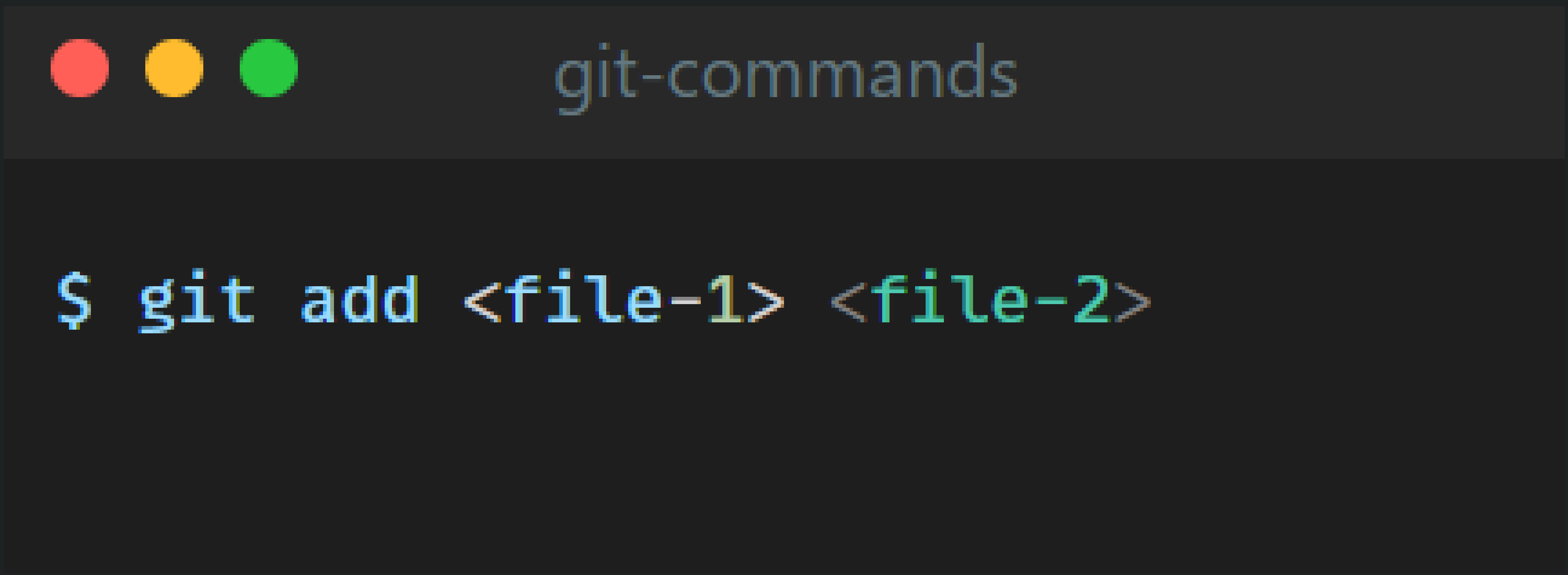


```
$ git init
```

Ibrahem Magdy

2- git add

This command adds changes to the staging area, allowing them to be committed

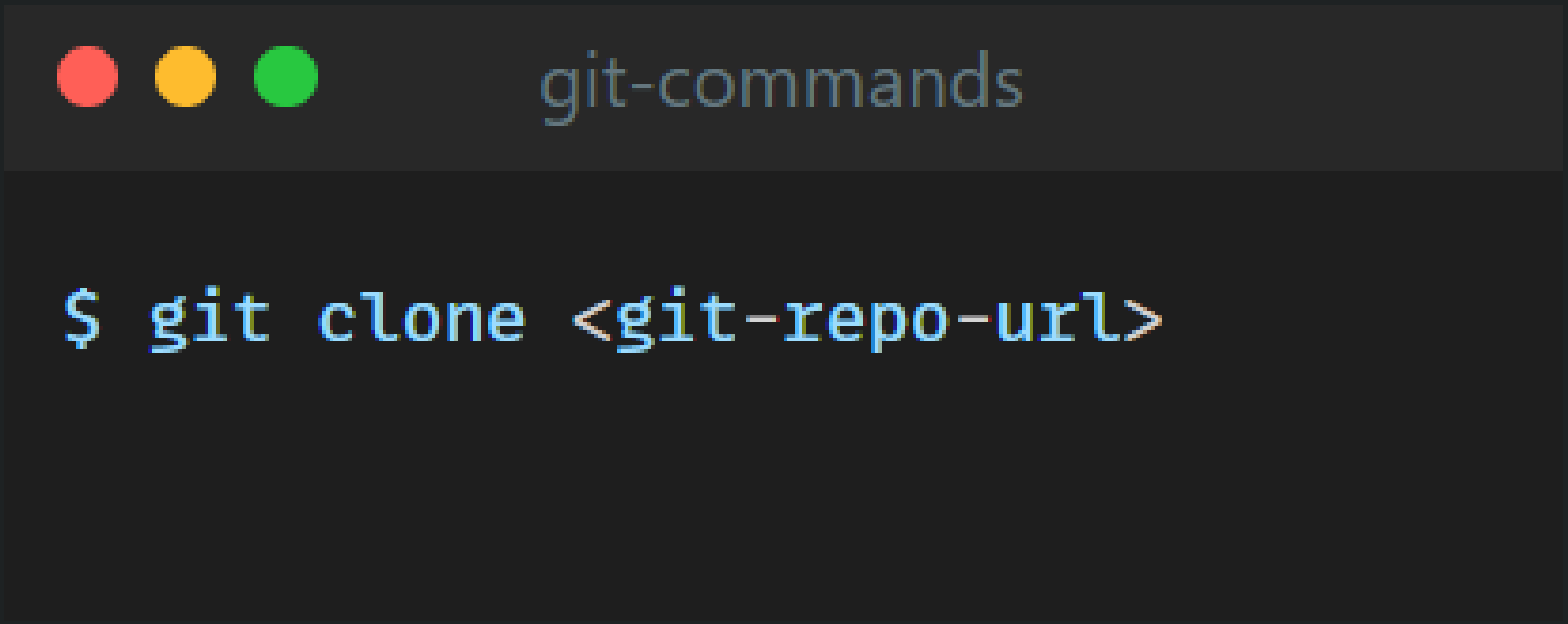


```
$ git add <file-1> <file-2>
```

Ibrahem Magdy

3- git clone

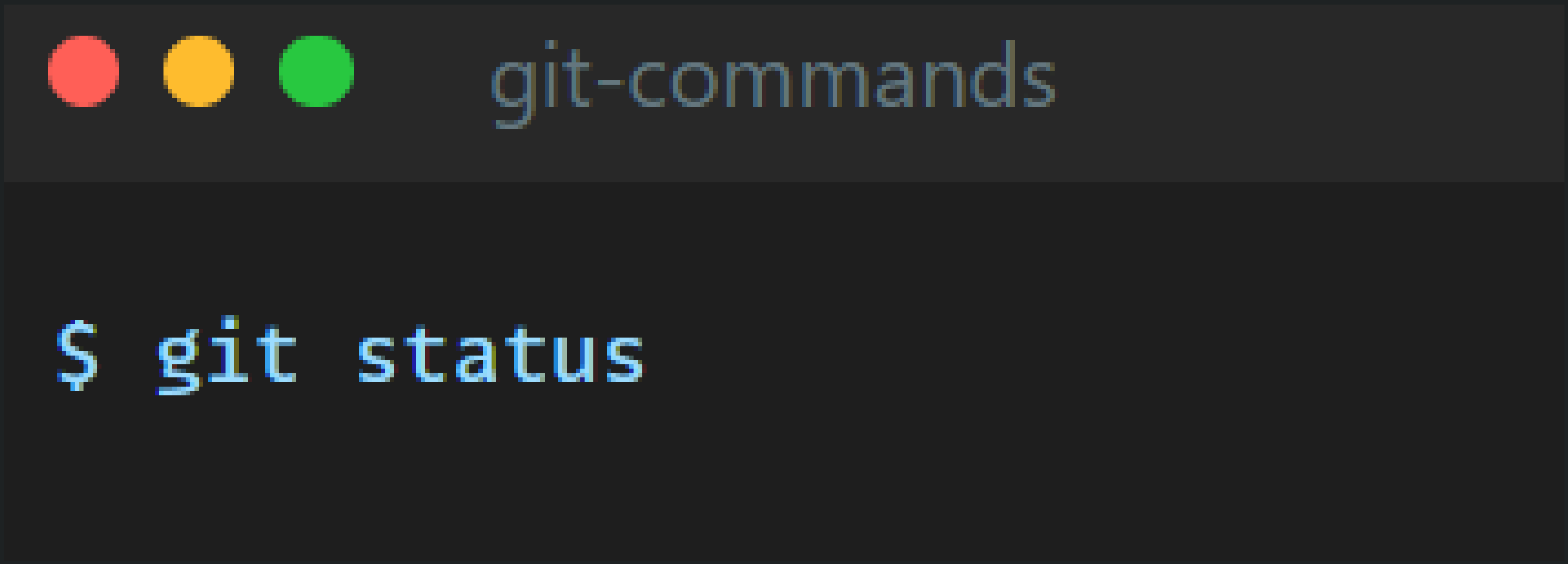
This command Creates a copy of a repository on your local machine. This command allows you to download an entire repository from a remote server and save it to your local machine.



```
$ git clone <git-repo-url>
```

4-git status

This command shows you the current state of your repository, including any changes that have been made to tracked files that have not yet been committed, and any untracked files that have been added to the working directory.

