

Git-it Electron

Porting a NodeSchool workshopper to Electron

Jessica Lord

Engineer at GitHub on Atom

Git-it

An open source tool for teaching Git and GitHub to newcomers.

Command line, fork, clone, pull, collaborate & Pull Request.

Built for Patchwork

Born in early 2013 for GitHub Patchwork events



Patchwork today



4,975+ Completions

Congrats, you did it!



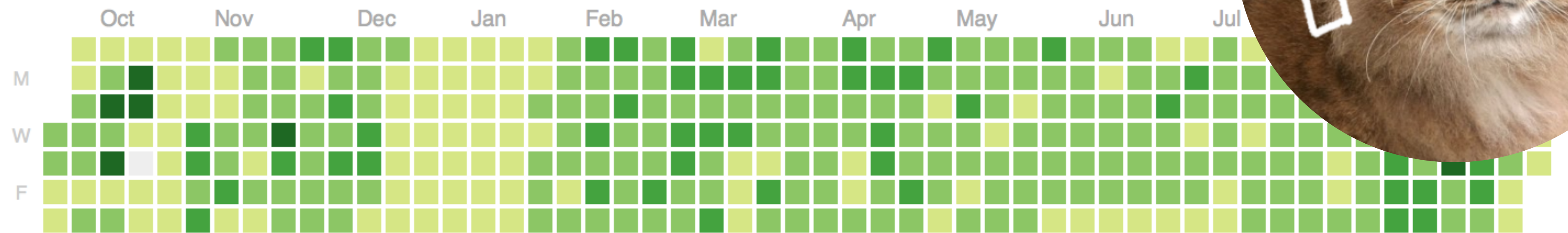
You now know alternate meanings for **fork** and **branch**. You've **collaborated** with someone whose location you're unaware of. You, friend, **pull request** with the best of 'em.

Your pulling and pushing compatriots (give or take 6 million):

Busy Robot



Public contributions



Contributions in the last year

32,736 total

Sep 24, 2014 – Sep 24, 2015

Longest streak

343 days

October 17 – September 24

Current streak

343 days

October 17 – September 24

Busy Robot

Oct. 16, 2014



Helpful Robot



reporobot commented 9 hours ago

Collaborator



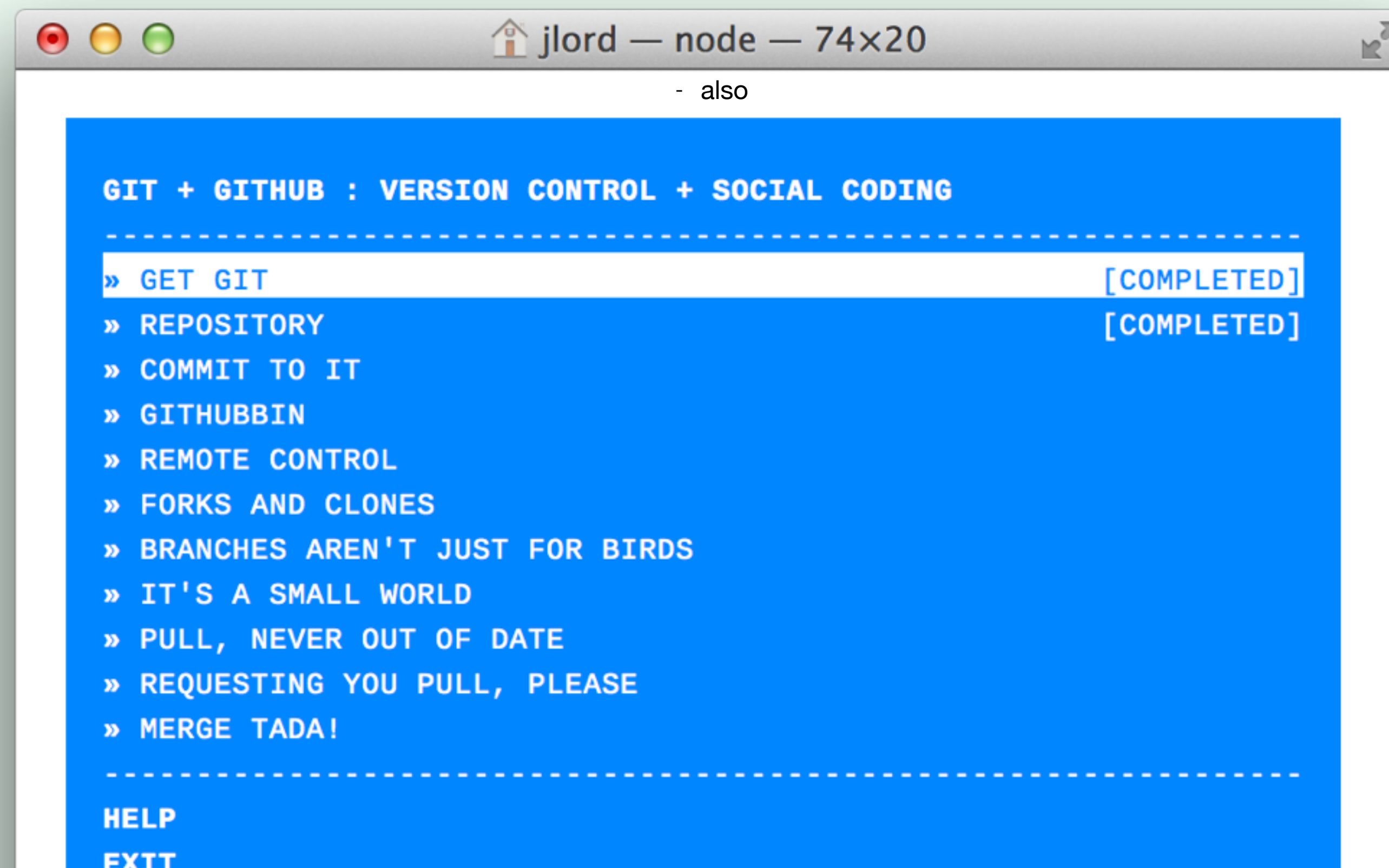
Hello! The name of the branch with this Pull Request is not of the `add-USERNAME` pattern so I'm skipping reviewing it for the Git-it challenges.

If you are completing the Git-it challenges, you'll need to close this Pull Request, rename your branch, push it to your fork of Patchwork on GitHub and then start a new Pull Request. Here's how to rename a branch and push it to GitHub from terminal:

```
$ git branch -m add-USERNAME  
$ git push origin add-USERNAME
```

Original Git-it

Based on NodeShool terminal workshops.

