JavaScript

Boitumelo Phetla[1,2,3] (PluralSight Google ScholarShip)[2,3]

[1] *Universidad Nacional Autónoma de México, Mexico City, Mexico*
[2] *University of Texas at Austin, Texas, United States of America*
[3] *LaTeXTemplates.com*

May 20, 2019

JavaScript, is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat.

1 Simplistic JavaScript 1

1.1 Command-line based programming

A simple project:

```
$bash: touch {index.html,script.js,style.css}
$bash: tree
   ---------- index.html
   ---------- script.js
   ---------- style.css
```

Include the script (**javascript**) and the page styling script (**cascading stylesheet**) files into the *index.html*.

```
<head>
      <script src="path/*.js"></script>
      <link rel="stylesheet"
          href="path/*.css">
</head>
```

Add some simple HTML markup code and launch a live-server of the code.

```
<!DOCTYPE>
<html>
      <head>
            <script src="script.js"></script>
            <link rel="stylesheet"
                href="style.css">
      </head>
            <body>
                  <div id="header">
                        <h1>Welcome to
                            JavaScript</h1>
                  </div>
            </body>
</html>
```

Launch the command-line (Terminal)

```
$bash: live-server
```

Figure 1: *Live-server*

1.2 Plunker

Or create an account on Plunker. Plunker sets up your working environment for you.

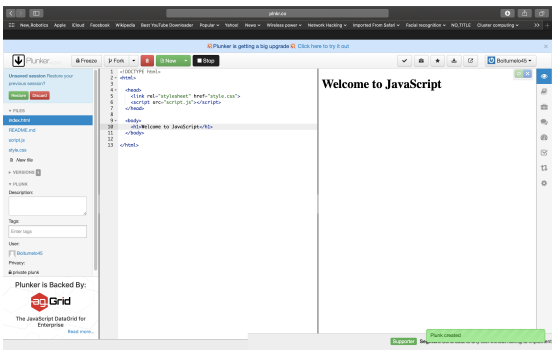

Figure 2: *Plunker*

1.3 Electron

Watch this video Electron.

```
# 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
```

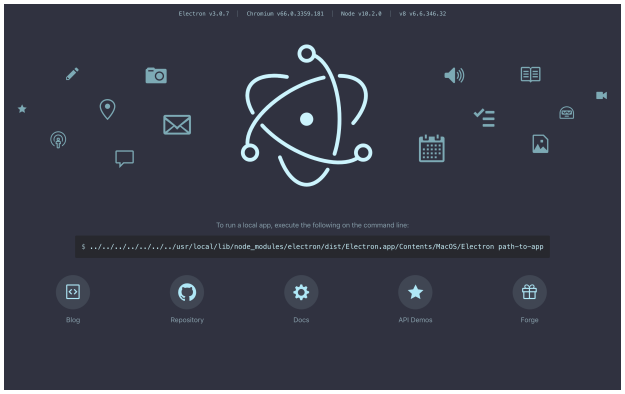

Figure 3: *Electron*

```
$bash: mkdir Electron1; cd Electron1; npm init
 1 {
 2   "name": "electron1",
 3   "version": "1.0.0",
 4   "description": "First App",
 5   "main": "index.js",
 6   "scripts": {
 7     "test": "echo \"Error: no test
      specified\" && exit 1"
 8   },
 9   "keywords": [
10     "Electron"
11   ],
12   "author": "Boitumelo Phetla",
13   "license": "ISC"
14 }
```

At this point, you'll need to install electron itself. The recommended way of doing so is to install it as a development dependency in your app, which allows you to work on multiple apps with different Electron versions. To do so, run the following command from your app's directory:

```
$bash: npm install --save-dev electron
$bash: tree -L 1
        .
        |____________node_modules
        |____________package-lock.json
        |____________package.json

1 directory, 2 files
```

All APIs and features found in Electron are accessible through the electron module, which can be required like any other Node.js module:

```
const electron = require('electron')
```

To avoid any huddles, try this simple example.

```
# Clone the repository
$ git clone
    https://github.com/electron/electron-quick-start
# Go into the repository
$ cd electron-quick-start
# Install dependencies
$ npm install
# Run the app
$ npm start
```

1.4 Meteor

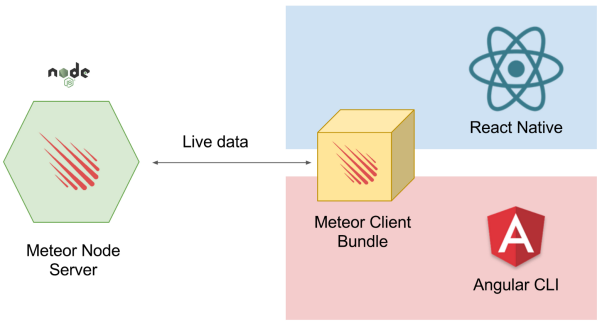

Figure 4: *Meteor*

To create the app, open your terminal and type:

```
$bash: meteor create simple-todos

output:

Created a new Meteor app in 'simple-todos'.

To run your new app:
  cd simple-todos
  meteor

If you are new to Meteor, try some of the
    learning resources here:
  https://www.meteor.com/tutorials

To start with a different app template, try one
    of the following:

  meteor create --bare  # to create an empty app
  meteor create --minimal # to create an app
      with as few Meteor packages as possible
  meteor create --full  # to create a more
      complete scaffolded app
```

1.5 JavaScript Editors (IDEs)

- Plunker
- Second item in a list
- Third item in a list