

Electron アプリを つくるう

2015-07-02

Electron?

Electron?

- Chromium + Node.js
 - でデスクトップアプリを作る
- Atom で使われている

似たプロダクト

- NW.js (a.k.a. node-webkit)
 - Chromium + Node.js の構成は同じ
- Electronの方が後発

Electronの強み

- Webフロントエンドの仕組みがそのまま使える
 - JavaScript
 - HTML / CSS
 - Node.js / npm

Electronの弱み

- Web技術に制約される
 - HTML / CSS / DOM の限界
 - JavaScriptの言語的限界
- サイズがでかい (1アプリ100MB~くらい)

Electronアプリを作る

Markdownエディタを作る

- Webっぽい × アプリっぽい
- Electron の利点が生きる

リポジトリ

- github.com/seanchas116/electron-md-editor

準備

準備

- Node を入れる
 - `brew install node`
- Electron を入れる
 - `npm install -g electron-prebuilt`

プロジェクト準備

- `mkdir electron-md-editor`
- `cd electron-md-editor`
- `npm init`
 - `main => main.js`

構成

renderer process (ウィンドウ)

index.html

scirpt.js

main.js

main process

構成

```
tree .
```

```
.
├── README.md
├── index.html
├── main.js
├── package.json
└── script.js
```

main.js

- 最初に読み込まれる
- ウィンドウを開く

```
var app = require('app');
var BrowserWindow = require('browser-window');

app.on('ready', function() {
  var mainWindow = new BrowserWindow({width: 800, height: 600});
  mainWindow.loadUrl('file://' + __dirname + '/index.html');
});
```

index.html

- ウィンドウに表示する
- script.js を読み込む

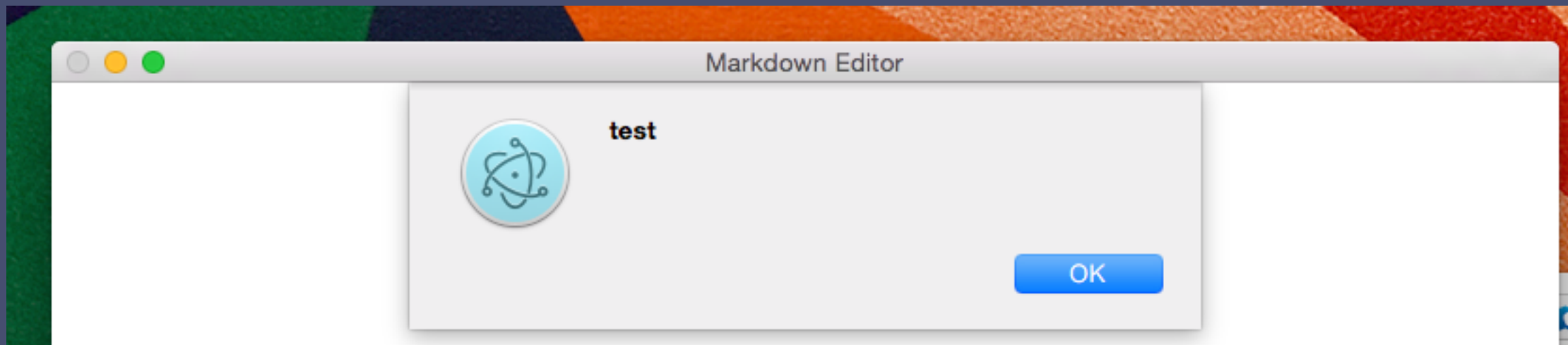
```
<!DOCTYPE html>
<html>
  <head>
    <title>Markdown Editor</title>
    <meta charset="UTF-8">
  </head>
  <body>
  </body>
  <script src="./script.js"></script>
</html>
```


scripts.js

```
alert("test");
```

実行

```
$ electron .
```



ちゃんとしたアプリを
作りたい

vue.jsでmarkdownエディタ

```
$ npm install --save vue marked
```

vue.js公式に markdown エディタの例がある

script.js

スタイルはお好みで

...