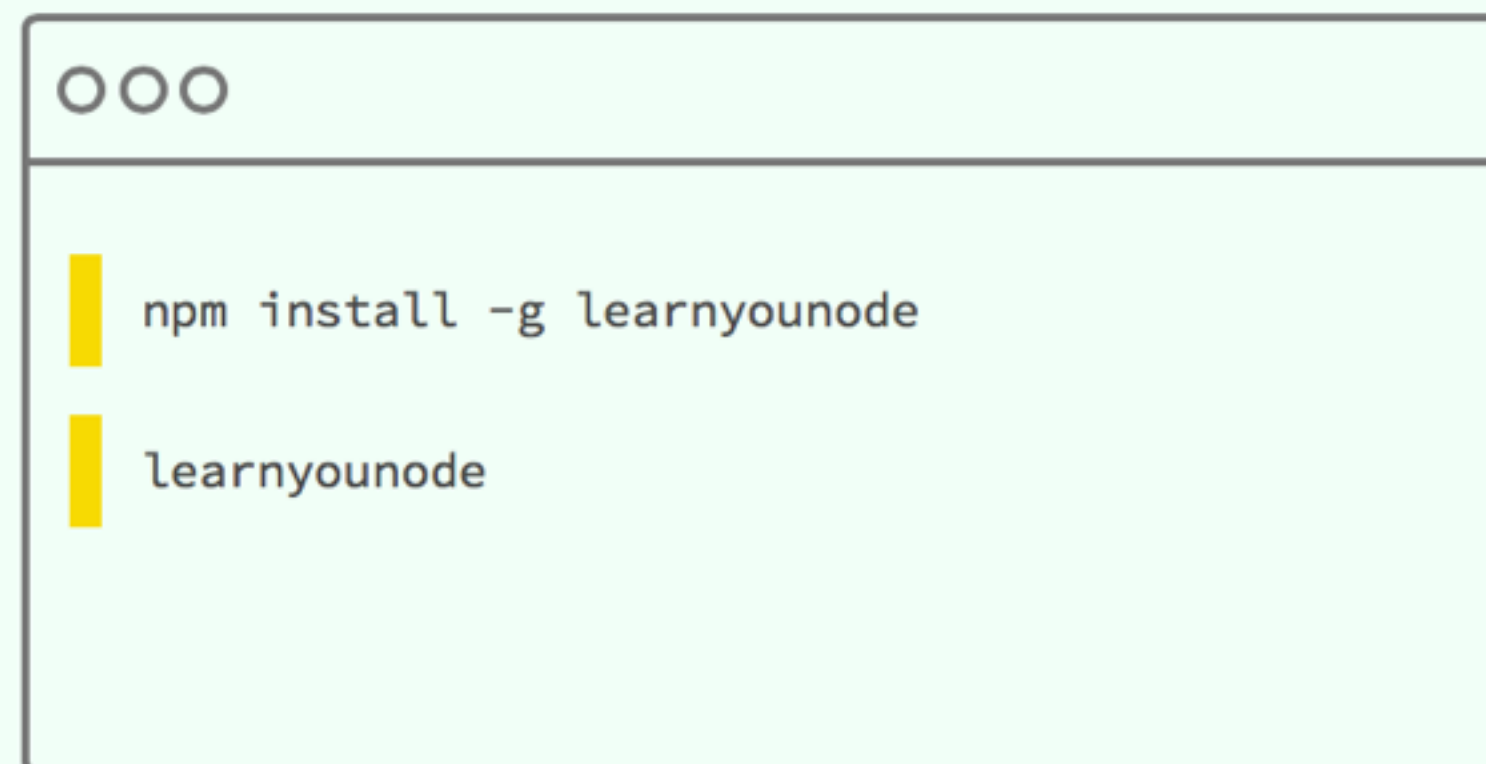


```
jlord — node — 74x20
- also

GIT + GITHUB : VERSION CONTROL + SOCIAL CODING
-----
» GET GIT [COMPLETED]
» REPOSITORY [COMPLETED]
» COMMIT TO IT
» GITHUBBIN
» REMOTE CONTROL
» FORKS AND CLONES
» BRANCHES AREN'T JUST FOR BIRDS
» IT'S A SMALL WORLD
» PULL, NEVER OUT OF DATE
» REQUESTING YOU PULL, PLEASE
» MERGE TADA!
-----
HELP
EXIT
```


NodeSchool.io

Awesome collection of **free** and **open source** interactive command line Node.js tutorials.



```
ooo  
  
| npm install -g learnyounode  
|  
| learnyounode
```



```
ooo  
  
» HELLO WORLD  
  
» BABY STEPS  
  
» MY FIRST I/O!
```

NodeSchool.io

Active community hosting local workshops all over the world.



Original Git-it Components

rvagg/
workshopper

Node.js scripts
for verifying

Node.js server
@reporobot

HTML
web guide

Original Git-it Components



rvagg/
workshopper

Node.js scripts
for verifying



Node.js server
@reporobot

HTML
web guide

Porting Git-it Components

Node.js scripts
for verifying

HTML
web guide

Electron



Node.js
scripts

The diagram consists of two adjacent rectangular boxes with blue borders. The left box contains the text 'Node.js scripts' and the right box contains the text 'HTML pages'. Both boxes are light blue and have a thin blue outline.