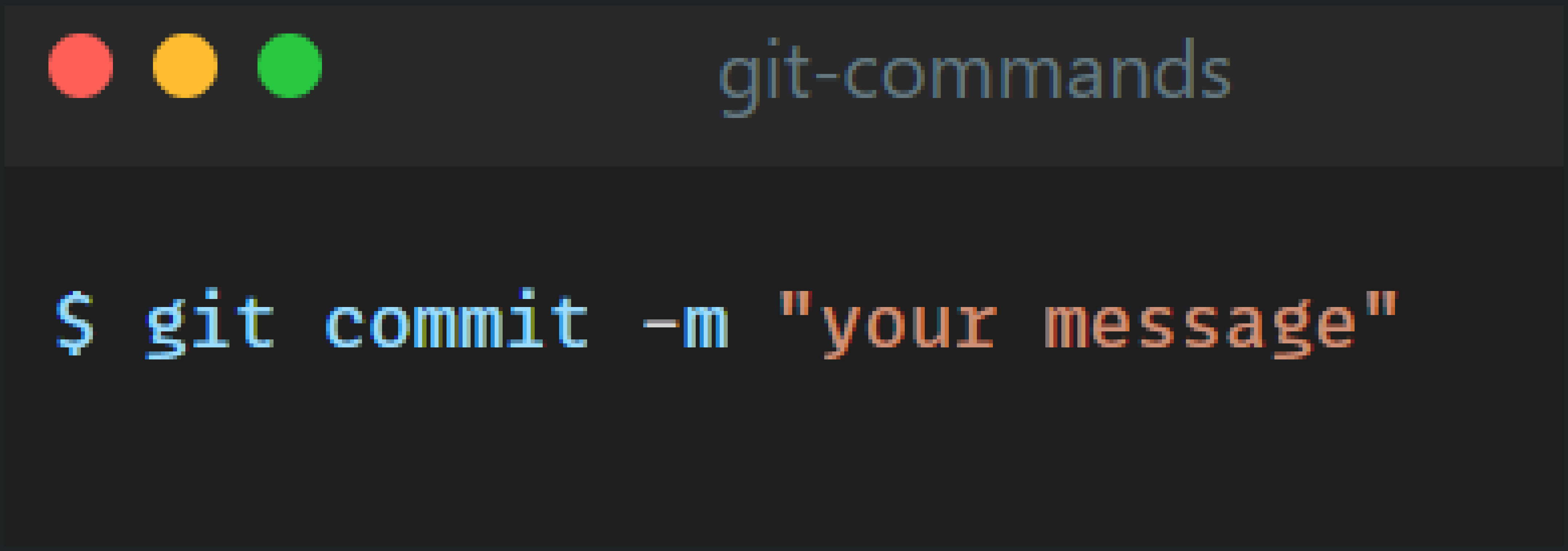


```
$ git status
```

Ibrahem Magdy

5- git commit

This command creates a snapshot of the changes made to the files in the staging area and records them in the repository's commit history along with a message that describes the changes made.

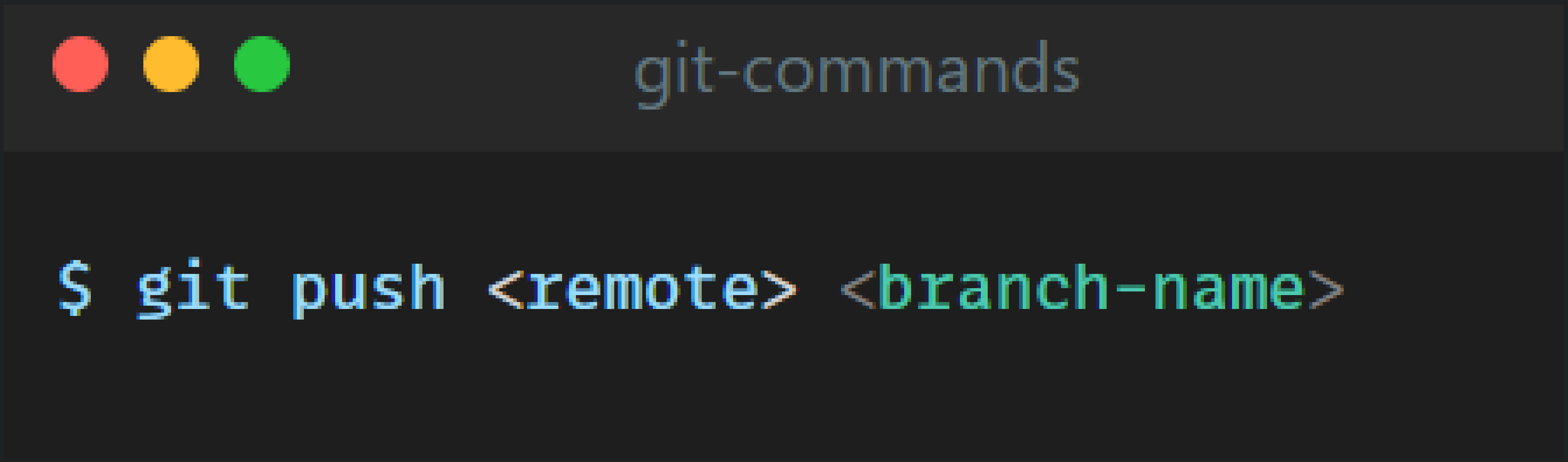


```
$ git commit -m "your message"
```

Ibrahem Magdy

6- git push

This command uploads the changes made to your local repository to a remote repository, allowing others to access and view the changes you've made.

