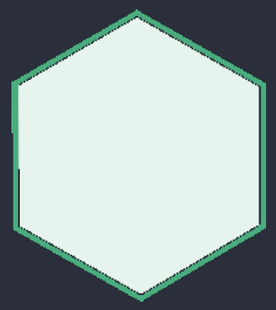
ELECTRON

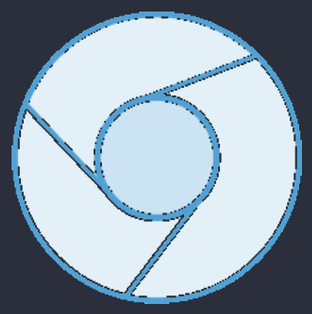
@codebytere

WHAT IS IT?

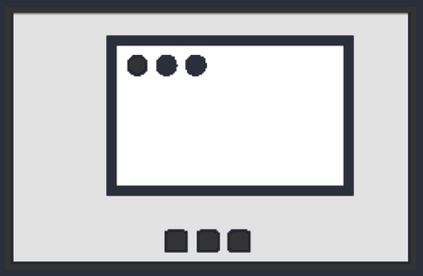
Node.js for filesystems and networks

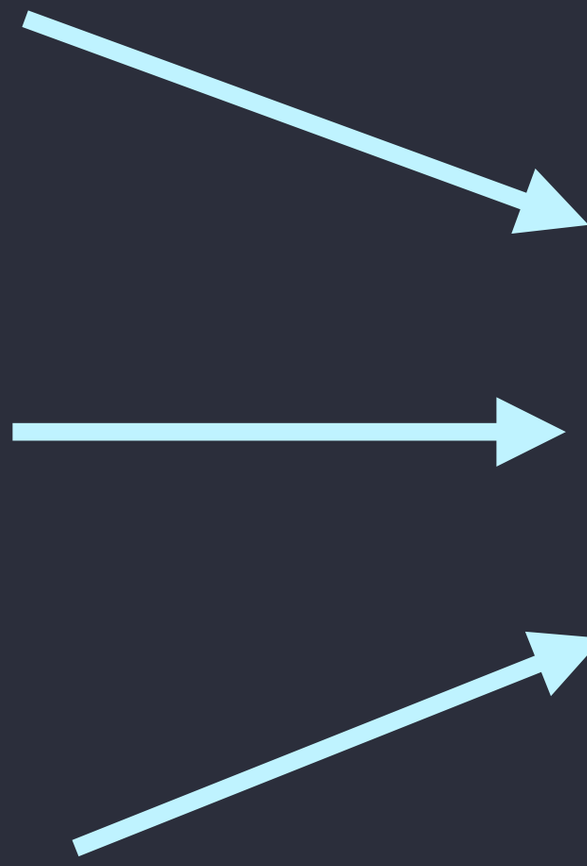


Chromium for making web pages

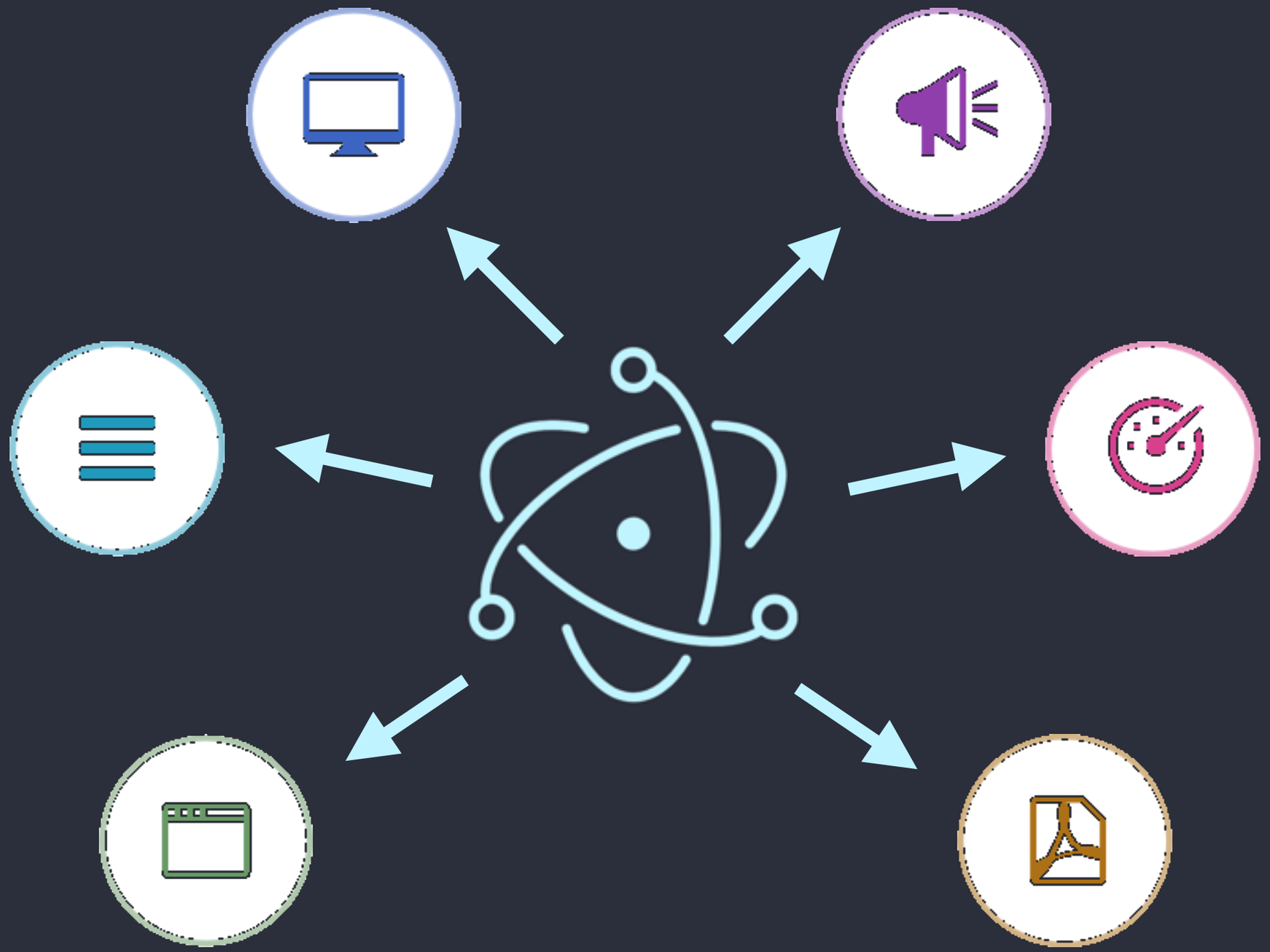


Native APIs for three systems





WHAT CAN I DO?