HTML
pages

Outlook: Good

Improve the experience

Changes in Experience

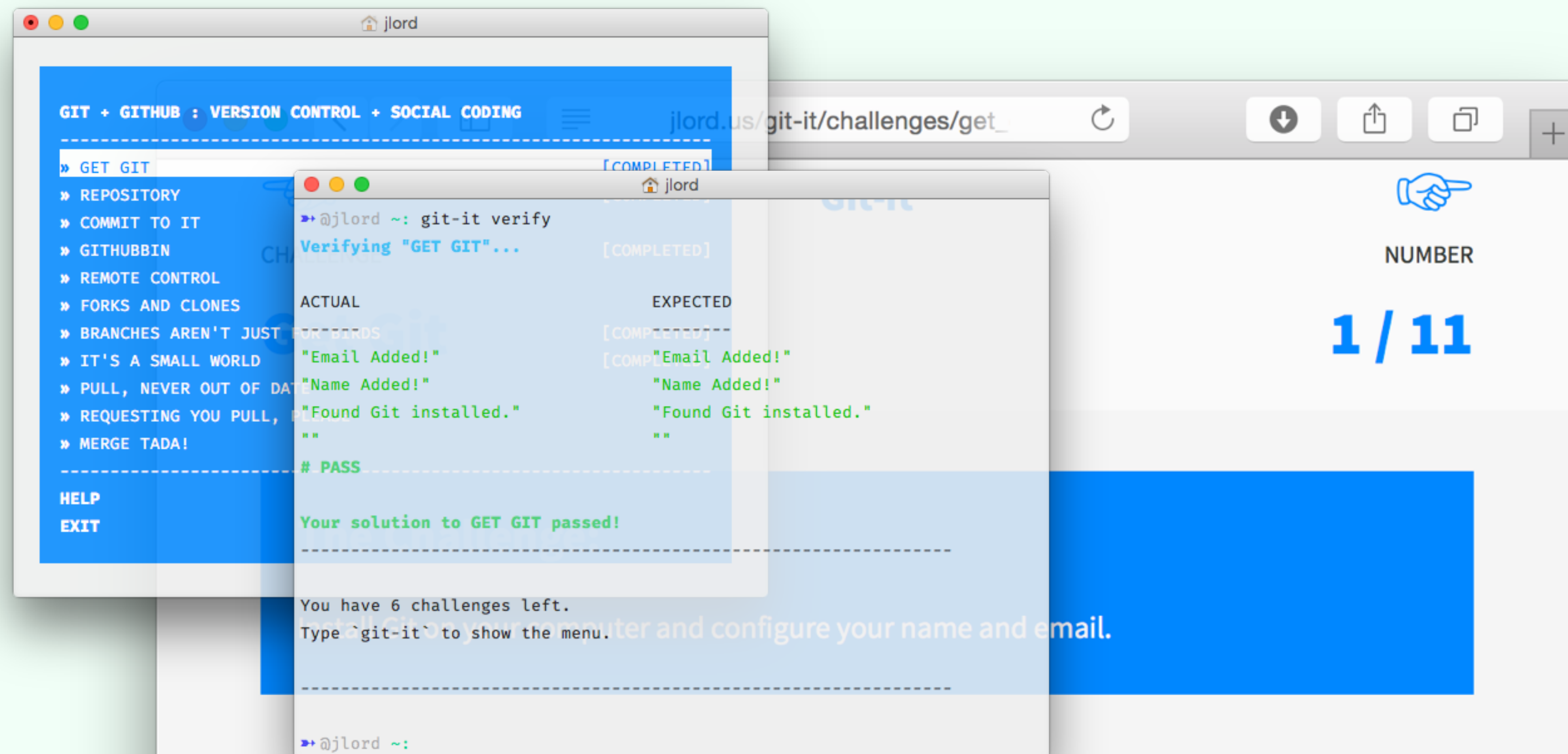
Original Git-it

Terminal for Git

Terminal for completing Git-it
Guide for Understanding

and

Installing Node then
installing Git-it from the
command line before you
learn the command line



Git

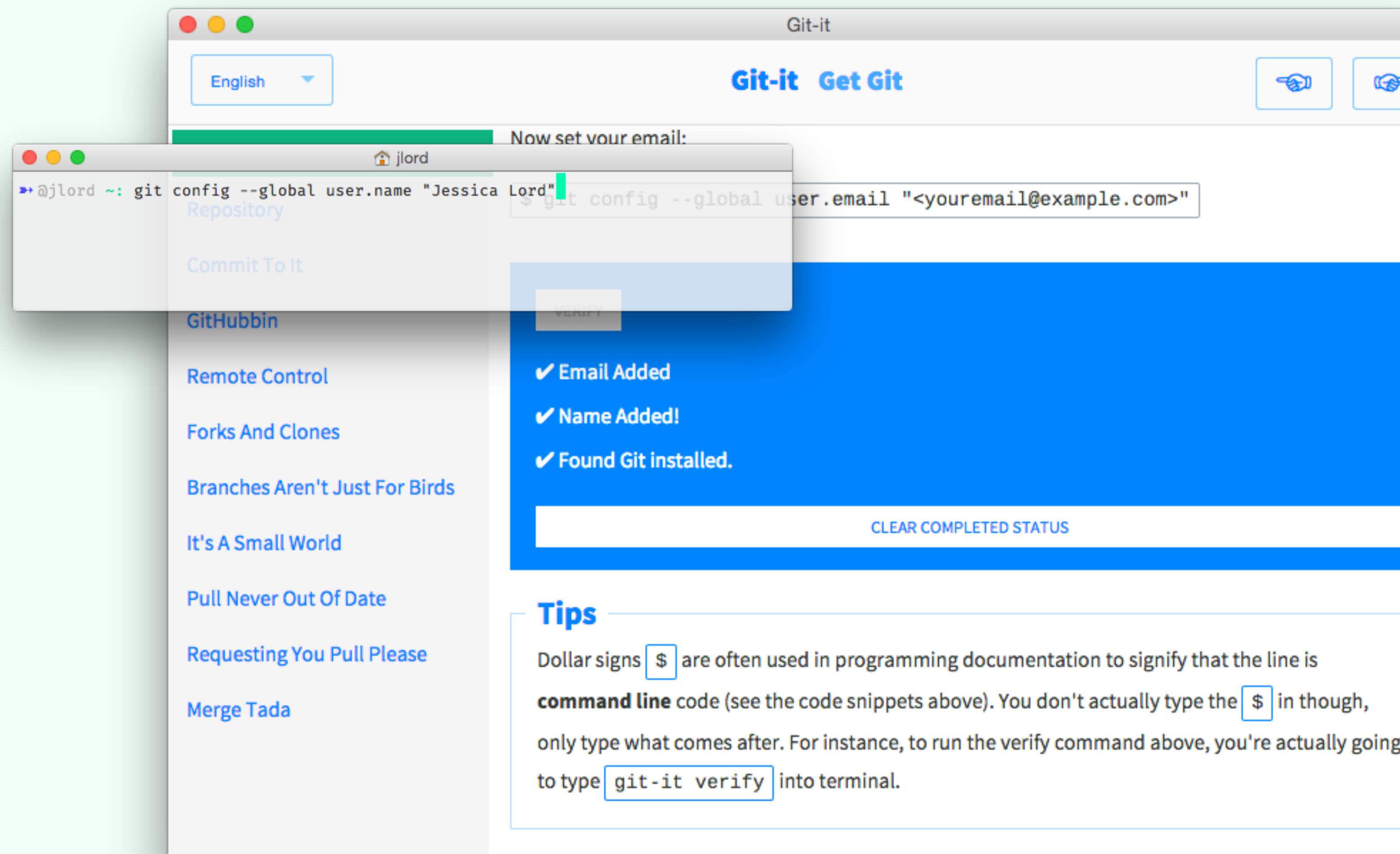
Git is **open source software** (free for anyone to use) written by Linus Torvalds who also wrote

