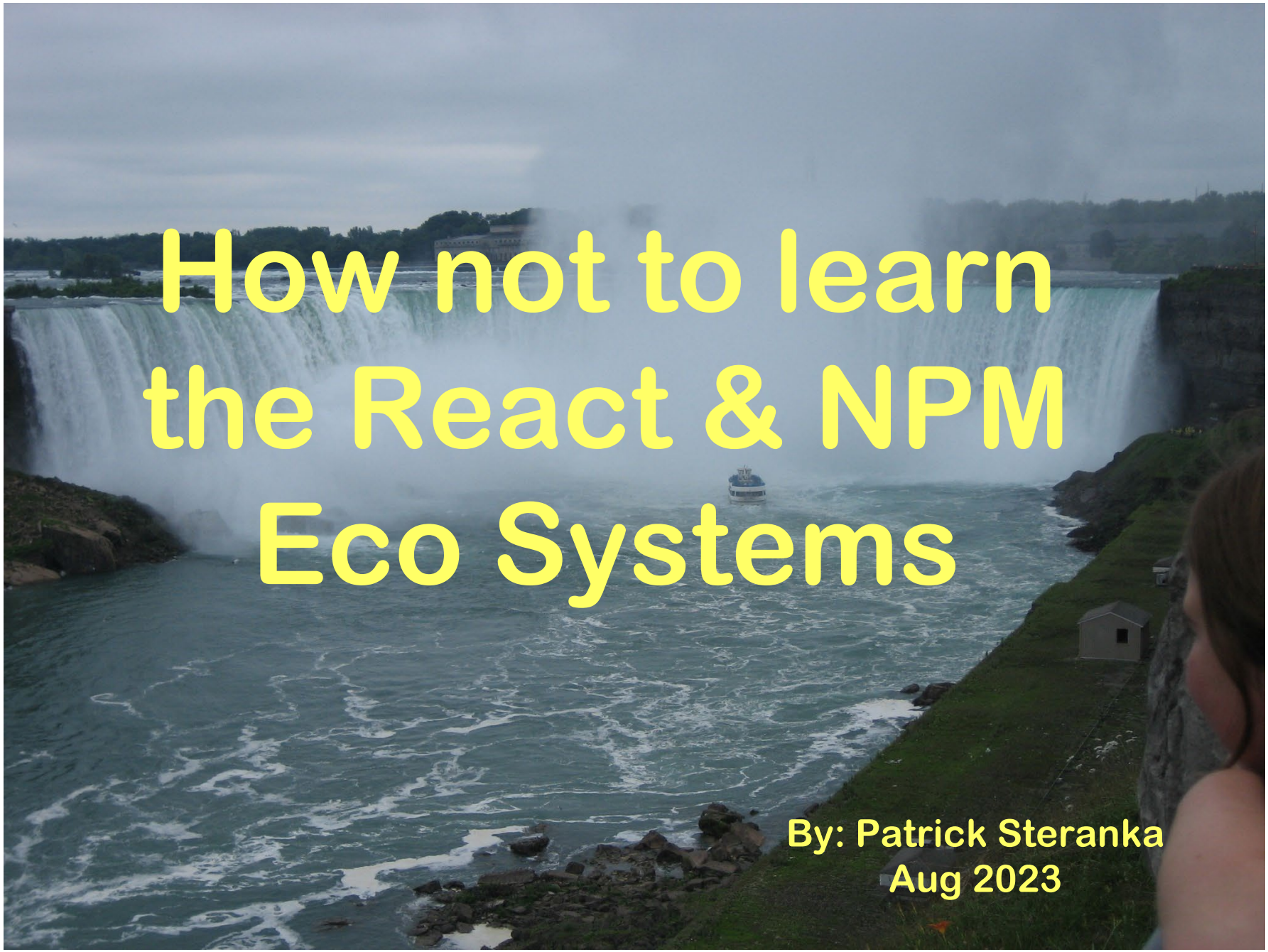

How not to learn
the React & NPM
Eco Systems
By: Patrick Steranka
Aug 2023

Goal of Presentation

- Help others avoid pitfalls I ran into
- Provide recommendations on how to learn Javascript and React
- Share my knowledge and experiences and give back to CharmCityJS

Patrick@Steranka.com

Agenda

- OLD: How I used to learn technologies
  - Big Picture
  - Learn technologies from bottom up
  - Read the specs, read books, take training
  - Know how it all works
- Review how I used to learn
  - What worked and what didn't
  - How my world view changed
- Review some examples of what I did
  - What failed, and suggest recommendations
- NEW: How I now learn technologies

