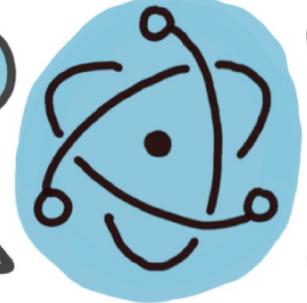


THE  
JOY OF  
DESKTOP  
APPS WITH

# ELECTRON



DAVID NEAL  
@REVERENTGEEK

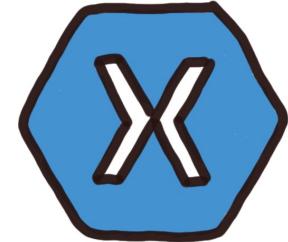


# Up Ahead

- What is Electron?
- Why desktop?
- Crash course
- Tips



# .NET + Mono + Xamarin



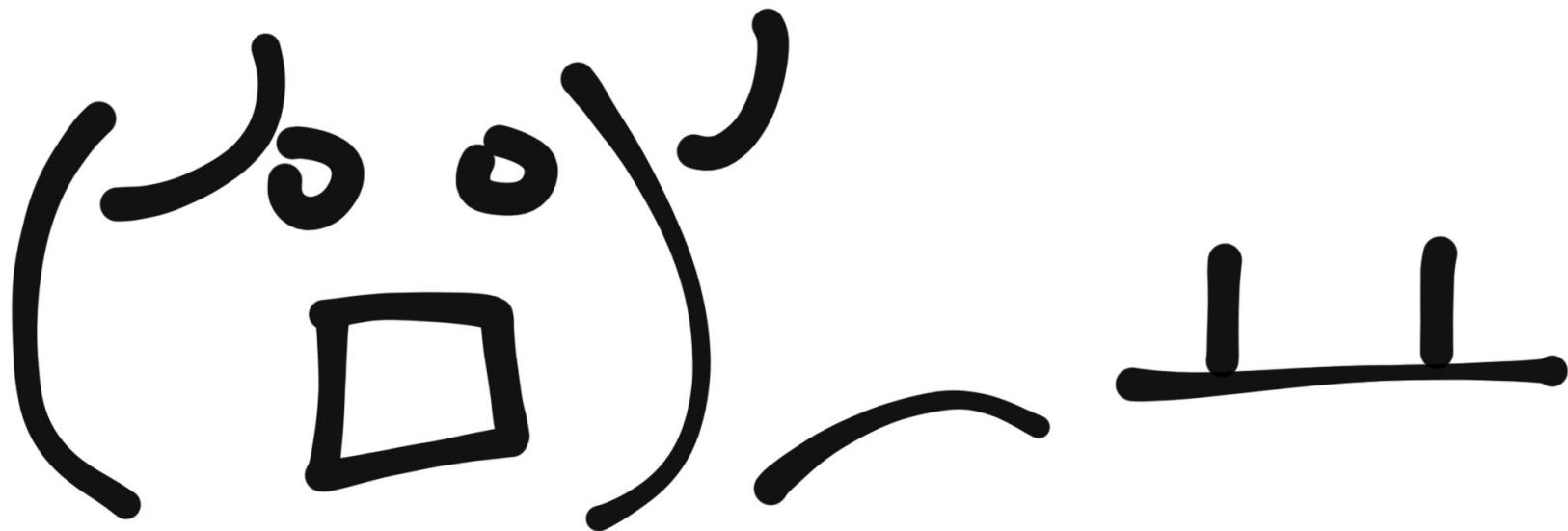
XAMARIN

## Pros:

- Shared .NET code base

## Cons:

- Xamarin Mac != Xamarin iOS/Android
- Native UI is hard
- Deployment
- ~~Licensing~~

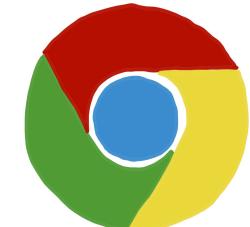
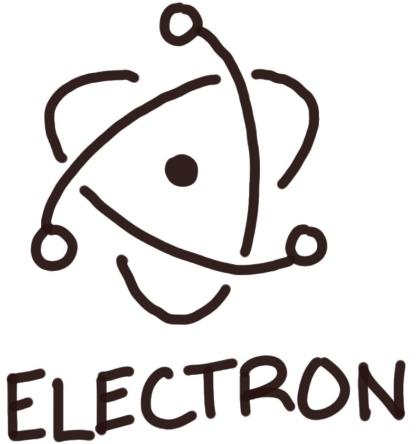
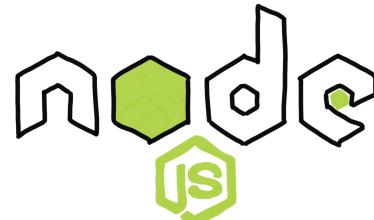
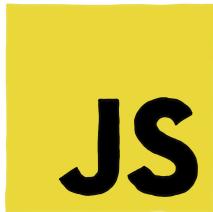