Changes in Experience

Git-it Electron

Use terminal for using Git

Use the app for learning
& completing challenges



Actually Porting Git-it to Electron

Started with the Quick Start Guide

```
var app = require('app'); // Module to control application life.
var BrowserWindow = require('browser-window'); // Module to create native
browser window.

// Report crashes to our server.
require('crash-reporter').start();

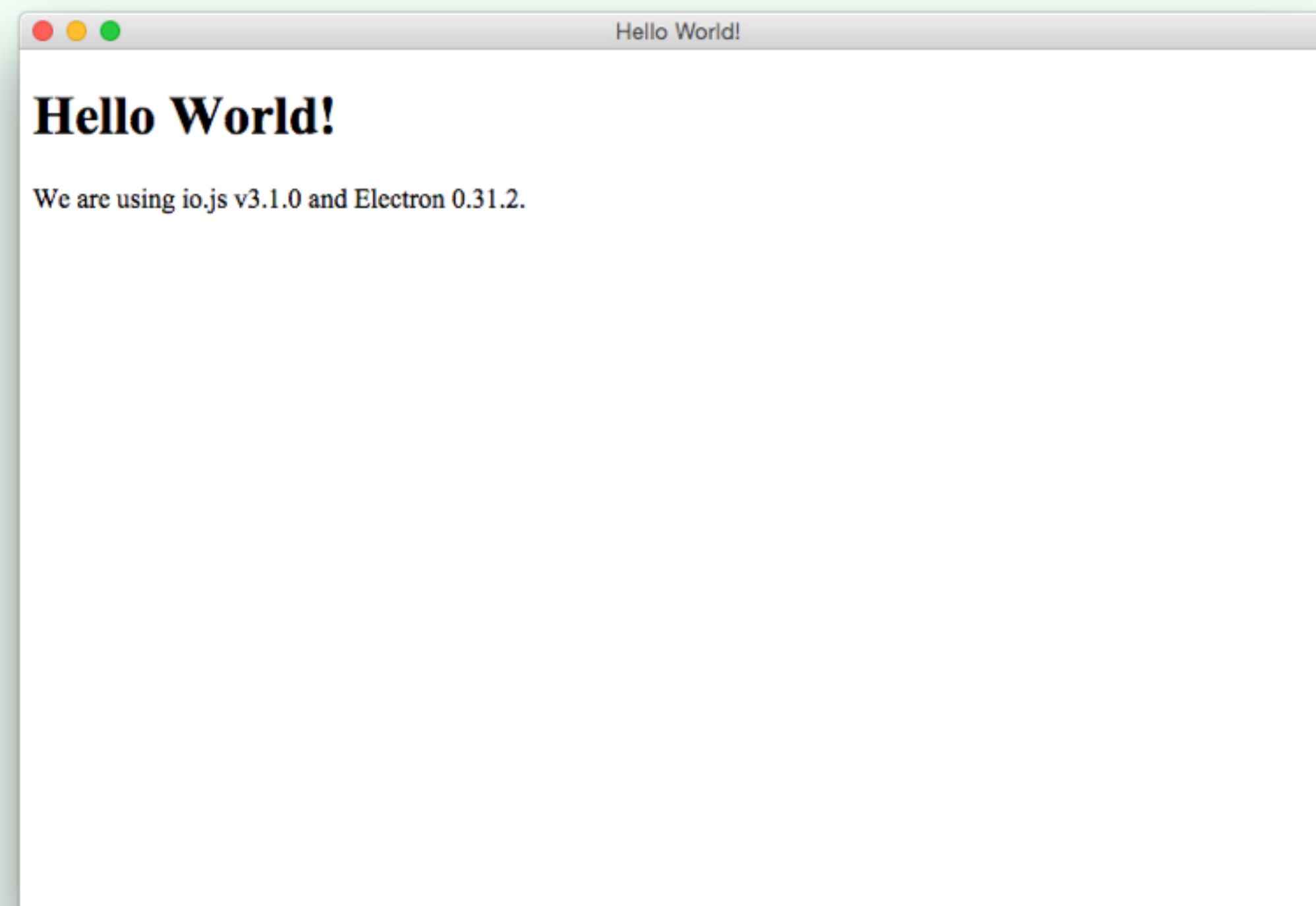
// Keep a global reference of the window object, if you don't, the window will
// be closed automatically when the JavaScript object is garbage collected.
var mainWindow = null;

// Quit when all windows are closed.
app.on('window-all-closed', function() {
  // On OS X it is common for applications and their menu bar
  // to stay active until the user quits explicitly with Cmd + Q
  if (process.platform !== 'darwin') {
    app.quit();
  }
});

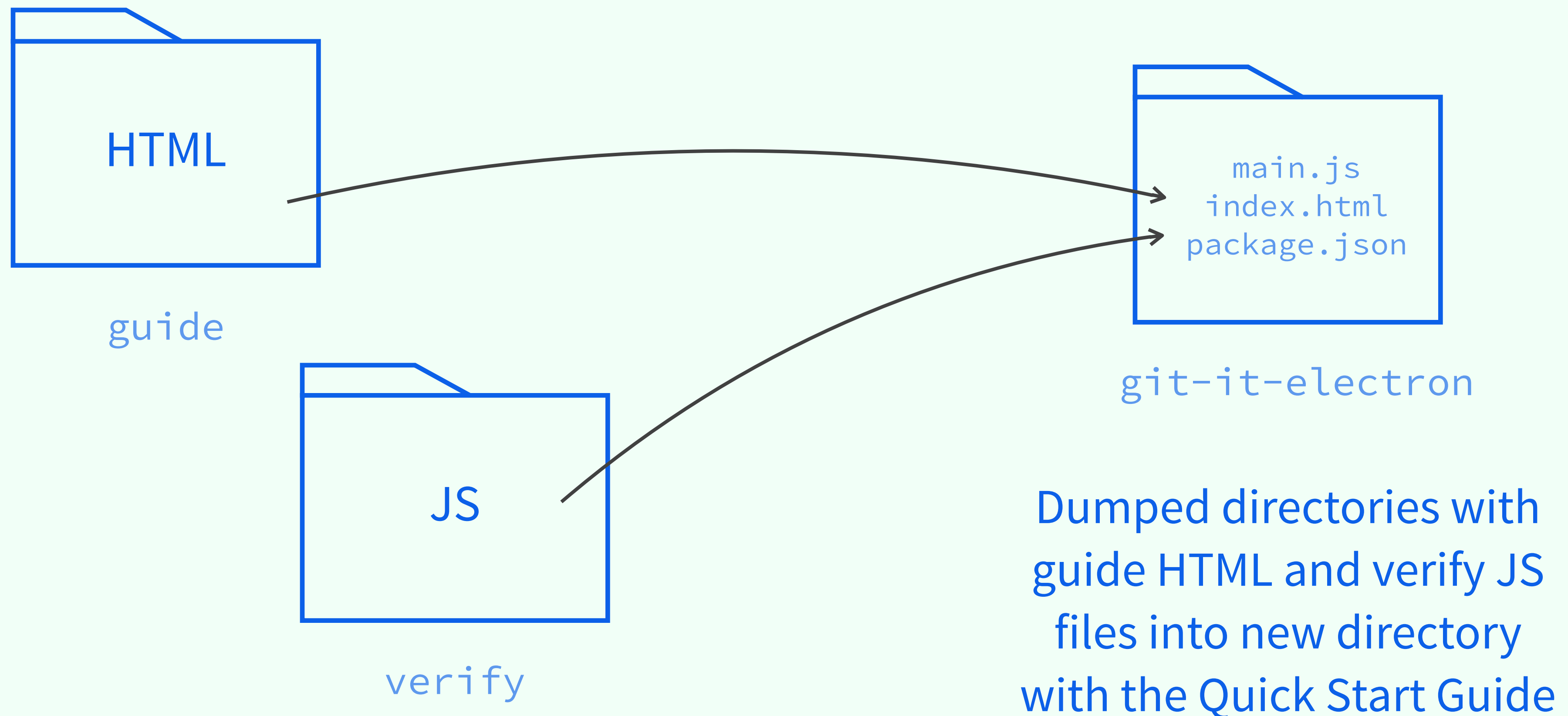
// This method will be called when Electron has finished
// initialization and is ready to create browser windows.
app.on('ready', function() {
  // Create the browser window.
```


