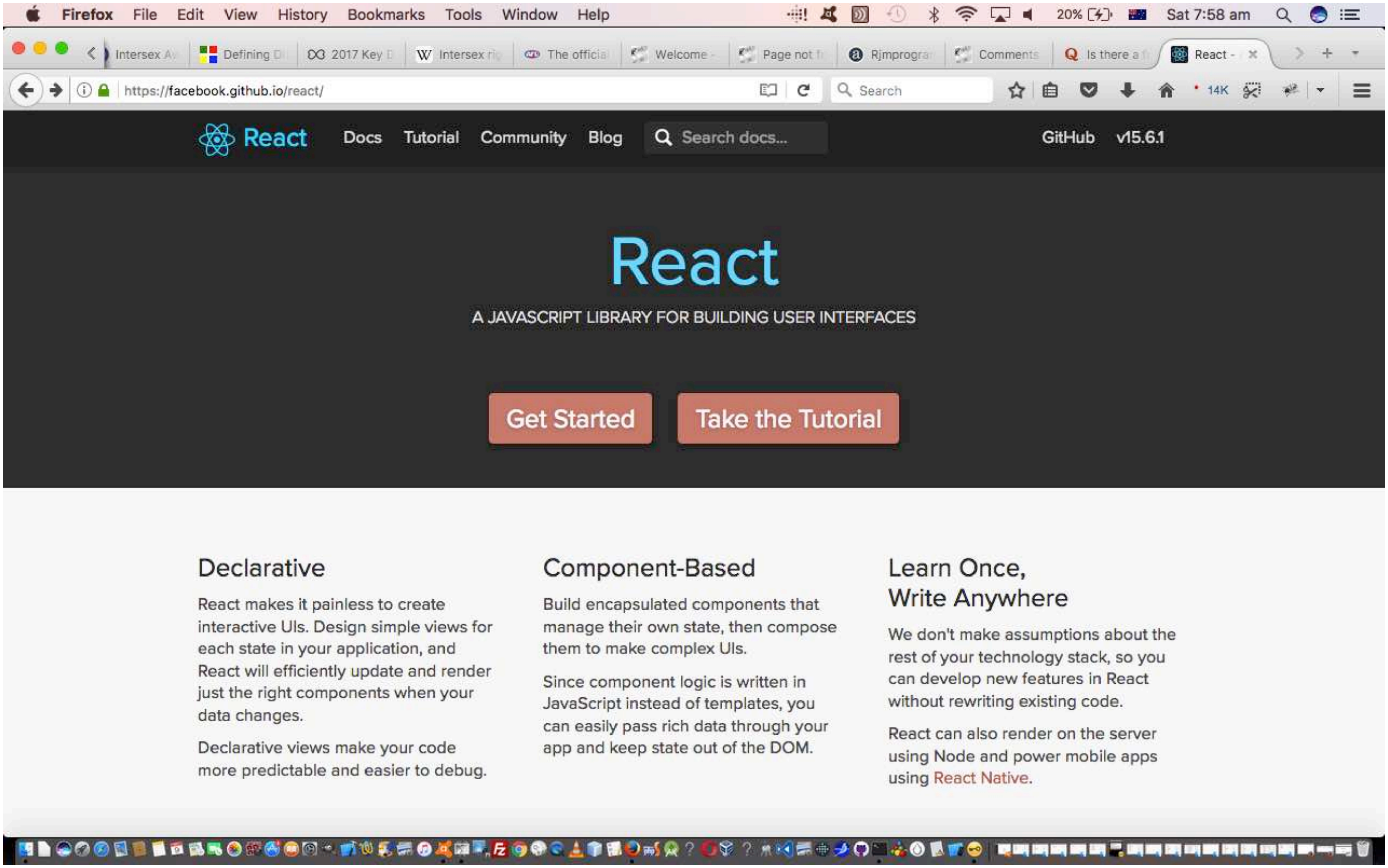


React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Component-Based

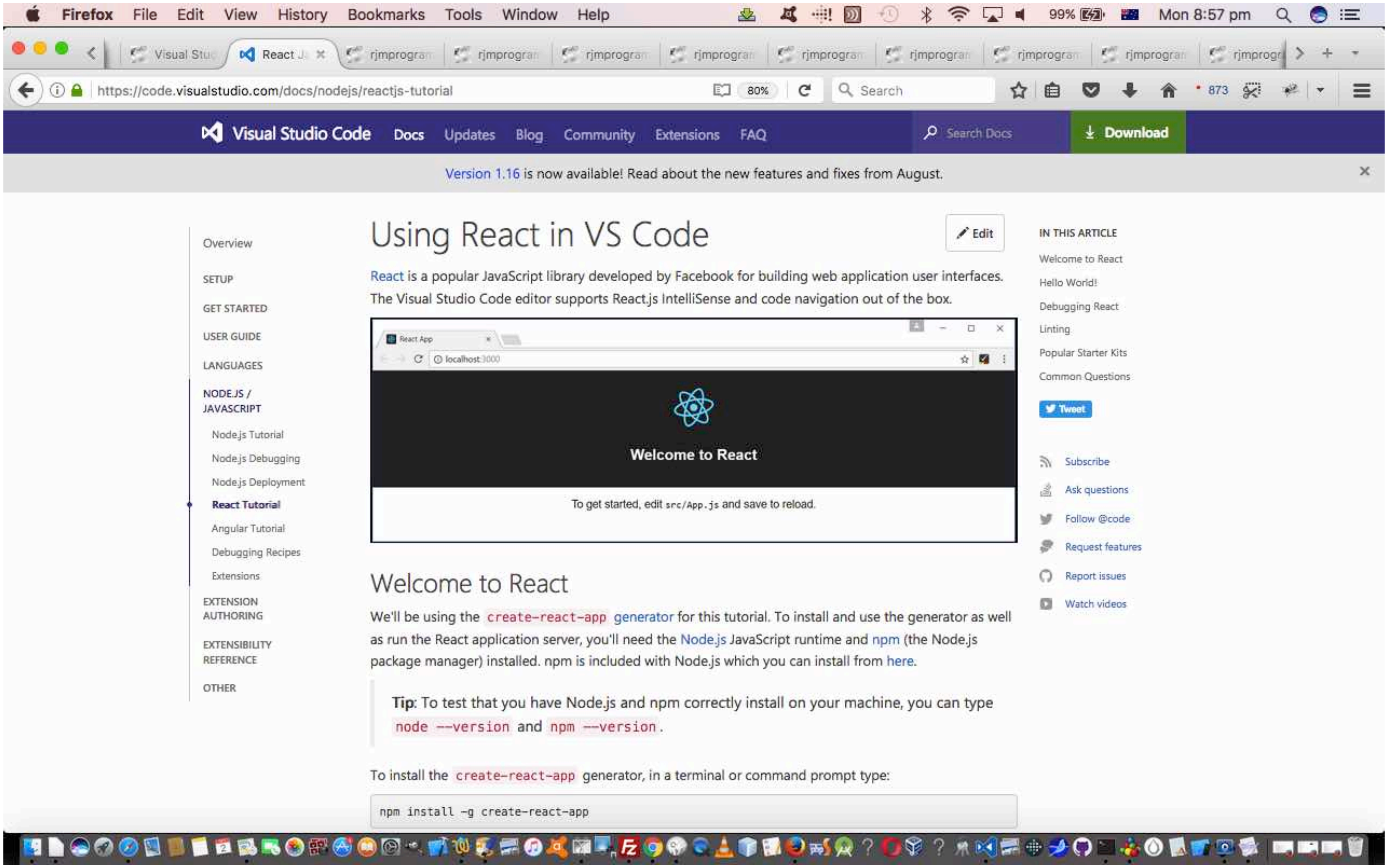
Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 8:57 pm
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
Visual Studio Code Docs Updates Blog Community Extensions FAQ
Search Docs
Download

Version 1.16 is now available! Read about the new features and fixes from August.

Using React in VS Code

React is a popular JavaScript library developed by Facebook for building web application user interfaces. The Visual Studio Code editor supports React.js IntelliSense and code navigation out of the box.

To get started, edit `src/App.js` and save to reload.

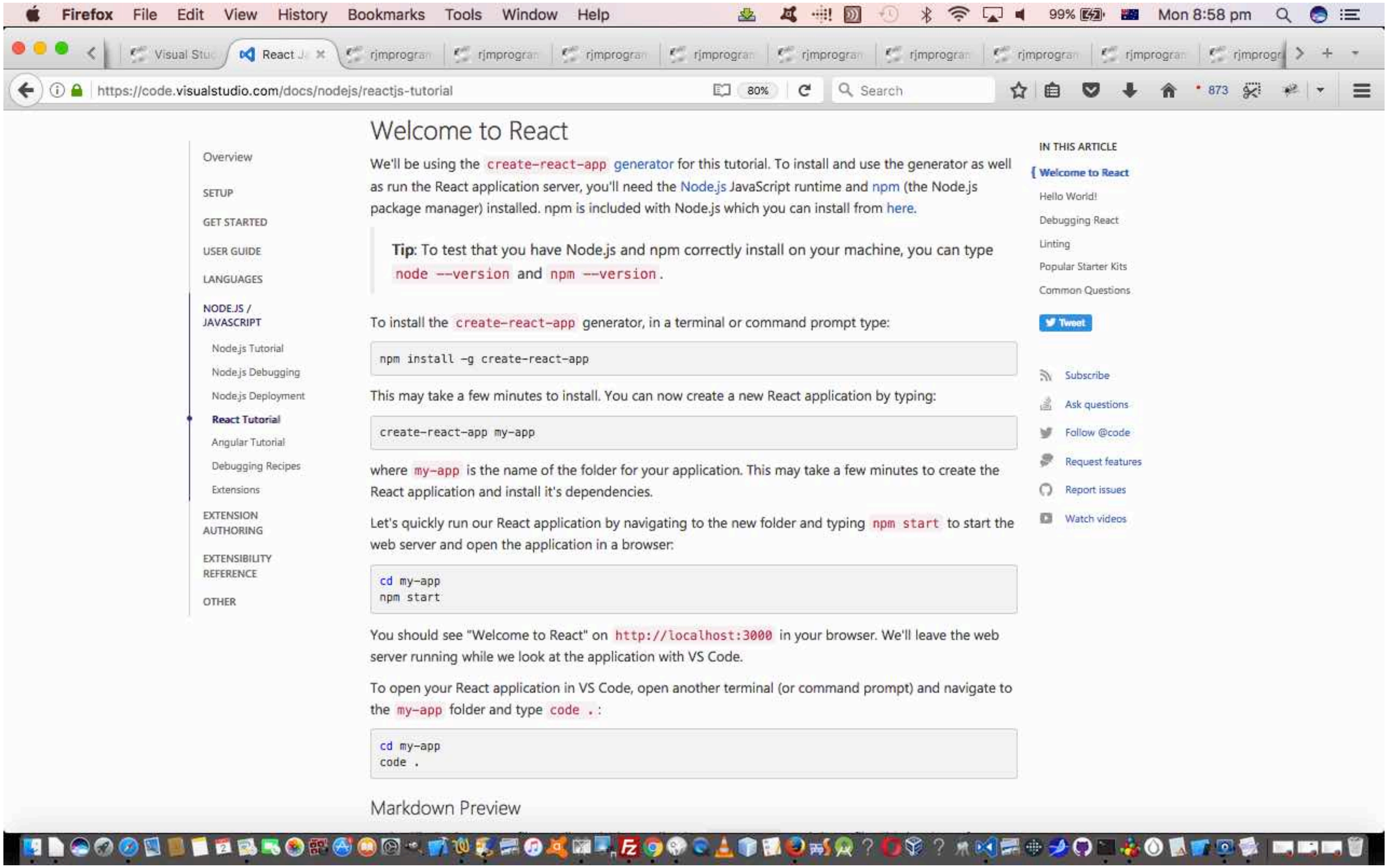
Welcome to React

We'll be using the `create-react-app` generator for this tutorial. To install and use the generator as well as run the React application server, you'll need the Node.js JavaScript runtime and npm (the Node.js package manager) installed. npm is included with Node.js which you can install from here.

> **Tip**: To test that you have Node.js and npm correctly install on your machine, you can type `node --version` and `npm --version`.

To install the `create-react-app` generator, in a terminal or command prompt type:

```
npm install -g create-react-app
```


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 8:58 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Welcome to React

We'll be using the `create-react-app` generator for this tutorial. To install and use the generator as well as run the React application server, you'll need the Node.js JavaScript runtime and npm (the Node.js package manager) installed. npm is included with Node.js which you can install from here.

> **Tip**: To test that you have Node.js and npm correctly install on your machine, you can type `node --version` and `npm --version`.

To install the `create-react-app` generator, in a terminal or command prompt type:

```
npm install -g create-react-app
```

This may take a few minutes to install. You can now create a new React application by typing:

```
create-react-app my-app
```

where `my-app` is the name of the folder for your application. This may take a few minutes to create the React application and install it's dependencies.

Let's quickly run our React application by navigating to the new folder and typing `npm start` to start the web server and open the application in a browser:

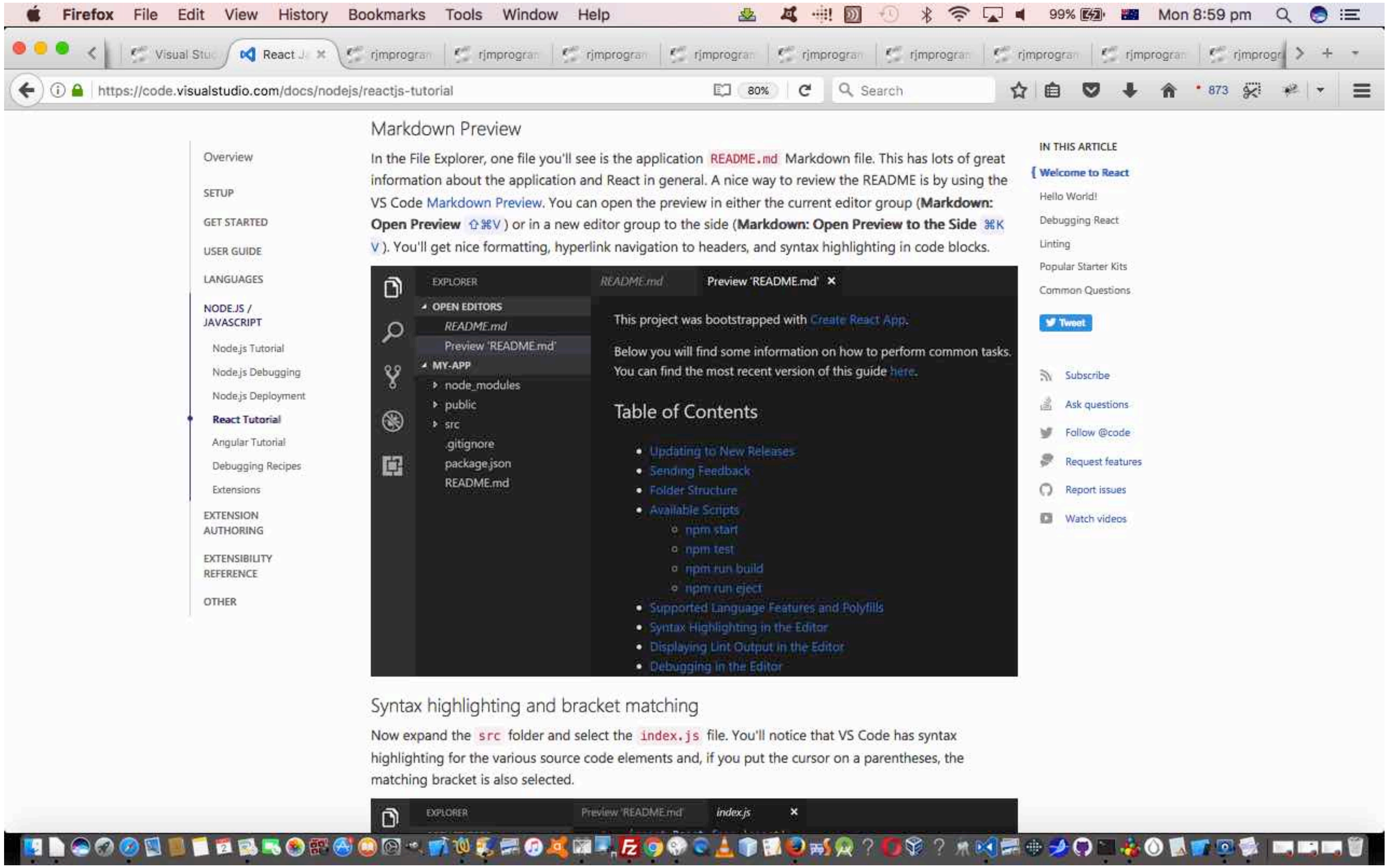
```
cd my-app
npm start
```

You should see "Welcome to React" on `http://localhost:3000` in your browser. We'll leave the web server running while we look at the application with VS Code.

To open your React application in VS Code, open another terminal (or command prompt) and navigate to the `my-app` folder and type `code .`:

```
cd my-app
code .
```

Markdown Preview


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 8:59 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Markdown Preview

In the File Explorer, one file you'll see is the application `README.md` Markdown file. This has lots of great information about the application and React in general. A nice way to review the README is by using the VS Code Markdown Preview. You can open the preview in either the current editor group (**Markdown: Open Preview** ⇧⌘V) or in a new editor group to the side (**Markdown: Open Preview to the Side** ⌘K V). You'll get nice formatting, hyperlink navigation to headers, and syntax highlighting in code blocks.

This project was bootstrapped with Create React App.

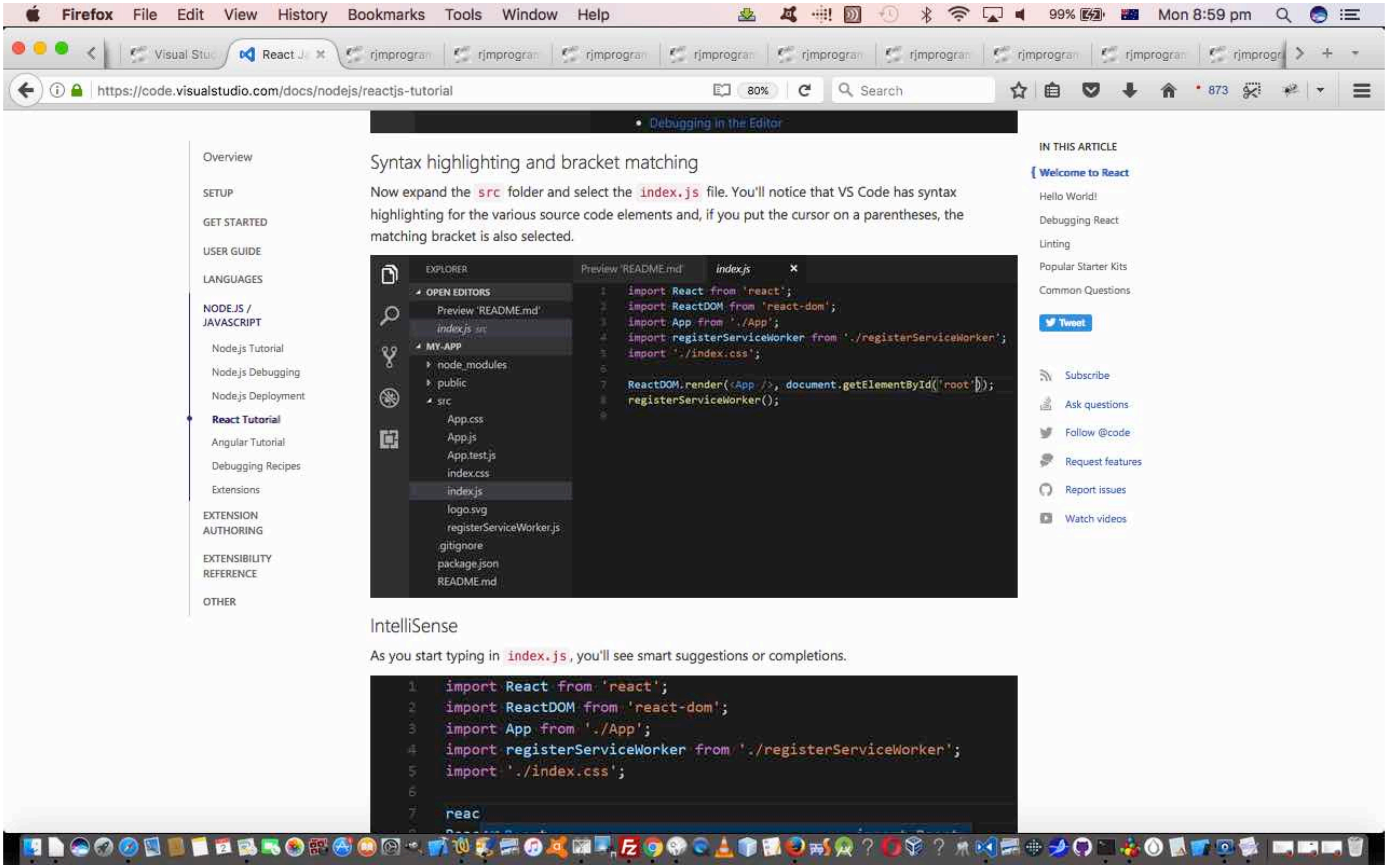
Below you will find some information on how to perform common tasks.
You can find the most recent version of this guide here.

Table of Contents

- Updating to New Releases
- Sending Feedback
- Folder Structure
- Available Scripts
  - npm start
  - npm test
  - npm run build
  - npm run eject
- Supported Language Features and Polyfills
- Syntax Highlighting in the Editor
- Displaying Lint Output in the Editor
- Debugging in the Editor

Syntax highlighting and bracket matching

Now expand the `src` folder and select the `index.js` file. You'll notice that VS Code has syntax highlighting for the various source code elements and, if you put the cursor on a parentheses, the matching bracket is also selected.


