

BUILDING DESKTOP APPLICATIONS WITH

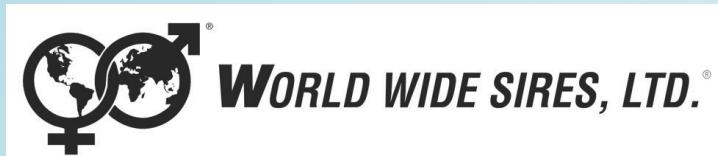


ELECTRON

By Abram Adams

WHO AM I?







TRY COLDFUSION

[!\[\]\(e03fe9a24fe5105c8f59e912ae7b8724_img.jpg\) BLOG](#)

[!\[\]\(b4572a044582c68c9e6e6b6b9b95c325_img.jpg\) HOME](#)

[!\[\]\(a4848a7f290c3fd14bf2276a5b09747a_img.jpg\) TUTORIALS \(17\)](#)

[!\[\]\(c1b926fcee63cdc7f15c603966ea3d76_img.jpg\) SCRATCH PAD \(+\)](#)

[!\[\]\(8bca340a5031aa30dac36202fe939cc1_img.jpg\) CF GITHUB REPOS](#)

[!\[\]\(3e3e4303b0632a7f1c9ba6d71d06a491_img.jpg\) COMMUNITY EVENTS](#)

[!\[\]\(92ce96eaea0d7284690928c496458da0_img.jpg\) HELP](#)

 HOME

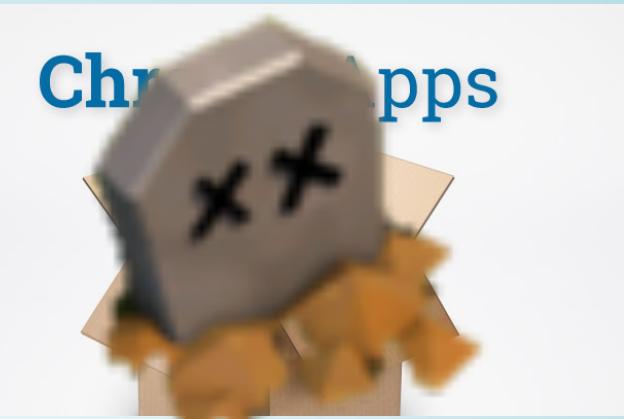
 WELCOME TO TRYCF.COM!

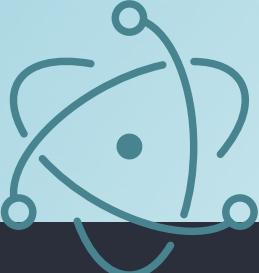
Try CF was designed to be a barrier-free introduction to the [CFML](#) programming language. Using the modern, interactive learning tools we've got here you'll quickly learn to code in CFML and be up to speed developing dynamic web apps in no time. If you just want a quick feel for the language, use the code box below to play around. Once you've had a chance to play and are ready for more, check out our ever growing list of [tutorials](#) designed to help you learn and master ColdFusion web development.

 CFML EDITOR

```
1 <cfscript>
2 // CFScript version
3 writeOutput(
4     'Hello world from CFScript! ' &
5     'The server time is ' & now()
6 );
7 </cfscript>
```

Results

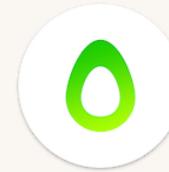
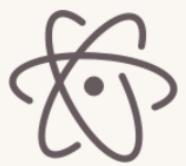




ELECTRON



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



jibo



DEMO!

WHY?



BENEFITS OF DESKTOP APPS

- Local File System Access
- Network Access
- Frameless Windows
- Multi-Threaded Processes
- Access to OS Features (clipboard, notifications, etc.)
- Database Access (Directly via NodeJS)
- Start/Stop/Interact with Native/OS Services
- Develop Native-Feeling User Experiences
- Fixes Age Old "Back Button" Issue
- NO MORE BROWSER WARS!!
- Too Many To List

HOW?

```
# Clone the Quick Start repository
$ git clone https://github.com/electron/electron-quick-start

# Go into the repository
$ cd electron-quick-start

# Install the dependencies and run
$ npm install && npm start
```

DEMO!