Electron Quick Start Guide

`package.json, main.js, index.html`



Actually Porting Git-it to Electron



Hurray, Almost Done!

But not quite.

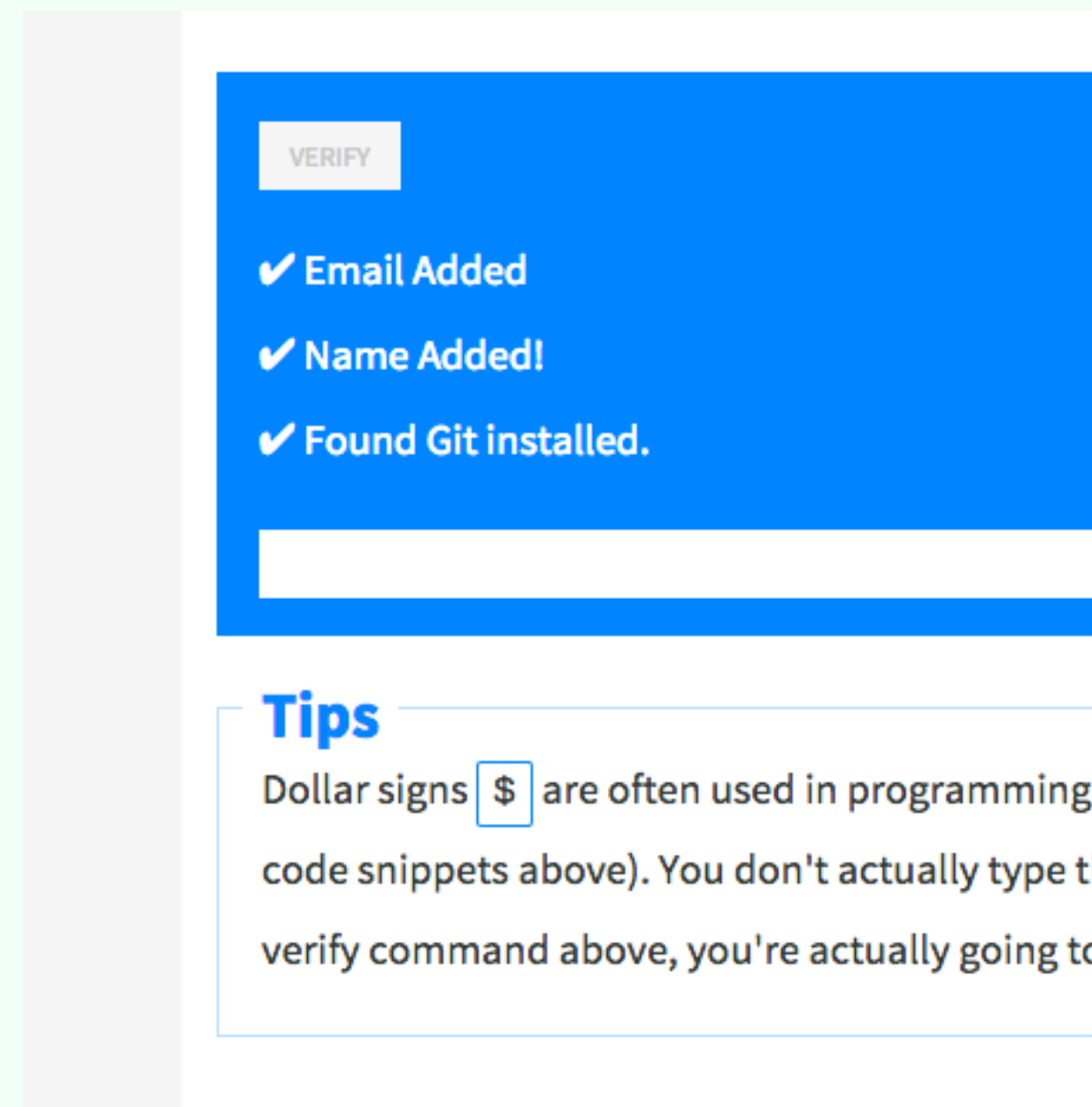
Build verifying and UI into
the application

Started with verify scripts as is.