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 8:59 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

• Debugging in the Editor

Syntax highlighting and bracket matching

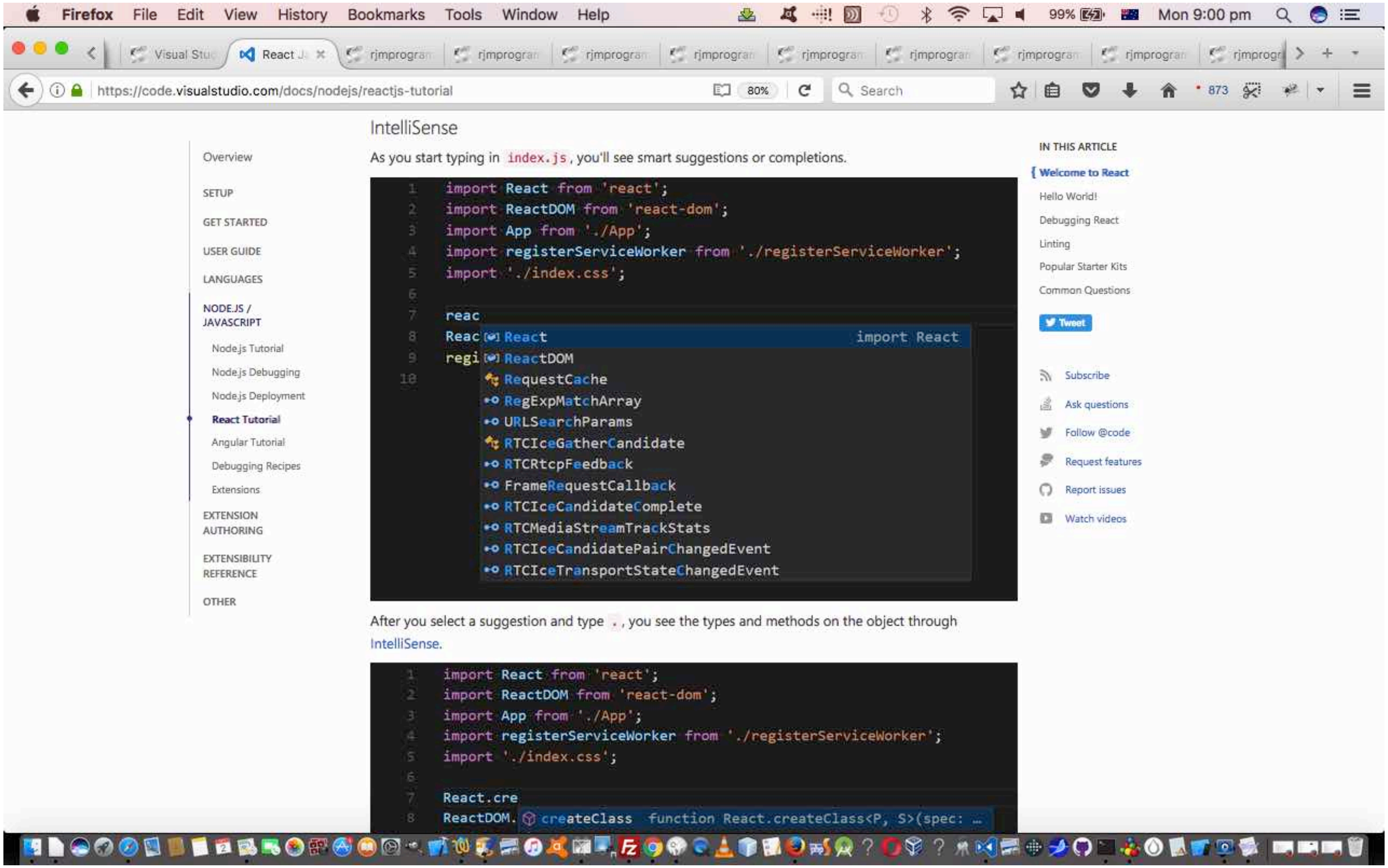
Now expand the `src` folder and select the `index.js` file. You'll notice that VS Code has syntax highlighting for the various source code elements and, if you put the cursor on a parentheses, the matching bracket is also selected.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

ReactDOM.render(<App />, document.getElementById('root'));
registerServiceWorker();
```

IntelliSense

As you start typing in `index.js`, you'll see smart suggestions or completions.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

reac
```


Firefox File Edit View History Bookmarks Tools Window Help
Mon 9:00 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search

Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS / JAVASCRIPT

Node.js Tutorial

Node.js Debugging

Node.js Deployment

React Tutorial

Angular Tutorial

Debugging Recipes

Extensions

EXTENSION AUTHORING

EXTENSIBILITY REFERENCE

OTHER

IntelliSense

As you start typing in `index.js`, you'll see smart suggestions or completions.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

reac
Reac  React                                      import React
regi  ReactDOM
      RequestCache
      RegExpMatchArray
      URLSearchParams
      RTCIceGatherCandidate
      RTCRtcpFeedback
      FrameRequestCallback
      RTCIceCandidateComplete
      RTCMediaStreamTrackStats
      RTCIceCandidatePairChangedEvent
      RTCIceTransportStateChangedEvent
```

After you select a suggestion and type `.`, you see the types and methods on the object through IntelliSense.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

React.cre
ReactDOM.  createClass  function React.createClass<P, S>(spec: …
```

IN THIS ARTICLE

Welcome to React

Hello World!

Debugging React

Linting

Popular Starter Kits

Common Questions

Tweet

Subscribe

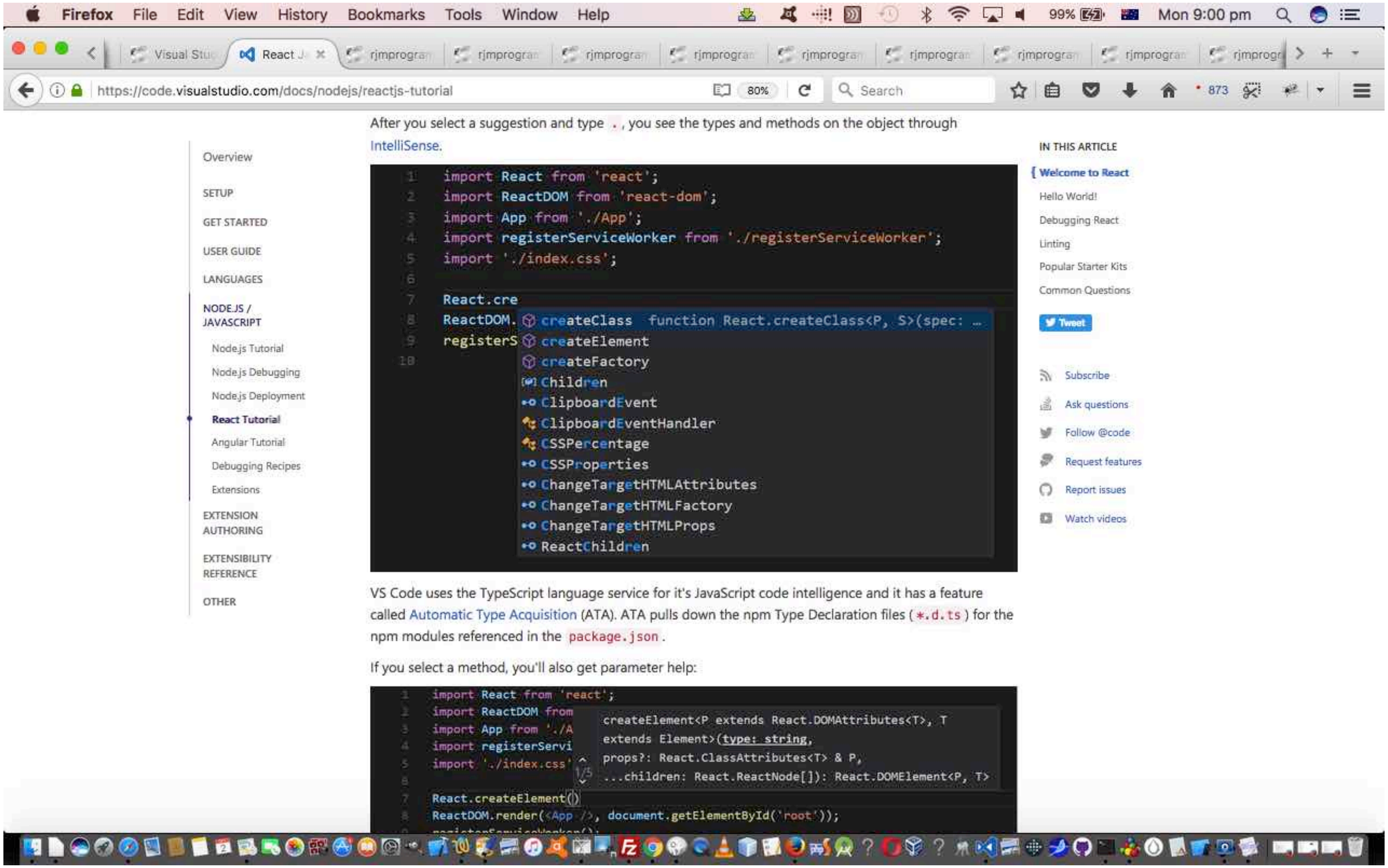
Ask questions

Follow @code

Request features

Report issues

Watch videos


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:00 pm
Visual Stu
React J
rjmprogram
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

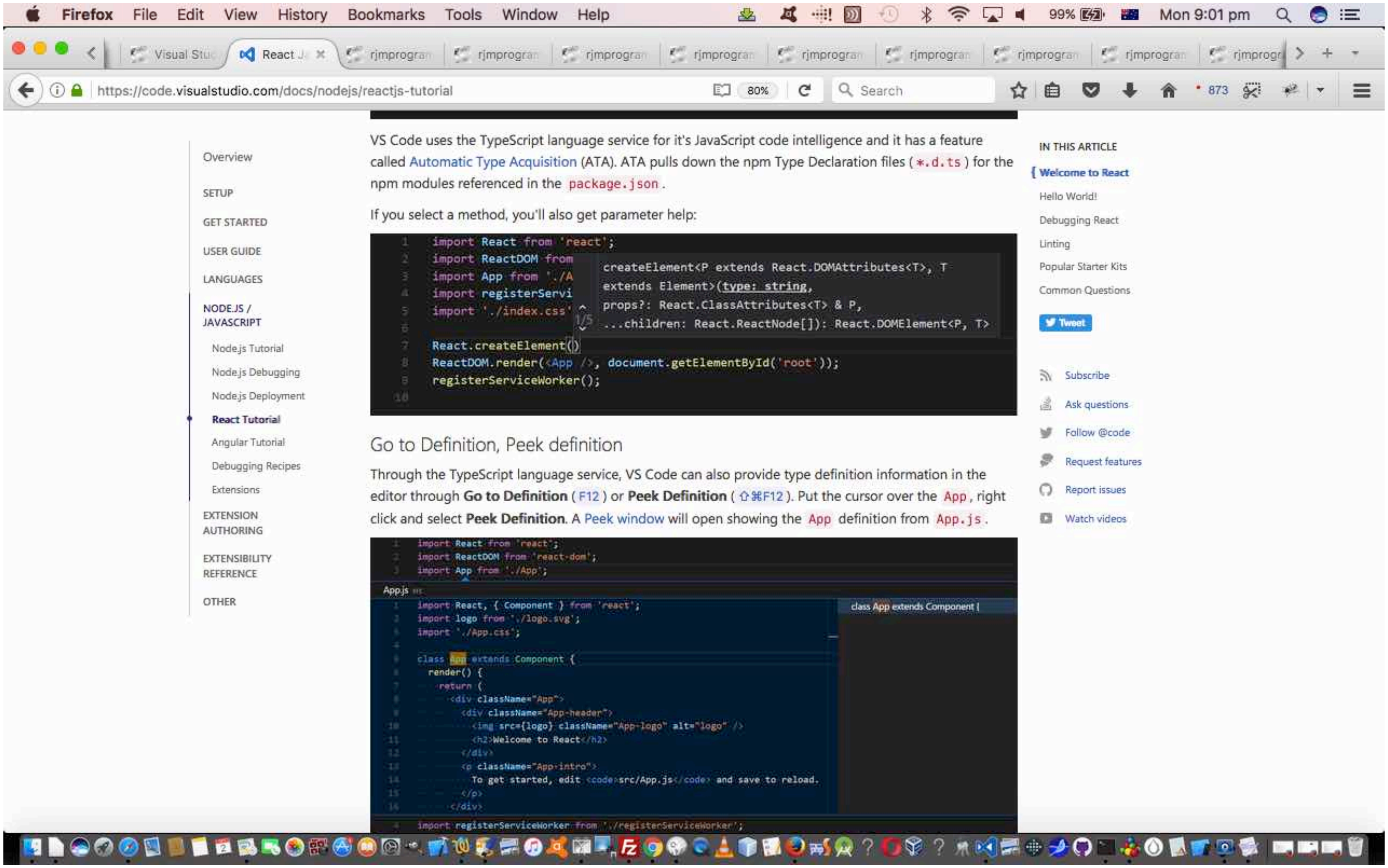
After you select a suggestion and type `.`, you see the types and methods on the object through IntelliSense.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

React.cre
ReactDOM. createClass   function React.createClass<P, S>(spec: ...
registerS createElement
          createFactory
          Children
          ClipboardEvent
          ClipboardEventHandler
          CSSPercentage
          CSSProperties
          ChangeTargetHTMLAttributes
          ChangeTargetHTMLFactory
          ChangeTargetHTMLProps
          ReactChildren
```

VS Code uses the TypeScript language service for it's JavaScript code intelligence and it has a feature called Automatic Type Acquisition (ATA). ATA pulls down the npm Type Declaration files (`*.d.ts`) for the npm modules referenced in the `package.json`.

If you select a method, you'll also get parameter help:

```
import React from 'react';
import ReactDOM from     createElement<P extends React.DOMAttributes<T>, T
import App from './A     extends Element>(type: string,
import registerServi     props?: React.ClassAttributes<T> & P,
import './index.css'  1/5 ...children: React.ReactNode[]): React.DOMElement<P, T>

React.createElement()
ReactDOM.render(<App />, document.getElementById('root'));
```


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:01 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

VS Code uses the TypeScript language service for it's JavaScript code intelligence and it has a feature called Automatic Type Acquisition (ATA). ATA pulls down the npm Type Declaration files (`*.d.ts`) for the npm modules referenced in the `package.json`.

If you select a method, you'll also get parameter help:

```
import React from 'react';
import ReactDOM from
import App from './A
import registerServi
import './index.css'

React.createElement()
ReactDOM.render(<App />, document.getElementById('root'));
registerServiceWorker();
```

```
createElement<P extends React.DOMAttributes<T>, T
extends Element>(type: string,
props?: React.ClassAttributes<T> & P,
...children: React.ReactNode[]): React.DOMElement<P, T>
```

Go to Definition, Peek definition

Through the TypeScript language service, VS Code can also provide type definition information in the editor through **Go to Definition** (F12) or **Peek Definition** (⇧⌘F12). Put the cursor over the `App`, right click and select **Peek Definition**. A Peek window will open showing the `App` definition from `App.js`.

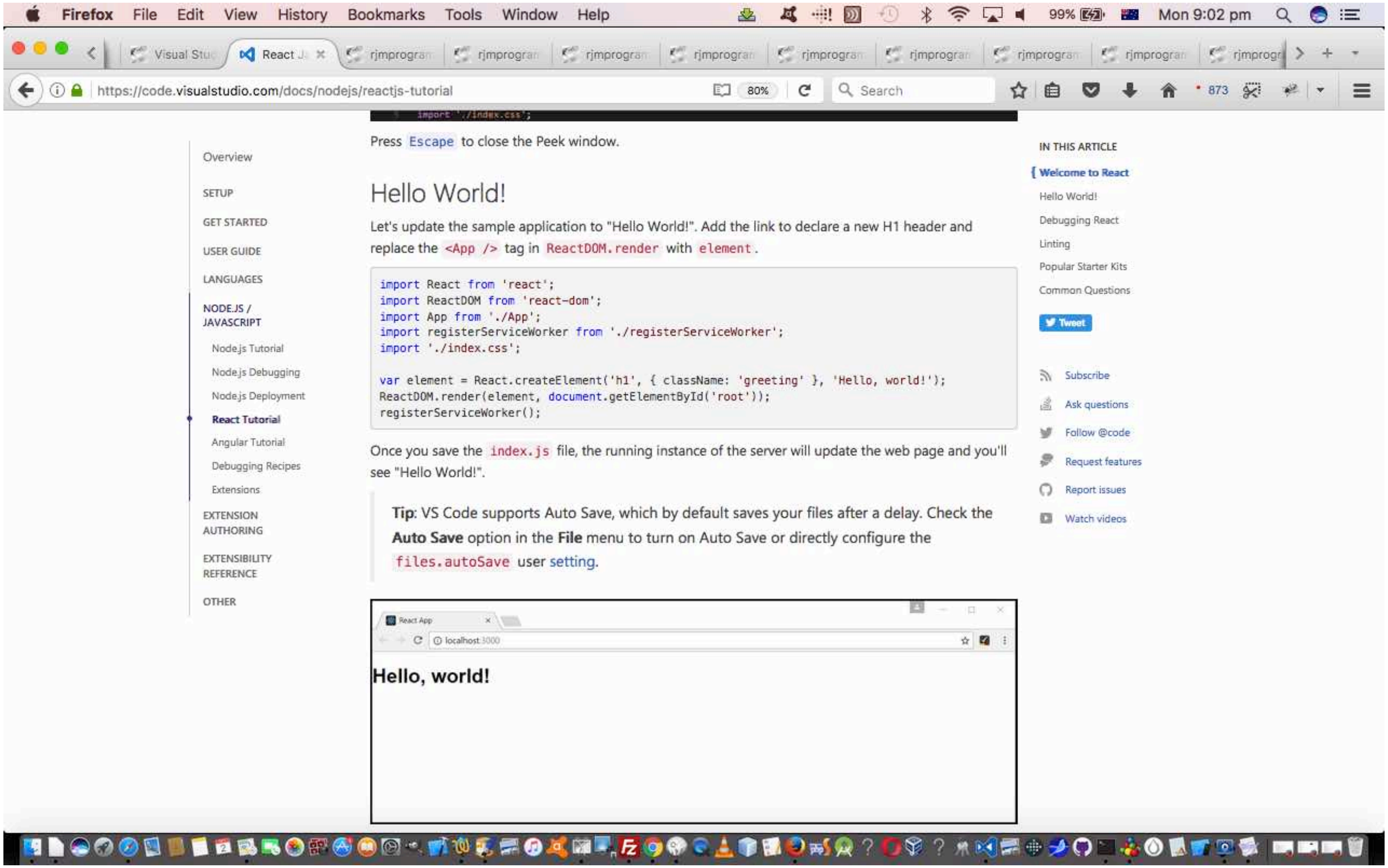
```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
```

App.js

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <div className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h2>Welcome to React</h2>
        </div>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and save to reload.
        </p>
      </div>
```

class App extends Component {

```
import registerServiceWorker from './registerServiceWorker';
```



Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:02 pm
Visual Stu
React J
rjmprogram
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Press `Escape` to close the Peek window.

Hello World!

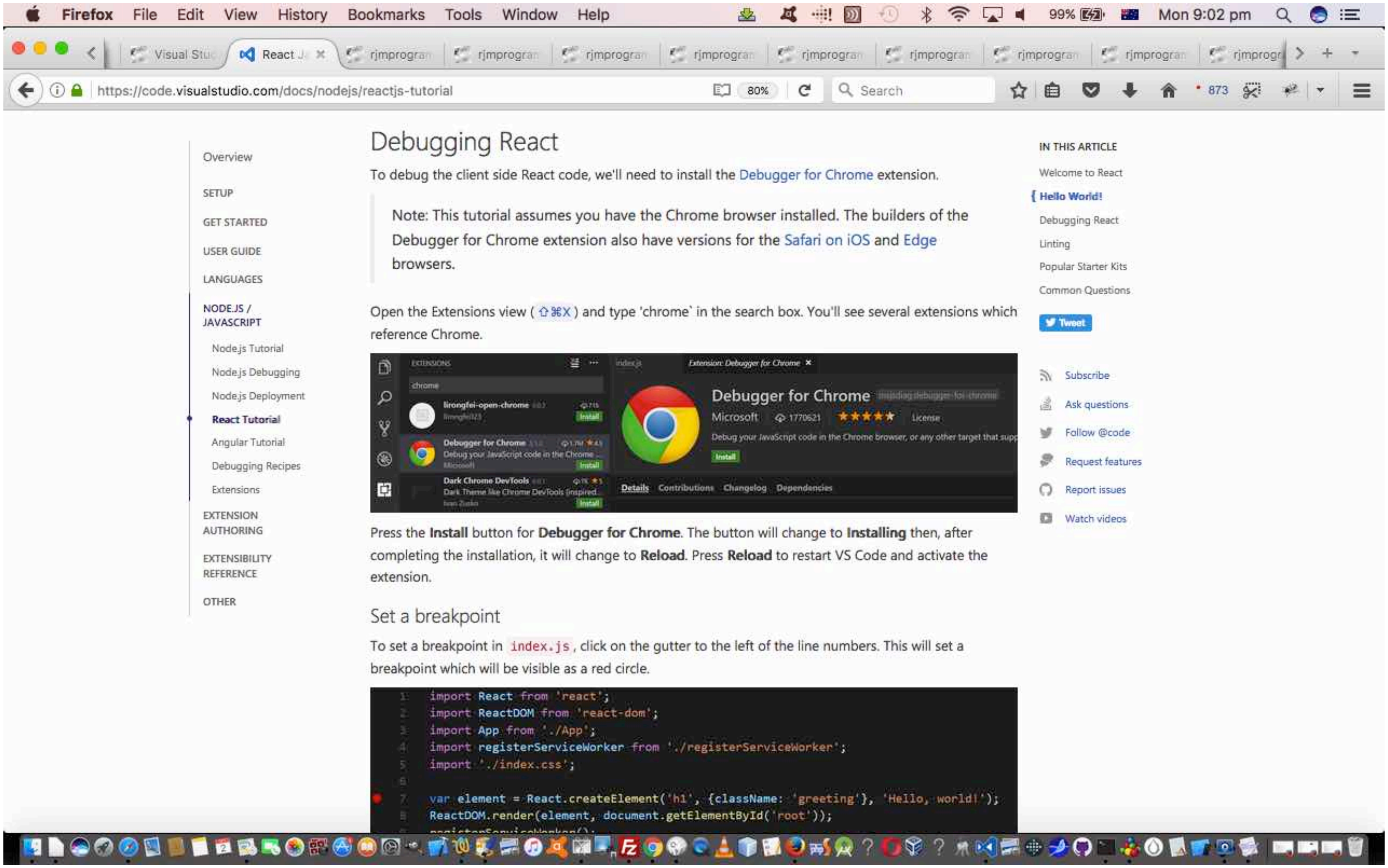
Let's update the sample application to "Hello World!". Add the link to declare a new H1 header and replace the `<App />` tag in `ReactDOM.render` with `element`.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', { className: 'greeting' }, 'Hello, world!');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```

Once you save the `index.js` file, the running instance of the server will update the web page and you'll see "Hello World!".