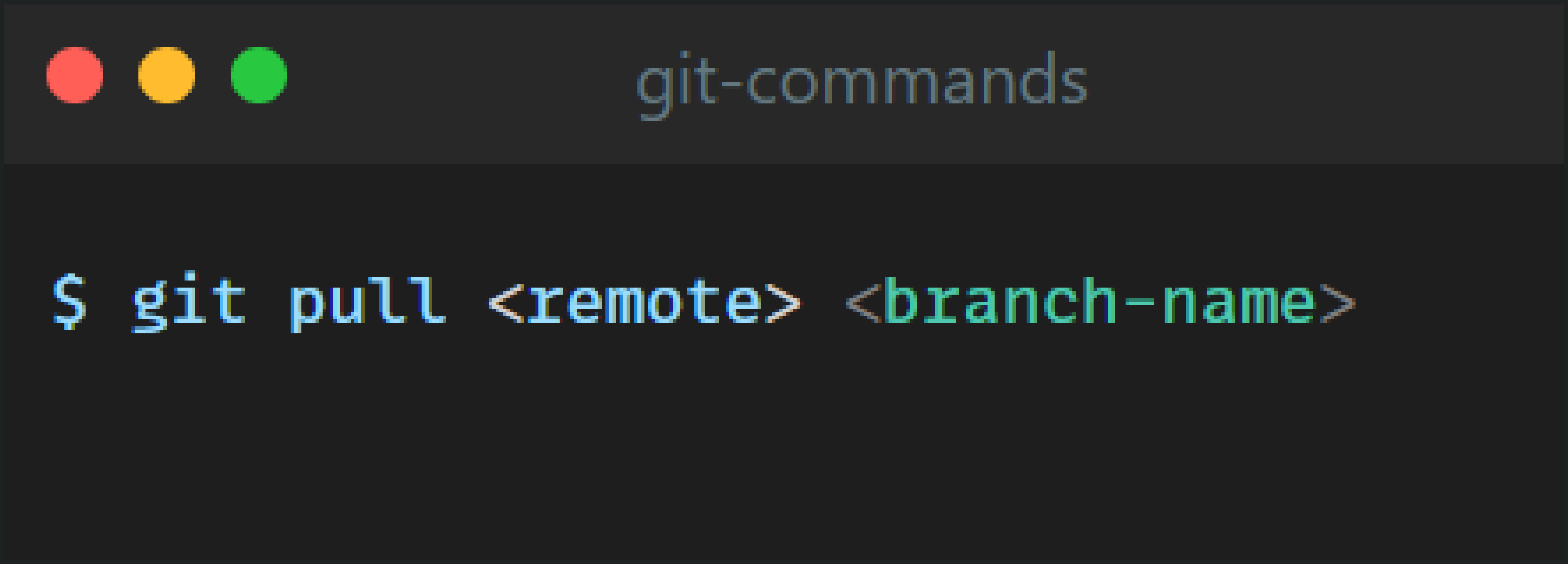


```
$ git push <remote> <branch-name>
```

Ibrahem Magdy

7- git pull

This command downloads changes from a remote repository and merges them with your local repository, allowing you to stay up-to-date with changes made by others.

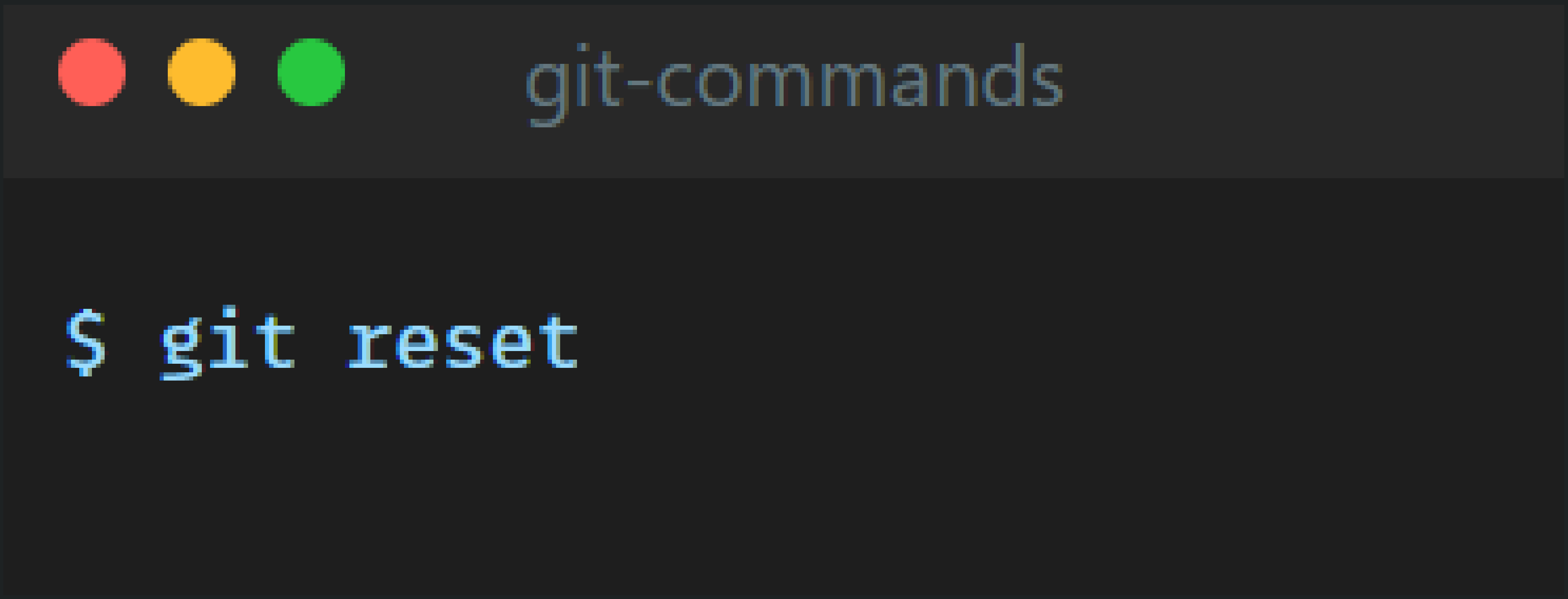


```
$ git pull <remote> <branch-name>
```

