

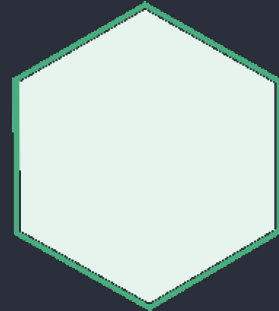


ELECTRON

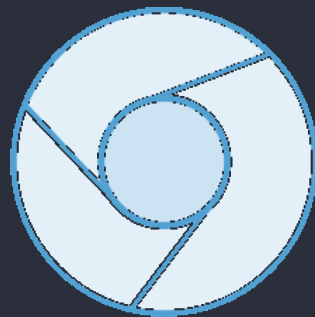
@codebytere

WHAT IS IT?

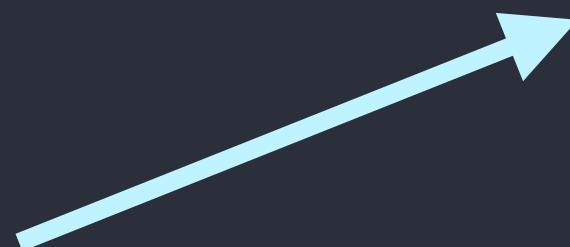
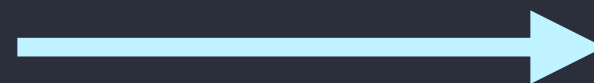
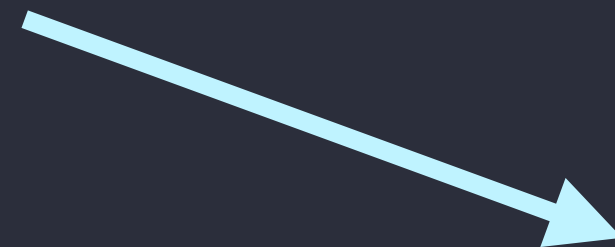
Node.js for
filesystems
and networks



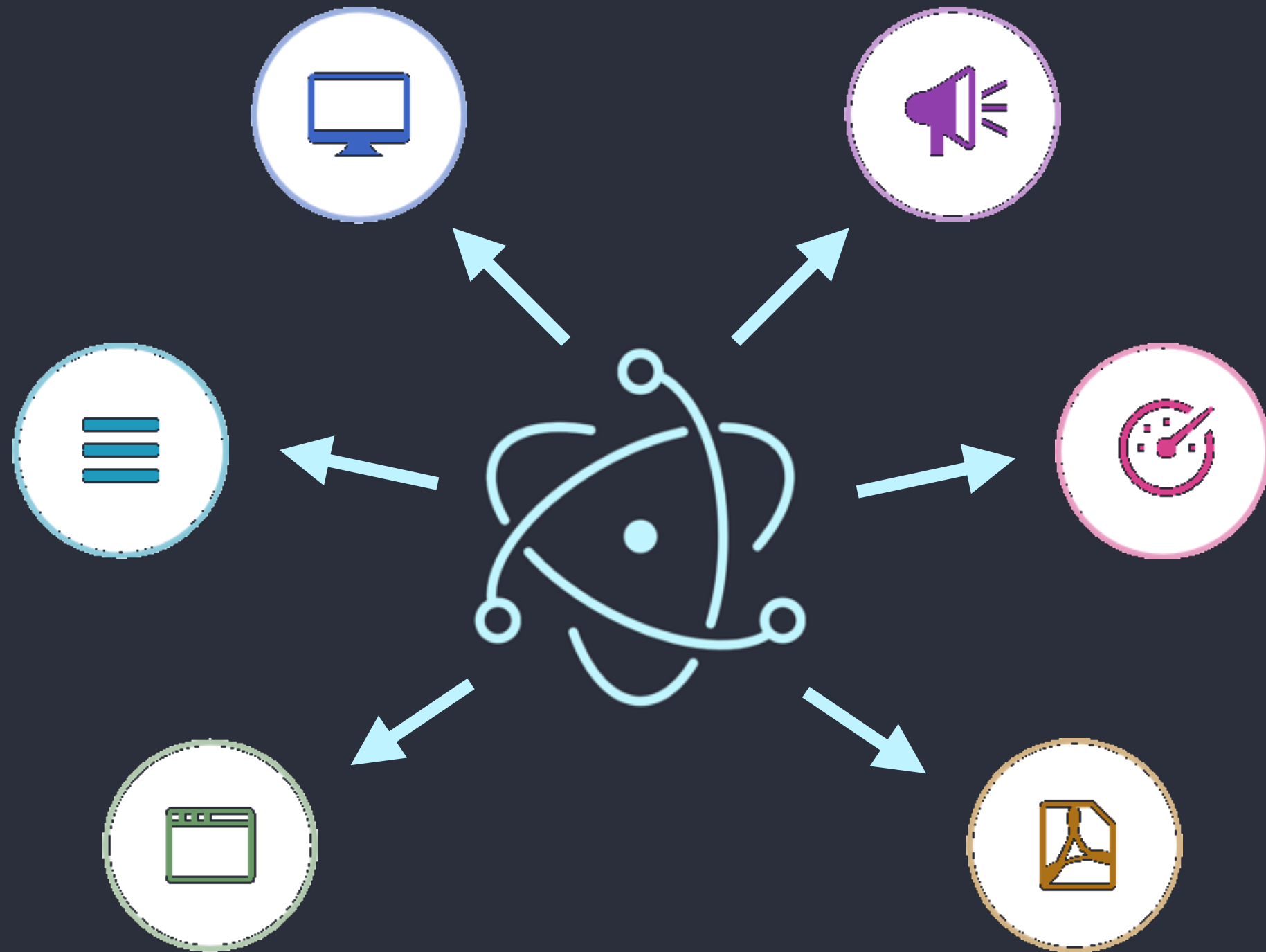
Chromium
for making
web pages



Native APIs
for three
systems



WHAT CAN I DO?