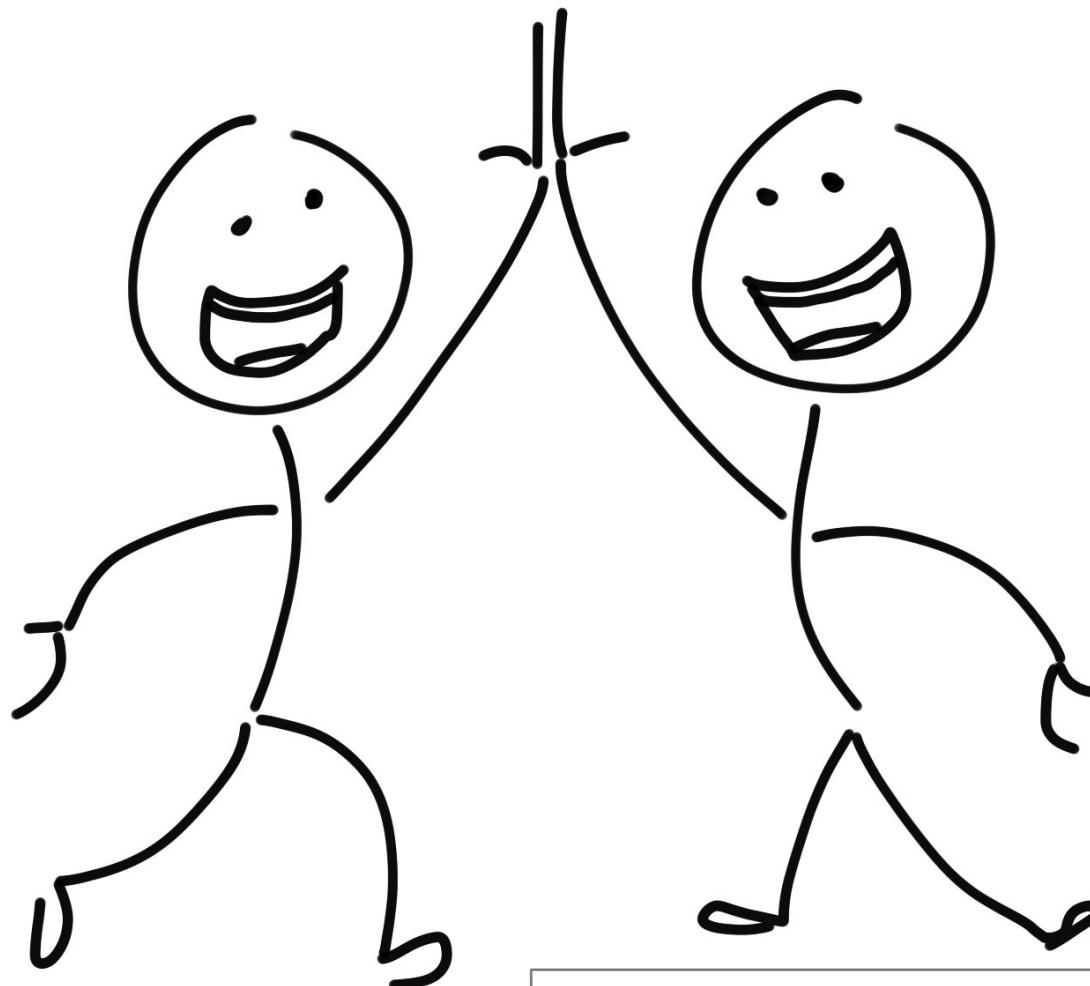
DAVID NEAL @REVERENTGEEK

# Electron

Pros:

- HTML + CSS + JavaScript
- Node.js + Chrome
- No deployment dependencies





DAVID NEAL @REVERENTGEEK

Not Your Dad's



JAVASCRIPT

DAVID NEAL @REVERENTGEEK

# Electron

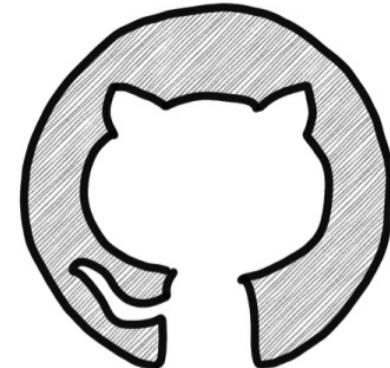
Cons:

- HTML + CSS + JavaScript



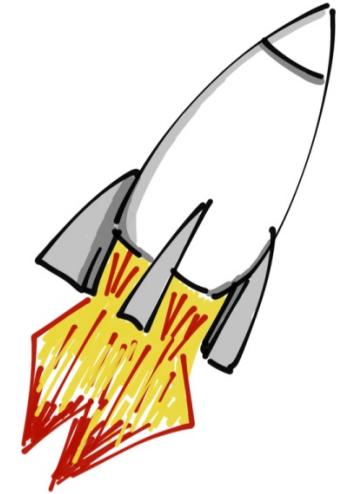
# History

- Created by GitHub for Atom
- Formerly Atom Shell
- Active since January 2013
- 1.0 release May 2016
- 1.4.3 release as of Oct 6<sup>th</sup>



# Electron Features

- Rapid development
- Themes
- Shared code/UI
- Deployment + “silent” updates
- Native UX



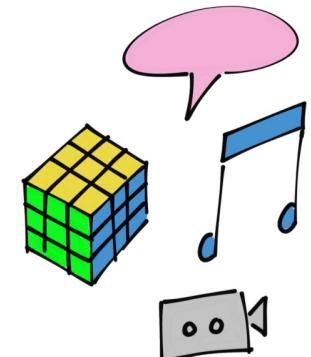
# Why Desktop Apps?

- Offline
- Printers, devices, other local hardware
- On-premises
- Internal, LOB
- Edit local files
- App Store
- Kiosk
- Desktop > Intranet
- Sometimes it “just feels right”