MAIN PROCESS

Node.js APIs

Main Process Modules

main.js

Create Renderer procs

Call native elements

Start/quit

RENDERER PROCESS

Node.js APIs

DOM APIs

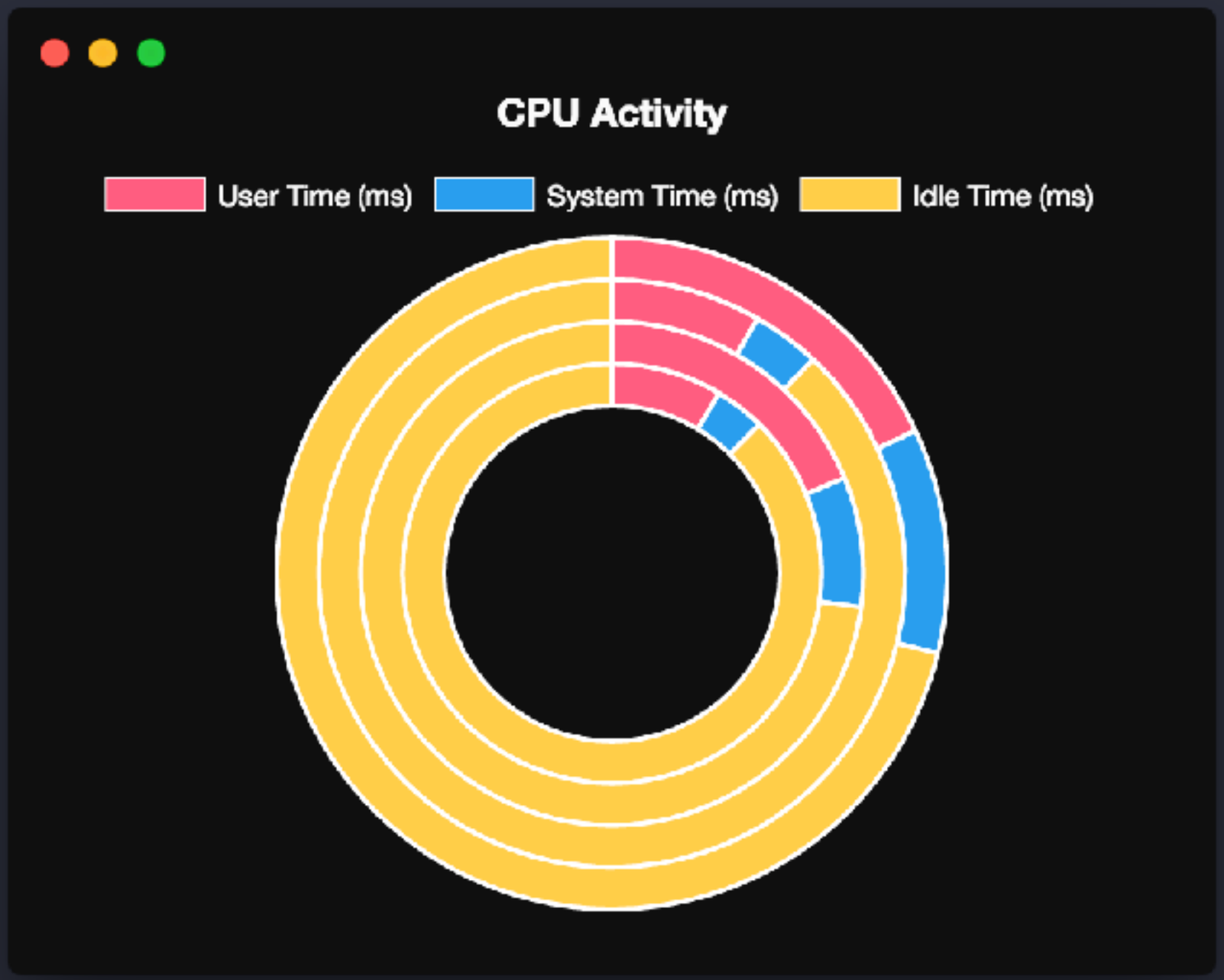
Renderer Modules

index.html

JS page interactions

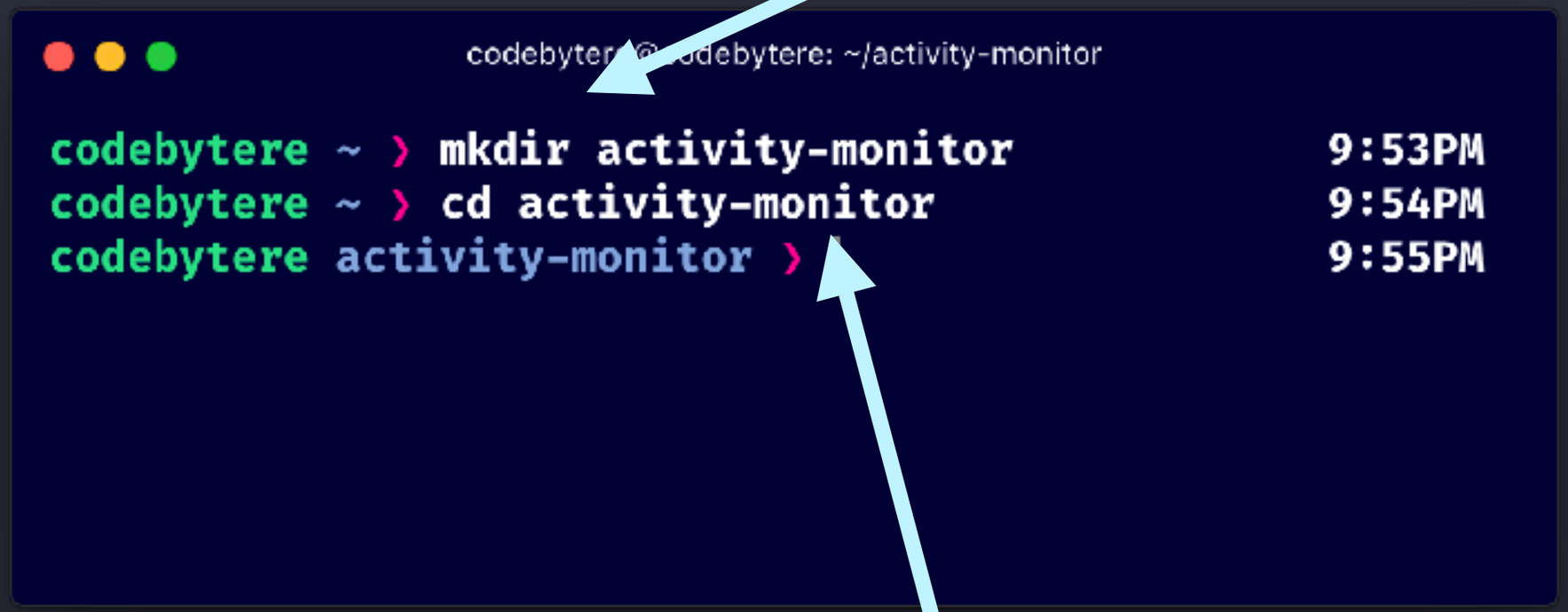
Design pages with HTML & CSS

ACTIVITY MONITOR



CREATING THE DIRECTORY

`MKDIR` CREATES A DIRECTORY IN THE CURRENT DIRECTORY



`CD` MOVES INTO THE SPECIFIED DIRECTORY

CREATING PACKAGE.JSON

CREATE PACKAGE.JSON WITH THE NODE PACKAGE MANAGER

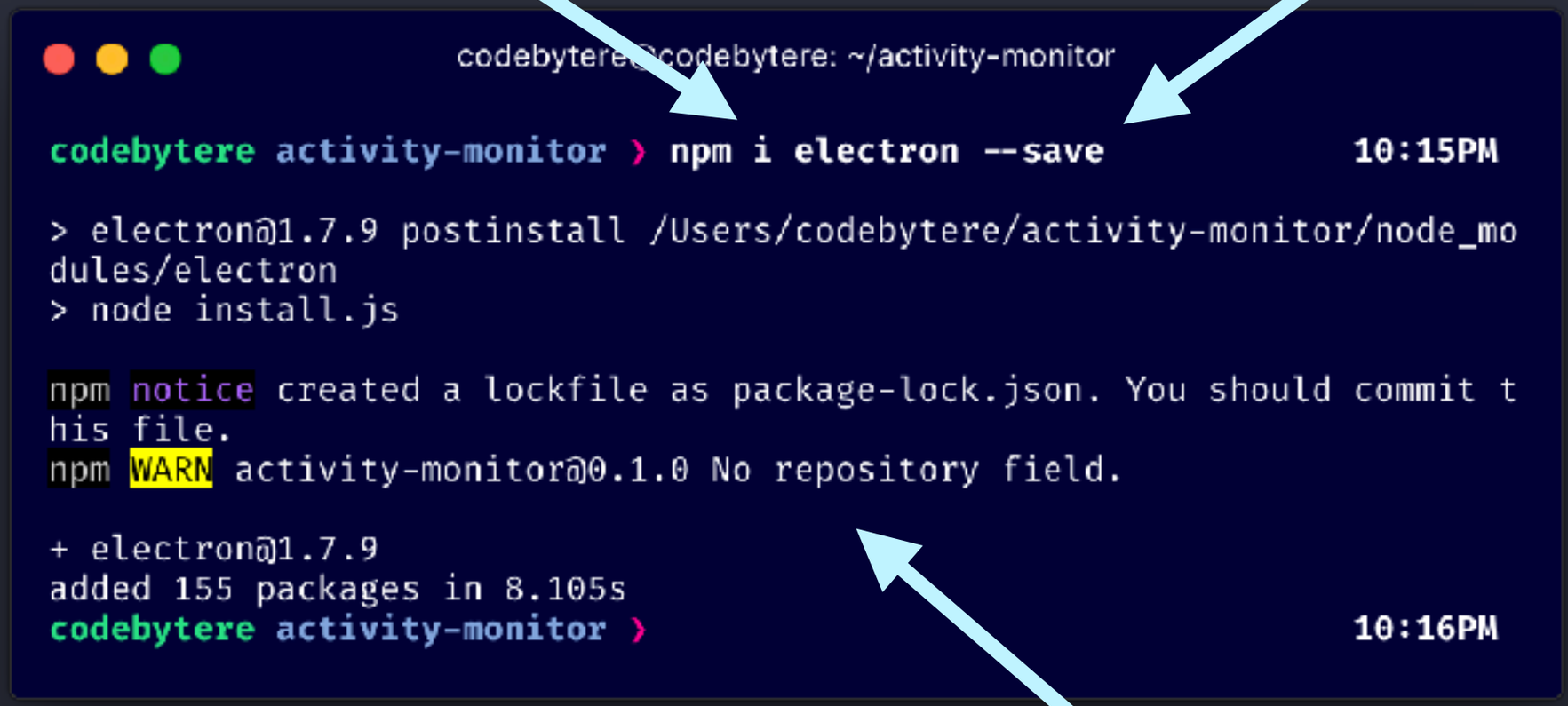
```
npm init

codebytere activity-monitor ❯ npm init                              10:08PM
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defau