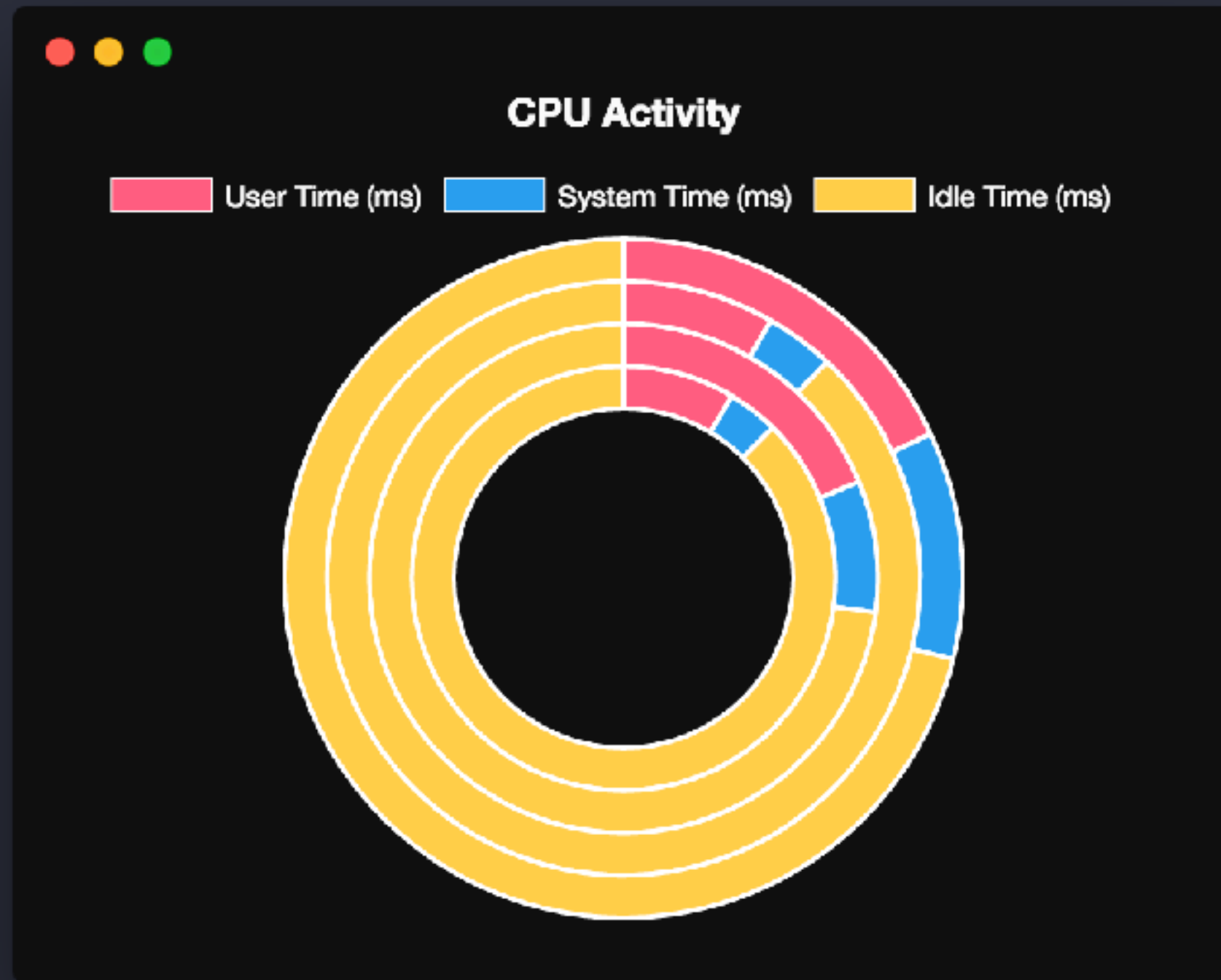
MAIN PROCESS



RENDERER PROCESS



ACTIVITY MONITOR



CREATING THE DIRECTORY

‘MKDIR’ CREATES A
DIRECTORY IN THE CURRENT
DIRECTORY

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

codebytere ~ > mkdir activity-monitor          9:53PM
codebytere ~ > cd activity-monitor              9:54PM
codebytere activity-monitor >                  9:55PM
```

‘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 defaults.  
  
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

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

codebytere activity-monitor > npm i electron --save 10:15PM

> electron@1.7.9 postinstall /Users/codebytere/activity-monitor/node_modules/electron
> node install.js

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN activity-monitor@0.1.0 No repository field.

+ electron@1.7.9
added 155 packages in 8.105s
codebytere activity-monitor > 10:16PM
```