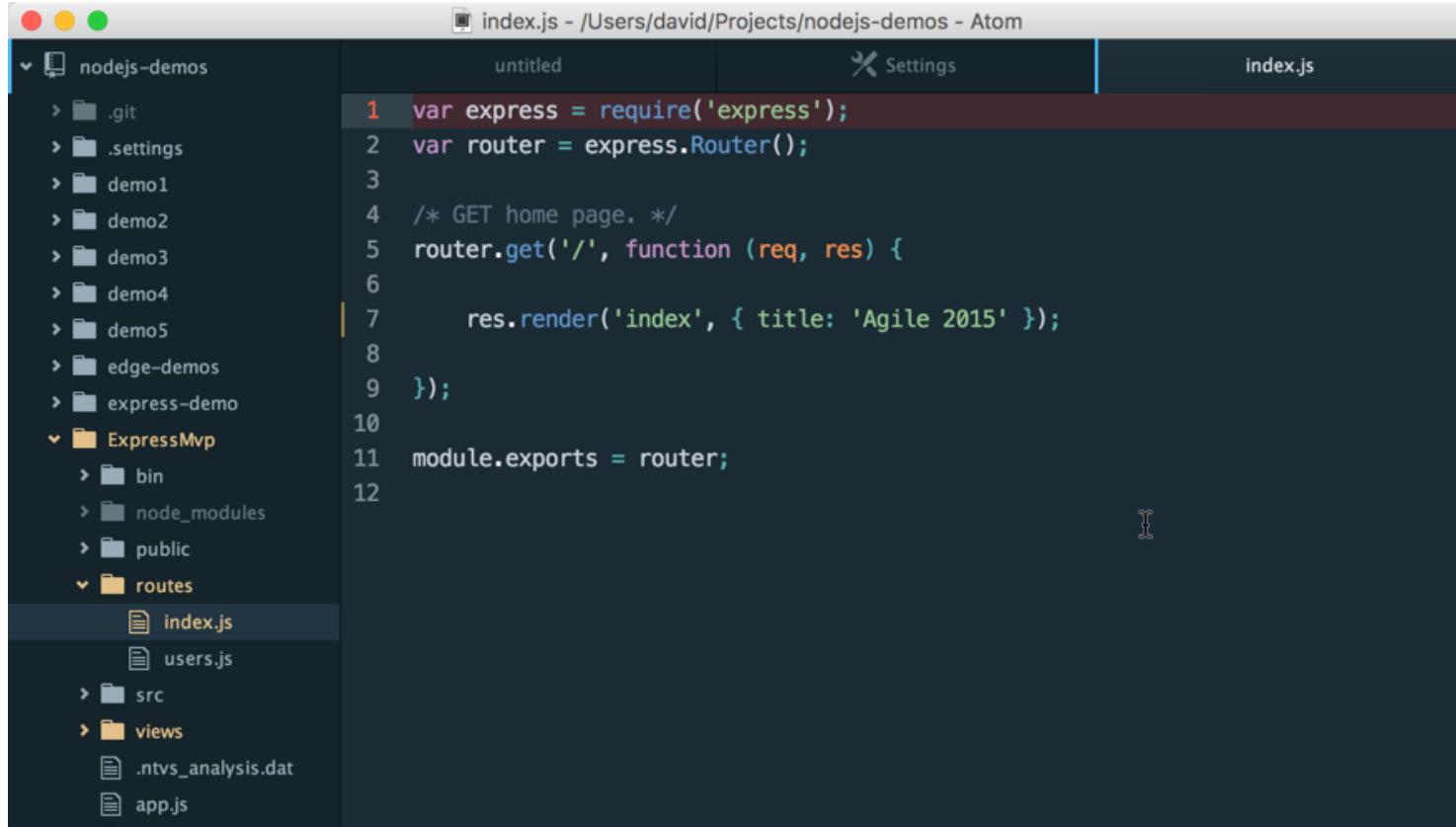


# Desktop App Ideas

- Disconnected data entry
  - Editor
  - Time management
  - Media player
  - Email client
  - Messaging, collaboration
  - Kiosk
  - Mapping, route planner
  - Social media client
  - Calendar
- Bulk media editor
  - File management, backup
  - Document generation, reading
  - Audio/video conferencing
  - Games



# Atom



The screenshot shows the Atom code editor interface. The title bar reads "index.js - /Users/david/Projects/nodejs-demos - Atom". The left sidebar displays a file tree for the project "nodejs-demos". The main editor area contains the following code:

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function (req, res) {
6
7     res.render('index', { title: 'Agile 2015' });
8
9 });
10
11 module.exports = router;
12
```

The file tree on the left shows the following structure:

- nodejs-demos
  - .git
  - .settings
  - demo1
  - demo2
  - demo3
  - demo4
  - demo5
  - edge-demos
  - express-demo
  - ExpressMvp
    - bin
    - node\_modules
    - public
    - routes
      - index.js
      - users.js
    - src
    - views
  - .ntvs\_analysis.dat
  - app.js
  - ExpressMvp.njsproj
  - package.json
  - README.md

DAVID NEAL @REVERENTGEEK

# Slack

Slack

LeanKit David Neal CHANNELS # all-hands-play # all-hands-work # bike-club # book-club # emergingleaders # franklin-visits # leankit-jam # leankit-pro-tips # learning-and-developm # nerdcore # news # pd # random # support # teamkit # the-playbook +51 More... DIRECT MESSAGES slackbot

ermahgerd-it\_s-cats

September 18th

David Neal 3:26 PM Uploaded an image: I WILL have my REVENGE



Mohammed Khalid 3:31 PM @david.neal: saw this earlier on my ca the ever scheming cat!

DAVID NEAL @REVERENTGEEK

# Visual Studio Code

The screenshot shows the Visual Studio Code interface with the title bar "index.js - nodejs-demos - Visual Studio Code". The left sidebar displays the file structure under "WORKING FILES" and "NODEJS-DEMOS". The main editor area shows the code for "index.js" in the "ExpressMvp/routes" directory:

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET home page. */
5 router.get('/', function (req, res) {
6
7   res.render('index', { title: 'Agile 2015' });
8
9 });
10
11 module.exports = router;
12
```

The status bar at the bottom indicates "master" and "0 ▲ 0".

DAVID NEAL @REVERENTGEEK

# Postman

Postman

Builder Team Library IN SYNC David No environment

Search History Collections All Me Team

LeanKit API 54 requests Bandit Software Board API GET Get Boards GET Get Board (selected) GET Get Board Identifiers GET Get Board Archive GET Get Newer if Exists GET Check for Updates GET Get Board History Since GET Get Board Backlog GET Get Board Archive GET Get Board Snapshot

Get Board

GET https://reverentgeek.leankit.com/kanban/api/boards/256064019 Params Send Save

Authorization Headers (3) Body Pre-request Script Tests Manage Cookies Generate Code Bulk Edit Presets

✓ Authorization [REDACTED]  
✓ Accept application/json  
✓ Content-Type application/json  
key value

Status: 200 OK Time: 327 ms

Body Cookies (1) Headers (10) Tests

Pretty Raw Preview JSON Save Response

```
1 { "ReplyData": [ 2 { 3 "IconConfiguration": { 4 "IconPathTemplate": "/customicons/{0}/{1}/{2}.{3}" , 5 "AllowedIconImages": [ 6 ] } } ] }
```

# Nylas N1

The screenshot displays the Nylas N1 email application interface. On the left, a sidebar lists various mailboxes: **Inbox** (7 messages), **Starred**, **Important**, **Sent Mail**, **All Mail**, and **Drafts**. Below these are **Views** (**Plugins**), **Labels** (**Housekeeping**, **Iceland**, **Photos**, **Product Updates**, **Travel**), and a search bar at the top.