Added helpers to interact with the DOM and save user progress.

Verify Challenges within the app

Unique verify scripts rely on shared helpers



Adding
completion
elements to DOM

Verify Challenges within the app

Unique verify scripts rely on shared helpers

```
if (gitOutput.match('git version')) {  
  counter++  
  addToList('Found Git installed.', true)  
}
```

Adding completion elements to DOM

Verify Challenges within the app

Unique verify scripts rely on shared helpers



challenge in terminal). Have fun!

GET STARTED!

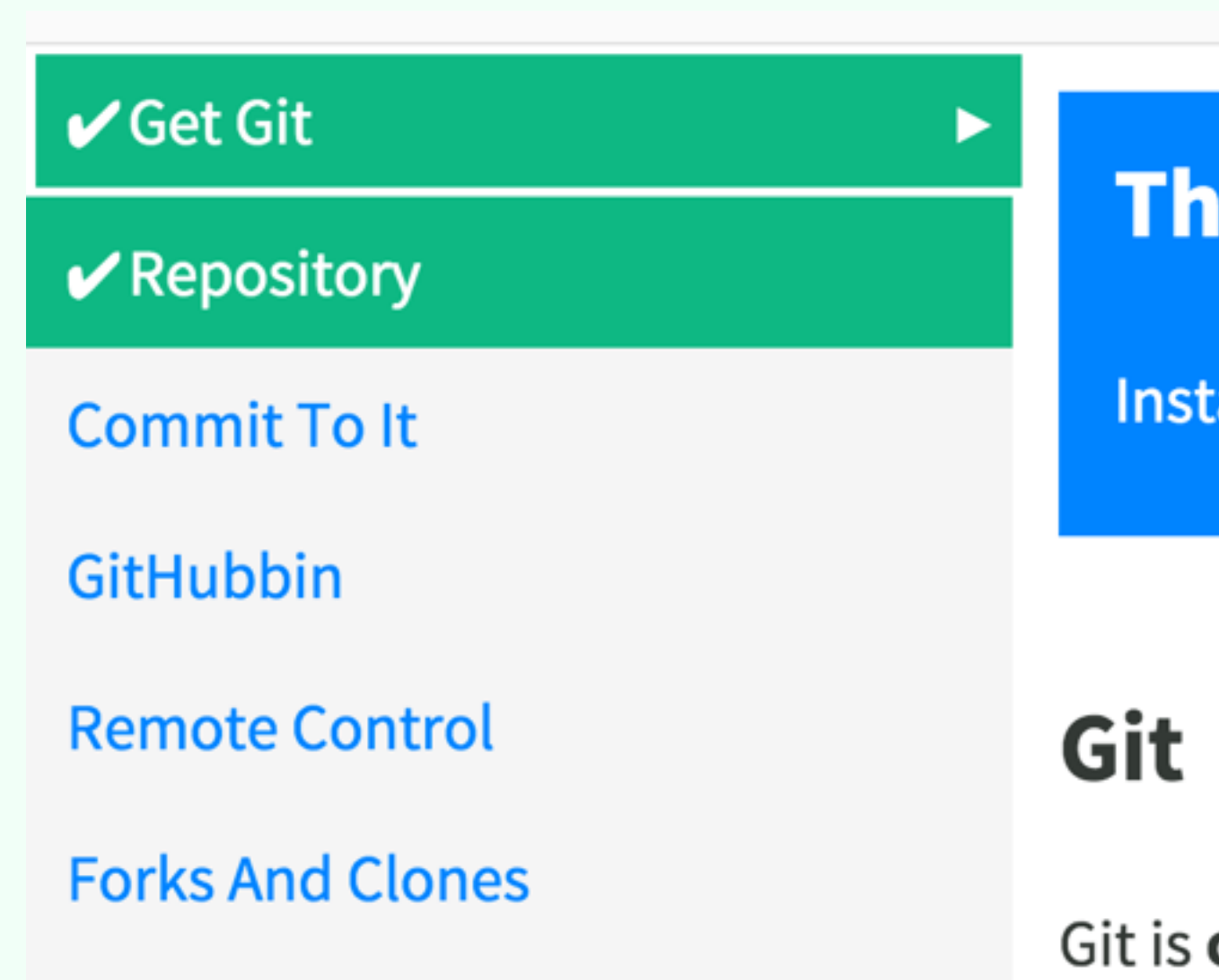
Git-it Challenges

1. ✓ **Get Git** Install and configure Git
2. ✓ **Repository** Create a local repository
3. **Commit to it** Check status, add and commit c
4. **GitHubbin** Get a GitHub account

Reading and
writing user data
and updating DOM

Verify Challenges within the app

Unique verify scripts rely on shared helpers



Reading and
writing user data
and updating DOM

Verify Challenges within the app

Unique verify scripts rely on shared helpers

```
var getData = function () {  
  var data = {}  
  data.path = ipc.sendSync('getUserDataPath', null)  
  data.contents = JSON.parse(fs.readFileSync(data.path))  
  return data  
}
```

Get user data from **main process** with **ipc**

Verify Challenges within the app

Unique verify scripts rely on shared helpers

```
ipc.on('getUserDataPath', function (event) {  
  event.returnValue = userDataPath  
})
```

Return the path from **main process** to the **render process**

Verify Challenges within the app

Unique verify scripts rely on shared helpers

```
function checkCompletedness (challenge) {  
  data = userData.getData()  
  if (data.contents[challenge].completed) {  
    // Do DOM things  
  }  
}
```

Checking challenge completion status and updating DOM

Verify Challenges within the app

Unique verify scripts rely on shared helpers

```
var updateData = function (challenge) {  
  var data = getData()  
  data.contents[challenge].completed = true  
  
  fs.writeFile(data.path, JSON.stringify(data.contents, null, ' '), cb)  
}
```

Updating user data by writing to the file system

Those are the basics of verifying challenges, showing completion and writing completion status.

More fun stuff

Spawn Exec

Most challenges verification involves spawning Git and comparing output

```
exec('config user.email', function (err, stdout, stderr) {  
  if (err) {  
    addToList('Error: ' + err.message, false)  
    return helper.challengeIncomplete()  
  }  
  var email = stdout.trim()  
  if (email === '') {  
    addToList('No email found.', false)  
    helper.challengeIncomplete()  
  } else {  
    addToList('Email Added', true)  
  }  
})
```

Portable Git/ Spawn Git

Windows won't find Git by default, so we provide our own.