NPM RUNS THE
INSTALLATION
PROCESS

CREATING FILES

CREATE JAVASCRIPT
FILES WITH TOUCH

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

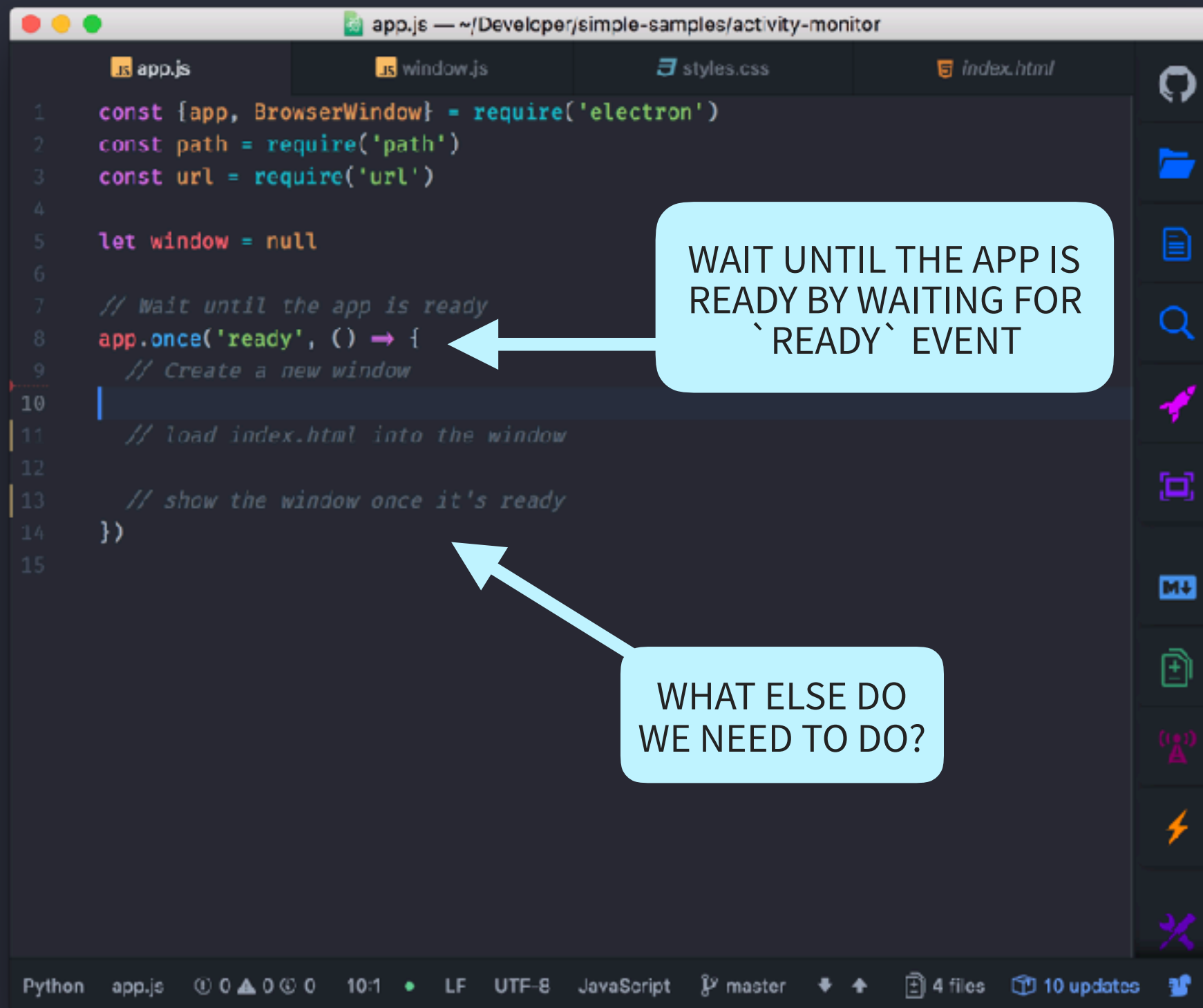
codebytere activity-monitor > touch app.js window.js      10:25PM
codebytere activity-monitor > touch style.css index.html  10:25PM
codebytere activity-monitor > ls                          10:25PM
app.js              package-lock.json window.js
index.html          package.json
node_modules        style.css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

CREATE STYLESHEET
AND INDEX HTML
FILE

SEE FILES IN YOUR
CURRENT DIRECTORY
WITH `LS`



APP.JS CODE



The image shows a code editor window with the title bar "app.js — ~/Developer/simple-samples/activity-monitor". The editor displays the contents of the `app.js` file, which is the active tab. Other tabs visible are `window.js`, `styles.css`, and `index.html`. The code in `app.js` is as follows:

```
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
```

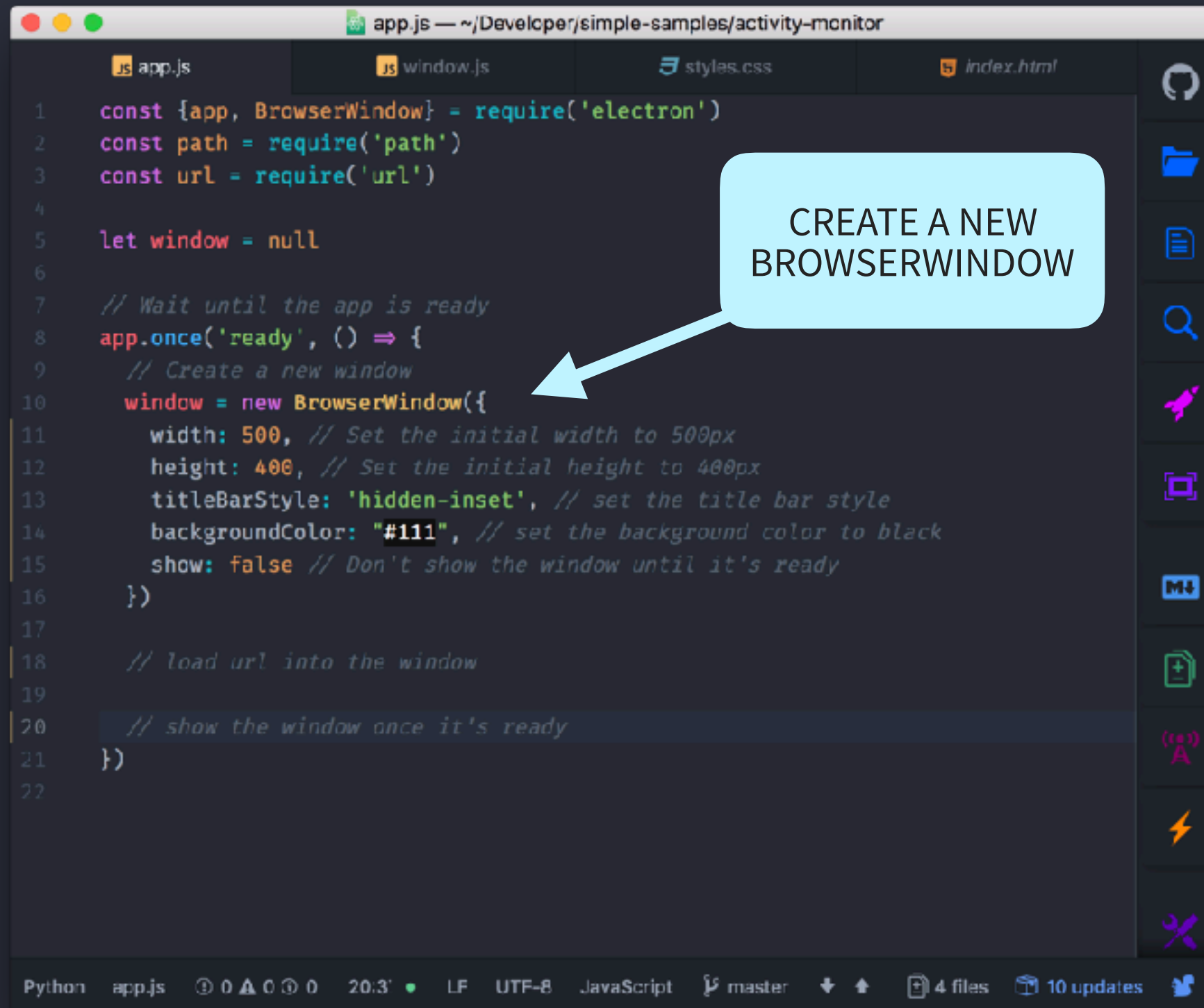
Two callout boxes with arrows pointing to the code are present:

- A light blue box with the text "WAIT UNTIL THE APP IS READY BY WAITING FOR 'READY' EVENT" has an arrow pointing to the `app.once('ready', () => {` line (line 8).
- A light blue box with the text "WHAT ELSE DO WE NEED TO DO?" has an arrow pointing to the closing brace of the `ready` event listener (line 14).

The bottom status bar of the editor shows: Python app.js 0 0 0 10:1 • LF UTF-8 JavaScript master 4 files 10 updates.



APP.JS CODE

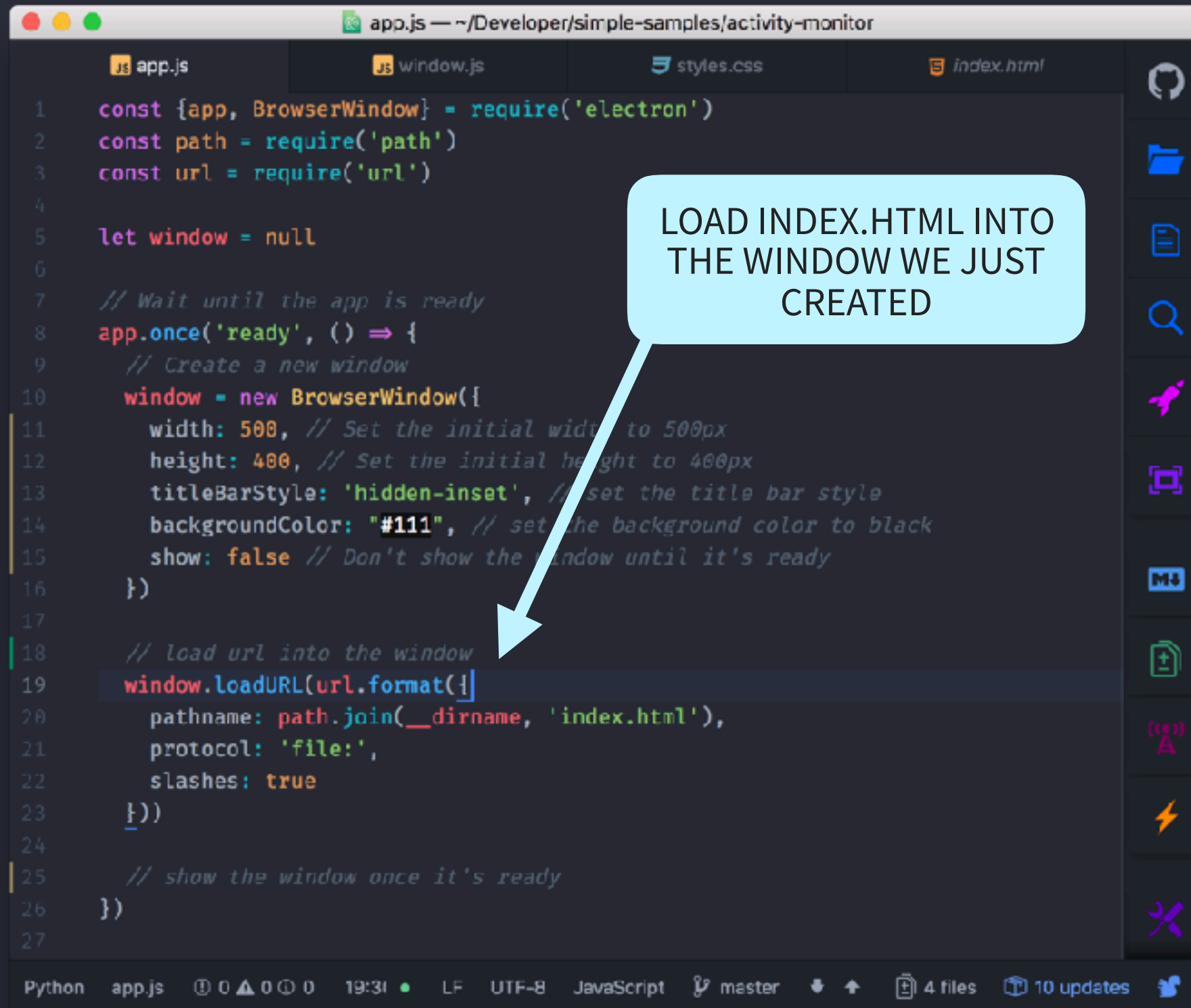


```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10   window = new BrowserWindow({
11     width: 500, // Set the initial 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



APP.JS CODE

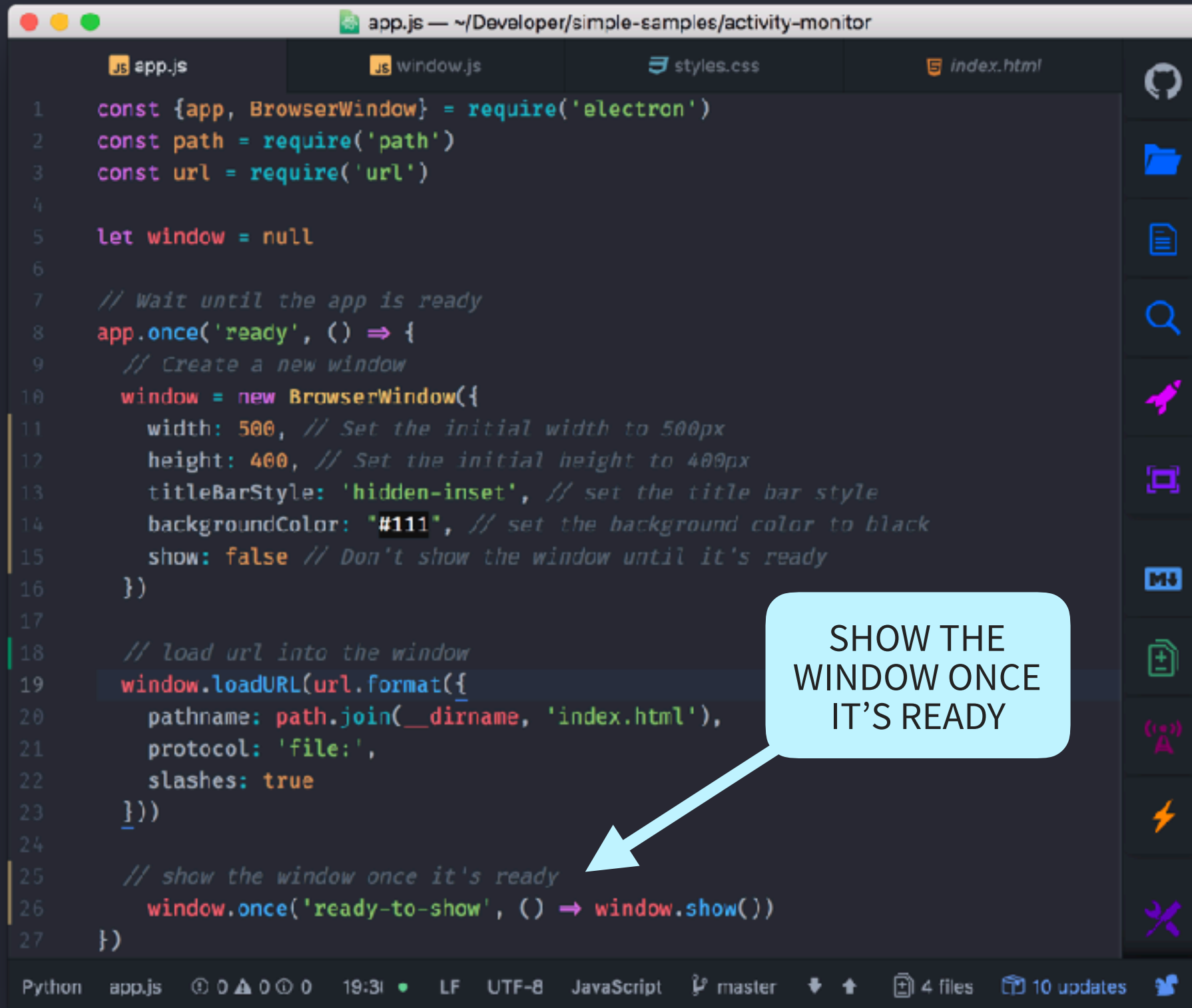