Ibrahem Magdy

8- git reset

This command removes changes made to tracked files from the staging area, allowing you to unstage changes that were added by mistake. It can also be used to undo changes made in the working directory.

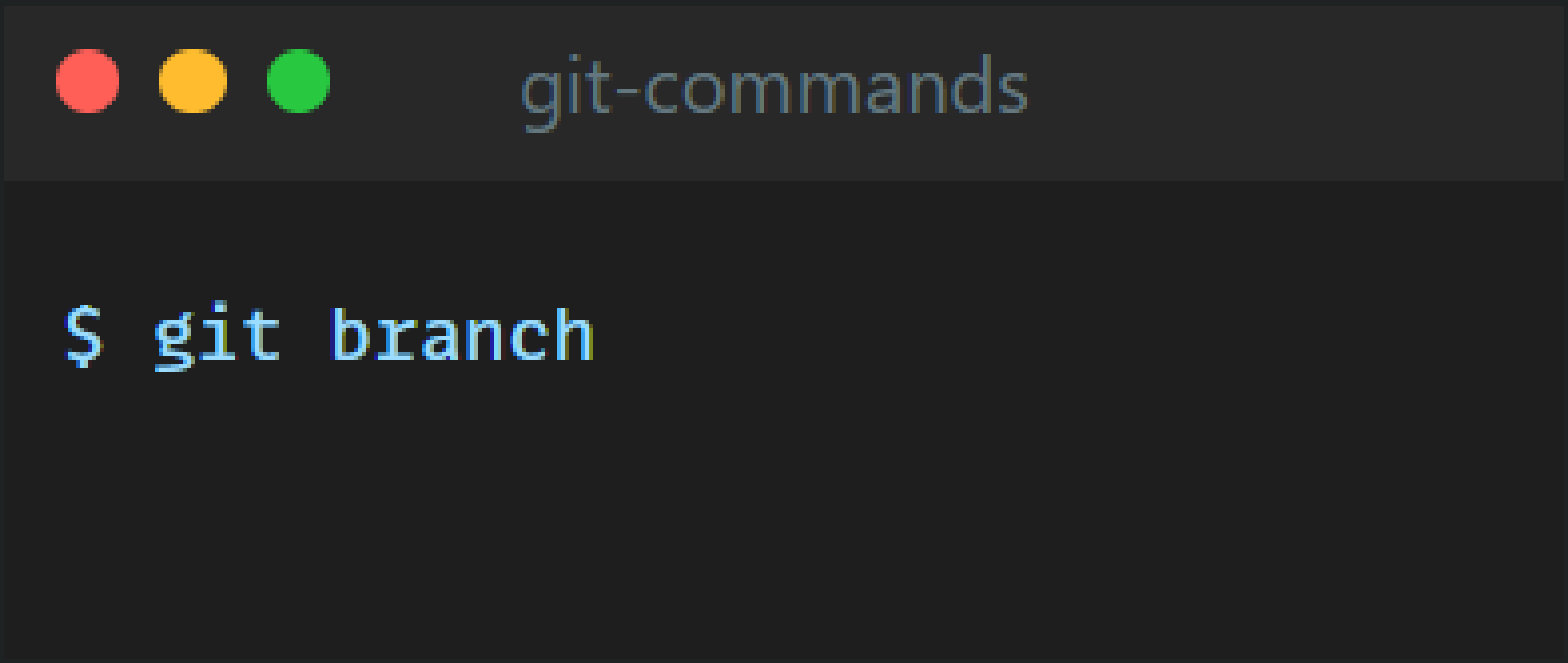


```
$ git reset
```

Ibrahem Magdy

9- git branch

This command shows you a list of all branches in your repository, including the current branch you're working on.