> **Tip**: VS Code supports Auto Save, which by default saves your files after a delay. Check the **Auto Save** option in the **File** menu to turn on Auto Save or directly configure the `files.autoSave` user setting.

Hello, world!


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:02 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Debugging React

To debug the client side React code, we'll need to install the Debugger for Chrome extension.

> Note: This tutorial assumes you have the Chrome browser installed. The builders of the Debugger for Chrome extension also have versions for the Safari on iOS and Edge browsers.

Open the Extensions view (⇧⌘X) and type 'chrome` in the search box. You'll see several extensions which reference Chrome.

Press the **Install** button for **Debugger for Chrome**. The button will change to **Installing** then, after completing the installation, it will change to **Reload**. Press **Reload** to restart VS Code and activate the extension.

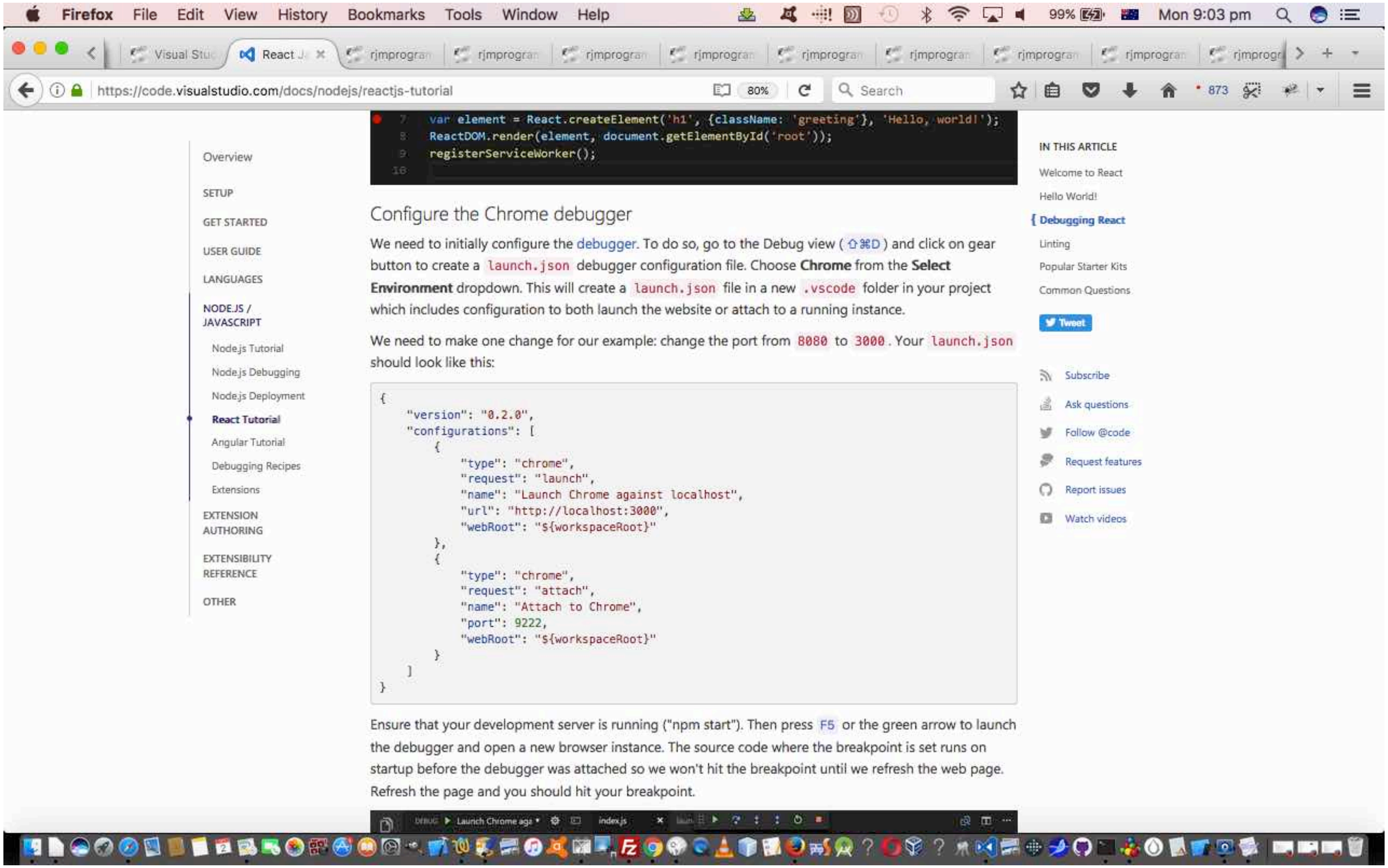
Set a breakpoint

To set a breakpoint in `index.js`, click on the gutter to the left of the line numbers. This will set a breakpoint which will be visible as a red circle.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', {className: 'greeting'}, 'Hello, world!');
ReactDOM.render(element, document.getElementById('root'));
```

Tweet

Subscribe
Ask questions
Follow @code
Request features
Report issues
Watch videos


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:03 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

```
var element = React.createElement('h1', {className: 'greeting'}, 'Hello, world!');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```

Overview
SETUP
GET STARTED
USER GUIDE
LANGUAGES
NODE.JS / JAVASCRIPT
Node.js Tutorial
Node.js Debugging
Node.js Deployment
React Tutorial
Angular Tutorial
Debugging Recipes
Extensions
EXTENSION AUTHORING
EXTENSIBILITY REFERENCE
OTHER

Configure the Chrome debugger

We need to initially configure the debugger. To do so, go to the Debug view (⇧⌘D) and click on gear button to create a `launch.json` debugger configuration file. Choose **Chrome** from the **Select Environment** dropdown. This will create a `launch.json` file in a new `.vscode` folder in your project which includes configuration to both launch the website or attach to a running instance.

We need to make one change for our example: change the port from `8080` to `3000`. Your `launch.json` should look like this:

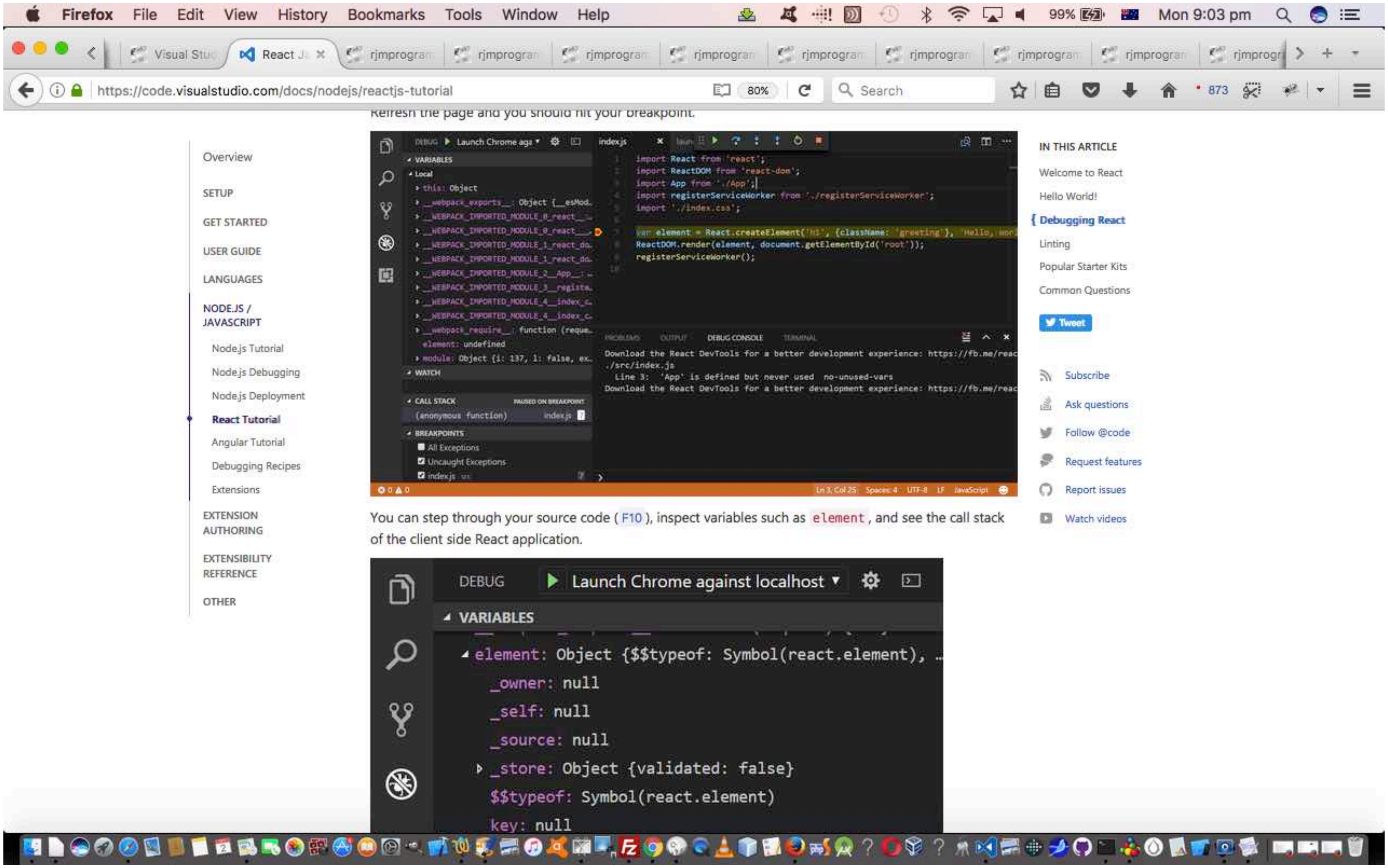
```
{
    "version": "0.2.0",
    "configurations": [
        {
            "type": "chrome",
            "request": "launch",
            "name": "Launch Chrome against localhost",
            "url": "http://localhost:3000",
            "webRoot": "${workspaceRoot}"
        },
        {
            "type": "chrome",
            "request": "attach",
            "name": "Attach to Chrome",
            "port": 9222,
            "webRoot": "${workspaceRoot}"
        }
    ]
}
```

Ensure that your development server is running ("npm start"). Then press `F5` or the green arrow to launch the debugger and open a new browser instance. The source code where the breakpoint is set runs on startup before the debugger was attached so we won't hit the breakpoint until we refresh the web page. Refresh the page and you should hit your breakpoint.

IN THIS ARTICLE
Welcome to React
Hello World!
Debugging React
Linting
Popular Starter Kits
Common Questions

Tweet

Subscribe
Ask questions
Follow @code
Request features
Report issues
Watch videos


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:03 pm
Visual Stu
React J
rjmprogram
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Refresh the page and you should hit your breakpoint.

```
this: Object
__webpack_exports__: Object {__esMod…
__WEBPACK_IMPORTED_MODULE_0_react__…
__WEBPACK_IMPORTED_MODULE_0_react___…
__WEBPACK_IMPORTED_MODULE_1_react_do…
__WEBPACK_IMPORTED_MODULE_1_react_do…
__WEBPACK_IMPORTED_MODULE_2__App__:…
__WEBPACK_IMPORTED_MODULE_3__registe…
__WEBPACK_IMPORTED_MODULE_4__index_c…
__WEBPACK_IMPORTED_MODULE_4__index_c…
__webpack_require__: function (reque…
element: undefined
module: Object {i: 137, l: false, ex…
```

```
(anonymous function)    index.js
```

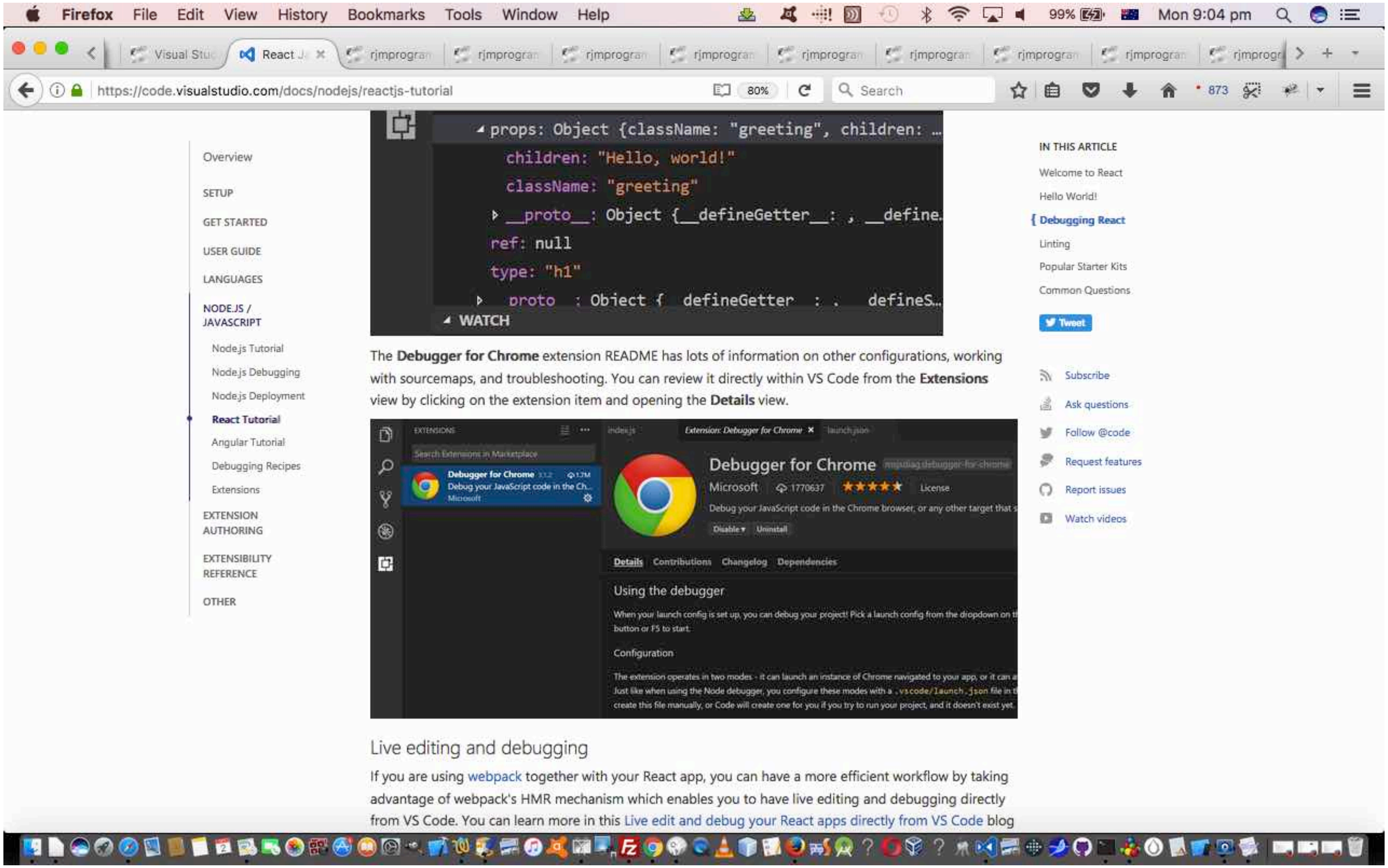
All Exceptions
Uncaught Exceptions
index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', {className: 'greeting'}, 'Hello, wor
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```

```
Download the React DevTools for a better development experience: https://fb.me/reac
./src/index.js
  Line 3:  'App' is defined but never used  no-unused-vars
Download the React DevTools for a better development experience: https://fb.me/reac
```

You can step through your source code (`F10`), inspect variables such as `element` , and see the call stack of the client side React application.

```
element: Object {$$typeof: Symbol(react.element), …
  _owner: null
  _self: null
  _source: null
  _store: Object {validated: false}
  $$typeof: Symbol(react.element)
  key: null
```


Firefox File Edit View History Bookmarks Tools Window Help
99% Mon 9:04 pm
Visual Stu
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873

Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS / JAVASCRIPT

Node.js Tutorial

Node.js Debugging

Node.js Deployment

React Tutorial

Angular Tutorial

Debugging Recipes

Extensions

EXTENSION AUTHORING

EXTENSIBILITY REFERENCE

OTHER

```
  ◢ props: Object {className: "greeting", children: …
      children: "Hello, world!"
      className: "greeting"
    ▶ __proto__: Object {__defineGetter__: , __define…
    ref: null
    type: "h1"
  ▶ __proto__: Object {__defineGetter__: , __defineS…
◢ WATCH
```

The **Debugger for Chrome** extension README has lots of information on other configurations, working with sourcemaps, and troubleshooting. You can review it directly within VS Code from the **Extensions** view by clicking on the extension item and opening the **Details** view.

EXTENSIONS
Search Extensions in Marketplace
Debugger for Chrome 3.1.2 1.7M
Debug your JavaScript code in the Ch...
Microsoft
index.js
Extension: Debugger for Chrome
launch.json
Debugger for Chrome msjsdiag.debugger-for-chrome
Microsoft 1770637 ★★★★★ License
Debug your JavaScript code in the Chrome browser, or any other target that s
Disable ▾ Uninstall
Details Contributions Changelog Dependencies

Using the debugger

When your launch config is set up, you can debug your project! Pick a launch config from the dropdown on t
button or F5 to start.

Configuration

The extension operates in two modes - it can launch an instance of Chrome navigated to your app, or it can a
Just like when using the Node debugger, you configure these modes with a `.vscode/launch.json` file in t
create this file manually, or Code will create one for you if you try to run your project, and it doesn't exist yet.

Live editing and debugging

If you are using webpack together with your React app, you can have a more efficient workflow by taking advantage of webpack's HMR mechanism which enables you to have live editing and debugging directly from VS Code. You can learn more in this Live edit and debug your React apps directly from VS Code blog

IN THIS ARTICLE

Welcome to React

Hello World!

Debugging React

Linting

Popular Starter Kits

Common Questions

Tweet

Subscribe

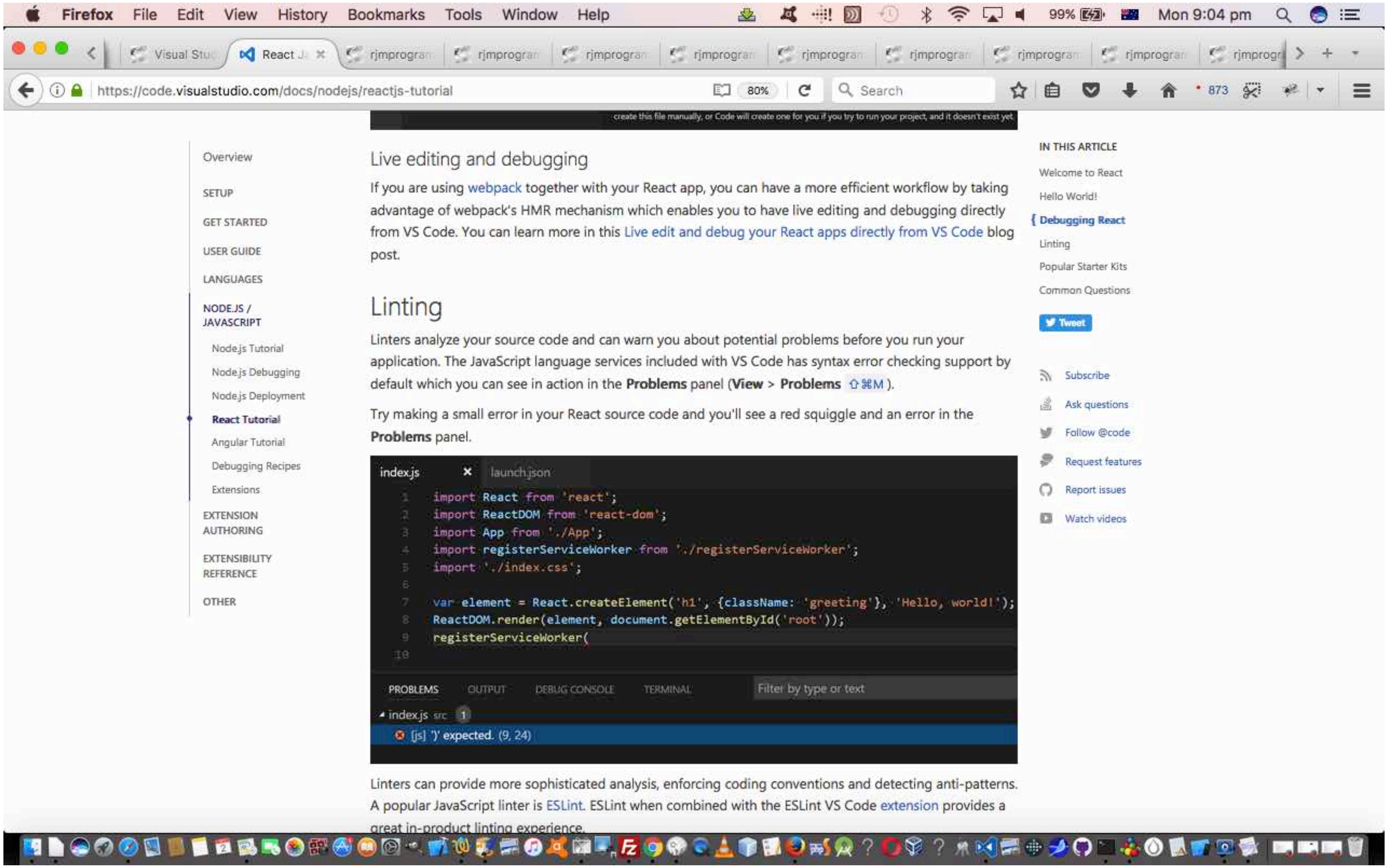
Ask questions

Follow @code

Request features

Report issues

Watch videos



Firefox File Edit View History Bookmarks Tools Window Help 99% Mon 9:04 pm

https://code.visualstudio.com/docs/nodejs/reactjs-tutorial

create this file manually, or Code will create one for you if you try to run your project, and it doesn't exist yet.

Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS / JAVASCRIPT

Node.js Tutorial

Node.js Debugging

Node.js Deployment

React Tutorial

Angular Tutorial

Debugging Recipes

Extensions

EXTENSION AUTHORING

EXTENSIBILITY REFERENCE

OTHER

Live editing and debugging

If you are using webpack together with your React app, you can have a more efficient workflow by taking advantage of webpack's HMR mechanism which enables you to have live editing and debugging directly from VS Code. You can learn more in this Live edit and debug your React apps directly from VS Code blog post.

Linting

Linters analyze your source code and can warn you about potential problems before you run your application. The JavaScript language services included with VS Code has syntax error checking support by default which you can see in action in the **Problems** panel (**View** > **Problems** ⇧⌘M).

Try making a small error in your React source code and you'll see a red squiggle and an error in the **Problems** panel.

index.js launch.json

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', {className: 'greeting'}, 'Hello, world!');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker(
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Filter by type or text

```
index.js src 1
  [js] ')' expected. (9, 24)
```

Linters can provide more sophisticated analysis, enforcing coding conventions and detecting anti-patterns. A popular JavaScript linter is ESLint. ESLint when combined with the ESLint VS Code extension provides a great in-product linting experience.

IN THIS ARTICLE

Welcome to React

Hello World!

Debugging React

Linting

Popular Starter Kits

Common Questions

Tweet

Subscribe

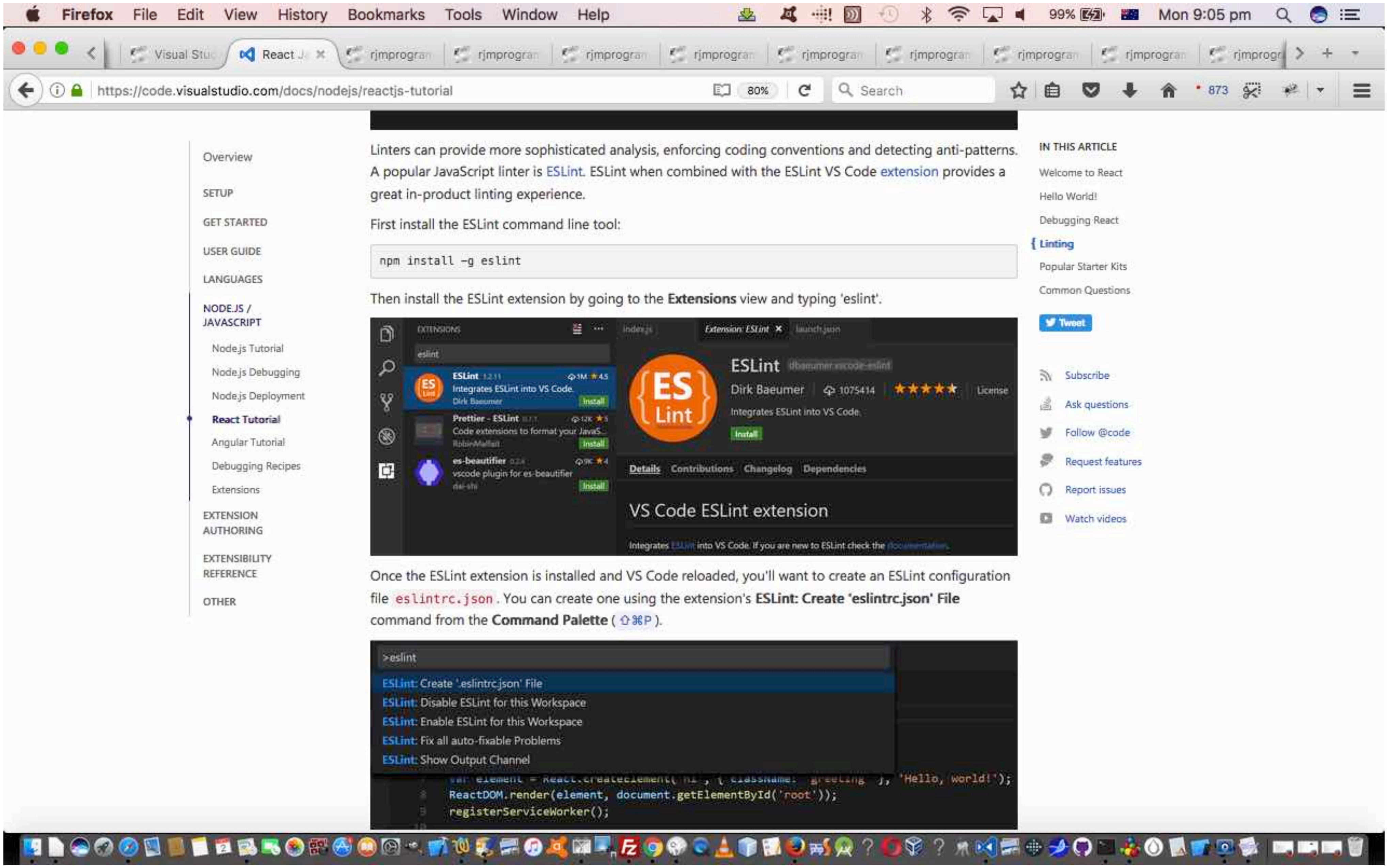
Ask questions

Follow @code

Request features

Report issues

Watch videos



Firefox File Edit View History Bookmarks Tools Window Help 99% Mon 9:05 pm

https://code.visualstudio.com/docs/nodejs/reactjs-tutorial

Overview
SETUP
GET STARTED
USER GUIDE
LANGUAGES
NODE.JS / JAVASCRIPT
Node.js Tutorial
Node.js Debugging
Node.js Deployment
React Tutorial
Angular Tutorial
Debugging Recipes
Extensions
EXTENSION AUTHORING
EXTENSIBILITY REFERENCE
OTHER

Linters can provide more sophisticated analysis, enforcing coding conventions and detecting anti-patterns. A popular JavaScript linter is ESLint. ESLint when combined with the ESLint VS Code extension provides a great in-product linting experience.

First install the ESLint command line tool:

```
npm install -g eslint
```

Then install the ESLint extension by going to the **Extensions** view and typing 'eslint'.

EXTENSIONS
eslint
ESLint 1.2.11 — Integrates ESLint into VS Code. — Dirk Baeumer — Install
Prettier - ESLint — Code extensions to format your JavaS... — RobinMalfait — Install
es-beautifier — vscode plugin for es-beautifier — dai-shi — Install

Extension: ESLint
ESLint
Dirk Baeumer | 1075414 | ★★★★★ | License
Integrates ESLint into VS Code.
Install
Details Contributions Changelog Dependencies
VS Code ESLint extension
Integrates ESLint into VS Code. If you are new to ESLint check the documentation.

Once the ESLint extension is installed and VS Code reloaded, you'll want to create an ESLint configuration file `eslintrc.json`. You can create one using the extension's **ESLint: Create 'eslintrc.json' File** command from the **Command Palette** (⇧⌘P).

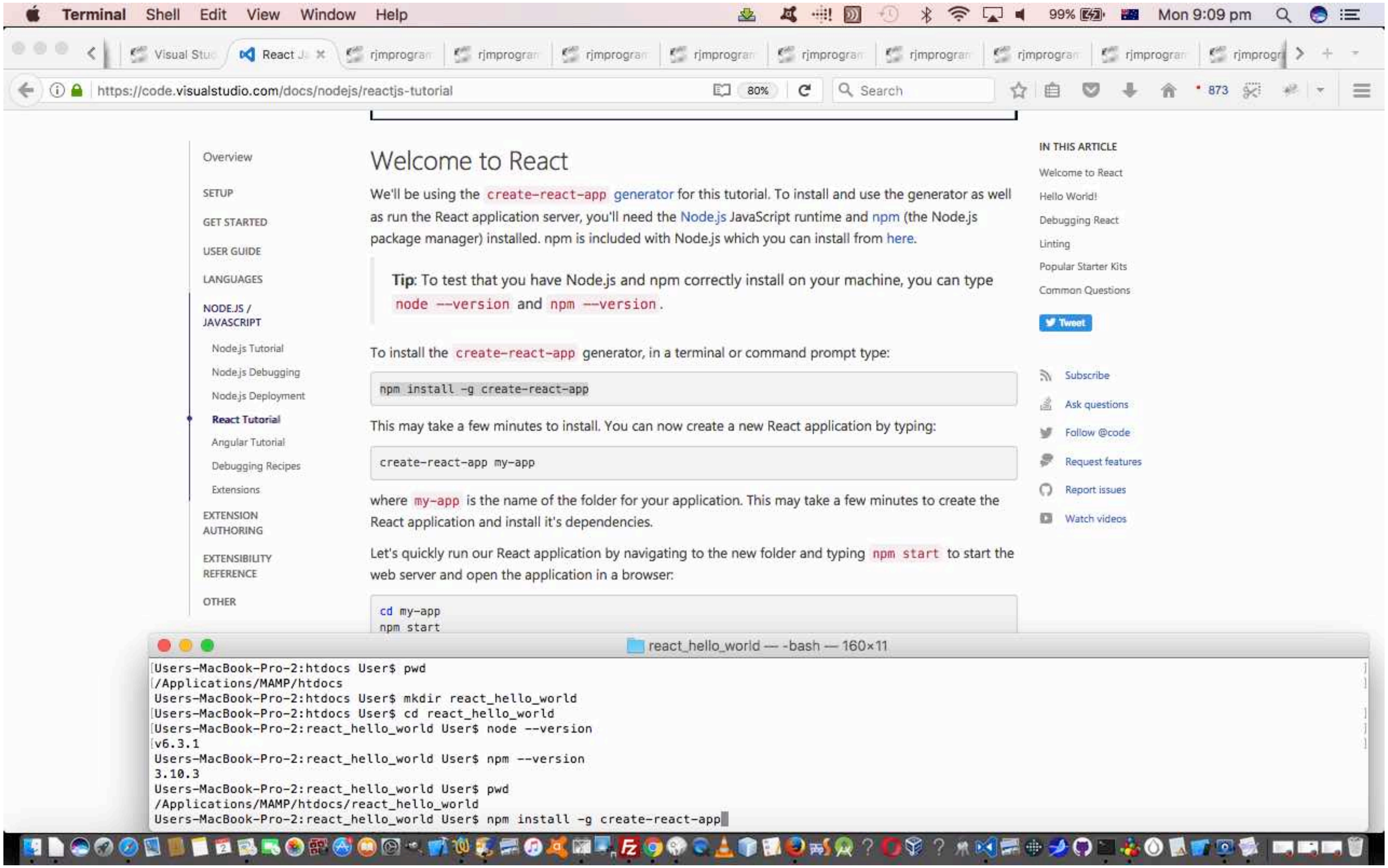
```
>eslint
ESLint: Create '.eslintrc.json' File
ESLint: Disable ESLint for this Workspace
ESLint: Enable ESLint for this Workspace
ESLint: Fix all auto-fixable Problems
ESLint: Show Output Channel
```

```
var element = React.createElement('h1', { className: 'greeting' }, 'Hello, world!');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```

IN THIS ARTICLE
Welcome to React
Hello World!
Debugging React
Linting
Popular Starter Kits
Common Questions

Tweet

Subscribe
Ask questions
Follow @code
Request features
Report issues
Watch videos



Welcome to React

We'll be using the `create-react-app` generator for this tutorial. To install and use the generator as well as run the React application server, you'll need the Node.js JavaScript runtime and npm (the Node.js package manager) installed. npm is included with Node.js which you can install from here.

> **Tip**: To test that you have Node.js and npm correctly install on your machine, you can type `node --version` and `npm --version`.

To install the `create-react-app` generator, in a terminal or command prompt type:

```
npm install -g create-react-app
```

This may take a few minutes to install. You can now create a new React application by typing:

```
create-react-app my-app
```

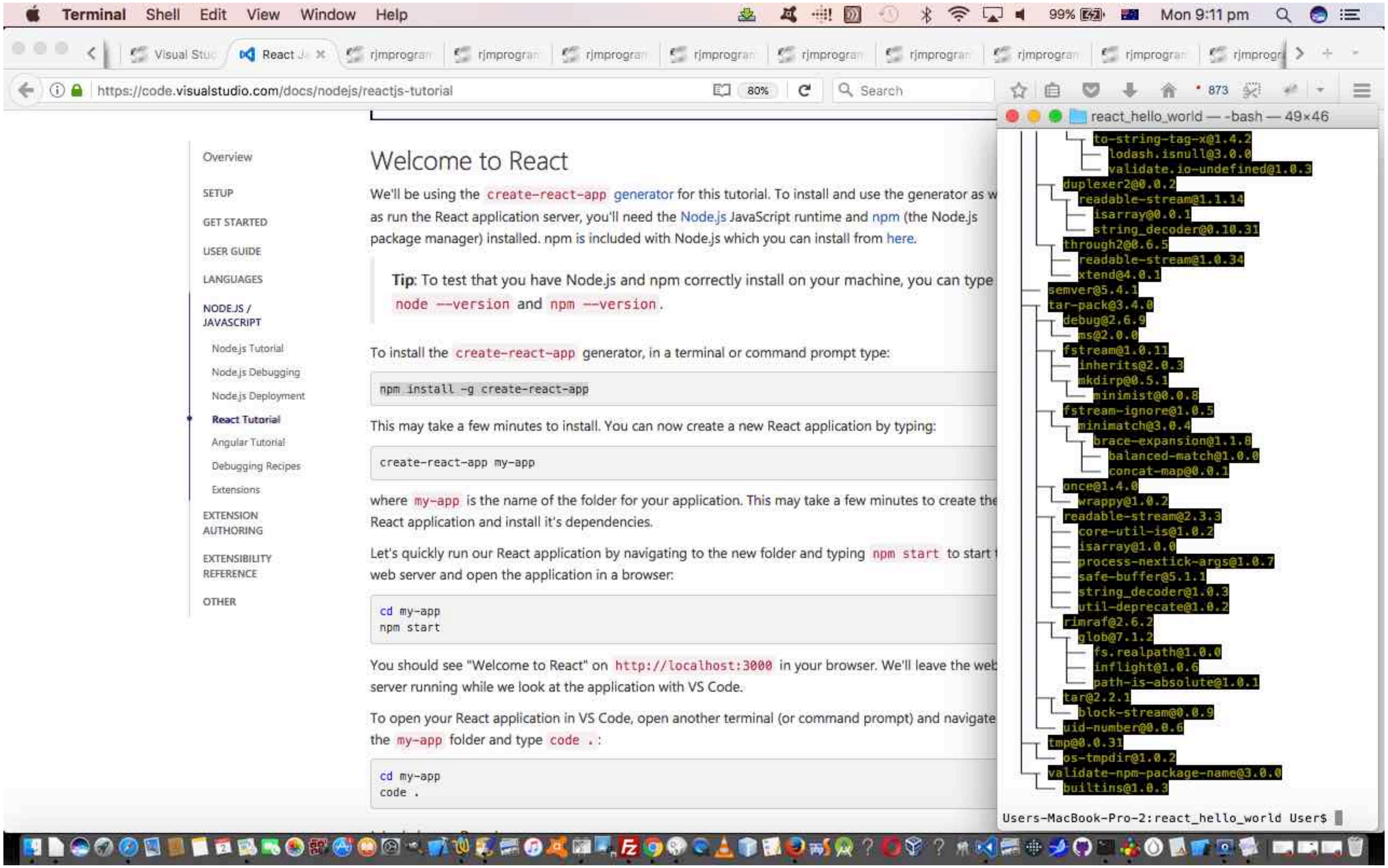
where `my-app` is the name of the folder for your application. This may take a few minutes to create the React application and install it's dependencies.

Let's quickly run our React application by navigating to the new folder and typing `npm start` to start the web server and open the application in a browser:

```
cd my-app
npm start
```

react_hello_world — -bash — 160×11

```
Users-MacBook-Pro-2:htdocs User$ pwd
/Applications/MAMP/htdocs
Users-MacBook-Pro-2:htdocs User$ mkdir react_hello_world
Users-MacBook-Pro-2:htdocs User$ cd react_hello_world
Users-MacBook-Pro-2:react_hello_world User$ node --version
v6.3.1
Users-MacBook-Pro-2:react_hello_world User$ npm --version
3.10.3
Users-MacBook-Pro-2:react_hello_world User$ pwd
/Applications/MAMP/htdocs/react_hello_world
Users-MacBook-Pro-2:react_hello_world User$ npm install -g create-react-app
```



Welcome to React

We'll be using the `create-react-app` generator for this tutorial. To install and use the generator as w
as run the React application server, you'll need the Node.js JavaScript runtime and npm (the Node.js
package manager) installed. npm is included with Node.js which you can install from here.

> **Tip**: To test that you have Node.js and npm correctly install on your machine, you can type
> `node --version` and `npm --version`.

To install the `create-react-app` generator, in a terminal or command prompt type:

```
npm install -g create-react-app
```

This may take a few minutes to install. You can now create a new React application by typing:

```
create-react-app my-app
```

where `my-app` is the name of the folder for your application. This may take a few minutes to create the
React application and install it's dependencies.

Let's quickly run our React application by navigating to the new folder and typing `npm start` to start
web server and open the application in a browser:

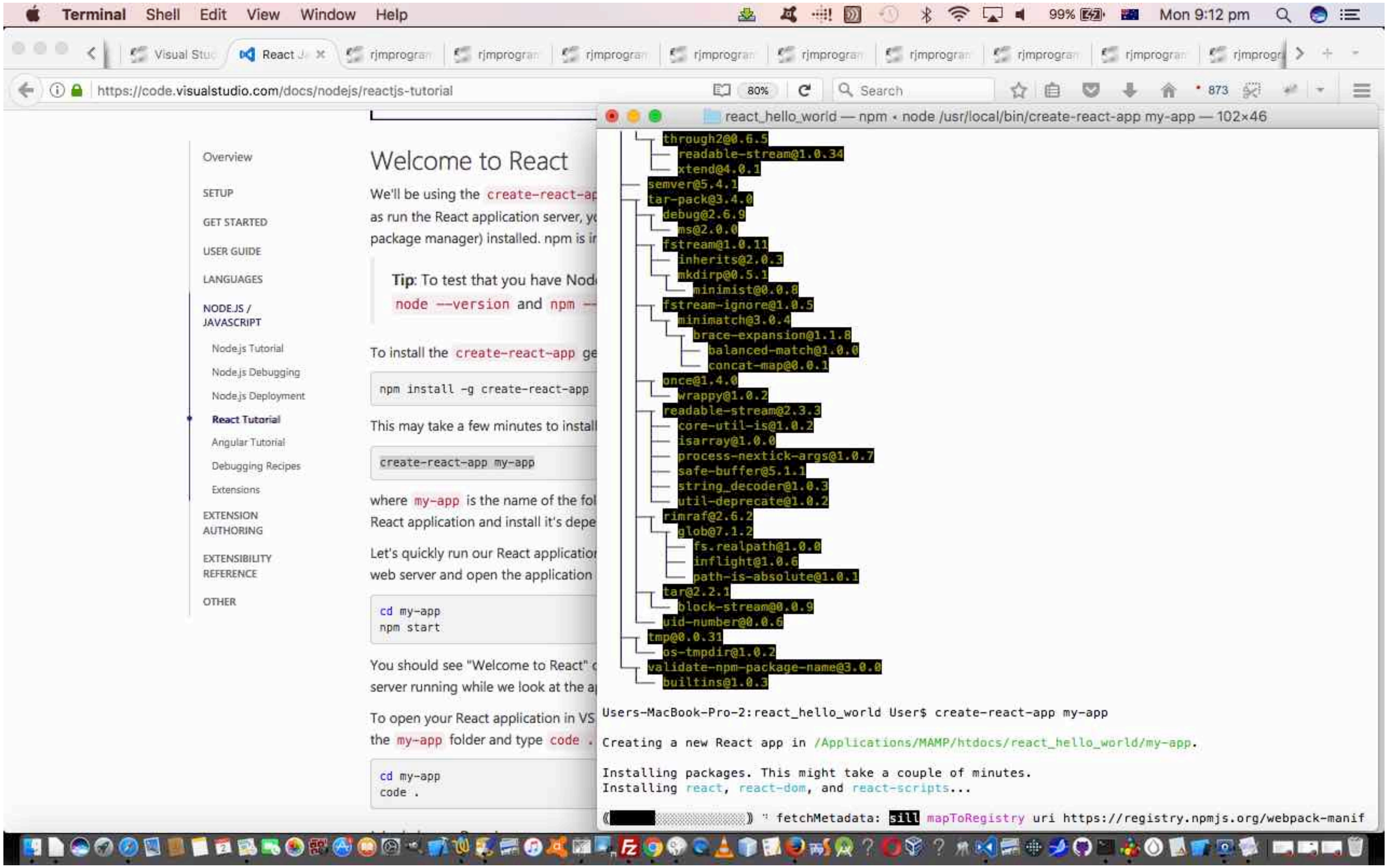
```
cd my-app
npm start
```

You should see "Welcome to React" on `http://localhost:3000` in your browser. We'll leave the web
server running while we look at the application with VS Code.

To open your React application in VS Code, open another terminal (or command prompt) and navigate
the `my-app` folder and type `code .`:

```
cd my-app
code .
```

```
│ │   └─┬ to-string-tag-x@1.4.2
│ │     ├── lodash.isnull@3.0.0
│ │     └── validate.io-undefined@1.0.3
│ ├─┬ duplexer2@0.0.2
│ │ └─┬ readable-stream@1.1.14
│ │   ├── isarray@0.0.1
│ │   └── string_decoder@0.10.31
│ └─┬ through2@0.6.5
│   ├── readable-stream@1.0.34
│   └── xtend@4.0.1
├── semver@5.4.1
├─┬ tar-pack@3.4.0
│ ├─┬ debug@2.6.9
│ │ └── ms@2.0.0
│ ├─┬ fstream@1.0.11
│ │ ├── inherits@2.0.3
│ │ └─┬ mkdirp@0.5.1
│ │   └── minimist@0.0.8
│ ├─┬ fstream-ignore@1.0.5
│ │ └─┬ minimatch@3.0.4
│ │   └─┬ brace-expansion@1.1.8
│ │     ├── balanced-match@1.0.0
│ │     └── concat-map@0.0.1
│ ├─┬ once@1.4.0
│ │ └── wrappy@1.0.2
│ ├─┬ readable-stream@2.3.3
│ │ ├── core-util-is@1.0.2
│ │ ├── isarray@1.0.0
│ │ ├── process-nextick-args@1.0.7
│ │ ├── safe-buffer@5.1.1
│ │ ├── string_decoder@1.0.3
│ │ └── util-deprecate@1.0.2
│ ├─┬ rimraf@2.6.2
│ │ └─┬ glob@7.1.2
│ │   ├── fs.realpath@1.0.0
│ │   ├── inflight@1.0.6
│ │   └── path-is-absolute@1.0.1
│ ├─┬ tar@2.2.1
│ │ └── block-stream@0.0.9
│ └── uid-number@0.0.6
├─┬ tmp@0.0.31
│ └── os-tmpdir@1.0.2
└─┬ validate-npm-package-name@3.0.0
  └── builtins@1.0.3

Users-MacBook-Pro-2:react_hello_world User$
```


Terminal Shell Edit View Window Help
99%
Mon 9:12 pm
Visual Stuc
React J
rjmprogran
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873
react_hello_world — npm • node /usr/local/bin/create-react-app my-app — 102×46

Overview
SETUP
GET STARTED
USER GUIDE
LANGUAGES
NODE.JS / JAVASCRIPT
Node.js Tutorial
Node.js Debugging
Node.js Deployment
React Tutorial
Angular Tutorial
Debugging Recipes
Extensions
EXTENSION AUTHORING
EXTENSIBILITY REFERENCE
OTHER

Welcome to React

We'll be using the `create-react-ap` as run the React application server, yo package manager) installed. npm is in

Tip: To test that you have Nod `node --version` and `npm --`

To install the `create-react-app` ge

```
npm install -g create-react-app
```

This may take a few minutes to install

```
create-react-app my-app
```

where `my-app` is the name of the fol React application and install it's depe

Let's quickly run our React applicatio web server and open the application

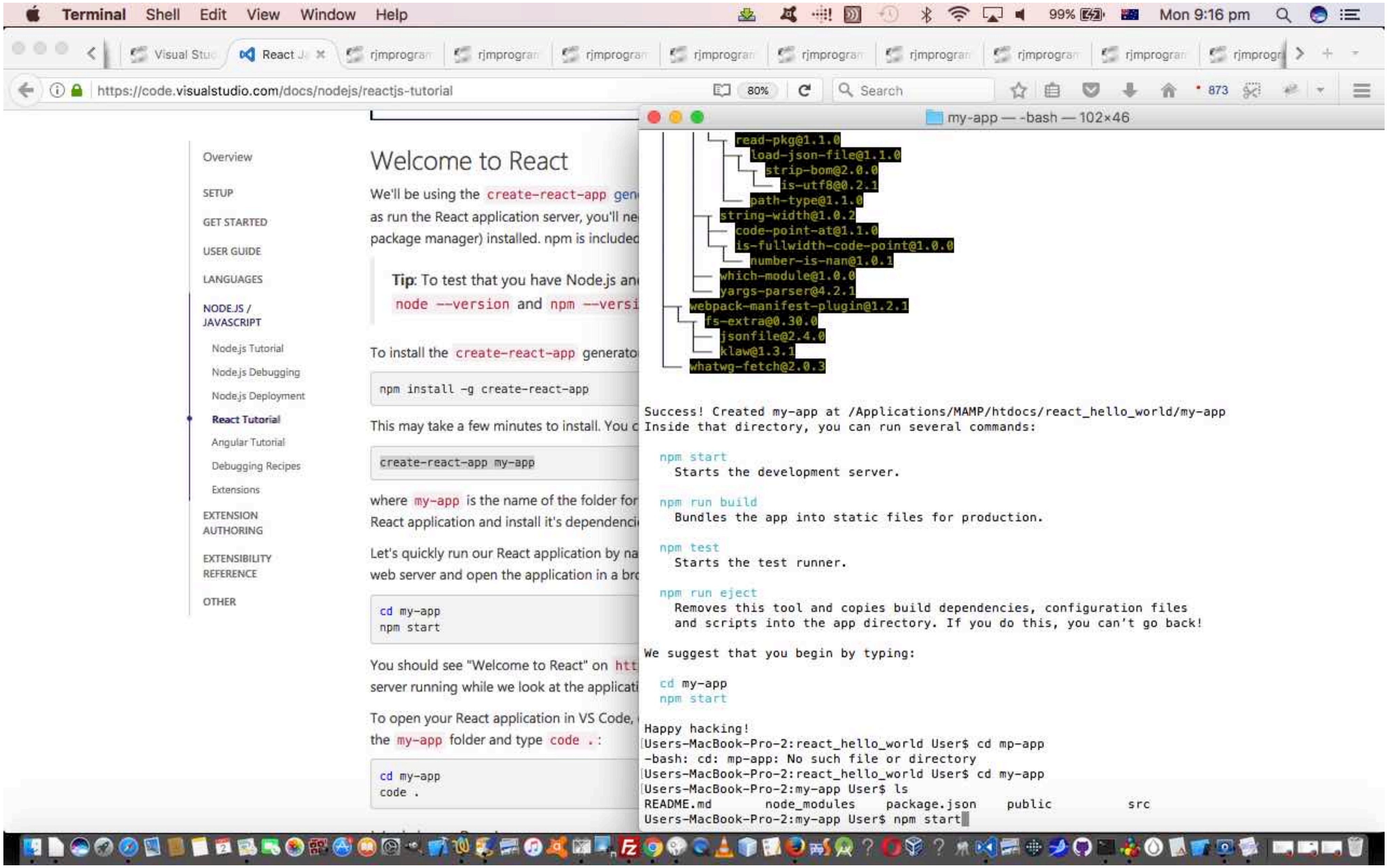
```
cd my-app
npm start
```

You should see "Welcome to React" c server running while we look at the a

To open your React application in VS the `my-app` folder and type `code .`

```
cd my-app
code .
```

```
│ └─┬ through2@0.6.5
│   ├── readable-stream@1.0.34
│   └── xtend@4.0.1
├── semver@5.4.1
├─┬ tar-pack@3.4.0
│ ├─┬ debug@2.6.9
│ │ └── ms@2.0.0
│ ├─┬ fstream@1.0.11
│ │ ├── inherits@2.0.3
│ │ └─┬ mkdirp@0.5.1
│ │   └── minimist@0.0.8
│ ├─┬ fstream-ignore@1.0.5
│ │ └─┬ minimatch@3.0.4
│ │   └─┬ brace-expansion@1.1.8
│ │     ├── balanced-match@1.0.0
│ │     └── concat-map@0.0.1
│ ├─┬ once@1.4.0
│ │ └── wrappy@1.0.2
│ ├─┬ readable-stream@2.3.3
│ │ ├── core-util-is@1.0.2
│ │ ├── isarray@1.0.0
│ │ ├── process-nextick-args@1.0.7
│ │ ├── safe-buffer@5.1.1
│ │ ├── string_decoder@1.0.3
│ │ └── util-deprecate@1.0.2
│ ├─┬ rimraf@2.6.2
│ │ └─┬ glob@7.1.2
│ │   ├── fs.realpath@1.0.0
│ │   ├── inflight@1.0.6
│ │   └── path-is-absolute@1.0.1
│ ├─┬ tar@2.2.1
│ │ └── block-stream@0.0.9
│ └── uid-number@0.0.6
├─┬ tmp@0.0.31
│ └── os-tmpdir@1.0.2
└─┬ validate-npm-package-name@3.0.0
  └── builtins@1.0.3

Users-MacBook-Pro-2:react_hello_world User$ create-react-app my-app

Creating a new React app in /Applications/MAMP/htdocs/react_hello_world/my-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

(█████░░░░░░░░░░░░░░░) ⠋ fetchMetadata: sill mapToRegistry uri https://registry.npmjs.org/webpack-manif
```



Welcome to React

We'll be using the `create-react-app` gen
as run the React application server, you'll ne
package manager) installed. npm is included

Tip: To test that you have Node.js an
`node --version` and `npm --versi`

To install the `create-react-app` generato

```
npm install -g create-react-app
```

This may take a few minutes to install. You c

```
create-react-app my-app
```

where `my-app` is the name of the folder for
React application and install it's dependenci

Let's quickly run our React application by na
web server and open the application in a bro

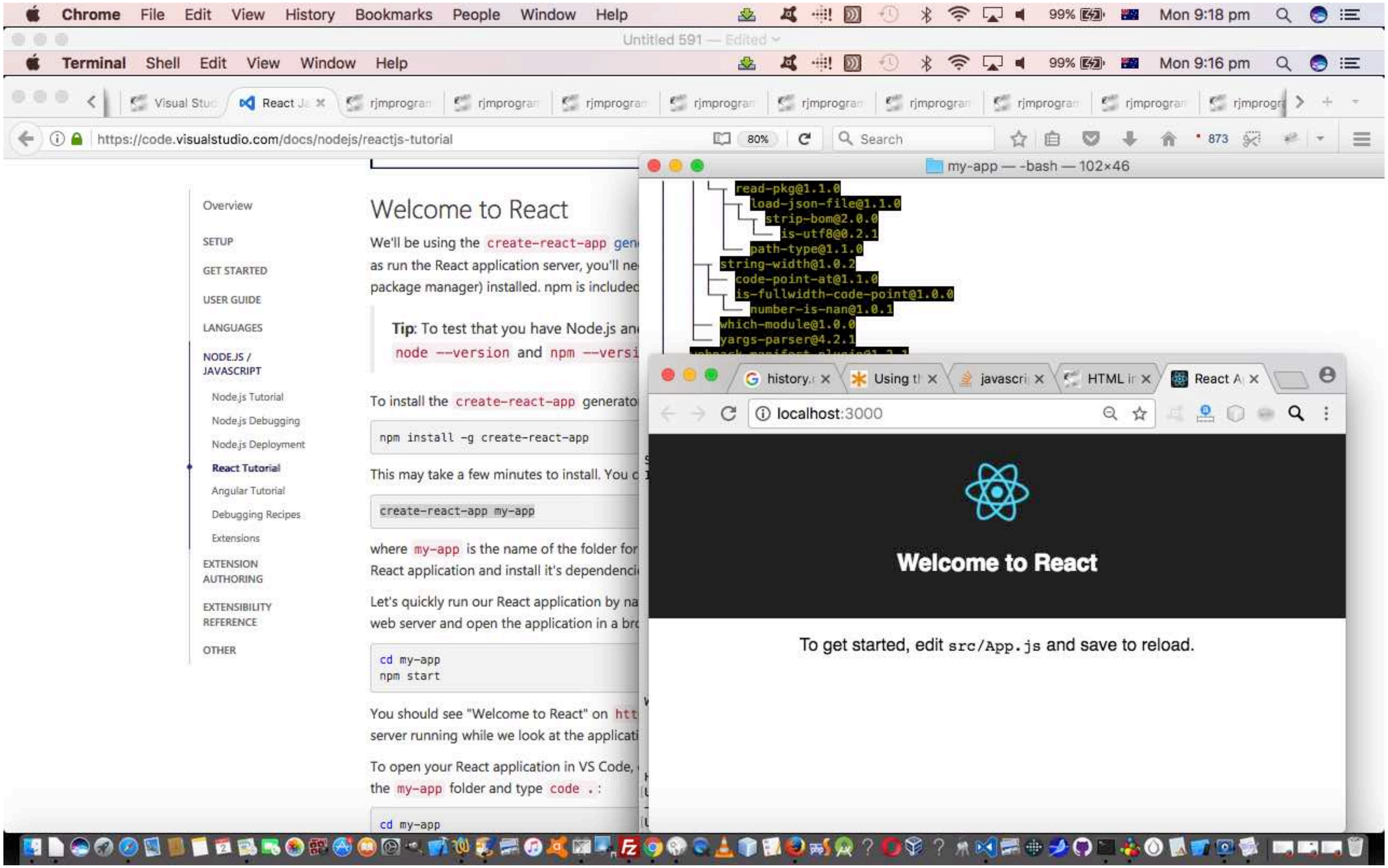
```
cd my-app
npm start
```

You should see "Welcome to React" on `htt`
server running while we look at the applicati

To open your React application in VS Code,
the `my-app` folder and type `code .`:

```
cd my-app
code .
```

```
  │  └─ read-pkg@1.1.0
  │     ├─┬ load-json-file@1.1.0
  │     │ └─┬ strip-bom@2.0.0
  │     │   └── is-utf8@0.2.1
  │     └── path-type@1.1.0
  │  ├─┬ string-width@1.0.2
  │  │ ├── code-point-at@1.1.0
  │  │ ├─┬ is-fullwidth-code-point@1.0.0
  │  │ │ └── number-is-nan@1.0.1
  │  ├── which-module@1.0.0
  │  └── yargs-parser@4.2.1
  ├─┬ webpack-manifest-plugin@1.2.1
  │ ├─┬ fs-extra@0.30.0
  │ │ ├── jsonfile@2.4.0
  │ │ └── klaw@1.3.1
  └── whatwg-fetch@2.0.3


Success! Created my-app at /Applications/MAMP/htdocs/react_hello_world/my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!
Users-MacBook-Pro-2:react_hello_world User$ cd mp-app
-bash: cd: mp-app: No such file or directory
Users-MacBook-Pro-2:react_hello_world User$ cd my-app
Users-MacBook-Pro-2:my-app User$ ls
README.md        node_modules     package.json     public           src
Users-MacBook-Pro-2:my-app User$ npm start
```



Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS / JAVASCRIPT

Node.js Tutorial

Node.js Debugging

Node.js Deployment

React Tutorial

Angular Tutorial

Debugging Recipes

Extensions

EXTENSION AUTHORING

EXTENSIBILITY REFERENCE

OTHER

Welcome to React

We'll be using the `create-react-app` gen
as run the React application server, you'll ne
package manager) installed. npm is included

Tip: To test that you have Node.js an
`node --version` and `npm --versi`

To install the `create-react-app` generato

```
npm install -g create-react-app
```

This may take a few minutes to install. You c

```
create-react-app my-app
```

where `my-app` is the name of the folder for
React application and install it's dependenci

Let's quickly run our React application by na
web server and open the application in a br

```
cd my-app
npm start
```

You should see "Welcome to React" on `htt`
server running while we look at the applicati

To open your React application in VS Code,
the `my-app` folder and type `code .`:

```
cd my-app
```

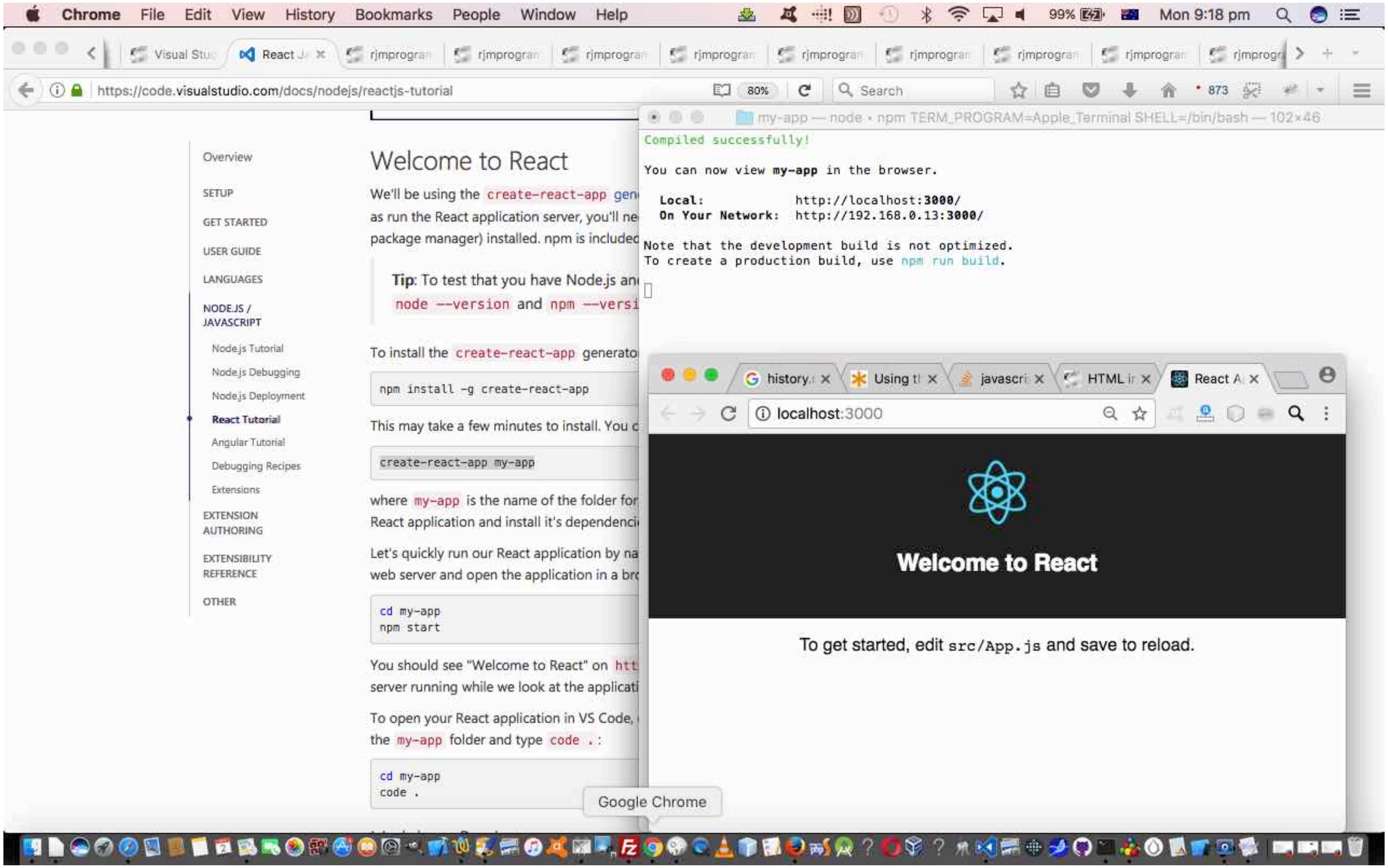
```
my-app — -bash — 102×46
  └─ read-pkg@1.1.0
     ├─ load-json-file@1.1.0
     │  └─ strip-bom@2.0.0
     │     └─ is-utf8@0.2.1
     └─ path-type@1.1.0
  string-width@1.0.2
  ├─ code-point-at@1.1.0
  ├─ is-fullwidth-code-point@1.0.0
  └─ number-is-nan@1.0.1
  which-module@1.0.0
  yargs-parser@4.2.1
```

localhost:3000

Welcome to React

To get started, edit `src/App.js` and save to reload.

Chrome File Edit View History Bookmarks People Window Help
99% Mon 9:18 pm
Untitled 591 — Edited
Terminal Shell Edit View Window Help
99% Mon 9:16 pm
Visual Stud
React Js
rjmprogram
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial
80%
Search
873
history.
Using th
javascri
HTML in
React A



Welcome to React

We'll be using the `create-react-app` gen
as run the React application server, you'll ne
package manager) installed. npm is included

Tip: To test that you have Node.js an
`node --version` and `npm --versi`

To install the `create-react-app` generato

```
npm install -g create-react-app
```

This may take a few minutes to install. You c

```
create-react-app my-app
```

where `my-app` is the name of the folder for
React application and install it's dependenci

Let's quickly run our React application by na
web server and open the application in a bro

```
cd my-app
npm start
```

You should see "Welcome to React" on `htt`
server running while we look at the applicati

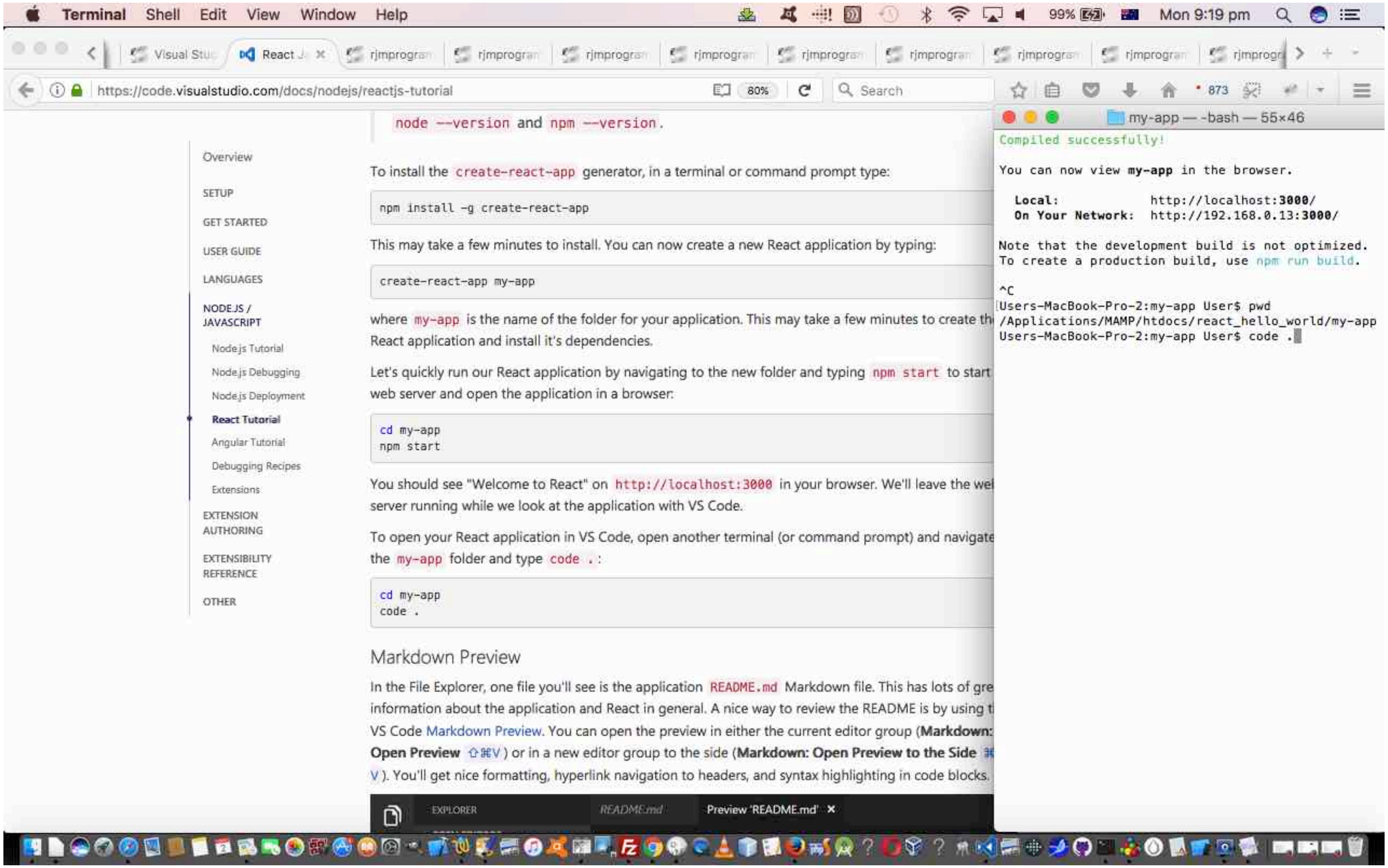
To open your React application in VS Code,
the `my-app` folder and type `code .`:

```
cd my-app
code .
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.13:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Welcome to React

To get started, edit `src/App.js` and save to reload.