```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10   window = new BrowserWindow({
11     width: 500, // Set the initial 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



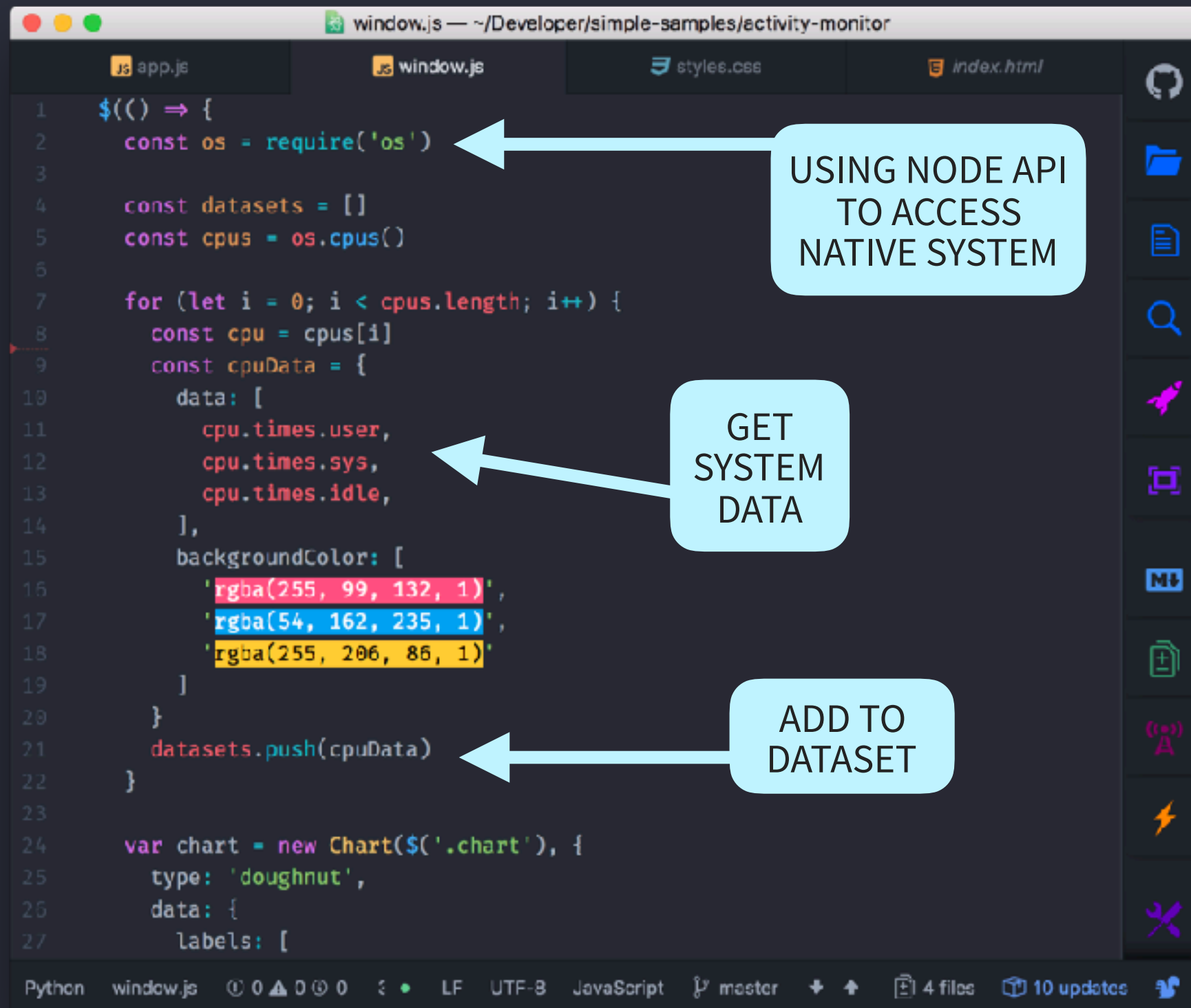
```
1  const {app, BrowserWindow} = require('electron')
2  const path = require('path')
3  const url = require('url')
4
5  let window = null
6
7  // Wait until the app is ready
8  app.once('ready', () => {
9    // Create a new window
10   window = new BrowserWindow({
11     width: 500, // Set the initial 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



The image shows a code editor window titled "window.js — ~/Developer/simple-samples/activity-monitor". The editor displays the contents of the "window.js" file, which is a JavaScript module using ES6 syntax. The code is as follows:

```
1  $((() => {
2    const os = require('os')
3
4    const datasets = []
5    const cpus = os.cpus()
6
7    for (let i = 0; i < cpus.length; i++) {
8      const cpu = cpus[i]
9      const cpuData = {
10        data: [
11          cpu.times.user,
12          cpu.times.sys,
13          cpu.times.idle,
14        ],
15        backgroundColor: [
16          'rgba(255, 99, 132, 1)',
17          'rgba(54, 162, 235, 1)',
18          'rgba(255, 206, 86, 1)'
19        ]
20      }
21      datasets.push(cpuData)
22    }
23
24    var chart = new Chart($('.chart'), {
25      type: 'doughnut',
26      data: {
27        labels: [
```

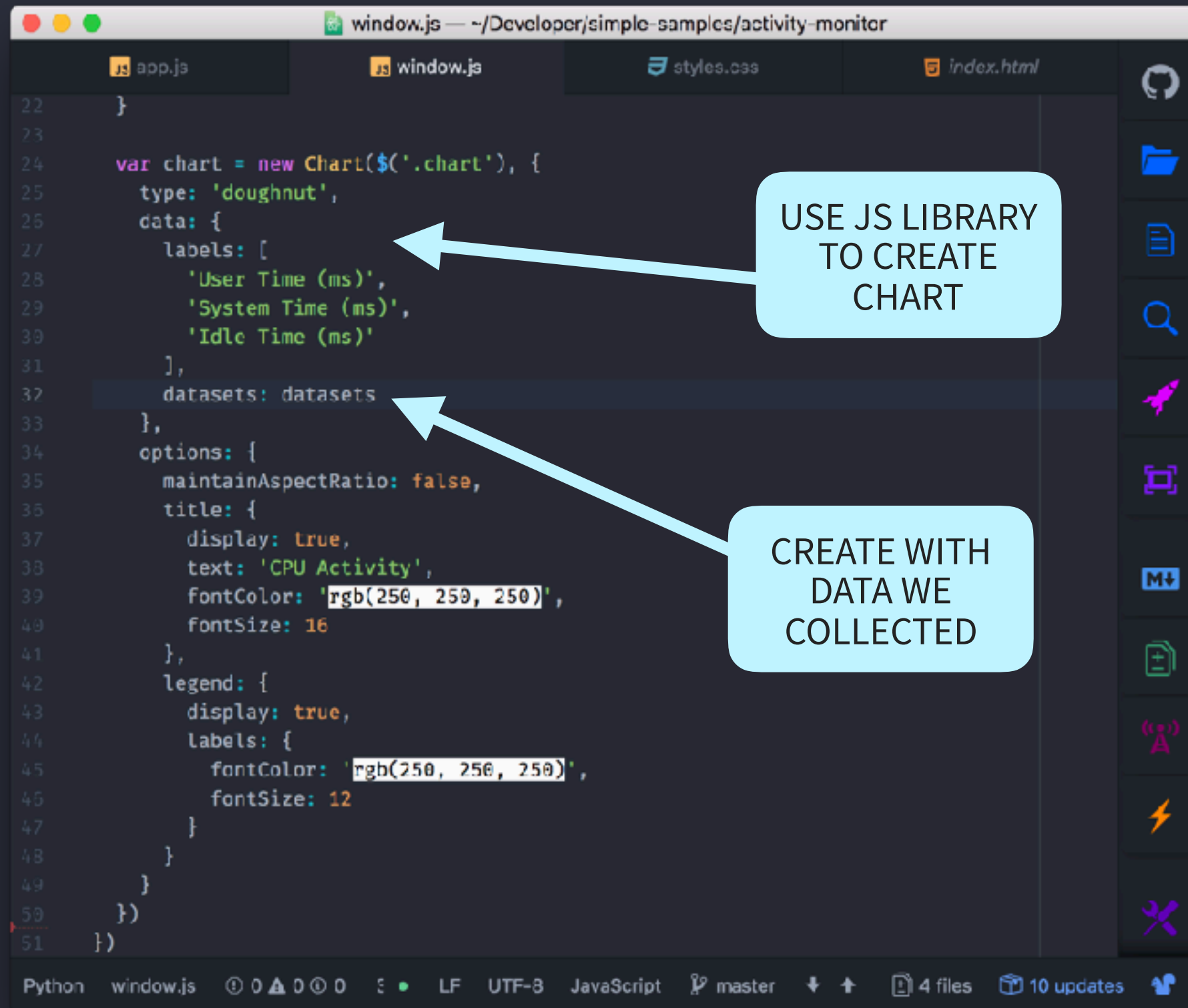
Three light blue callout boxes with arrows pointing to specific lines of code:

- USING NODE API TO ACCESS NATIVE SYSTEM** points to line 2: `const os = require('os')`
- GET SYSTEM DATA** points to line 11: `cpu.times.user`
- ADD TO DATASET** points to line 21: `datasets.push(cpuData)`

The editor's interface includes a file explorer on the right with icons for repository, files, search, and other tools. The bottom status bar shows the current file is "window.js", the encoding is "UTF-8", and the language is "JavaScript".



WINDOW.JS CODE



The screenshot shows a code editor with the file `window.js` open. The code defines a chart using a JS library. Two callouts provide context:


- USE JS LIBRARY TO CREATE CHART**: Points to the `new Chart` constructor on line 24.
- CREATE WITH DATA WE COLLECTED**: Points to the `datasets` property on line 37.