lts.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (activity-monitor)
version: (1.0.0) 0.1.0
description: see a cpu donut!
entry point: (index.js) app.js
test command:
git repository:
keywords:
author: Shelley Vohr
license: (ISC) MIT
```

FILL IN INFORMATION ABOUT YOUR APPLICATION - ENTER LEAVES THE FIELD DEFAULT

INSTALLING ELECTRON

`I` IS SHORTHAND FOR `INSTALL`

`--SAVE` SAVES ELECTRON INTO PACKAGE.JSON



NPM RUNS THE INSTALLATION PROCESS

CREATING FILES

CREATE JAVASCRIPT FILES WITH TOUCH

```
codebytere@codebytere: ~/activity-monitor

codebytere activity-monitor ❯ touch app.js window.js                10:25PM
codebytere activity-monitor ❯ touch style.css index.html            10:25PM
codebytere activity-monitor ❯ ls                                    10:25PM
app.js              package-lock.json window.js
index.html          package.json
node_modules        style.css
codebytere activity-monitor ❯                                       10:25PM
```

CREATE STYLESHEET AND INDEX HTML FILE

SEE FILES IN YOUR CURRENT DIRECTORY WITH `LS`

APP.JS CODE

app.js — ~/Develcper/simple-samples/activity-monitor

app.js | window.js | styles.css | index.html

```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10
11   // load index.html into the window
12
13   // show the window once it's ready
14 })
15
```

WAIT UNTIL THE APP IS READY BY WAITING FOR `READY` EVENT

WHAT ELSE DO WE NEED TO DO?

