

Table of Contents

Electron	3
-----------------------	---

Electron

(Atom shell)

<https://electronjs.org/>

Build cross platform desktop apps with JavaScript, HTML, and CSS

*Snippet from Wikipedia: **Electron (software framework)***

Electron (formerly known as **Atom Shell**) is a free and open-source software framework developed and maintained by OpenJS Foundation. The framework is designed to create desktop applications using web technologies (mainly HTML, CSS and JavaScript, although other technologies such as front-end frameworks and WebAssembly are possible) that are rendered using a version of the Chromium browser engine and a back end using the Node.js runtime environment. It also uses various APIs to enable functionality such as native integration with Node.js services and an inter-process communication module.

Electron was originally built for Atom and is the main GUI framework behind several other open-source projects including GitHub Desktop, Light Table, Visual Studio Code, WordPress Desktop and Eclipse Theia.

[Creative Commons Attribution-Share Alike 4.0](#)

GitHub Topics

- <https://github.com/topics/electron>

Electron is a desktop application framework developed by GitHub and created by Cheng Zhao. It allows for the development of desktop applications using front- and back-end infrastructure such as HTML, CSS, and JavaScript.



Main Process (main.js)

```
1 // Modules to control application life and create native browser window
2 const { app, BrowserWindow } = require('electron')
3 const path = require('path')
4
5 function createWindow () {
6   // Create the browser window.
7   const mainWindow = new BrowserWindow({
8     width: 800,
9     height: 600,
10    webPreferences: {
11      preload: path.join(__dirname, 'preload.js')
12    }
13  })
14
15 // and load the index.html of the app.
16 mainWindow.loadFile('index.html')
17
18 // Open the DevTools.
19 // mainWindow.webContents.openDevTools()
20 }
```

HTML (index.html)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <!-- https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP -->
6     <meta http-equiv="Content-Security-Policy" content="default-src
'self'; script-src 'self'">
7     <title>Hello World!</title>
8   </head>
9   <body>
10    <h1>Hello World!</h1>
11    We are using Node.js <span id="node-version"></span>,
12    Chromium <span id="chrome-version"></span>,
13    and Electron <span id="electron-version"></span>.
14
15    <!-- You can also require other files to run in this process -->
16    <script src="/>./renderer.js"></script>
17  </body>
18 </html>
```

Modules

Search...

Renderer Process (renderer.js)

```
1 // This file is required by the index.html file and will
2 // be executed in the renderer process for that window.
3 // No Node.js APIs are available in this process because
4 // 'nodeIntegration' is turned off. Use 'preload.js' to
5 // selectively enable features needed in the rendering
6 // process.
7
```

Preload (preload.js)

```
1 // All of the Node.js APIs are available in the preload process.
2 // It has the same sandbox as a Chrome extension.
3 window.addEventListener('DOMContentLoaded', () => {
4   const replaceText = (selector, text) => {
5     const element = document.getElementById(selector)
6     if (element) element.innerText = text
7   }
8
9   for (const type of ['chrome', 'node', 'electron']) {
10     replaceText(`${type}-version`, process.versions[type])
11   }
12 })
```

[Previous](#) [Next](#)

Related:

- Cross-platform

External links:

- <https://www.electronjs.org/fiddle>

tool, architecture, programming, devopscreate

From:
<https://www.almbok.com/> - **ALMBoK.com**



Permanent link:
<https://www.almbok.com/tools/electron>

Last update: **2024/04/13 12:53**