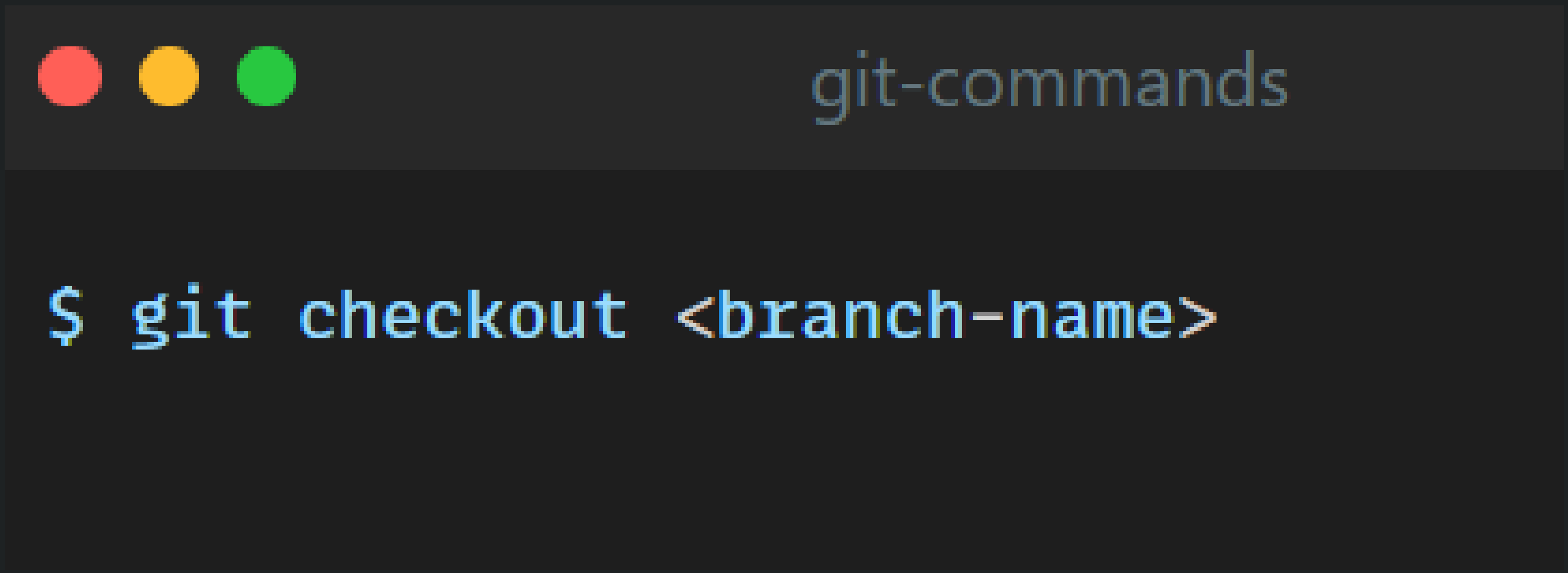


```
$ git branch
```

Ibrahem Magdy

10- git checkout

This command allows you to switch between different branches in your repository.