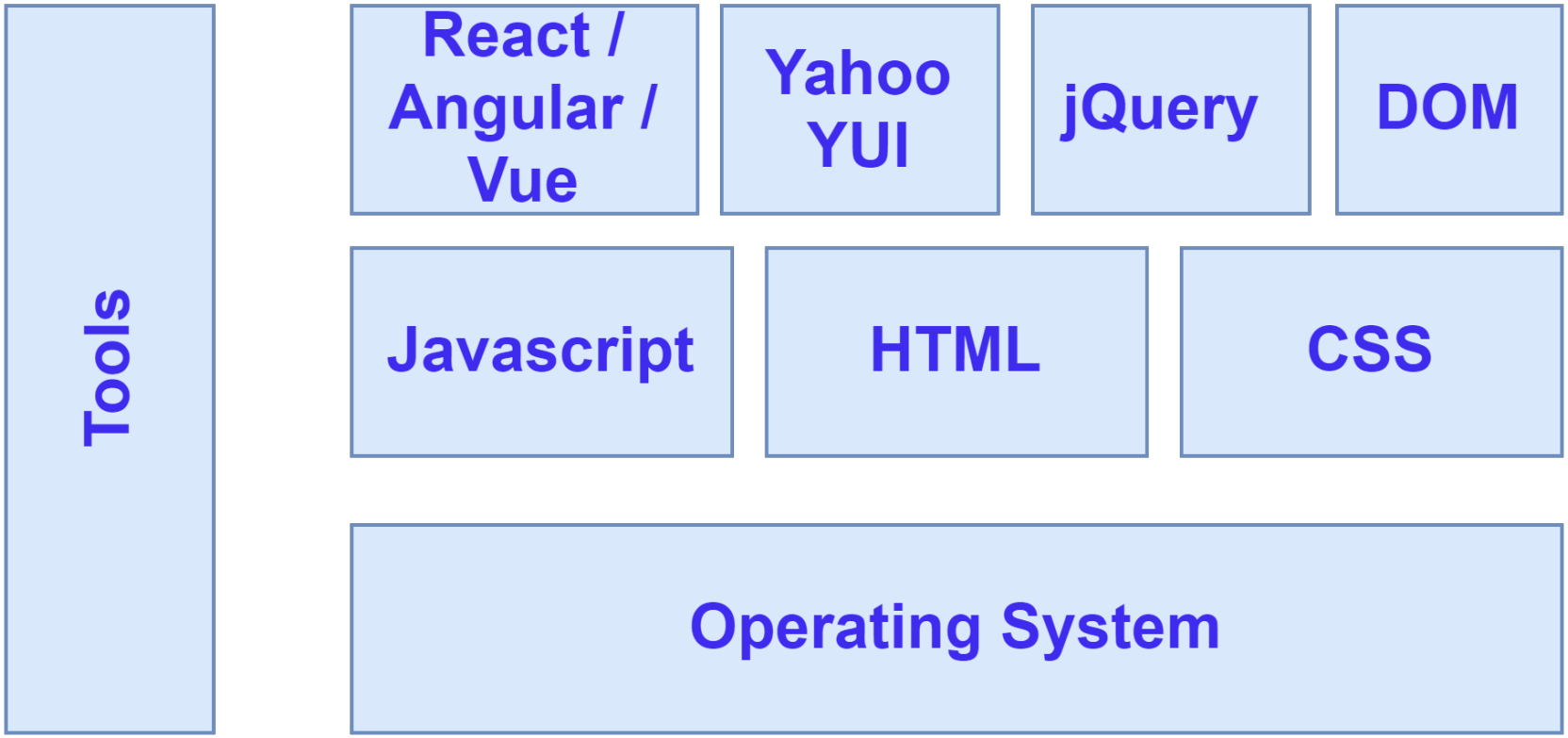
Patrick@Steranka.com

How I used to learn technologies

- Understand Big Picture
- Learn technologies from the bottom up
- Read the specs, books, take training
- Get tools and learn the tools
- Attend meetups & talk with other developers
- For each concept
  - Create small stand-alone sample programs to illustrate a specific concept
- Create diagrams and documents to capture what I've learned