The main area shows the **Inbox** with the following messages:

- You, Evan, You (3)** 4:01 pm  
Re: Iceland and all the pings!!  
Wow! Iceland looks awesome. On Sa...
- Groupon** Oct 2  
Nyla's Burger Basket — \$50 Grou...
- Expedia Reservations** Oct 2
- Expedia travel confirmation - Se...** Sep 16 - Your flight to San Francisco...
- Christie Zhao** Oct 2
- Labor Day (potential) BBQ or Pi...** Hi All! - Steffi and I were talking tonig...
- Venmo** Oct 2  
Mark Ballesteros paid you \$31.30  
Mark Ballesteros - 707 for Horace an...
- Expedia Cancellations** Oct 2
- Itinerary No: 1117736000918 - C...** Canceled Flight Credit Voucher
- Songkick** Oct 2  
New concert for The Weeknd  
Here is a new event for stuff in your tr...
- Project Fi** Oct 2  
Project Fi is now available on m...

A message from **Evan Morikawa** is selected, showing the following content:

You Where are the Iceland photos?! 🐾 🐾 🐾 🐾 🐾

Evan To: You Cc: Lana

Here's one! More coming soon.

... Show previous

New Message

To: \_\_\_\_\_ Cc: \_\_\_\_\_

Subject: \_\_\_\_\_

**DAVID NEAL @REVERENTGEEK**

Jibo

Hello! I'm Jibo.



DAVID NEAL @REVERENTGEEK



DAVID NEAL @REVERENTGEEK

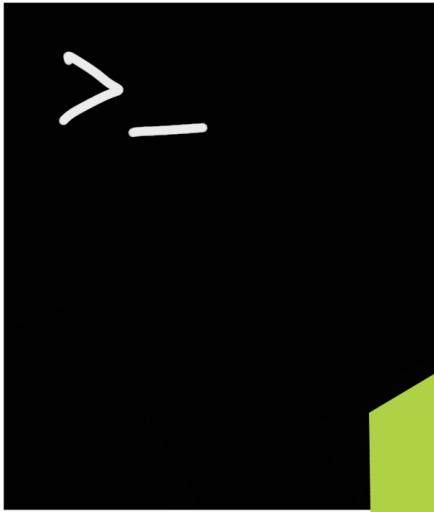
```
> npm install electron  
> touch main.js  
> touch index.html
```



PROCESS

( main.js )

DAVID NEAL @REVERENTGEEK



PROCESS

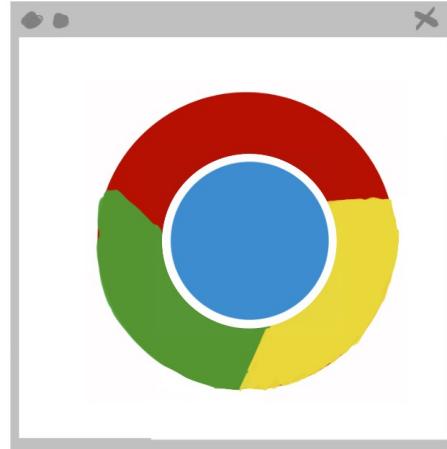
( main.js )

DAVID NEAL @REVERENTGEEK



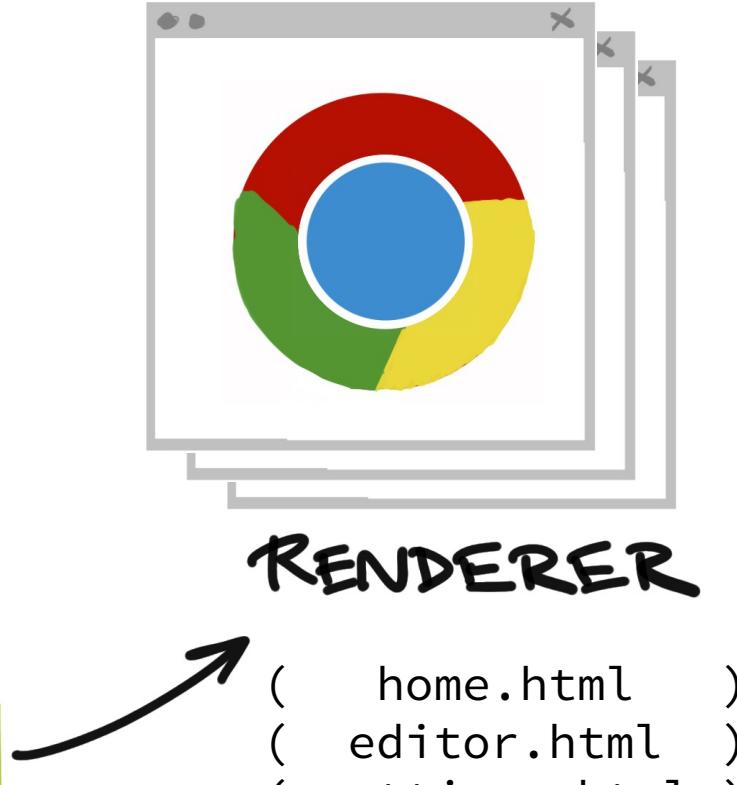
PROCESS

( main.js )



RENDERER  
( index.html )

DAVID NEAL @REVERENTGEEK



# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600, // center, resizable, kiosk
9         height: 800 // x, y, min-width, max-width
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12 } );
```

# index.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Demo App</title>
6    </head>
7    <body>
8      <h1>Hello from Electron!</h1>
9    </body>
10   </html>
```

