

MARCH 1, 2018

How to make the desktop app with JavaScript (Electron) in 30 minutes

javascript beginner productivity electron

READ NEXT

Using Snippet notes to make programming...

4593 views



Florida Collins
florida18



Junpei Shimo...
j unp1234

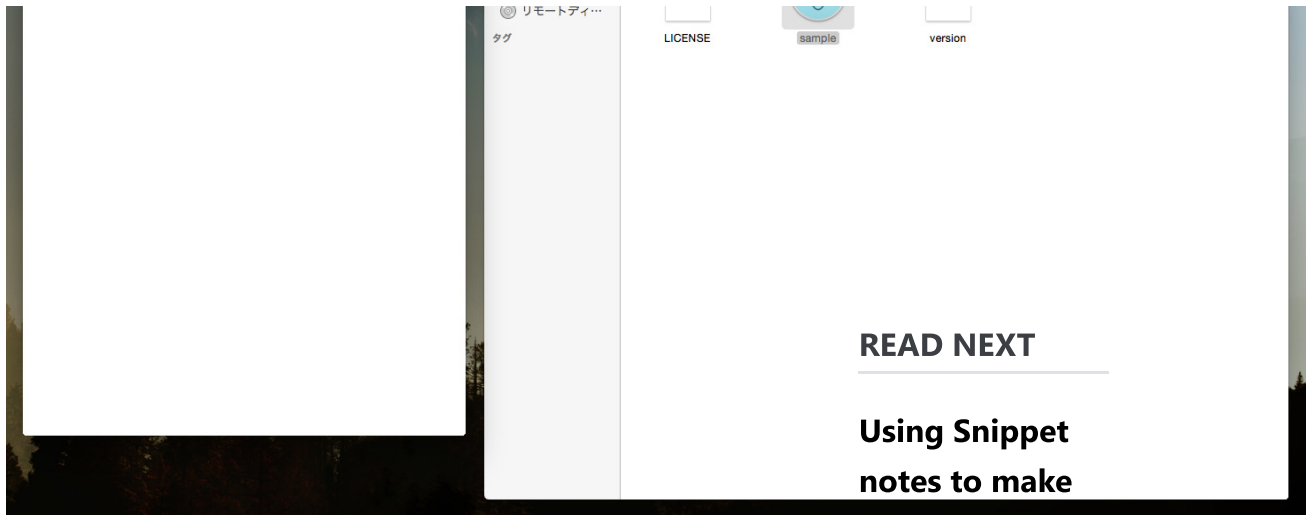
Overview

Swift Weekly
Trending
Repositories

Electron is a tool that you can create desktop applications with JavaScript. Because it is an MIT license, it can be used for free and commercial use is also possible. Atom and Slack applications were also made with Electron. Below, simple features.

- You can make applications with WEB technology called Node.js + HTML + CSS.
- Since it has a built-in Chromium browser (open source version of Chrome), it does not differ from usual writing taste.
- With this one you can make applications for Windows, Mac and Linux

Here we will introduce the installation of Electron to the distribution of applications. Below is an application that only issues "Hello World", but it can be done in 30 minutes until the first time from zero.



READ NEXT

Using Snippet
notes to make
programming...



Junpei Shimo...
j unp1234

Installation

First, install what you need.

Installation of node

For Windows, download and install msi from [here](#). For Mac, install with the following command.

```
$ brew install node
```

Swift Weekly
Trending
Repositories



Junpei Shimo...
j unp1234



1

Installation of Electron

After installing node, the next is Electron installation.

```
$ npm -g install electron-prebuilt
```

Create an application with Electron

Boostlog is an online community for developers
who want to share ideas and grow each other.



[Sign in with GitHub.](#)

"index.js" by default, but since Electron Quick Start reads "main.js", it is set to "main.js".

```
$ mkdir sample
$ cd sample
$ npm init -y
```

Based on the input contents "package.json" like the following will be completed.

```
{
  "name": "sample",
  "version": "0.0.0",
  "description": "Sample",
  "main": "main.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git@github.com:Sample/sample.git"
  },
  "author": "Nyanchu",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/Sample/sample/issues"
  }
}
```

Preparation is complete.