Python app.js ⓘ 0 ⚠ 0 ⓔ 0 10:1 LF UTF-8 JavaScript master 4 files 10 updates

APP.JS CODE

app.js — ~/Develcper/simple-samples/activity-monitor

app.js | window.js | styles.css | index.html

```
1   const {app, BrowserWindow} = require('electron')
2   const path = require('path')
3   const url = require('url')
4
5   let window = null
6
7   // Wait until the app is ready
8   app.once('ready', () => {
9     // Create a new window
10    window = new BrowserWindow({
11      width: 500, // Set the initial width to 500px
12      height: 400, // Set the initial height to 400px
13      titleBarStyle: 'hidden-inset', // set the title bar style
14      backgroundColor: "#111", // set the background color to black
15      show: false // Don't show the window until it's ready
16    })
17
18    // load url into the window
19
20    // show the window once it's ready
21  })
22
```

CREATE A NEW BROWSERWINDOW

Python app.js ⓘ 0 ⚠ 0 ⓘ 0 20:3 LF UTF-8 JavaScript master 4 files 10 updates

APP.JS CODE

app.js — ~/Developer/simple-samples/activity-monitor

app.js | window.js | styles.css | index.html

```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10   window = new BrowserWindow({
11     width: 500, // Set the initial width to 500px
12     height: 400, // Set the initial height to 400px
13     titleBarStyle: 'hidden-inset', // set the title bar style
14     backgroundColor: "#111", // set the background color to black
15     show: false // Don't show the window until it's ready
16   })
17
18   // load url into the window
19   window.loadURL(url.format({
20     pathname: path.join(__dirname, 'index.html'),
21     protocol: 'file:',
22     slashes: true
23   }))
24
25   // show the window once it's ready
26 })
27
```