`node --version` and `npm --version`.

To install the `create-react-app` generator, in a terminal or command prompt type:

```
npm install -g create-react-app
```

This may take a few minutes to install. You can now create a new React application by typing:

```
create-react-app my-app
```

where `my-app` is the name of the folder for your application. This may take a few minutes to create th
React application and install it's dependencies.

Let's quickly run our React application by navigating to the new folder and typing `npm start` to start
web server and open the application in a browser:

```
cd my-app
npm start
```

You should see "Welcome to React" on `http://localhost:3000` in your browser. We'll leave the we
server running while we look at the application with VS Code.

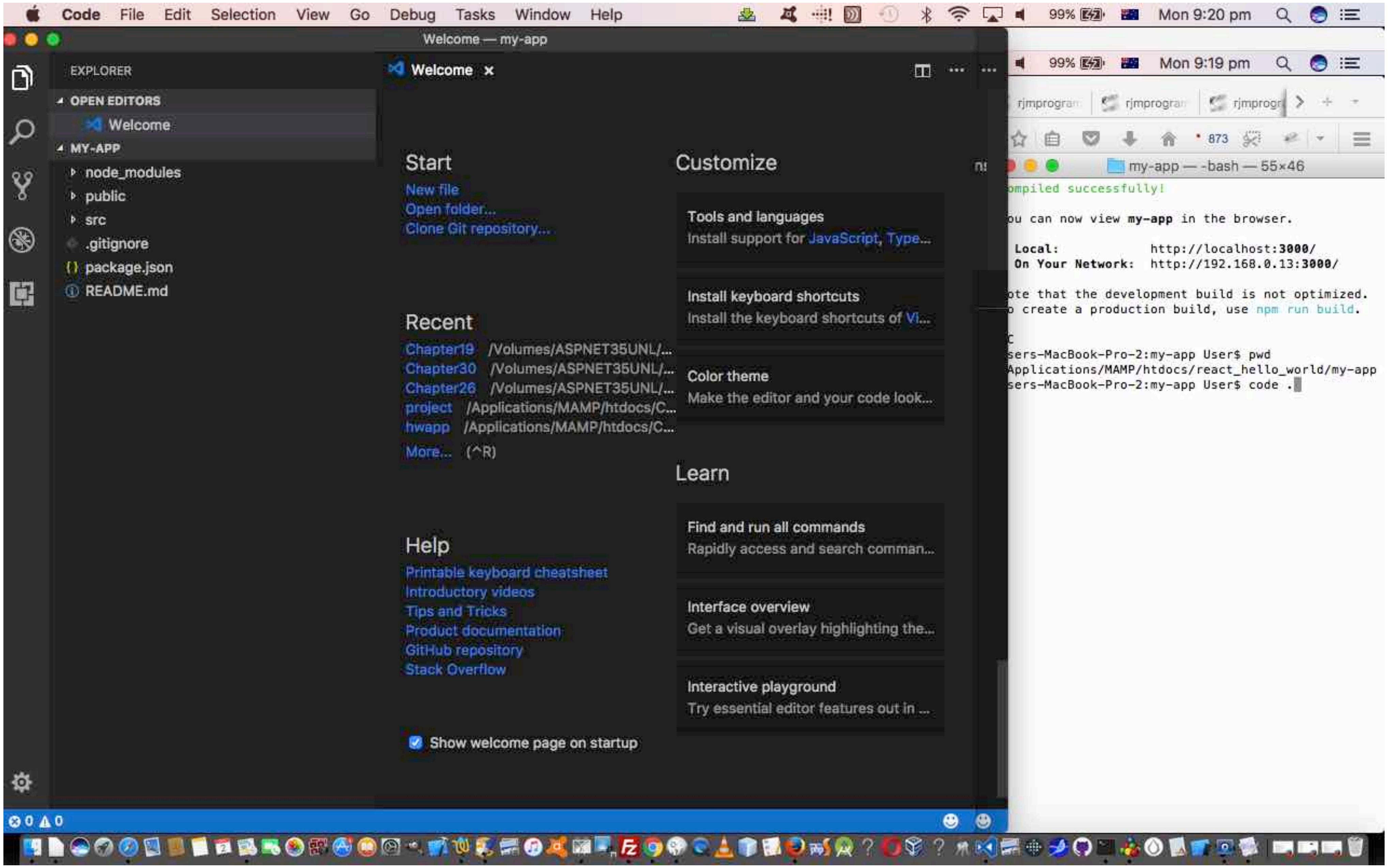
To open your React application in VS Code, open another terminal (or command prompt) and navigate
the `my-app` folder and type `code .`:

```
cd my-app
code .
```

Markdown Preview

In the File Explorer, one file you'll see is the application `README.md` Markdown file. This has lots of gre
information about the application and React in general. A nice way to review the README is by using t
VS Code Markdown Preview. You can open the preview in either the current editor group (**Markdown:
Open Preview** ⇧⌘V) or in a new editor group to the side (**Markdown: Open Preview to the Side** ⌘
V). You'll get nice formatting, hyperlink navigation to headers, and syntax highlighting in code blocks.

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.13:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.

^C
Users-MacBook-Pro-2:my-app User$ pwd
/Applications/MAMP/htdocs/react_hello_world/my-app
Users-MacBook-Pro-2:my-app User$ code .
```



Start
New file
Open folder...
Clone Git repository...

Recent
Chapter19 /Volumes/ASPNET35UNL/...
Chapter30 /Volumes/ASPNET35UNL/...
Chapter26 /Volumes/ASPNET35UNL/...
project /Applications/MAMP/htdocs/C...
hwapp /Applications/MAMP/htdocs/C...
More... (^R)

Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow

✓ Show welcome page on startup

Customize

Tools and languages
Install support for JavaScript, Type...

Install keyboard shortcuts
Install the keyboard shortcuts of Vi...

Color theme
Make the editor and your code look...

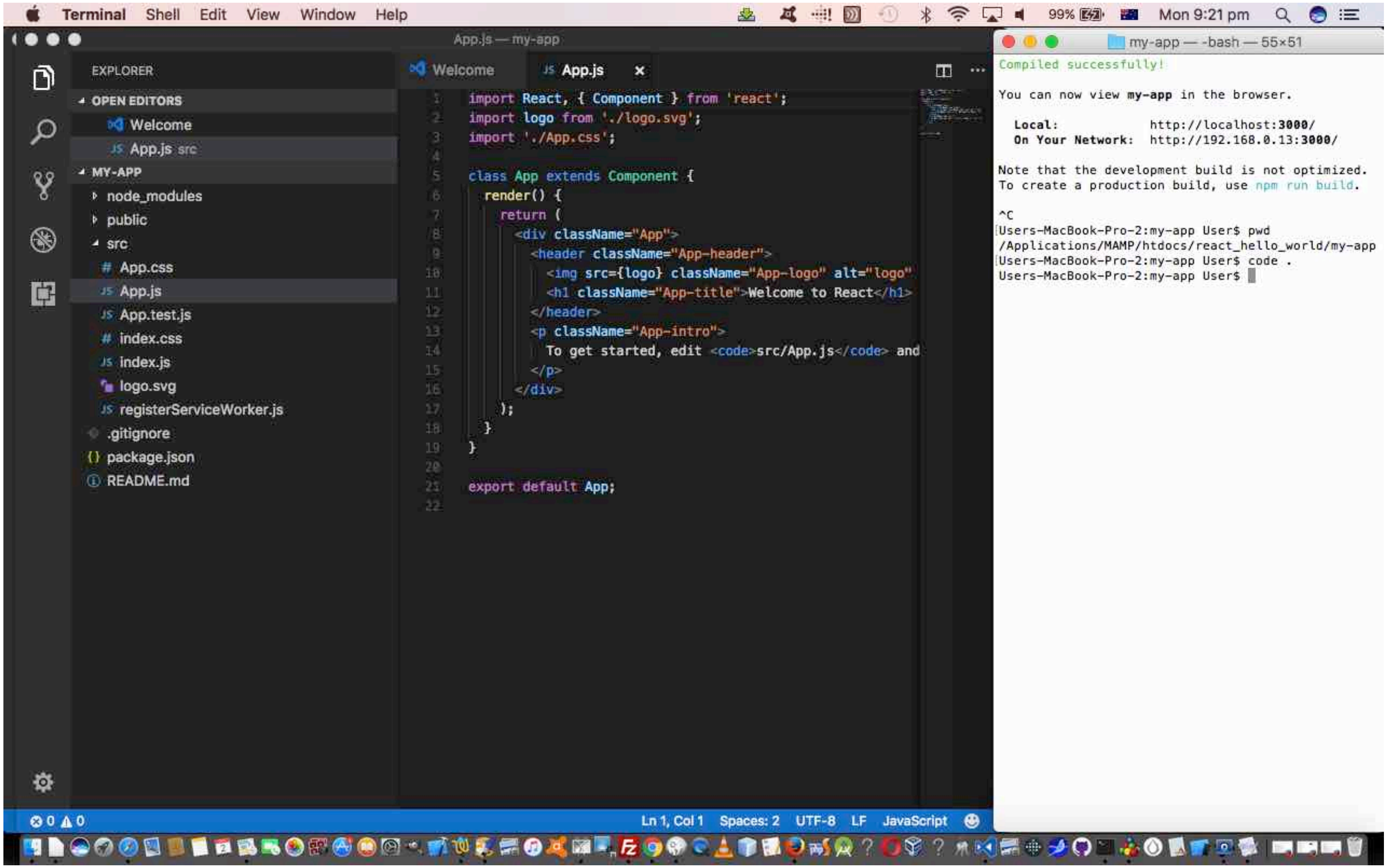
Learn

Find and run all commands
Rapidly access and search comman...

Interface overview
Get a visual overlay highlighting the...

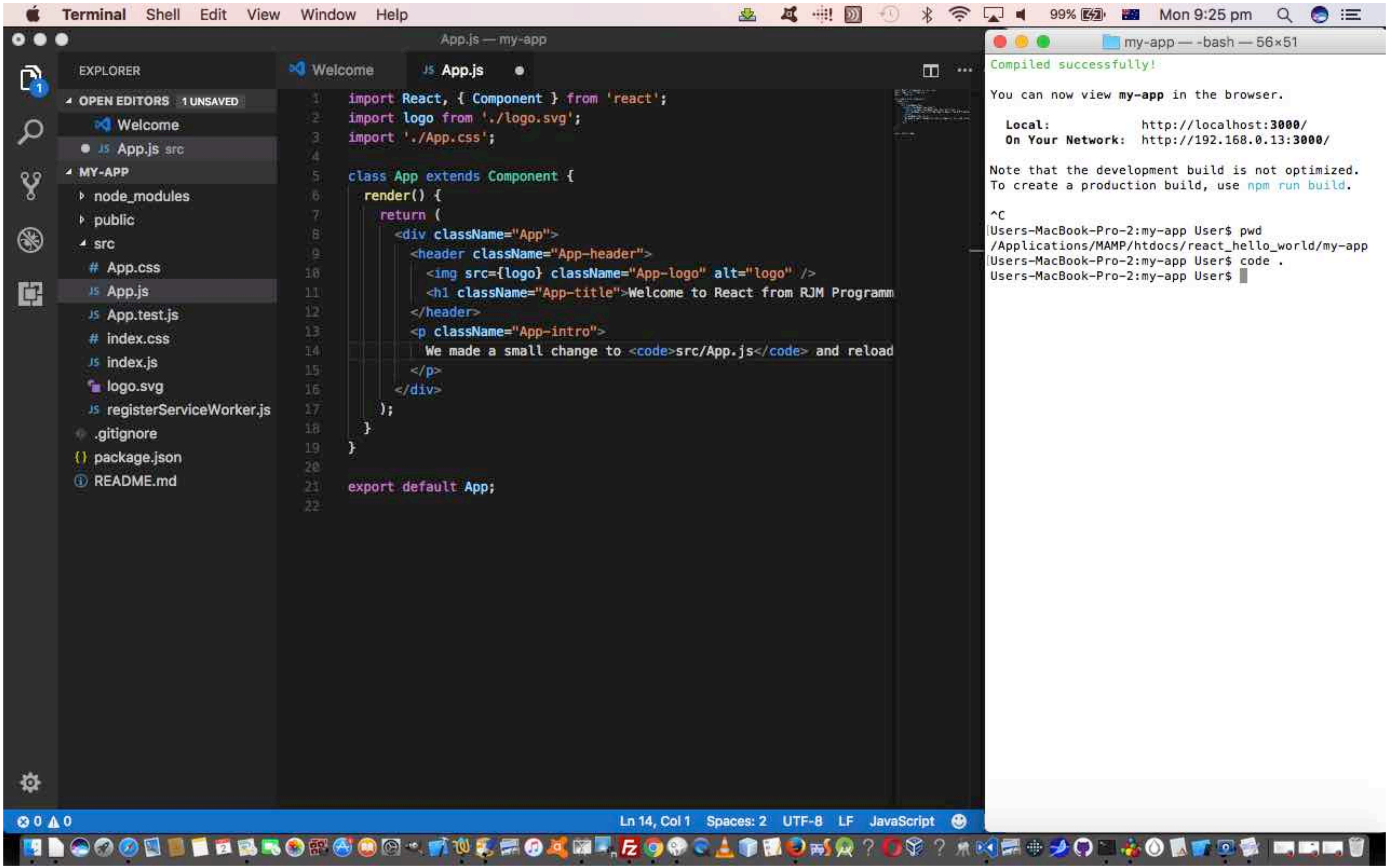
Interactive playground
Try essential editor features out in ...

```
ompiled successfully!

ou can now view my-app in the browser.

 Local:            http://localhost:3000/
 On Your Network:  http://192.168.0.13:3000/

ote that the development build is not optimized.
o create a production build, use npm run build.

C
sers-MacBook-Pro-2:my-app User$ pwd
Applications/MAMP/htdocs/react_hello_world/my-app
sers-MacBook-Pro-2:my-app User$ code .
```



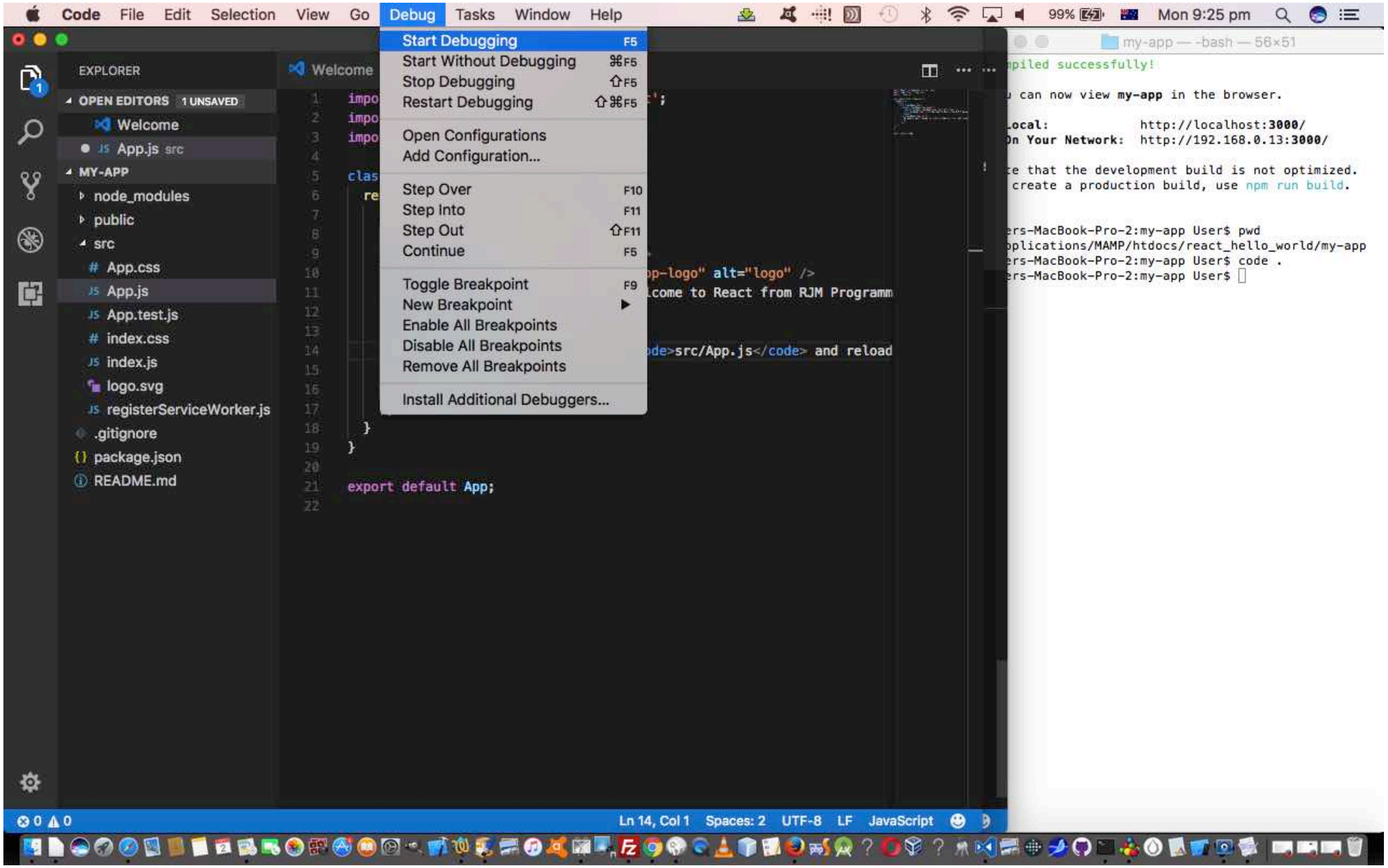
```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo"
          <h1 className="App-title">Welcome to React</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code> and
        </p>
      </div>
    );
  }
}

export default App;
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.13:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.