READ NEXT

Using Snippet notes to make programming...



Junpei Shimo...
junp1234

Swift Weekly Trending Repositories



Junpei Shimo...
junp1234



1

Output Hello World

Let's output "Hello World". First, write a description that only displays HTML in main.js .

```
'use strict';

// Mod of Electron
```

Boostlog is an online community for developers
who want to share ideas and grow each other.



Sign in with GitHub.

```
// Global declaration so that main window is not GC
let mainWindow;
```

```
// Close all windows closed
app.on('window-all-closed', function () {
  if (process.platform !== 'darwin') {
    app.quit();
  }
});
```

```
// Execute after completion of initialization of Electron
app.on('ready', function () {
  // Main screen display. You can specify the width and height of the window
  mainWindow = new BrowserWindow({width: 800, height: 600});
  mainWindow.loadURL('file://' + __dirname + '/index.html');

  // Close the application when the window is closed
  mainWindow.on('closed', function () {
    mainWindow = null;
  });
});
```

Next, I will write "Hello World" in index.html.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sample</title>
</head>
<body>
  <p>Hello World</p>
</body>
</html>
```

Now let's quick start the application. Type the following command.

```
# If you are in the sample directory, go back to the upper level
$ cd ../
# quick start
$ electron sample/
```

READ NEXT

Using Snippet

notes to make


programming...

 Junpei Shimo...
junp1234

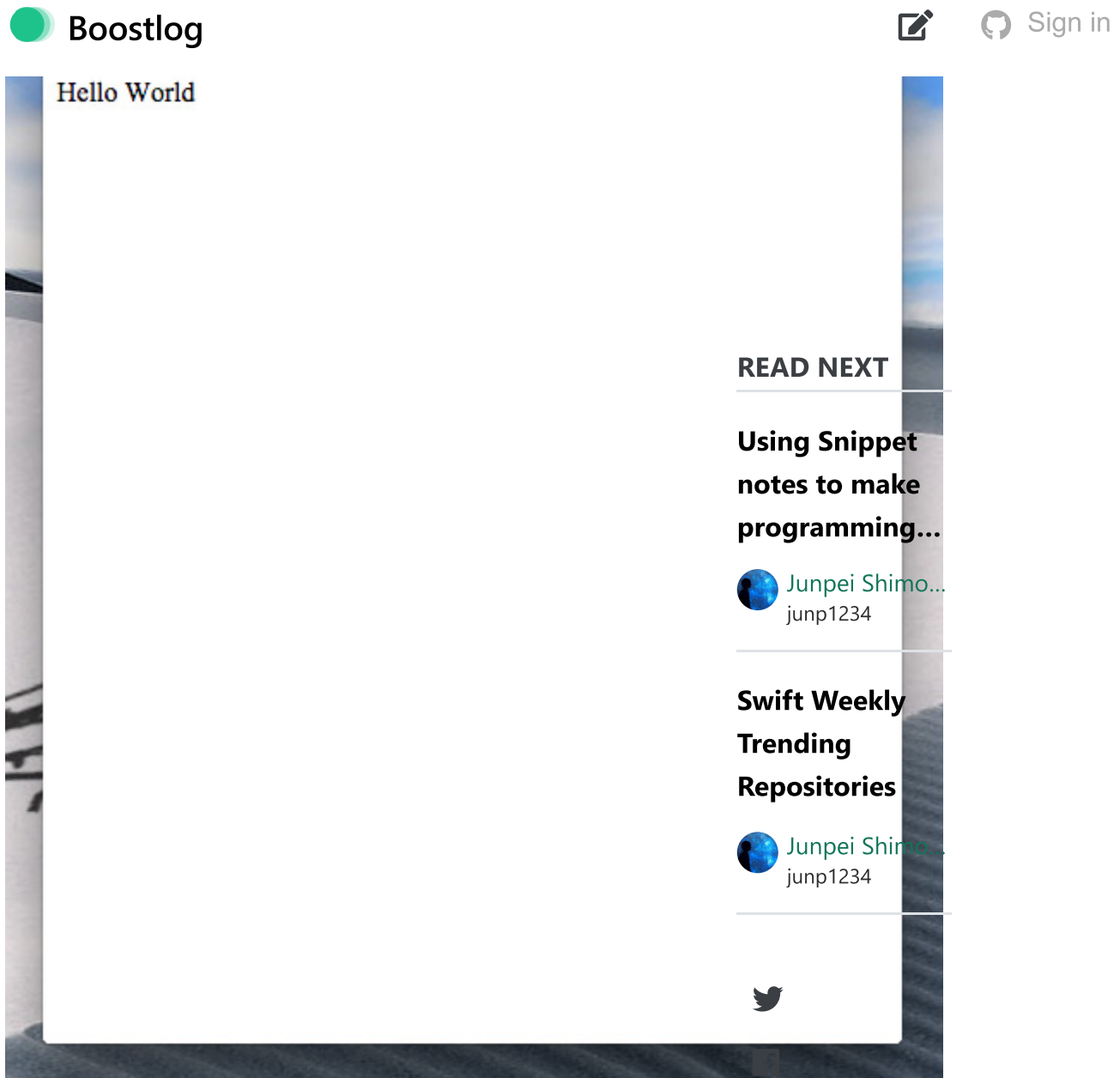
Swift Weekly

Trending

Repositories

 Junpei Shimo...
junp1234





I think the window has come up and "Hello World" is displayed. It's very easy, is not it? If you get an error, it is js error or something. I think that you can understand by reading a message. Close it by closing application window or enter Control + C.

Archive the application

Well, let's archive the application we created this time. To archive an application created with Electron, use a tool called `asar`. First let's install npm.

Boostlog is an online community for developers who want to share ideas and grow each other.

 [Sign in with GitHub.](#)