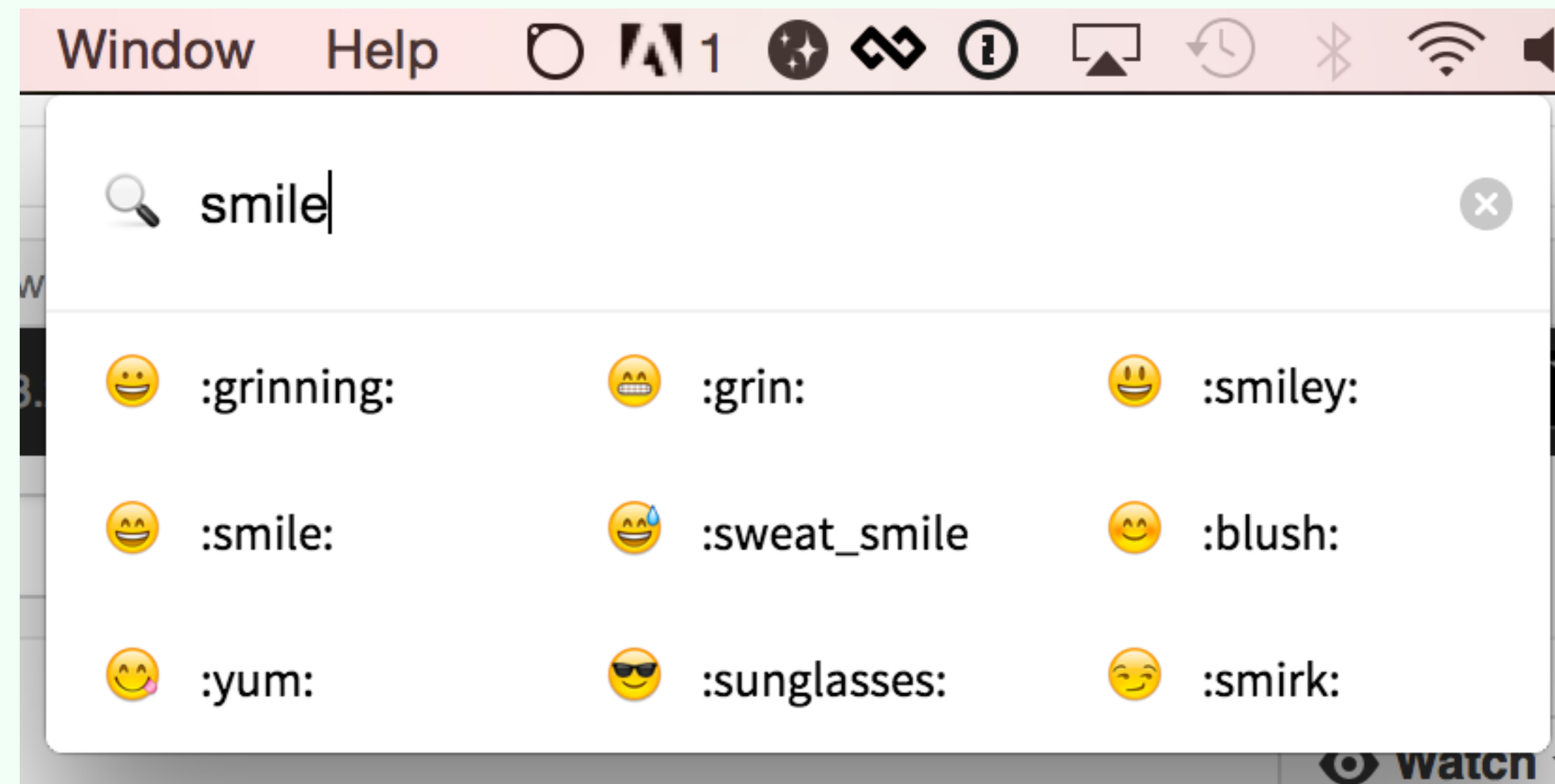
```
var winGit = path.join(__dirname, '../assets/PortableGit/bin/git.exe  ')\n\nmodule.exports = function spawnGit (command, options, callback) {\n  if (typeof options === 'function') {\n    callback = options\n    options = null\n  }\n  if (os.platform() === 'win32') {\n    exec('"' + winGit + '" ' + command, options, callback)\n  } else {\n    exec('git ' + command, options, callback)\n  }\n}
```