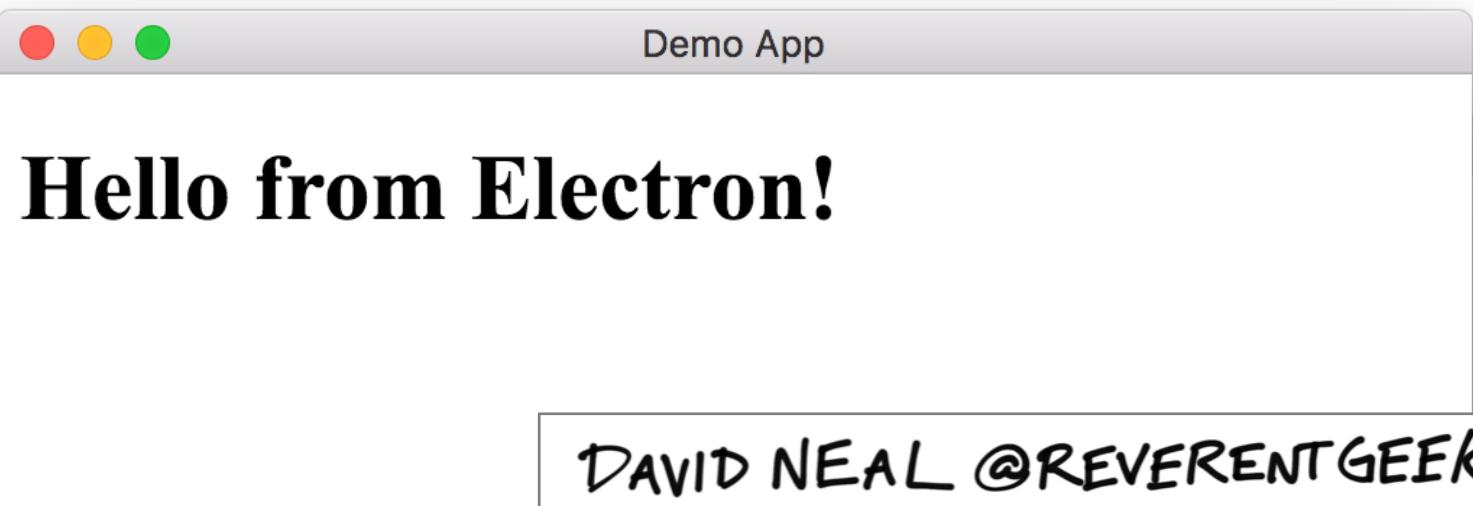
# index.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Demo App</title>
6    </head>
7    <body>
8      <h1>Hello from Electron!</h1>
9    </body>
10   </html>
```

```
> electron main.js
```

# index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo App</title>
6      </head>
7      <body>
8          <h1>Hello from Electron!</h1>
9      </body>
10 </html>
```



# index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo App</title>
6      </head>
7      <body>
8          <h1>Hello from Electron!</h1>
9      </body>
10 </html>
```



Hello from Electron!

# index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo App</title>
6      </head>
7      <body>
8          <h1>Hello from Electron!</h1>
9      </body>
10 </html>
```



# index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo App</title>
6          <script>
7              window.onload = function() {
8                  let msg = "I'm running Node.js " + process.versions.node
9                      + ", Electron " + process.versions.electron
10                     + " and Chrome " + process.versions.chrome;
11                     document.querySelector("h1").textContent = msg;
12             }
13         </script>
14     </head>
```

# index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo App</title>
6          <script>
7              window.onload = function() {
8                  let msg = "I'm running Node.js " + process.versions.node
9                      + ", Electron " + process.versions.electron
10                     + " and Chrome " + process.versions.chrome;
11                     document.querySelector("h1").textContent = msg;
12             }
13         </script>
14     </head>
```

# index.html

```
11      document.querySelector("h1").textContent = msg;  
12  
13      const fs = require("fs");  
14      fs.readdir(process.env.HOME, (err, entries) => {  
15          entries.forEach( entry => {  
16              if ( !entry.startsWith( "." ) ) {  
17                  let li = document.createElement("li");  
18                  li.textContent = entry;  
19                  document.querySelector("ul").appendChild(li);  
20              }  
21          });  
22      });
```

# index.html

```
11   document.querySelector("h1").textContent = msg;  
12  
13   const fs = require("fs");  
14   fs.readdir(process.env.HOME, (err, entries) => {  
15     entries.forEach( entry => {  
16       if ( !entry.startsWith( "." ) ) {  
17         let li = document.createElement("li");  
18         li.textContent = entry;  
19         document.querySelector("ul").appendChild(li);  
20       }  
21     });  
22   });
```

# index.html

```
11   document.querySelector("h1").textContent = msg;  
12  
13   const fs = require("fs");  
14   fs.readdir(process.env.HOME, (err, entries) => {  
15     entries.forEach( entry => {  
16       if ( !entry.startsWith( "." ) ) {  
17         let li = document.createElement("li");  
18         li.textContent = entry;  
19         document.querySelector("ul").appendChild(li);  
20       }  
21     });  
22   });
```

```
6 <script>
7   window.onload = function() {
8     let msg = "I'm running Node.js " + process.versions.node
9       + ", Electron " + process.versions.electron
10      + " and Chrome " + process.versions.chrome;
11      document.querySelector("h1").textContent = msg;
12
13    const fs = require("fs");
14    fs.readdir(process.env.HOME, (err, entries) => {
15      err
16
17      I'm running Node.js 6.1.0, Electron 1.2.7
18      and Chrome 51.0.2704.106
19
20    })
21  }
22  });
23 }
24 </script>
25 </head>
```



Demo App

- Applications
- Creative Cloud Files
- Desktop
- Documents
- Downloads

DAVID NEAL @REVERENTGEEK

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12    mainWindow.openDevTools( { detach: true } );
13} );
```

# main.js

```
1 const electron = require( "electron" );
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
8         width: 600,
9         height: 800
10    } );
11    mainWindow.loadURL( "file://" + __dirname + "/index.html" );
12    mainWindow.openDevTools( { detach: true } );
13 } );
```

# main.js

```
2 const app = electron.app;
3 const BrowserWindow = electron.BrowserWindow;
4 let mainWindow;
5
6 app.on( "ready", function() {
7     mainWindow = new BrowserWindow( {
```

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar lists files: 'top', 'file:///' (containing 'index.html'), 'Users/david/Projects/demos/electron-demo' (containing 'index.html'), and '(no domain)'. The main pane displays the content of 'index.html' with line numbers 2 through 14. Line 14 is highlighted with a blue box. The right sidebar shows the execution context for line 13:

- Watch: Not Paused
- Call Stack: Not Paused
- Scope: Not Paused
- Breakpoints: Not Paused