^C
Users-MacBook-Pro-2:my-app User$ pwd
/Applications/MAMP/htdocs/react_hello_world/my-app
Users-MacBook-Pro-2:my-app User$ code .
Users-MacBook-Pro-2:my-app User$
```



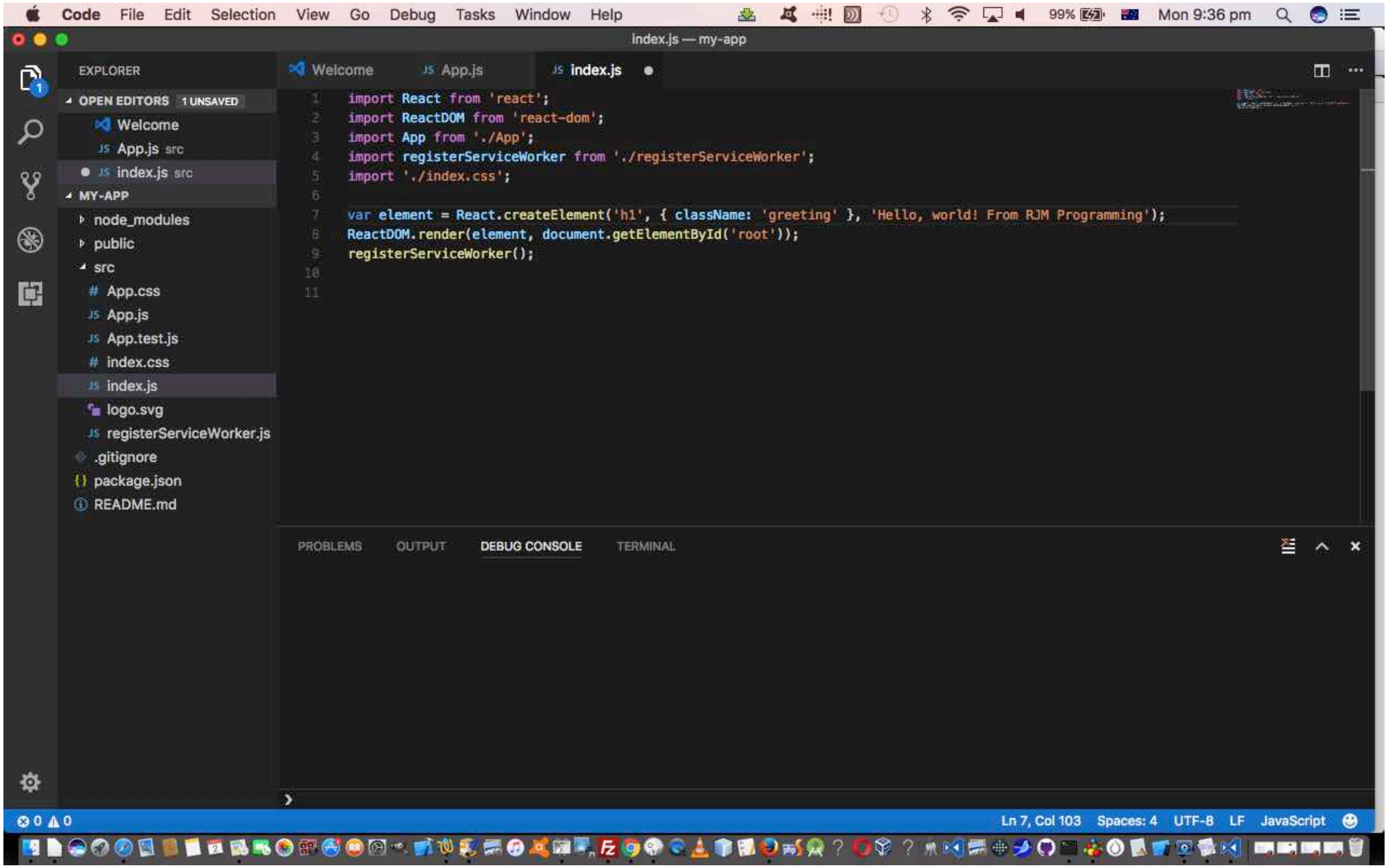
```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React from RJM Programm
        </header>
        <p className="App-intro">
          We made a small change to <code>src/App.js</code> and reload
        </p>
      </div>
    );
  }
}

export default App;
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.13:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.

^C
Users-MacBook-Pro-2:my-app User$ pwd
/Applications/MAMP/htdocs/react_hello_world/my-app
Users-MacBook-Pro-2:my-app User$ code .
Users-MacBook-Pro-2:my-app User$
```

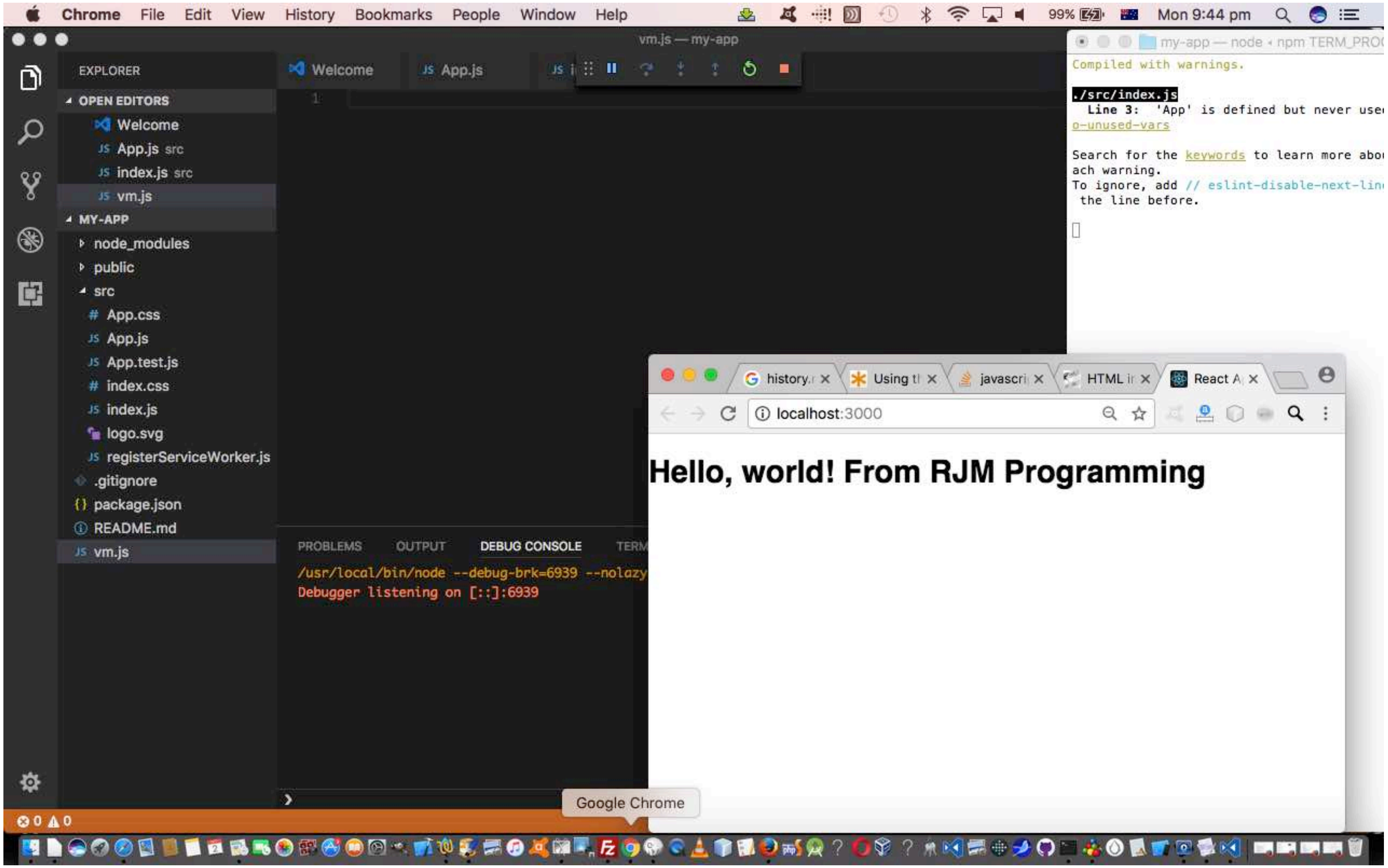


```
impo                                  ';
impo
impo

clas
  re




                                   p-logo" alt="logo" />
                                   come to React from RJM Programm


                                   de>src/App.js</code> and reload



  }
}

export default App;
```

```
piled successfully!

u can now view my-app in the browser.

ocal:            http://localhost:3000/
n Your Network:  http://192.168.0.13:3000/

e that the development build is not optimized.
 create a production build, use npm run build.


ers-MacBook-Pro-2:my-app User$ pwd
pplications/MAMP/htdocs/react_hello_world/my-app
ers-MacBook-Pro-2:my-app User$ code .
ers-MacBook-Pro-2:my-app User$ 
```



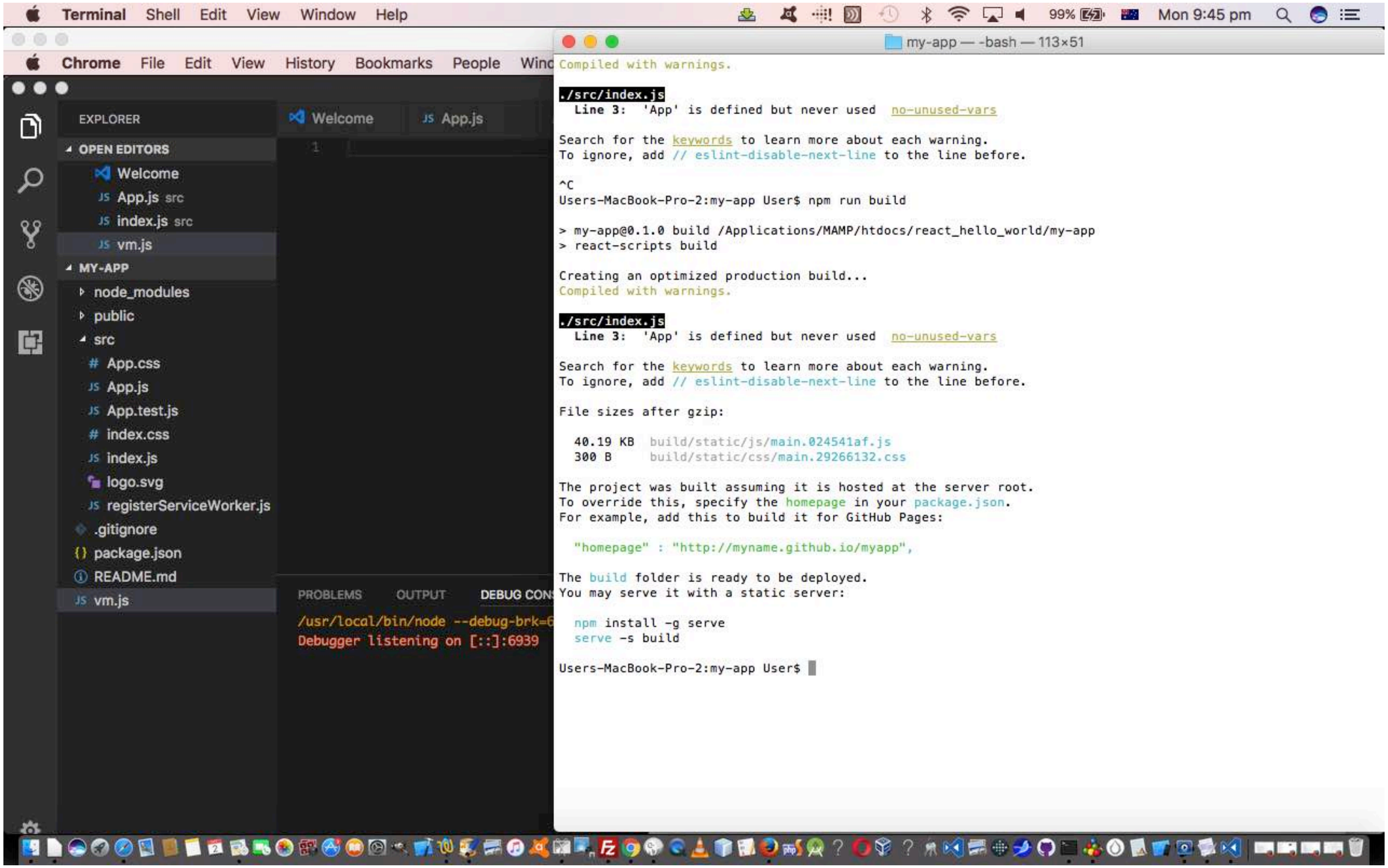
```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', { className: 'greeting' }, 'Hello, world! From RJM Programming');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```



```
Compiled with warnings.

./src/index.js
  Line 3:  'App' is defined but never use
o-unused-vars

Search for the keywords to learn more abo
ach warning.
To ignore, add // eslint-disable-next-lin
 the line before.
```

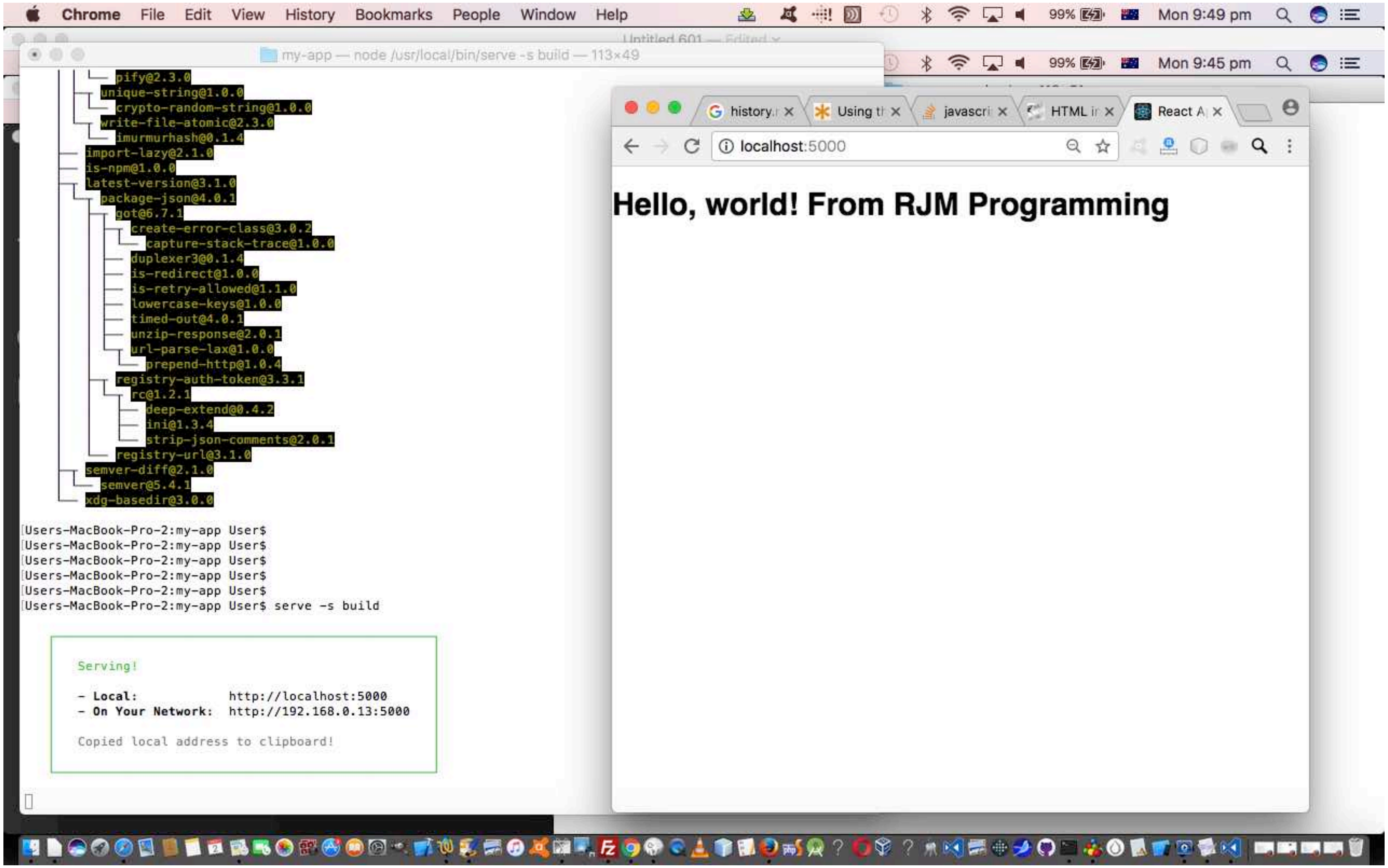
```
/usr/local/bin/node --debug-brk=6939 --nolazy
Debugger listening on [::]:6939
```

Hello, world! From RJM Programming



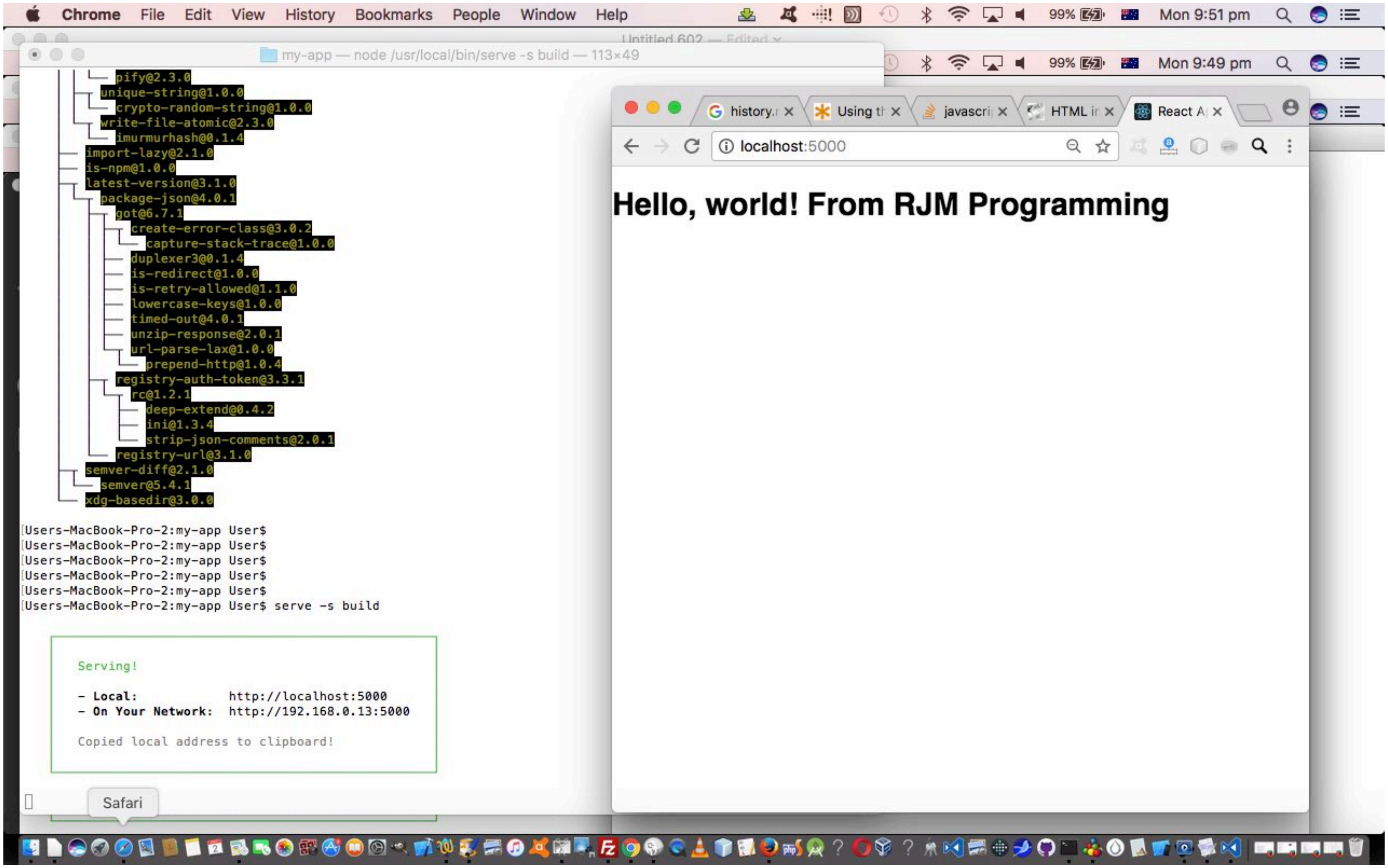
```
/usr/local/bin/node --debug-brk=6
Debugger listening on [::]:6939
```

```
Compiled with warnings.

./src/index.js
  Line 3:  'App' is defined but never used  no-unused-vars

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

^C
Users-MacBook-Pro-2:my-app User$ npm run build

> my-app@0.1.0 build /Applications/MAMP/htdocs/react_hello_world/my-app
> react-scripts build

Creating an optimized production build...
Compiled with warnings.

./src/index.js
  Line 3:  'App' is defined but never used  no-unused-vars

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

  40.19 KB  build/static/js/main.024541af.js
  300 B     build/static/css/main.29266132.css

The project was built assuming it is hosted at the server root.
To override this, specify the homepage in your package.json.
For example, add this to build it for GitHub Pages:

  "homepage" : "http://myname.github.io/myapp",

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Users-MacBook-Pro-2:my-app User$
```


Chrome File Edit View History Bookmarks People Window Help
99% Mon 9:49 pm
Untitled 601 — Edited
99% Mon 9:45 pm
my-app — node /usr/local/bin/serve -s build — 113×49
history.
Using t
javascri
HTML i
React A
localhost:5000