```
$ # asar pack {application directory} {output file name} .asar
$ asar pack ./sample ./sample.asar
```

Let's execute the archived file.

```
$ electron sample.asar
```

As before, I think that the window got up and I could make the "Hello World".

Distribute the application

Files archived with asar can only be started by users installing Electron. Therefore, in order to make it executable for everyone, it is necessary to pack it along with the Electron's execution environment. In this time I will use a tool called electron-packager that automatically does my neighborhood. Install first.

```
$ npm i electron-packager -g
```

And packaging it. For packaging for Mac and Windows, for example below.

```
$ # electron-packager <sourcedir> <appname> --platform=<platform> --arch=<arch> --ver
$ electron-packager ./sample sample --platform=darwin,win32 --arch=x64 --electronVers
```

I will also explain the arguments with ease.

- platform ... Select all, linux, win32, darwin. Separate commas if you put more than one.
- arch ... Select all, ia32, x64.
- electronVersion ... Specify the version of Electron. After executing the above command for a while, two directories "sample - darwin - x64" and "sample - win32 - x64" are made. Let's take a look at the contents of the "sample-darwin-x64" directory.

Conclusion

Anyway it's easy. I'm very happy to be able to make desktop applications just by knowledge of WEB. Next time I will try to arrange the appearance of the application a little more.

AUTHOR



Florida Collins
florida18

Using Snippet notes to make programming...



Junpei Shimo...
junp1234

Swift Weekly Trending

Repositories



1

[Sign in with Github](#)

Junpei Shimo...
junp1234



Boostlog is an online community for developers
who want to share ideas and grow each other.



 [Sign up with GitHub.](#)

READ NEXT

Jan 24 2018

Boostlog is an online community for developers
who want to share ideas and grow each other.



[Sign in with GitHub.](#)

Boostlog

junp1234

Sign in

0

Jan 24 2018

Swift Weekly Trending Repositories

swift github opensource +1

Junpei Shimotsu

junp1234

READ NEXT

Using Snippet notes to make programming...

0

Junpei Shimo...

junp1234

Swift Weekly Trending Repositories

Junpei Shimo...

junp1234



1