```
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title>Demo App</title>
6         <script>
7             window.onload = () => {
8                 let msg = "I'm running Node.js
9                     + ", Electron " + process.v
10                    + " and Chrome " + process.
11                    document.querySelector("h1"
12
13
14
```

DAVID NEAL @REVERENTGEEK

There are  hard problems  
in software development:

There are  2 hard problems  
in software development:

## 1. CACHE INVALIDATION

There are  2 hard problems  
in software development:

1. CACHE INVALIDATION
2. NAMING THINGS

There are  hard problems  
in software development:

1. CACHE INVALIDATION
2. NAMING THINGS
3. OFF-BY-ONE ERRORS

4

There are ~~3~~<sup>4</sup> hard problems  
in software development:

1. CACHE INVALIDATION
2. NAMING THINGS
3. OFF-BY-ONE ERRORS

4

There are ~~3~~<sup>4</sup> hard problems  
in software development:

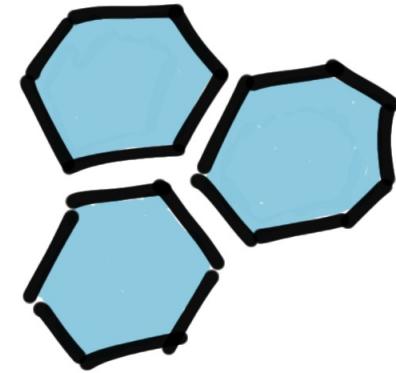
1. CACHE INVALIDATION
2. NAMING THINGS
3. OFF-BY-ONE ERRORS
4. CHOOSING THE PERFECT  
ANIMATED GIF!



DAVID NEAL @REVERENTGEEK

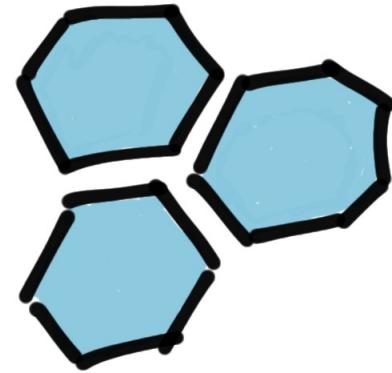
# Process modules

- app
- ipc
- dialog
- menu, menu-item
- power-monitor
- tray



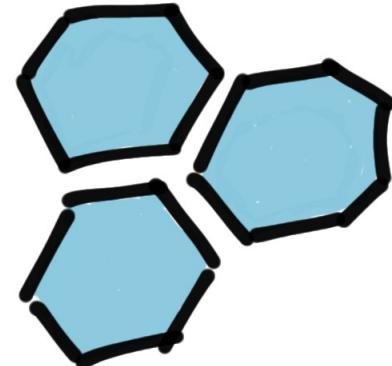
# Render modules

- ipc
- remote
- web-frame



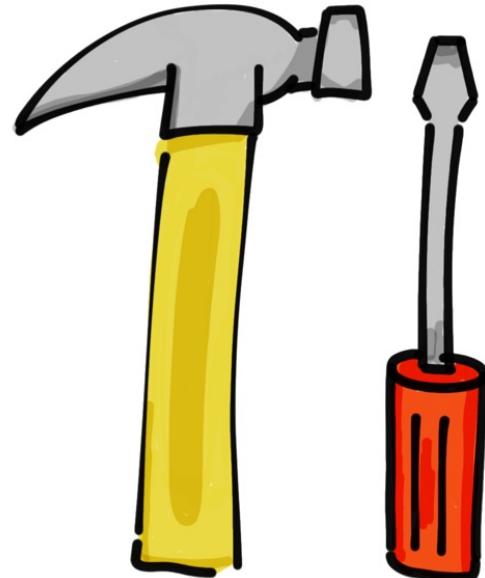
# Modules available to both

- clipboard
- crash-reporter
- native-image
- screen
- shell



# Recommended Tools

- electron-debug
- electron-reload
- electron-packager
- electron-builder
- electron-updater
- electron-mocha



# Devtron

The screenshot shows the Devtron developer tools interface. At the top, there's a navigation bar with tabs: Elements, Console, Sources, Network, Timeline, Profiles, Resources, Security, Audits, and Devtron. The Devtron tab is active. Below the navigation bar is a search bar labeled "Search modules" and a "Load Graph" button.

The main area displays a "Require Graph" for the "Renderer Process - 16 files, 57 KB". It includes sections for Event Listeners, IPC, Lint, and About. The "About" section lists the following modules:

Module	Version	Size	File
Electron	0.1.0	828 B	▼ index.html - /Users/david/Projects/demos/electron-demo
electron-prebuilt	1.0.1	1 KB	reset-search-paths.js - electron.asar/common
electron-prebuilt	1.0.1	2 KB	init.js - electron.asar/common
electron-prebuilt	1.0.1	767 B	► electron.js - electron.asar/renderer/api(exports)
electron-prebuilt	1.0.1	7 KB	override.js - electron.asar/renderer
electron-prebuilt	1.0.1	16 KB	► web-view.js - electron.asar/renderer/web-view
electron-prebuilt	1.0.1	10 KB	web-view-attributes.js - electron.asar/renderer/web-view
devtron	1.1.0	322 B	api.js - /Users/david/Projects/demos/electron-demo/node_modules/devtron

At the bottom, there's a "Console" tab with a log message:

```
> require('devtron').install()
Installing Devtron from /Users/david/Projects/demos/electron-
demo/node_modules/devtron
< undefined
>
```

DAVID NEAL @REVERENTGEEK

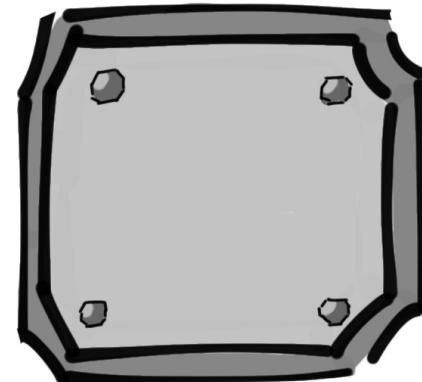


# SPECTRON

An Electron Testing Framework

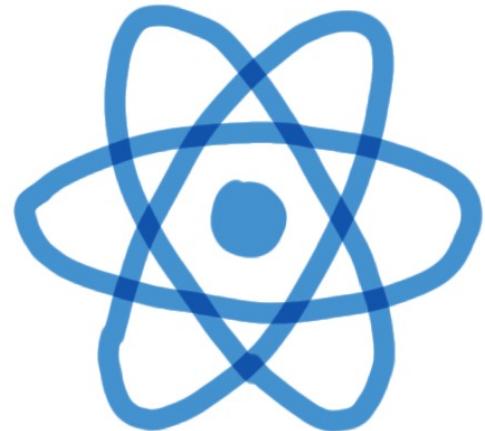
# Boilerplate Projects

- electron-boilerplate
- electron-react-boilerplate



# electron-prebuilt-compile

- ES6/ES7 (Babel), TypeScript, CoffeeScript
- React.js
- LESS
- Jade



Radar Report Generator

**Radar Report**

**Account Name**

**Email Address**

**Password**

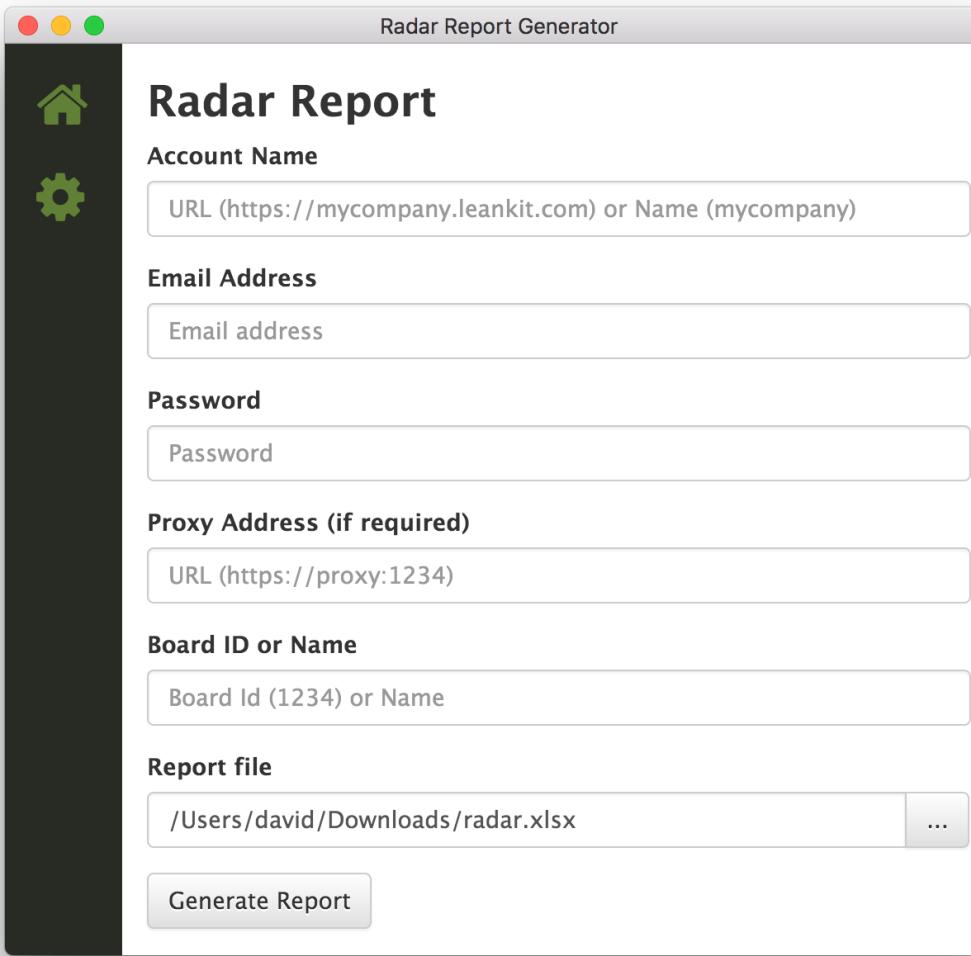
**Proxy Address (if required)**

**Board ID or Name**

**Report file**

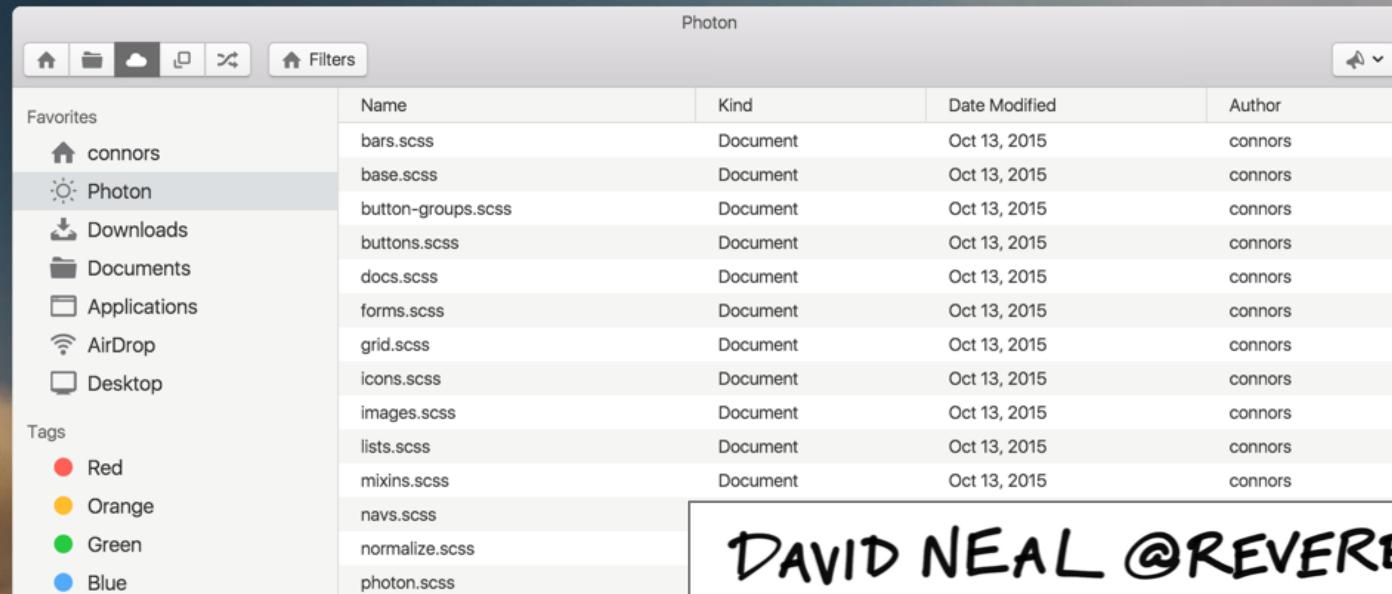
...

**Generate Report**



DAVID NEAL @REVERENTGEEK

The fastest way to build beautiful Electron apps using  
simple HTML and CSS



A screenshot of the Photon application interface. The window title is "Photon". The left sidebar contains a "Favorites" section with icons for "connors" (home), "Photon" (sun), "Downloads" (cloud), "Documents" (file folder), "Applications" (square), "AirDrop" (Wi-Fi), and "Desktop" (monitor). Below that is a "Tags" section with color-coded circles for "Red" (red), "Orange" (orange), "Green" (green), and "Blue" (blue). The main content area is a table listing files:

	Name	Kind	Date Modified	Author
	bars.scss	Document	Oct 13, 2015	connors
	base.scss	Document	Oct 13, 2015	connors
	button-groups.scss	Document	Oct 13, 2015	connors
	buttons.scss	Document	Oct 13, 2015	connors
	docs.scss	Document	Oct 13, 2015	connors
	forms.scss	Document	Oct 13, 2015	connors
	grid.scss	Document	Oct 13, 2015	connors
	icons.scss	Document	Oct 13, 2015	connors
	images.scss	Document	Oct 13, 2015	connors
	lists.scss	Document	Oct 13, 2015	connors