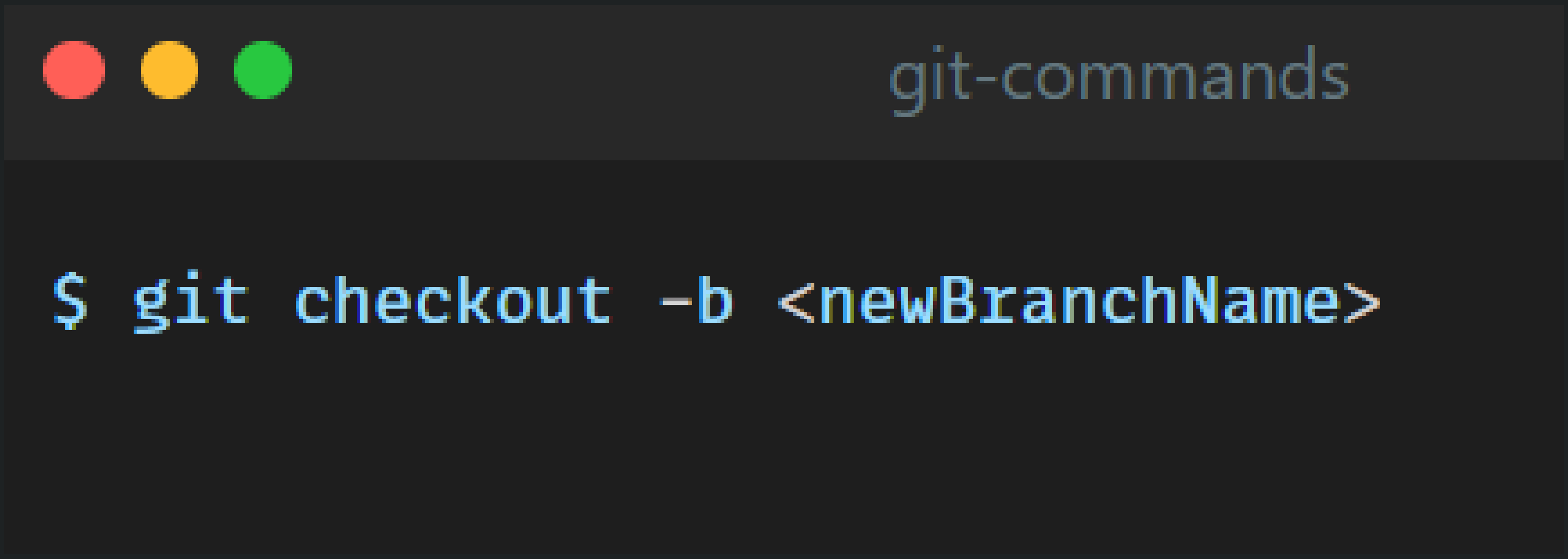


```
$ git checkout <branch-name>
```

Ibrahem Magdy

11- git checkout -b

This command creates a new branch in your Git repository and switches to it in one step. The -b option stands for "branch" and tells Git to create a new branch with the specified name.

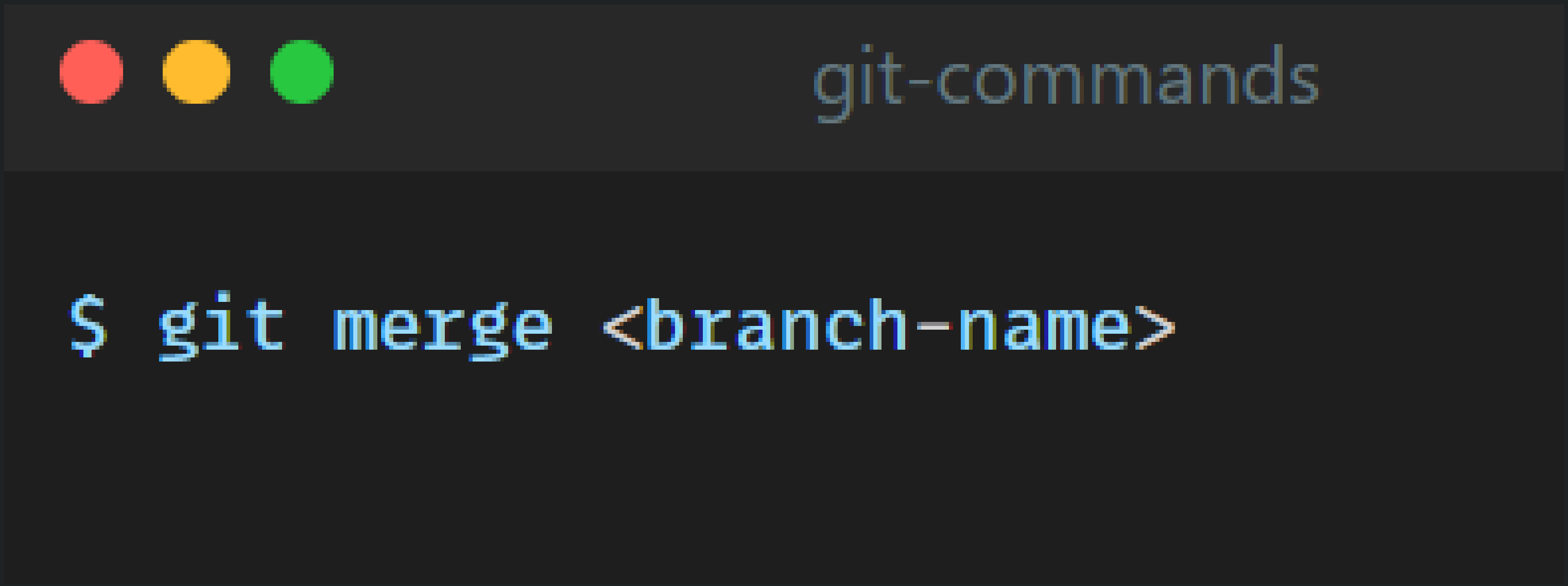


```
$ git checkout -b <newBranchName>
```

Ibrahem Magdy

12- git merge

This command combines changes made in one branch into another branch, allowing you to merge different branches of code into a single branch.