```
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 }
```

The bottom status bar indicates the file is `window.js`, using `UTF-8` encoding, `JavaScript` language, and is on the `master` branch. It also shows 4 files and 10 updates.



STYLE.CSS CODE



```
1  html, body, .container-fluid {
2    height: 100%;
3    background-color: #111;
4  }
5
6  html {
7    -webkit-app-region: drag;
8  }
9
10 .container-fluid {
11   padding: 25px;
12 }
13
```

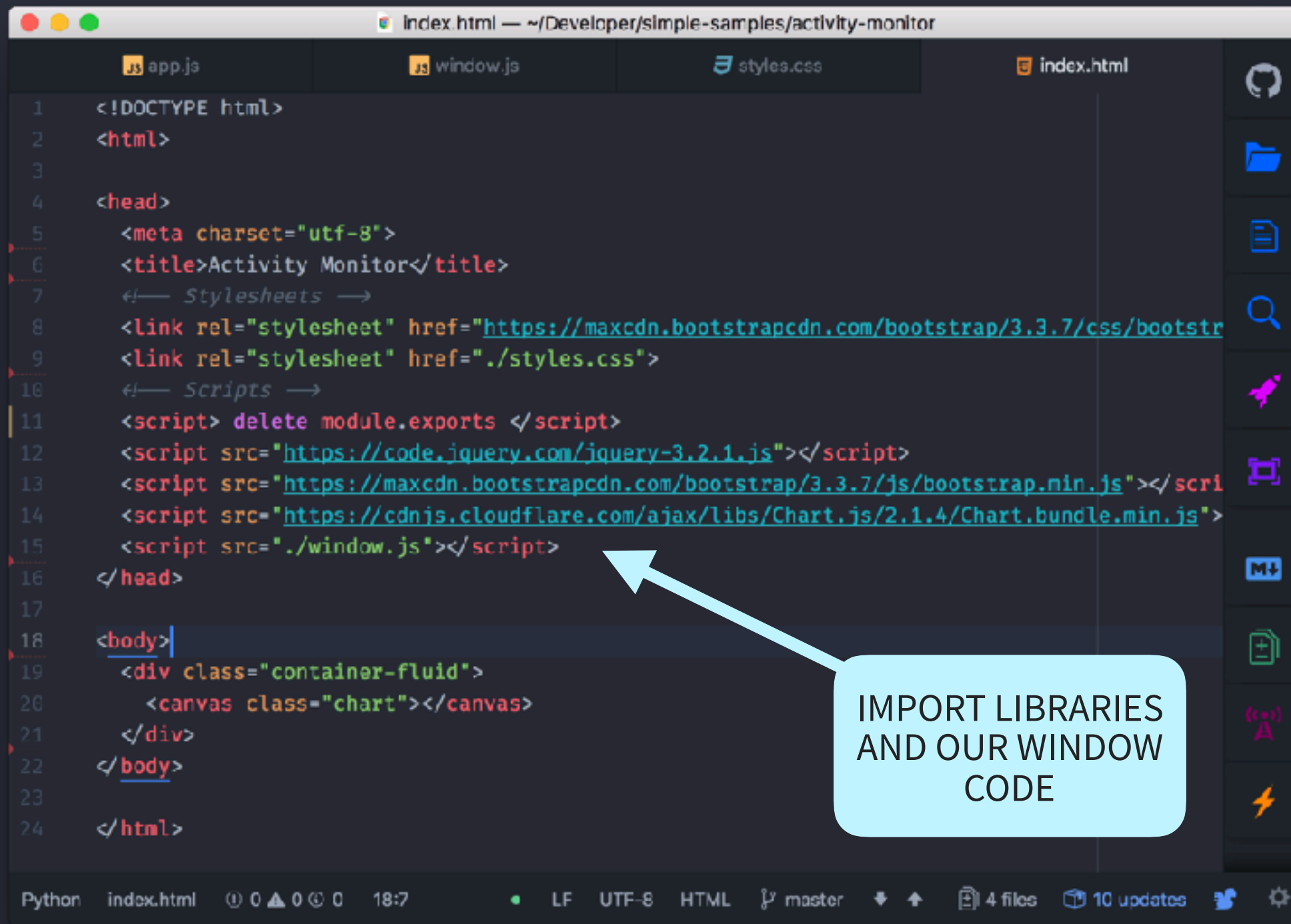
SET BACKGROUND COLOR AND HEIGHT OF CHART CONTAINER

The image shows a code editor window with the title 'styles.css — ~/Developer/simple-samples/activity-monitor'. The editor has four tabs: 'app.js', 'window.js', 'styles.css' (active), and 'index.html'. The code in 'styles.css' is as follows:

```
1  html, body, .container-fluid {
2    height: 100%;
3    background-color: #111;
4  }
5
6  html {
7    -webkit-app-region: drag;
8  }
9
10 .container-fluid {
11   padding: 25px;
12 }
13
```

A light blue arrow points from a text box containing the text 'SET BACKGROUND COLOR AND HEIGHT OF CHART CONTAINER' to the CSS code on lines 2 and 3. The status bar at the bottom shows 'Python', 'styles.css', '0 0 0 0', '1:1', 'LF', 'UTF-8', 'CSS', 'master', '4 files', '10 updates', and a settings icon.

INDEX.HTML CODE

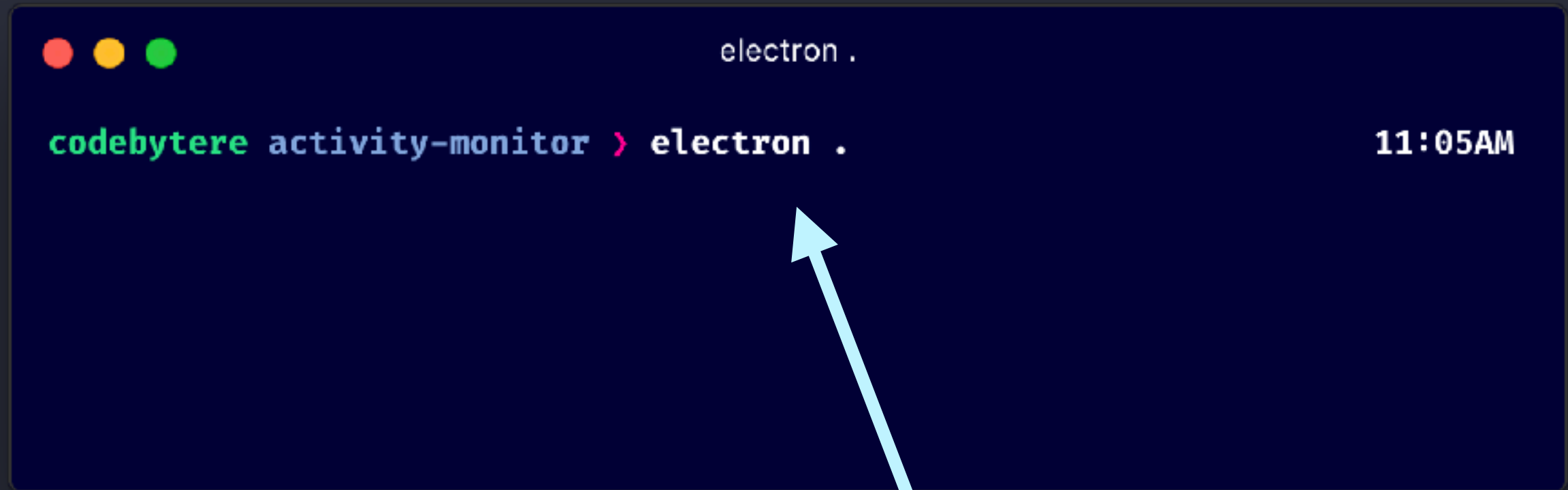


```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <title>Activity Monitor</title>
7   <!-- Stylesheets -->
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
9   <link rel="stylesheet" href="./styles.css">
10  <!-- Scripts -->
11  <script> delete module.exports </script>
12  <script src="https://code.jquery.com/jquery-3.2.1.js"></script>
13  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
14  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.4/Chart.bundle.min.js"></script>
15  <script src="./window.js"></script>
16 </head>
17
18 <body>
19   <div class="container-fluid">
20     <canvas class="chart"></canvas>
21   </div>
22 </body>
23
24 </html>
```

IMPORT LIBRARIES
AND OUR WINDOW
CODE



RUNNING YOUR APP



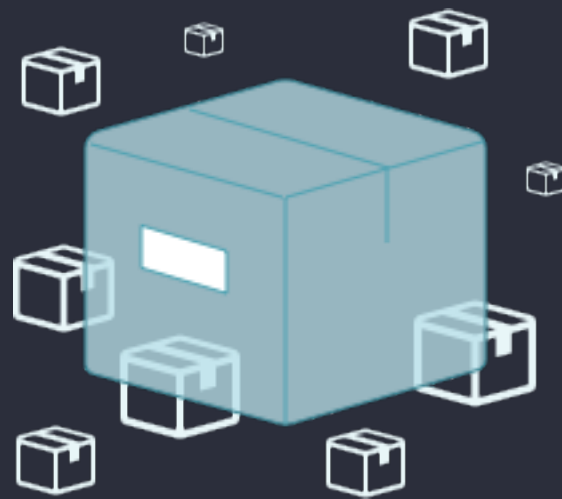
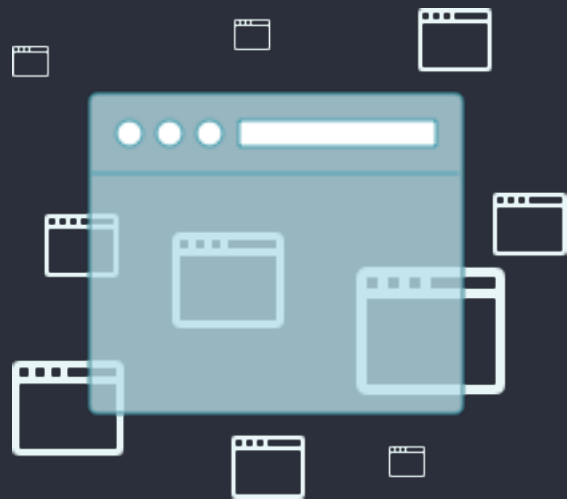
A terminal window with a dark blue background. The title bar shows three colored circles (red, yellow, green) and the text "electron .". The terminal content shows the command "codebytere activity-monitor > electron ." in a monospaced font. The time "11:05AM" is displayed in the top right corner. A light blue arrow points from a callout box below to the "electron ." part of the command.

```
electron .  
  
codebytere activity-monitor > electron . 11:05AM
```

CALL ELECTRON ON ALL
FILES IN CURRENT
DIRECTORY

COMPLETE APP

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



THANK YOU!