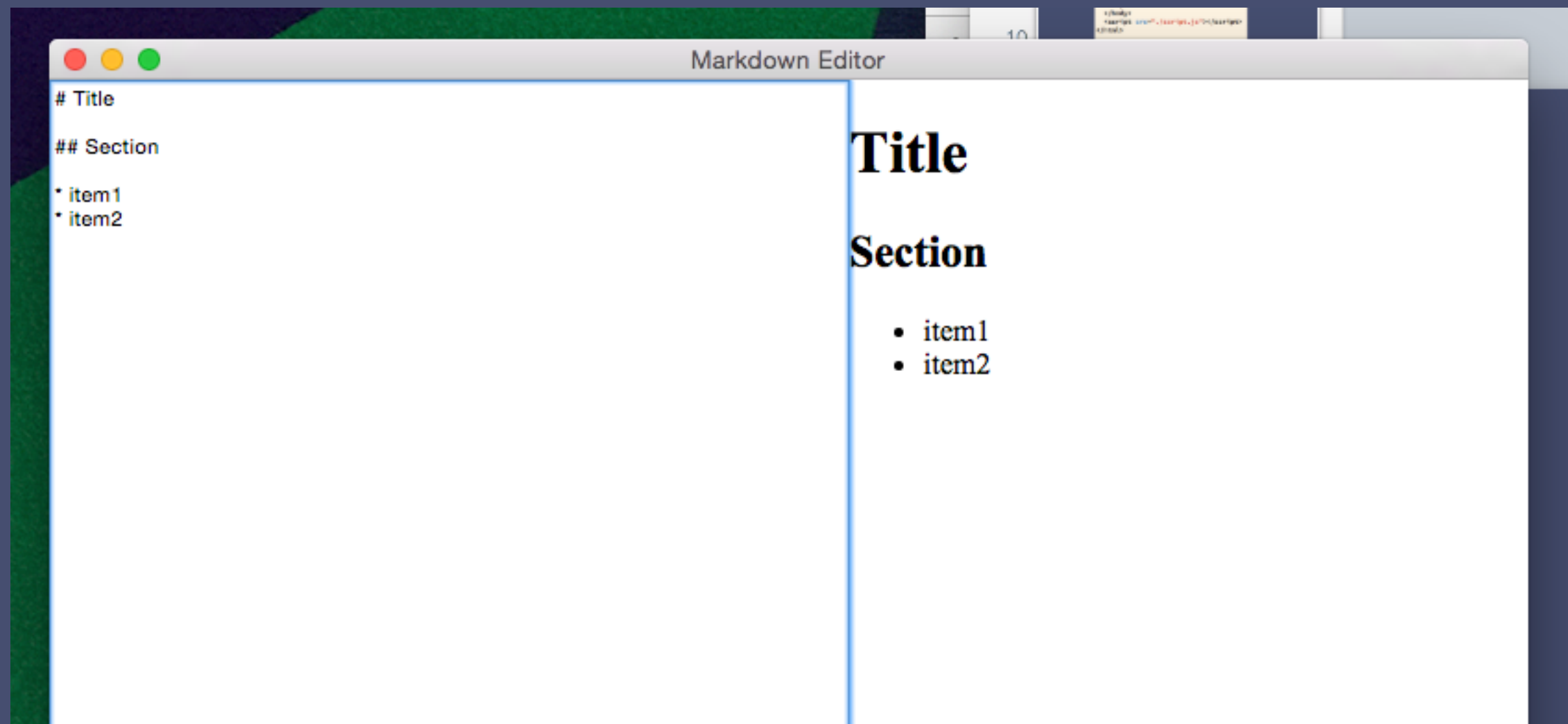
```
<body>
  <div id="editor">
    <textarea v-model="input"></textarea>
    <div v-html="input | marked"></div>
  </div>
</body>
```

script.js

```
var Vue = require("vue");
var marked = require("marked");

var editor = new Vue({
  el: '#editor',
  data: {
    input: '# hello',
    filename: null
  },
  filters: {
    marked: marked
  }
});
```

実行



デスクトップアプリらしい
機能を

ファイルを開く・保存

- Webアプリにはできない
- 使うもの
 - node の fs モジュール
 - electron の dialog モジュール

script.js: Fileメニューを追加

```
var remote = require('remote');
var Menu = remote.require('menu');
var template = [
  {
    label: 'Markdown Editor',
  },
  {
    label: 'File',
    submenu: [
      {
        label: 'Open',
        click: openFileDialog
      },
      {
        label: 'Save',
        click: saveFileDialog
      },
    ],
  }
];
var menu = Menu.buildFromTemplate(template);
Menu.setApplicationMenu(menu);
```

require("remote") ??

- "menu" は main process 用のモジュール
- renderer process から読み込みたい!
 - remote.require("menu")
 - 間接的に読み込む

script.js: openFileDialog

```
var fs = require("fs");
var dialog = remote.require("dialog");

function openFileDialog() {
  dialog.showOpenDialog(function (filenames) {
    console.log(filenames);
    var filename = filenames[0];
    if (filename) {
      fs.readFile(filename, "utf8", function
(err, data) {
        if (err) { throw err; }
        editor.$data.filename = filename;
        editor.$data.input = data;
      });
    }
  });
}
```

script.js: saveFileDialog

```
function saveFileDialog() {  
    function save(filename) {  
        fs.writeFile(filename, editor.$data.input, function  
(err) {  
            if (err) { throw err; }  
            alert("Saved!");  
        });  
    }  
  
    if (editor.$data.filename) {  
        save(editor.$data.filename);  
    } else {  
        dialog.showSaveDialog(function (filename) {  
            if (filename) {  
                save(filename);  
                editor.$data.filename = filename;  
            }  
        });  
    }  
}
```

アプリを
パッケージングしたい

electron-packager

```
$ npm install -g electron-packager  
  
$ electron-packager ./ MarkdownEditor  
  --platform darwin  
  --arch x64  
  --version 0.28.3
```



まとめ

まとめ

- Electron で Markdown エディタ作成
- パッケージング
- Web技術 + Node.jsでアプリが作れる