PROCESSES

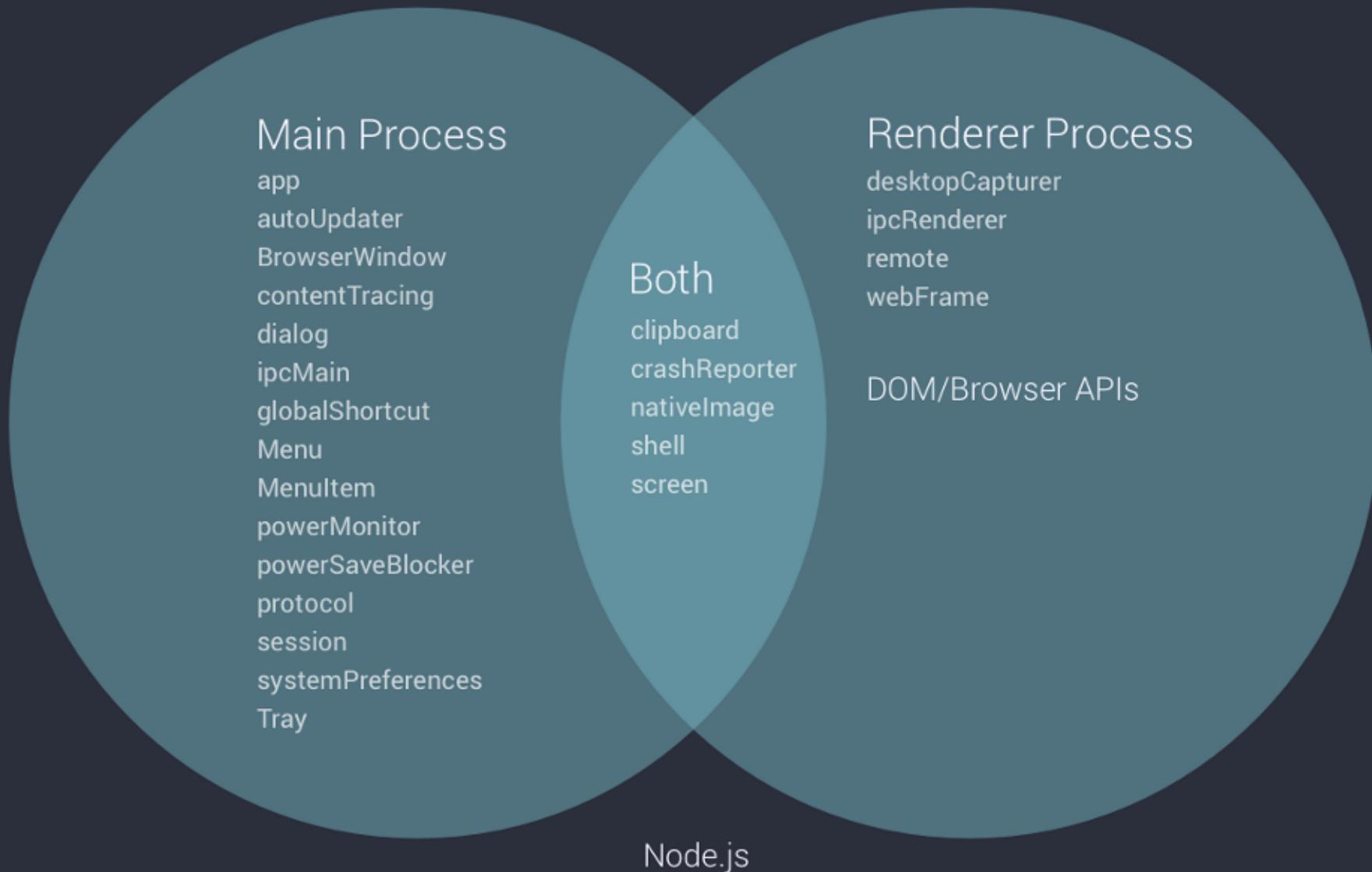
MAIN PROCESS

The main process is responsible for creating and managing *BrowserWindow* instances and various application events.

RENDER PROCESS

The render process is responsible for running the user-interface of your app, or in other words, a web page which is an instance of *webContents*.

Credit: <https://medium.com/@ccnokes/deep-dive-into-electrons-main-and-renderer-processes-7a9599d5c9e2>



```
// main.js (MAIN PROCESS)
function createWindow () {
  // Create the browser window.
  mainWindow = new BrowserWindow({width: 800, height: 600}

  // and load the index.html of the app.
  mainWindow.loadURL(url.format({
    pathname: path.join(__dirname, 'index.html'),
    protocol: 'file:',
    slashes: true
  }))}

//...
}
```

```
<!-- index.html --->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello cf.Objective() World!</h1>
    <!-- All of the Node.js APIs are available in this ren-
        We are using Node.js <script>document.write(process.ve-
        Chromium <script>document.write(process.versions.chrom-
        and Electron <script>document.write(process.versions.e-
    </body>

    <script>
      // You can also require other files to run in this pro-
      require('./renderer.js')
    </script>
</html>
```

DEMO!

NODE.JS AS A FIRST CLASS CITIZEN

- ECMAScript 2015/ES6 support
- Leverage ~475k npm modules
- You can do what previously had to be done server side

DEMO

BUT WAIT, THERE'S MORE!

DEMO!

CONCLUSION

RESOURCES

- <https://electron.atom.io/community/>



<http://cfml-slack.herokuapp.com/>



<https://github.com/abramadams>



@Abram_Adams



aadams@cfxchange.com

SPA DEVELOPMENT FOR THE SERVER-SIDE DEVELOPER

By Abram Adams



<http://cfml-slack.herokuapp.com/>



<https://github.com/abramadams>



@Abram_Adams



aadams@cfxchange.com