LOAD INDEX.HTML INTO THE WINDOW WE JUST CREATED

Python app.js ⓧ 0 ⚠ 0 ⓘ 0 19:31 LF UTF-8 JavaScript master 4 files 10 updates

APP.JS CODE

app.js — ~/Developer/simple-samples/activity-monitor

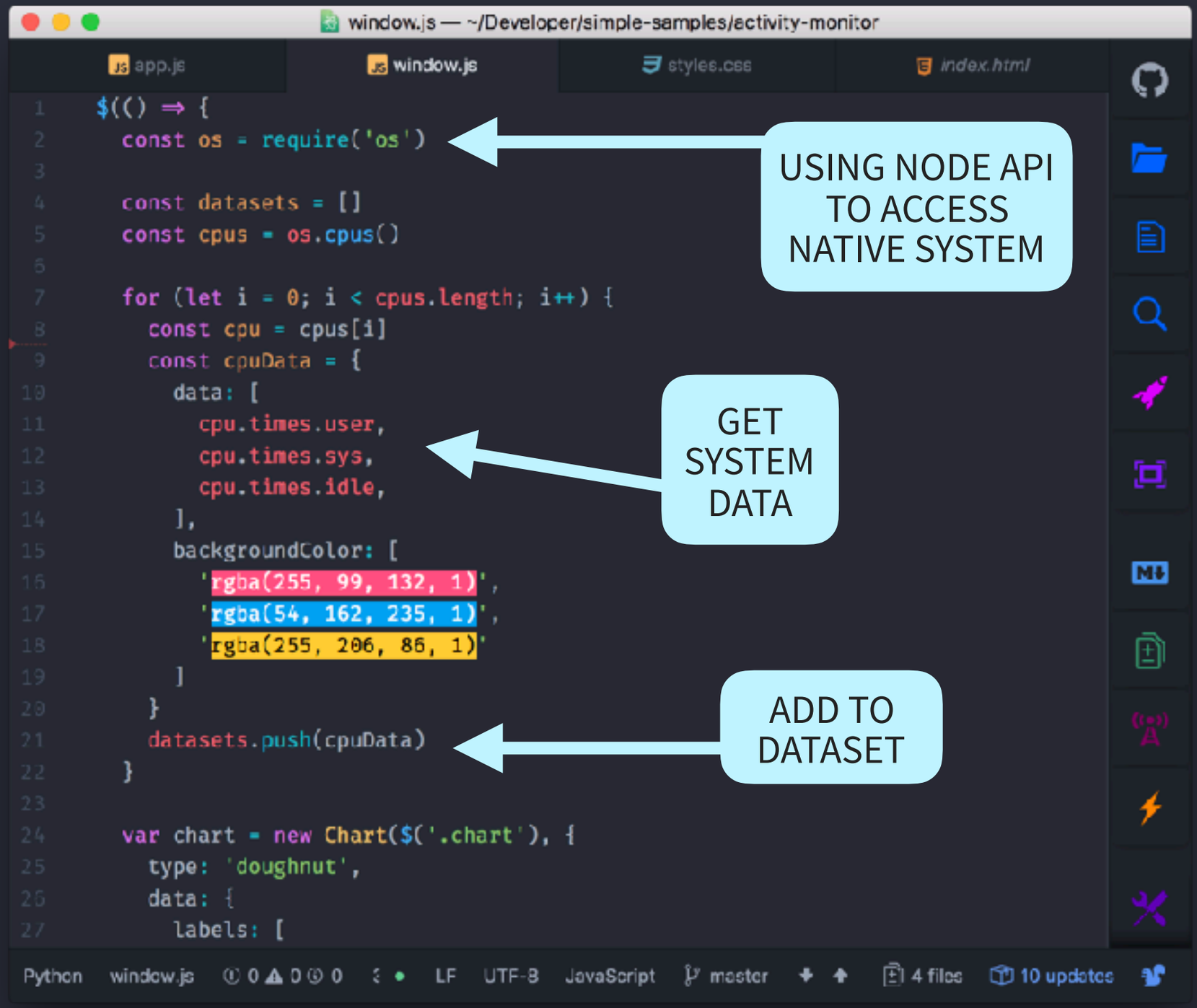
app.js | window.js | styles.css | index.html

```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10   window = new BrowserWindow({
11     width: 500, // Set the initial width to 500px
12     height: 400, // Set the initial height to 400px
13     titleBarStyle: 'hidden-inset', // set the title bar style
14     backgroundColor: "#111", // set the background color to black
15     show: false // Don't show the window until it's ready
16   })
17
18   // load url into the window
19   window.loadURL(url.format({
20     pathname: path.join(__dirname, 'index.html'),
21     protocol: 'file:',
22     slashes: true
23   }))
24
25   // show the window once it's ready
26   window.once('ready-to-show', () => window.show())
27 })
```