	mixins.scss	Document	Oct 13, 2015	connors
	navs.scss	Document	Oct 13, 2015	connors
	normalize.scss	Document	Oct 13, 2015	connors
	photon.scss	Document	Oct 13, 2015	connors

DAVID NEAL @REVERENTGEEK

LeanKit Data Utility

Login Export Import Board Info

Choose Board and Lanes to Export

Select A Board

Board  Backlog

Archive  Include cards archived more than 14 days ago

Filter cards by search (optional)

Optional search term(s)

Include comments in search

Include tags in search

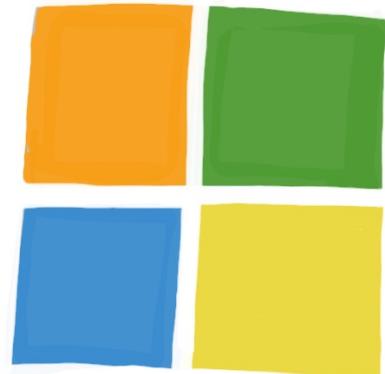
Next



DAVID NEAL @REVERENTGEEK

# Building for Windows

- Windows 7/Server 2008 R2 minimum  
(<https://dev.modern.ie>)
- Visual Studio 2015 Community
- Python 2.7
- Node.js
- Git



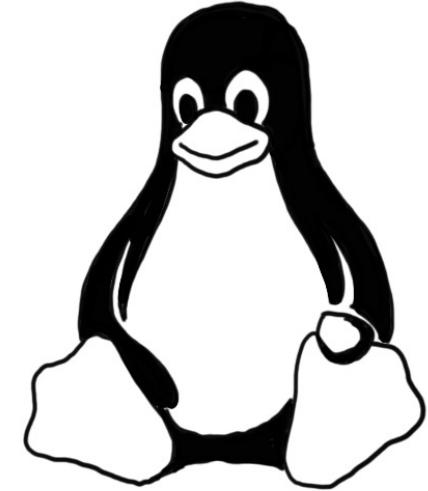
# Building for Mac

- OS X 10.8+
- Xcode 5.1+
- Node.js
- pyobjc (only if using Python installed with Homebrew)



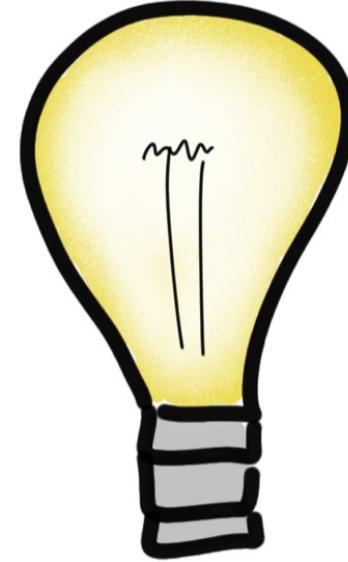
# Building for Linux

- Python 2.7
- Node.js
- Clang 3.4+
- GTK+ & libnotify dev headers



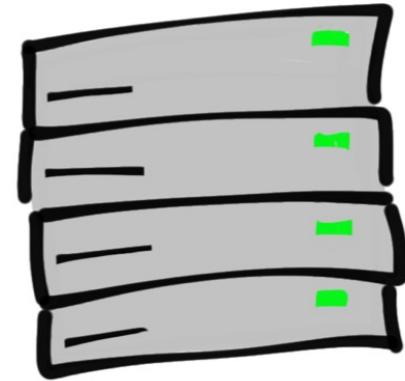
# Tips

- Use CSS default cursor
- `-webkit-user-drag:none`
- `-webkit-user-drag:text`
- Keep windows open
- Offload work to `main` over `ipc`



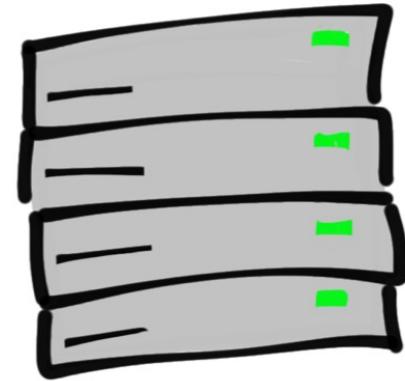
# Local Storage

- Read/write .json files
  - Pro tip: use fs-jetpack
- nedb
- pouchdb



# Relational DB Storage

- seriate
- pg
- mysql
- oracledb



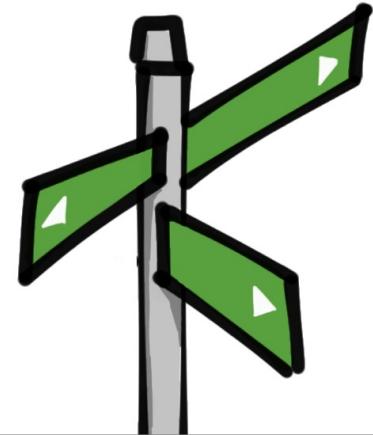
# From here...

<https://github.com/sindresorhus/awesome-electron>

Pluralsight Course: Rob Conery's "Electron Playbook"

<http://electron.atom.io>

<http://atom-slack.herokuapp.com/>



You Don't NEED  
PERMISSION

TO BE

AWESOME!

# THANK You!

DAVID NEAL  
@Reverent Geek  
[david@reverentgeek.com](mailto:david@reverentgeek.com)