Traditional Chinese

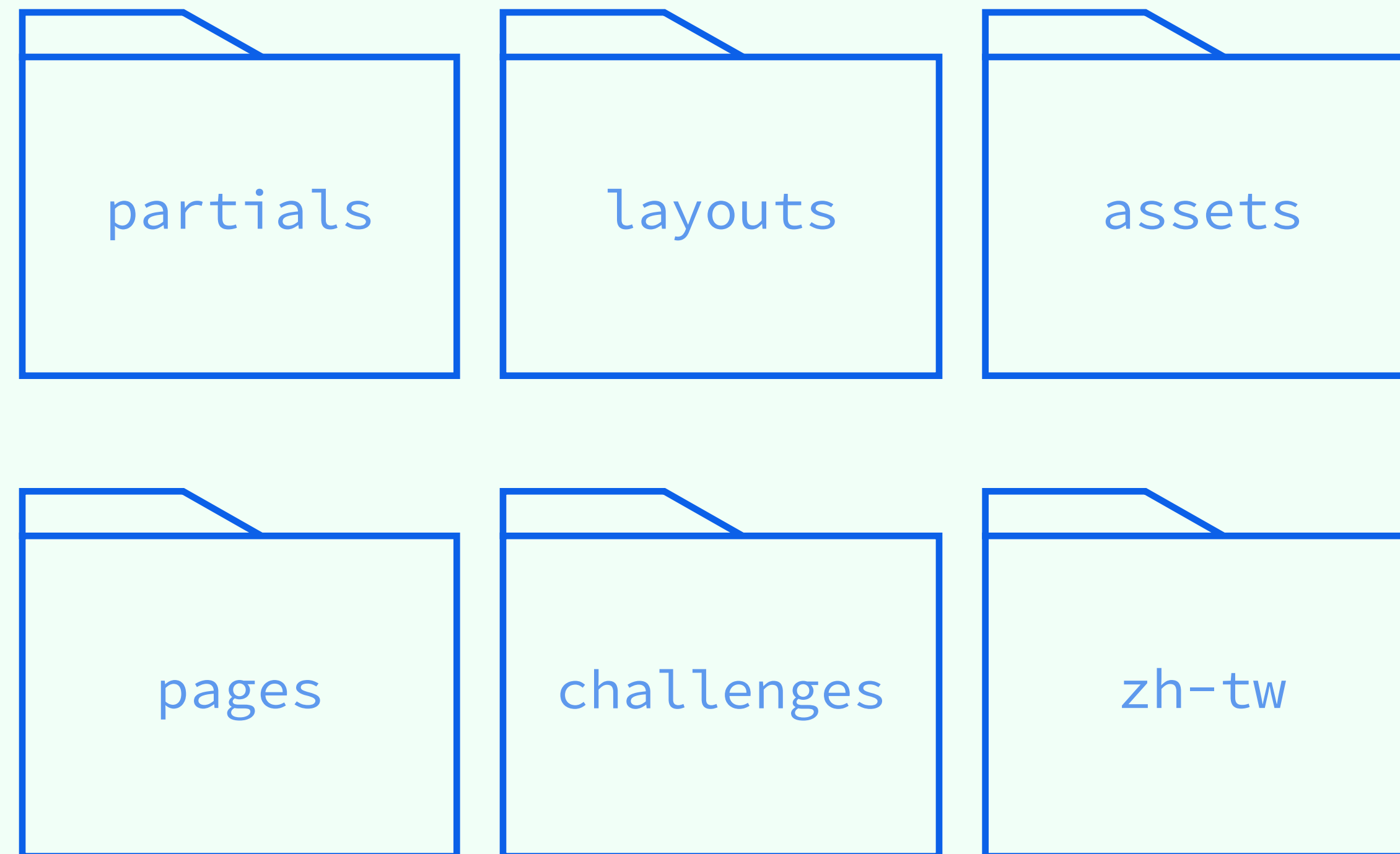


Shoutout to Mu-An

github.com/muan/mojibar



Static Site Generator



NPM SCRIPTS

Make a general framework for creating tutorials.

```
"scripts": {  
  "start": "electron .",  
  "test": "standard **/*.js",  
  "build-chals": "node lib/build-challenges.js",  
  "build-pages": "node lib/build-pages.js",  
  "build-zhtw": "node lib/build-challenges.js zhtw",  
  "build-all": "npm run build-chals && npm run build-pages && npm run build-zhtw",  
  "package": "electron-packager . Git-it --platform=win32,darwin --arch=x64 ...",  
  "pack-win": "electron-packager . Git-it --platform=win32 --arch=ia32 ...",  
  "pack-lin": "electron-packager . Git-it --platform=linux --arch=x64 ...",  
}
```

Next to-dos

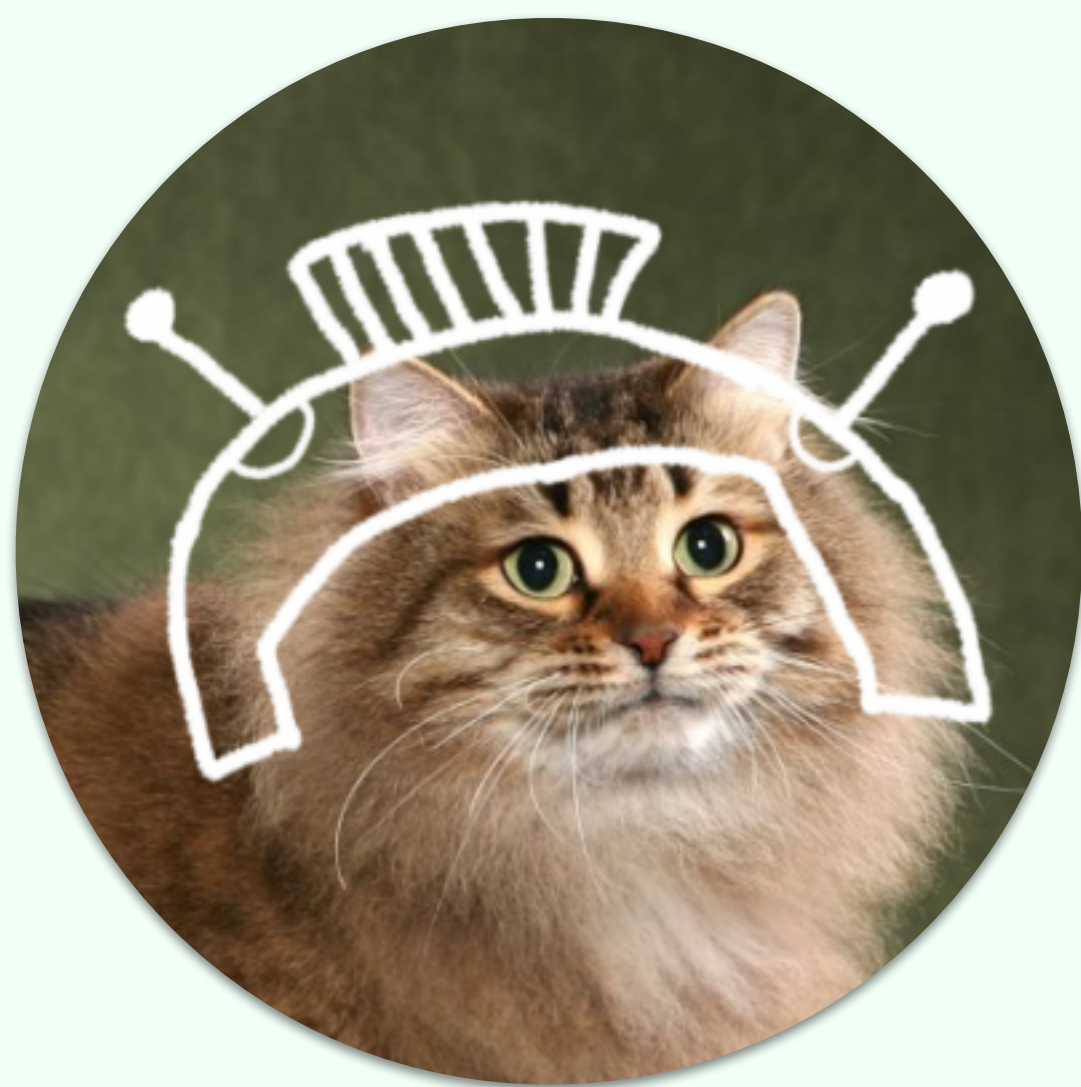
Make a general framework for creating tutorials.

Log helpful errors for users to the console.

More polish and tests.

Other ...

Thanks!



@reporobot