SHOW THE WINDOW ONCE IT'S READY

Python app.js ⓘ 0 ⚠ 0 ⓘ 0 19:31 LF UTF-8 JavaScript master 4 files 10 updates

WINDOW.JS CODE



WINDOW.JS CODE

window.js — ~/Developer/simple-samples/activity-monitor

app.js | window.js | styles.css | index.html

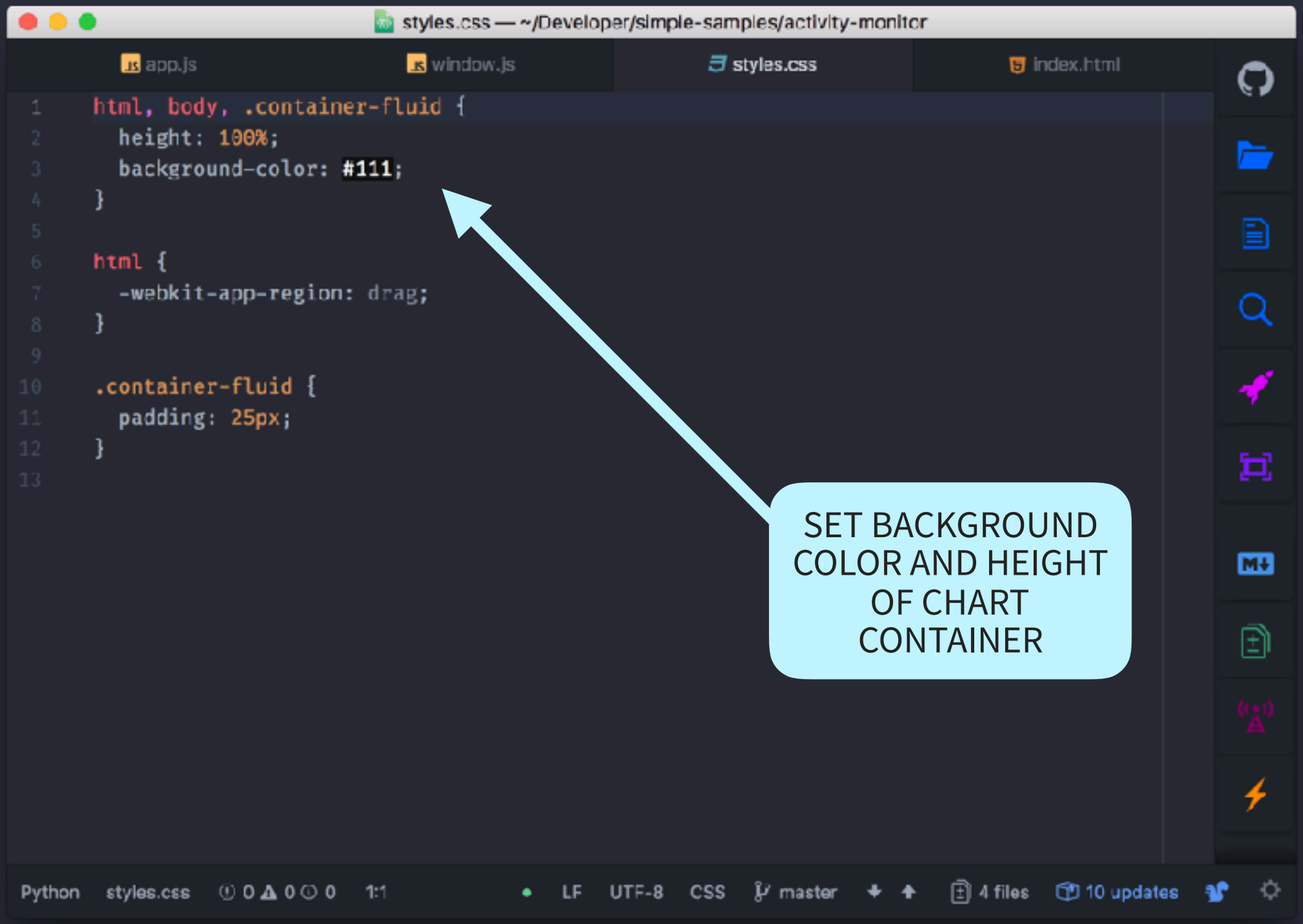
```
22    }
23
24    var chart = new Chart($('.chart'), {
25      type: 'doughnut',
26      data: {
27        labels: [
28          'User Time (ms)',
29          'System Time (ms)',
30          'Idle Time (ms)'
31        ],
32        datasets: datasets
33      },
34      options: {
35        maintainAspectRatio: false,
36        title: {
37          display: true,
38          text: 'CPU Activity',
39          fontColor: 'rgb(250, 250, 250)',
40          fontSize: 16
41        },
42        legend: {
43          display: true,
44          labels: {
45            fontColor: 'rgb(250, 250, 250)',
46            fontSize: 12
47          }
48        }
49      }
50    })
51  })
```