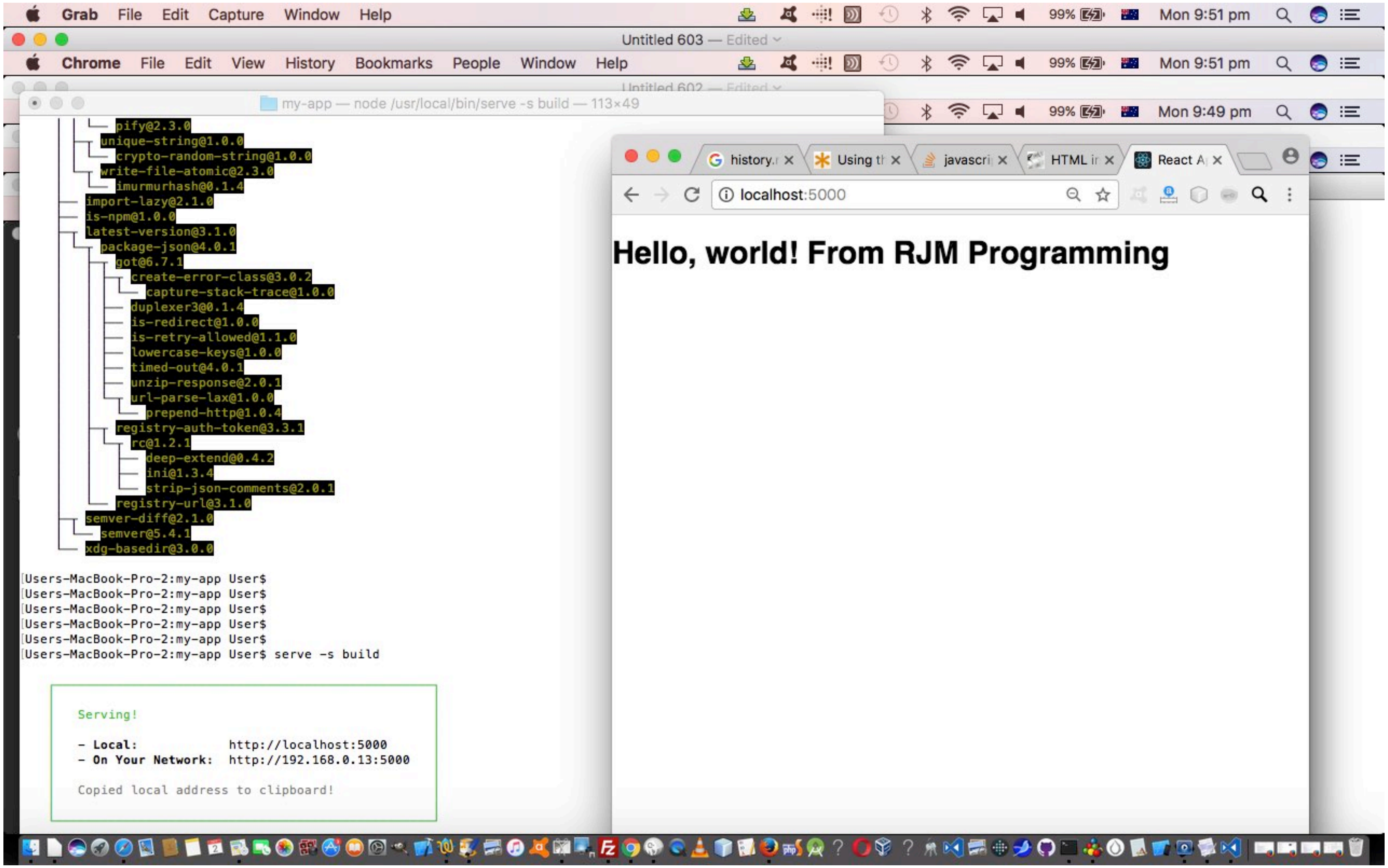
```
  │   └── pify@2.3.0
  │ ├─┬ unique-string@1.0.0
  │ │ └── crypto-random-string@1.0.0
  │ └─┬ write-file-atomic@2.3.0
  │   └── imurmurhash@0.1.4
  ├── import-lazy@2.1.0
  ├── is-npm@1.0.0
  ├─┬ latest-version@3.1.0
  │ └─┬ package-json@4.0.1
  │   ├─┬ got@6.7.1
  │   │ ├─┬ create-error-class@3.0.2
  │   │ │ └── capture-stack-trace@1.0.0
  │   │ ├── duplexer3@0.1.4
  │   │ ├── is-redirect@1.0.0
  │   │ ├── is-retry-allowed@1.1.0
  │   │ ├── lowercase-keys@1.0.0
  │   │ ├── timed-out@4.0.1
  │   │ ├── unzip-response@2.0.1
  │   │ └─┬ url-parse-lax@1.0.0
  │   │   └── prepend-http@1.0.4
  │   ├─┬ registry-auth-token@3.3.1
  │   │ └─┬ rc@1.2.1
  │   │   ├── deep-extend@0.4.2
  │   │   ├── ini@1.3.4
  │   │   └── strip-json-comments@2.0.1
  │   └── registry-url@3.1.0
  ├─┬ semver-diff@2.1.0
  │ └── semver@5.4.1
  └── xdg-basedir@3.0.0

Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$ serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!
```

Hello, world! From RJM Programming


Chrome File Edit View History Bookmarks People Window Help
99% Mon 9:51 pm
my-app — node /usr/local/bin/serve -s build — 113×49
history.r
Using th
javascri
HTML ir
React A
localhost:5000
Safari

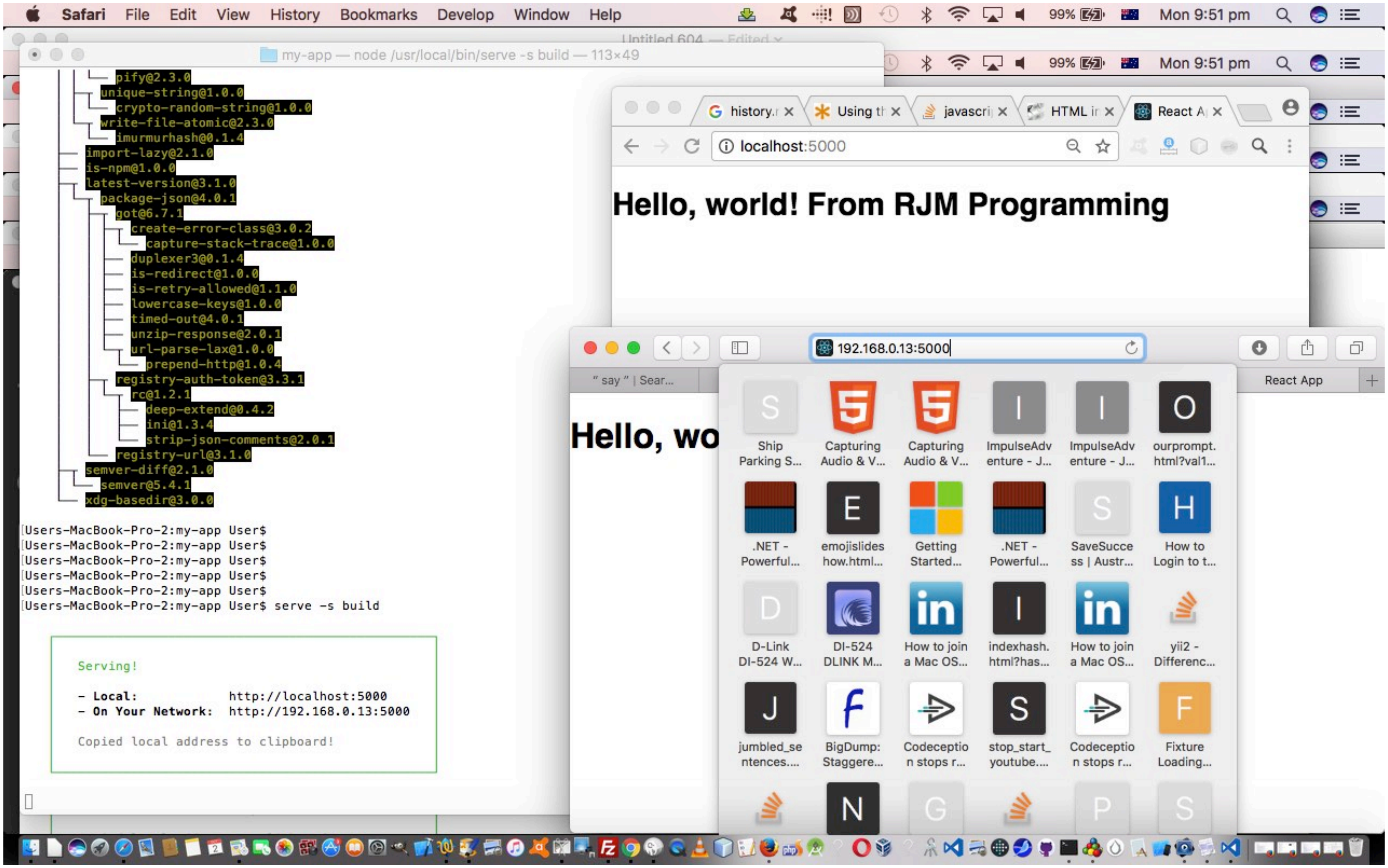
```
    │   └── pify@2.3.0
    ├── unique-string@1.0.0
    │   └── crypto-random-string@1.0.0
    └── write-file-atomic@2.3.0
        └── imurmurhash@0.1.4
├── import-lazy@2.1.0
├── is-npm@1.0.0
├── latest-version@3.1.0
│   └── package-json@4.0.1
│       ├── got@6.7.1
│       │   ├── create-error-class@3.0.2
│       │   │   └── capture-stack-trace@1.0.0
│       │   ├── duplexer3@0.1.4
│       │   ├── is-redirect@1.0.0
│       │   ├── is-retry-allowed@1.1.0
│       │   ├── lowercase-keys@1.0.0
│       │   ├── timed-out@4.0.1
│       │   ├── unzip-response@2.0.1
│       │   └── url-parse-lax@1.0.0
│       │       └── prepend-http@1.0.4
│       ├── registry-auth-token@3.3.1
│       │   └── rc@1.2.1
│       │       ├── deep-extend@0.4.2
│       │       ├── ini@1.3.4
│       │       └── strip-json-comments@2.0.1
│       └── registry-url@3.1.0
├── semver-diff@2.1.0
│   └── semver@5.4.1
└── xdg-basedir@3.0.0

Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$ serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!
```

Hello, world! From RJM Programming


Grab File Edit Capture Window Help
Untitled 603 — Edited
Chrome File Edit View History Bookmarks People Window Help
Mon 9:51 pm
my-app — node /usr/local/bin/serve -s build — 113×49
localhost:5000

```
    │   └── pify@2.3.0
    │  ├── unique-string@1.0.0
    │  │ └── crypto-random-string@1.0.0
    │  └── write-file-atomic@2.3.0
    │    └── imurmurhash@0.1.4
    ├── import-lazy@2.1.0
    ├── is-npm@1.0.0
    ├── latest-version@3.1.0
    │ └── package-json@4.0.1
    │   ├── got@6.7.1
    │   │ ├── create-error-class@3.0.2
    │   │ │ └── capture-stack-trace@1.0.0
    │   │ ├── duplexer3@0.1.4
    │   │ ├── is-redirect@1.0.0
    │   │ ├── is-retry-allowed@1.1.0
    │   │ ├── lowercase-keys@1.0.0
    │   │ ├── timed-out@4.0.1
    │   │ ├── unzip-response@2.0.1
    │   │ └── url-parse-lax@1.0.0
    │   │   └── prepend-http@1.0.4
    │   ├── registry-auth-token@3.3.1
    │   │ └── rc@1.2.1
    │   │   ├── deep-extend@0.4.2
    │   │   ├── ini@1.3.4
    │   │   └── strip-json-comments@2.0.1
    │   └── registry-url@3.1.0
    ├── semver-diff@2.1.0
    │ └── semver@5.4.1
    └── xdg-basedir@3.0.0

Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$ serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!
```

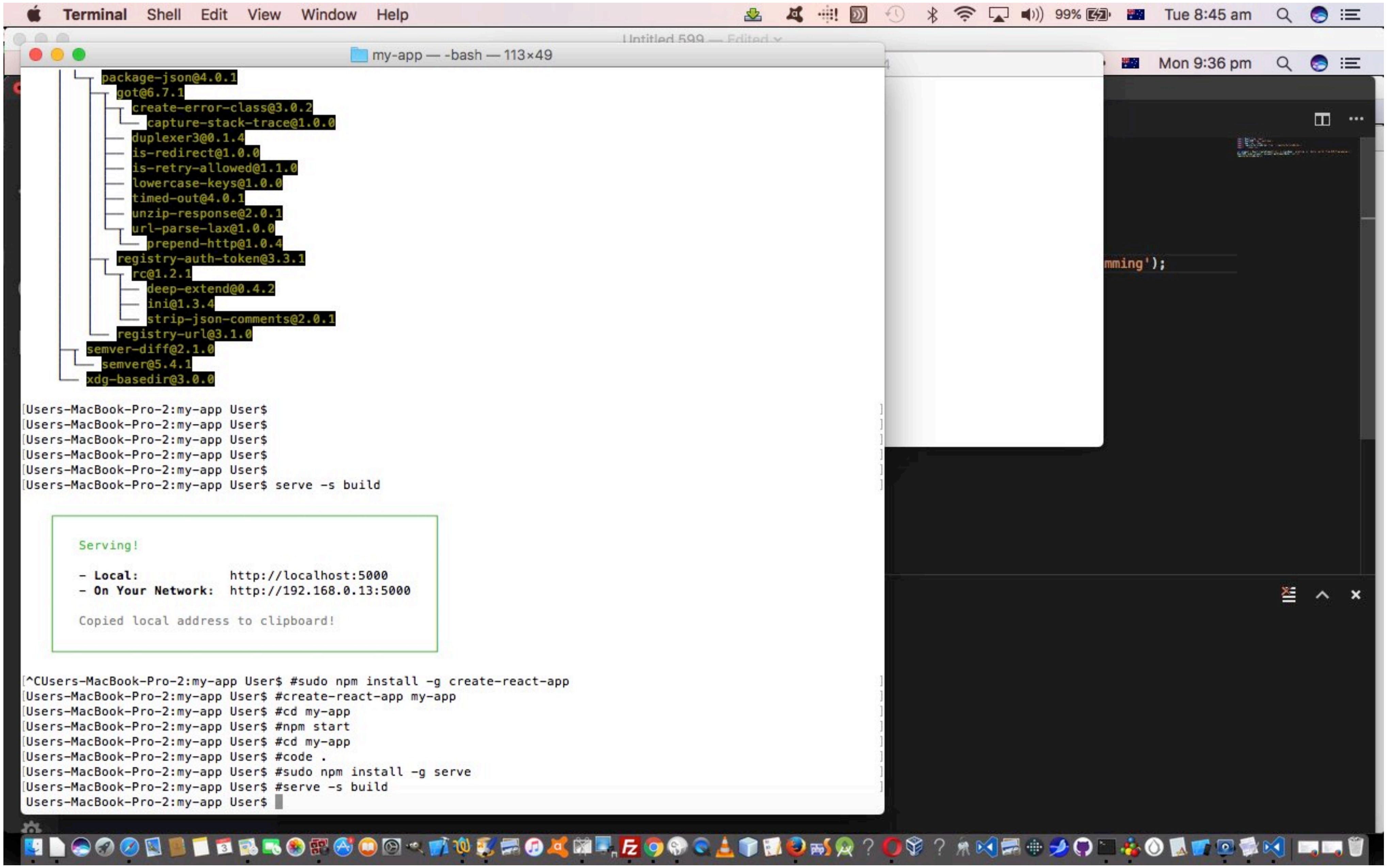
Hello, world! From RJM Programming



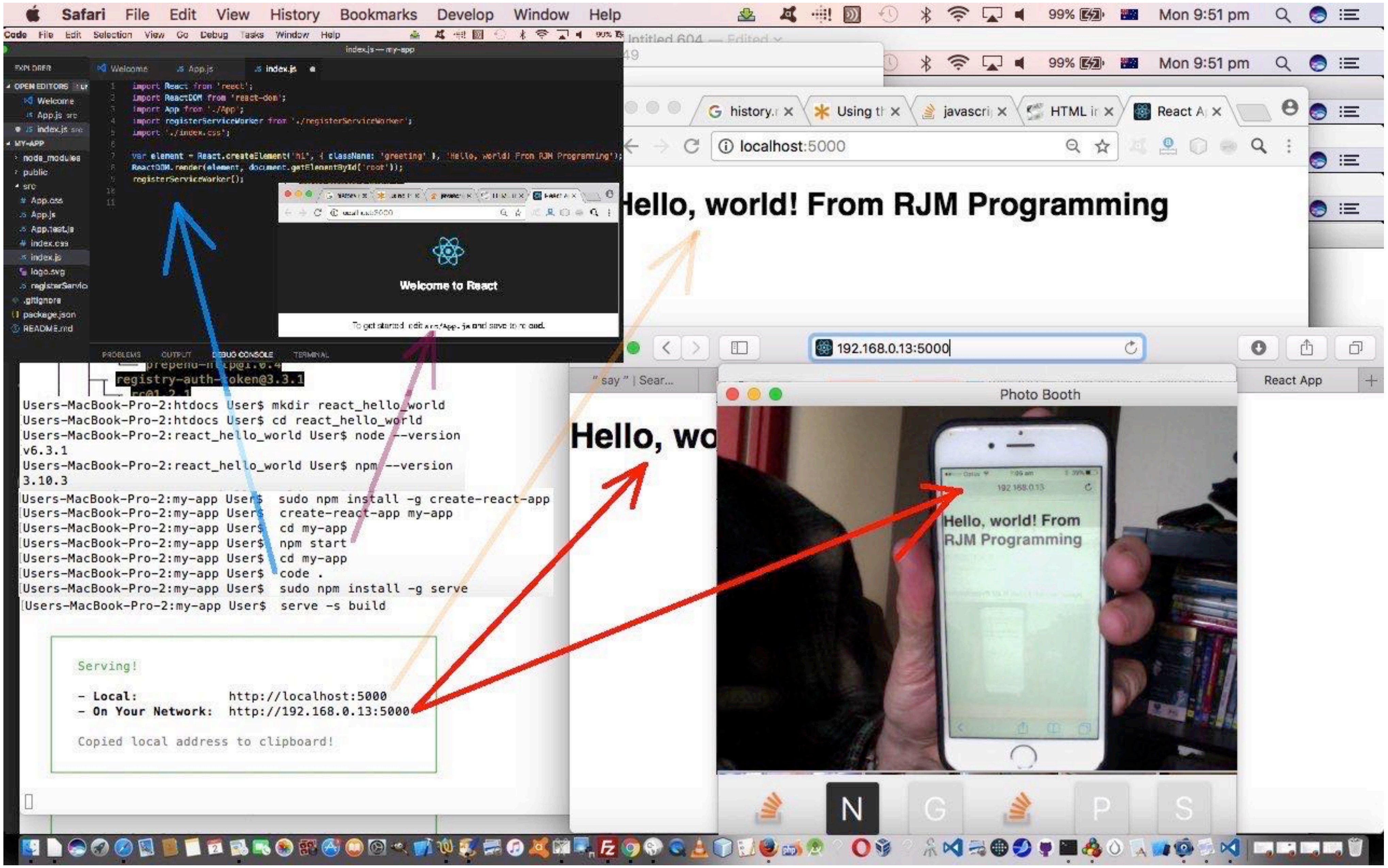
```
│   └── pify@2.3.0
├── unique-string@1.0.0
│   └── crypto-random-string@1.0.0
└── write-file-atomic@2.3.0
    └── imurmurhash@0.1.4
├── import-lazy@2.1.0
├── is-npm@1.0.0
├── latest-version@3.1.0
│   └── package-json@4.0.1
│       ├── got@6.7.1
│       │   ├── create-error-class@3.0.2
│       │   │   └── capture-stack-trace@1.0.0
│       │   ├── duplexer3@0.1.4
│       │   ├── is-redirect@1.0.0
│       │   ├── is-retry-allowed@1.1.0
│       │   ├── lowercase-keys@1.0.0
│       │   ├── timed-out@4.0.1
│       │   ├── unzip-response@2.0.1
│       │   └── url-parse-lax@1.0.0
│       │       └── prepend-http@1.0.4
│       ├── registry-auth-token@3.3.1
│       │   └── rc@1.2.1
│       │       ├── deep-extend@0.4.2
│       │       ├── ini@1.3.4
│       │       └── strip-json-comments@2.0.1
│       └── registry-url@3.1.0
├── semver-diff@2.1.0
│   └── semver@5.4.1
└── xdg-basedir@3.0.0

Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$ serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!
```

Hello, world! From RJM Programming

Hello, wo



```
    │  └─ package-json@4.0.1
    │     ├─ got@6.7.1
    │     │  ├─ create-error-class@3.0.2
    │     │  │  └─ capture-stack-trace@1.0.0
    │     │  ├─ duplexer3@0.1.4
    │     │  ├─ is-redirect@1.0.0
    │     │  ├─ is-retry-allowed@1.1.0
    │     │  ├─ lowercase-keys@1.0.0
    │     │  ├─ timed-out@4.0.1
    │     │  ├─ unzip-response@2.0.1
    │     │  └─ url-parse-lax@1.0.0
    │     │     └─ prepend-http@1.0.4
    │     ├─ registry-auth-token@3.3.1
    │     │  └─ rc@1.2.1
    │     │     ├─ deep-extend@0.4.2
    │     │     ├─ ini@1.3.4
    │     │     └─ strip-json-comments@2.0.1
    │     └─ registry-url@3.1.0
    ├─ semver-diff@2.1.0
    │  └─ semver@5.4.1
    └─ xdg-basedir@3.0.0

Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$
Users-MacBook-Pro-2:my-app User$ serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!

^CUsers-MacBook-Pro-2:my-app User$ #sudo npm install -g create-react-app
Users-MacBook-Pro-2:my-app User$ #create-react-app my-app
Users-MacBook-Pro-2:my-app User$ #cd my-app
Users-MacBook-Pro-2:my-app User$ #npm start
Users-MacBook-Pro-2:my-app User$ #cd my-app
Users-MacBook-Pro-2:my-app User$ #code .
Users-MacBook-Pro-2:my-app User$ #sudo npm install -g serve
Users-MacBook-Pro-2:my-app User$ #serve -s build
Users-MacBook-Pro-2:my-app User$ 
```



```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';

var element = React.createElement('h1', { className: 'greeting' }, 'Hello, world! From RJM Programming');
ReactDOM.render(element, document.getElementById('root'));
registerServiceWorker();
```

Hello, world! From RJM Programming

```
registry-auth-token@3.3.1
Users-MacBook-Pro-2:htdocs User$ mkdir react_hello_world
Users-MacBook-Pro-2:htdocs User$ cd react_hello_world
Users-MacBook-Pro-2:react_hello_world User$ node --version
v6.3.1
Users-MacBook-Pro-2:react_hello_world User$ npm --version
3.10.3
Users-MacBook-Pro-2:my-app User$  sudo npm install -g create-react-app
Users-MacBook-Pro-2:my-app User$  create-react-app my-app
Users-MacBook-Pro-2:my-app User$  cd my-app
Users-MacBook-Pro-2:my-app User$  npm start
Users-MacBook-Pro-2:my-app User$  cd my-app
Users-MacBook-Pro-2:my-app User$  code .
Users-MacBook-Pro-2:my-app User$  sudo npm install -g serve
Users-MacBook-Pro-2:my-app User$  serve -s build

   Serving!

   - Local:            http://localhost:5000
   - On Your Network:  http://192.168.0.13:5000

   Copied local address to clipboard!
```