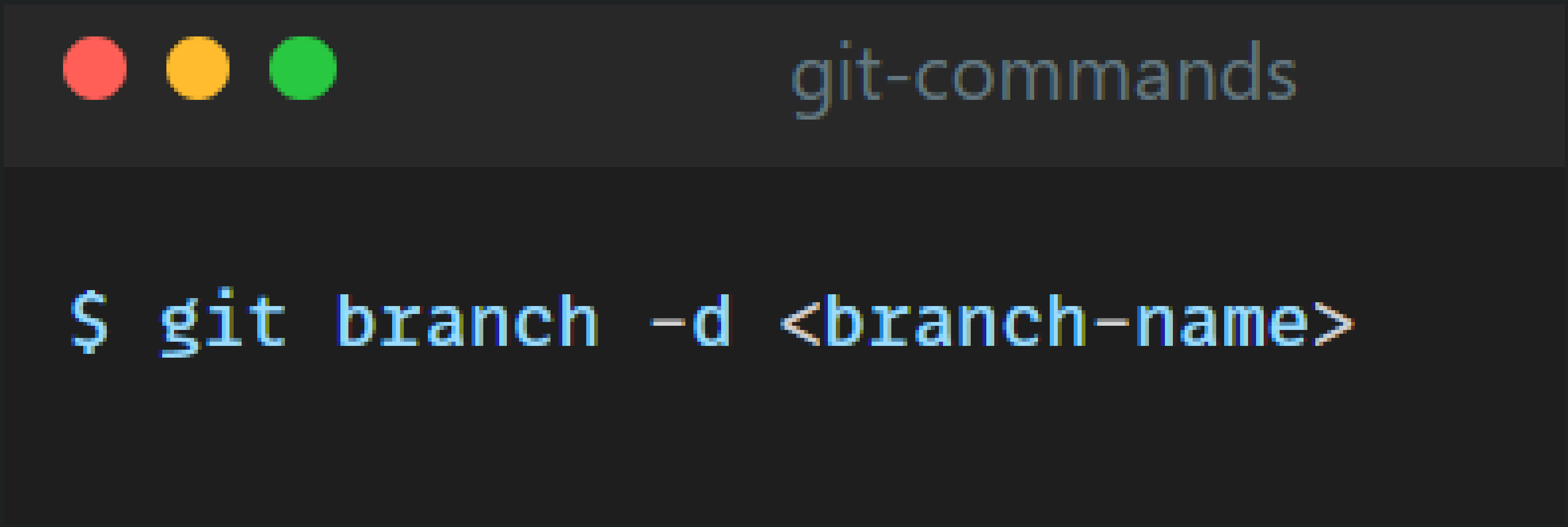


```
$ git merge <branch-name>
```

Ibrahem Magdy

13- delete branch

This command is used to delete a branch in your Git repository. The -d option stands for "delete" and tells Git to delete the specified branch.

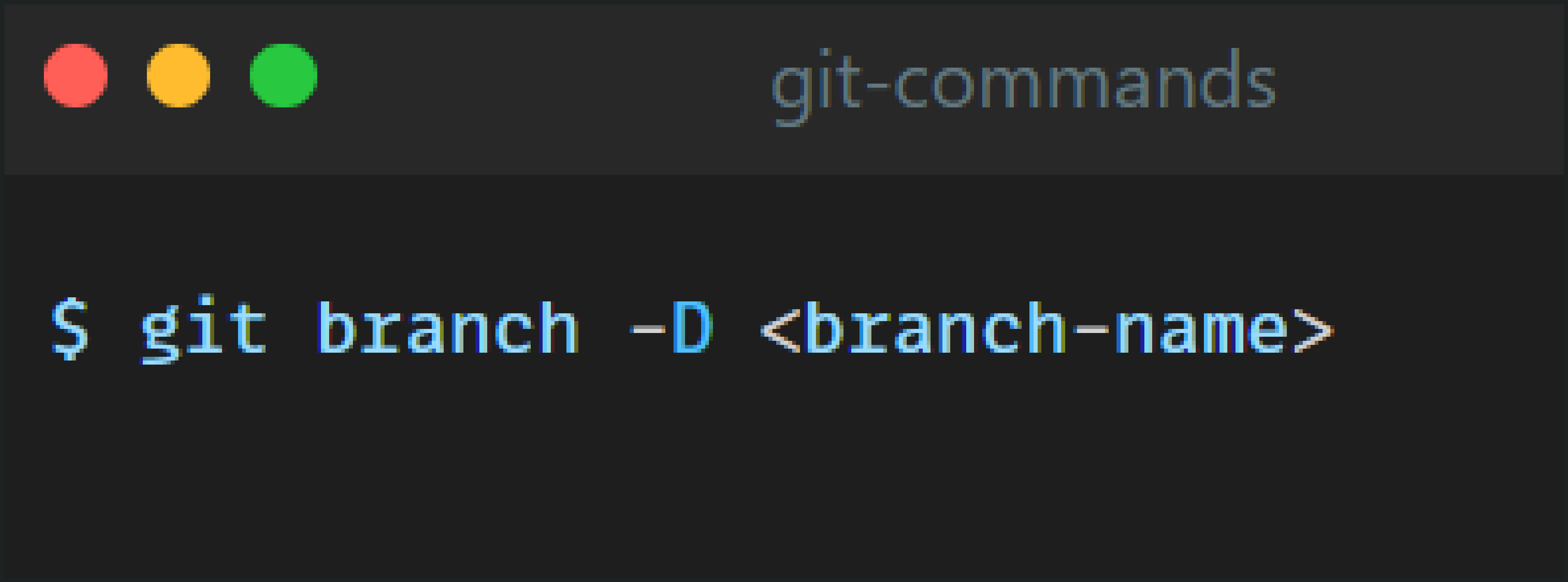


```
$ git branch -d <branch-name>
```

Ibrahem Magdy

14- delete branch by force

This command is used to force delete a branch in your Git repository. The -D option is a shorthand for the **--delete --force** options and tells Git to delete the specified branch even if it contains unmerged changes.



```
$ git branch -D <branch-name>
```

Ibrahem Magdy