USE JS LIBRARY TO CREATE CHART

CREATE WITH DATA WE COLLECTED

Python window.js 0 0 0 LF UTF-8 JavaScript master 4 files 10 updates

STYLE.CSS CODE



INDEX.HTML CODE

index.html — ~/Developer/simple-samples/activity-monitor

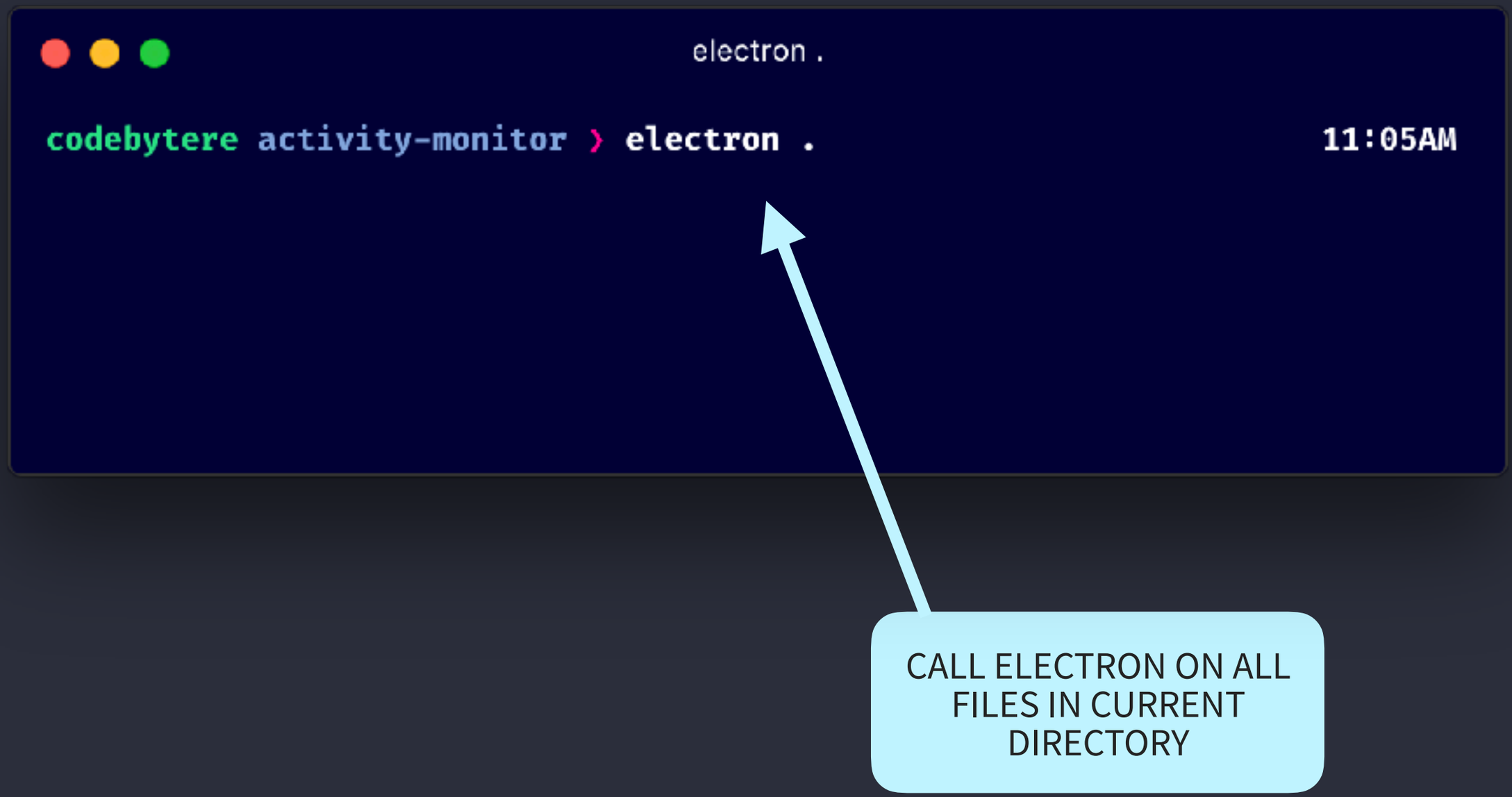
app.js | window.js | styles.css | index.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Activity Monitor</title>
  <!-- Stylesheets -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstr
  <link rel="stylesheet" href="./styles.css">
  <!-- Scripts -->
  <script> delete module.exports </script>
  <script src="https://code.jquery.com/jquery-3.2.1.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></scri
  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.4/Chart.bundle.min.js">
  <script src="./window.js"></script>
</head>

<body>
  <div class="container-fluid">
    <canvas class="chart"></canvas>
  </div>
</body>

</html>
```

IMPORT LIBRARIES AND OUR WINDOW CODE

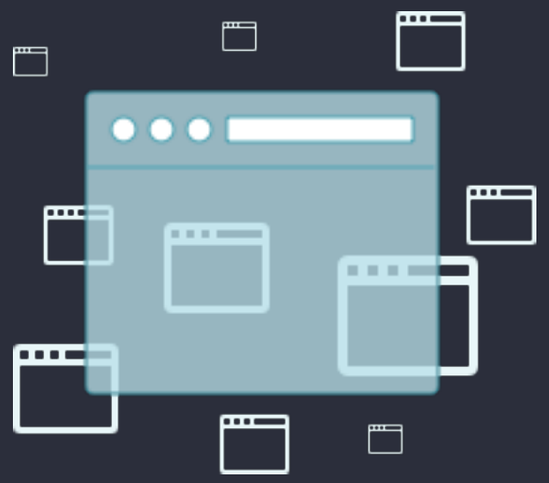
Python index.html ⓘ 0 ⚠ 0 ⓔ 0 18:7 LF UTF-8 HTML master 4 files 10 updates

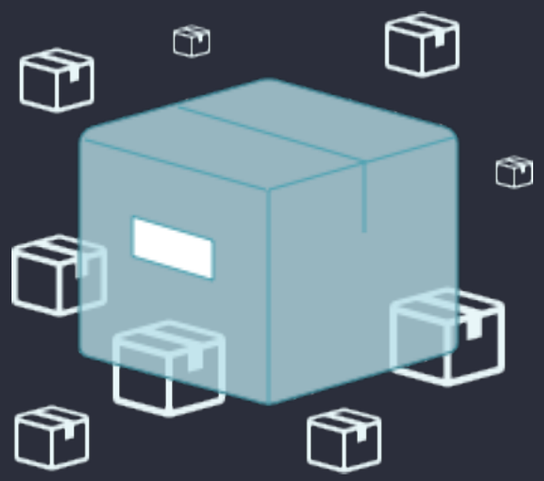
RUNNING YOUR APP



COMPLETE APP

https://github.com/electron/simple-samples/





THANK YOU!