Patrick@Steranka.com

Big Picture



Patrick@Steranka.com

Learning technologies from the bottom up

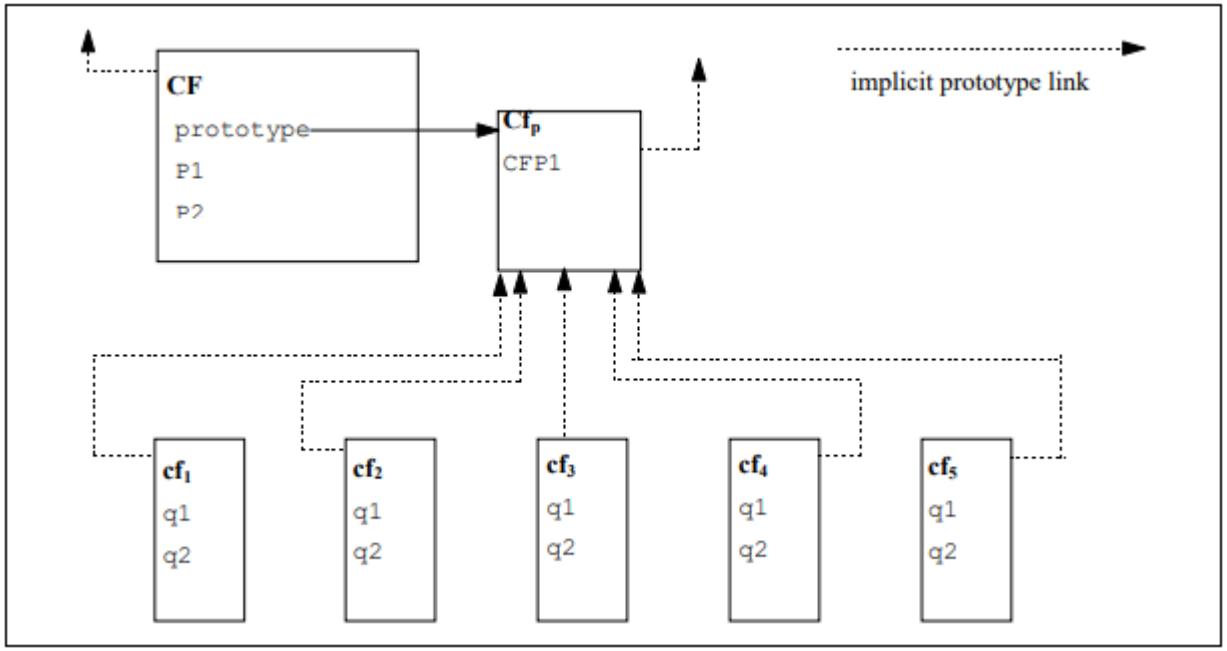
- Specs
  - Javascript ECMA-262 (110 pages), to 2023 (14th edition) (840 pages)
  - HTML5 [Web]
  - CSS3 Specs - MDN
  - React and other technologies have no specs
    - Old website: legacy.reactjs.org, New website: react.dev
- Books
  - JavaScript: The Definitive Guide, 7th Edition
  - HTML & CSS: Design and Build Websites
  - CSS: The definitive Guide
  - JavaScript: The good parts – Not a beginners book
  - Many others...
- Training / Podcasts / other
  - UDEMY

Patrick@Steranka.com

ECMA-262 Object Model

- Javascript Object
  - Custructors, Prototypes, inheritance

ECMAScript does not contain proper classes such as those in C++, Smalltalk, or Java, but rather, supports constructors which create objects by executing code that allocates storage for the objects and initializes all or part of them by assigning initial values to their properties.



Patrick@Steranka.com

What not to do

Learn technologies from the bottom up

- **Specs – Javascript, HTML, CSS**
  - **Waste of time for beginers. Many better (easier) references exist.**
    - **HTML standard its over 1400 pages!**
    - **CSS is a collection of dozens of standards. Forget it! Read a book on how to use CSS and learn the 20% of the 20 standards you want.**
      - **CSS Introduction dropped and replaced by CSS Snapshots (2007, 2010, 2015, 2017, 2018, 2020,21,22,23**
    - **Javascript spec is now 840 pages has undergone numerous changes in the past 15 years**
  - **So much to learn and so little time...**
    - **The web has been around 20-30 years now, there is a lot of cruft that doesn't need to be learned!**
    - **Focus on the 80% of HTML you need to learn**
    - **Learn CSS Box model**
- **Books & Video**
  - **Don't read/watch old React, Javascript, HTML, CSS books they will teach outdated and/or deprecated concepts**

Patrick@Steranka.com

What you should do

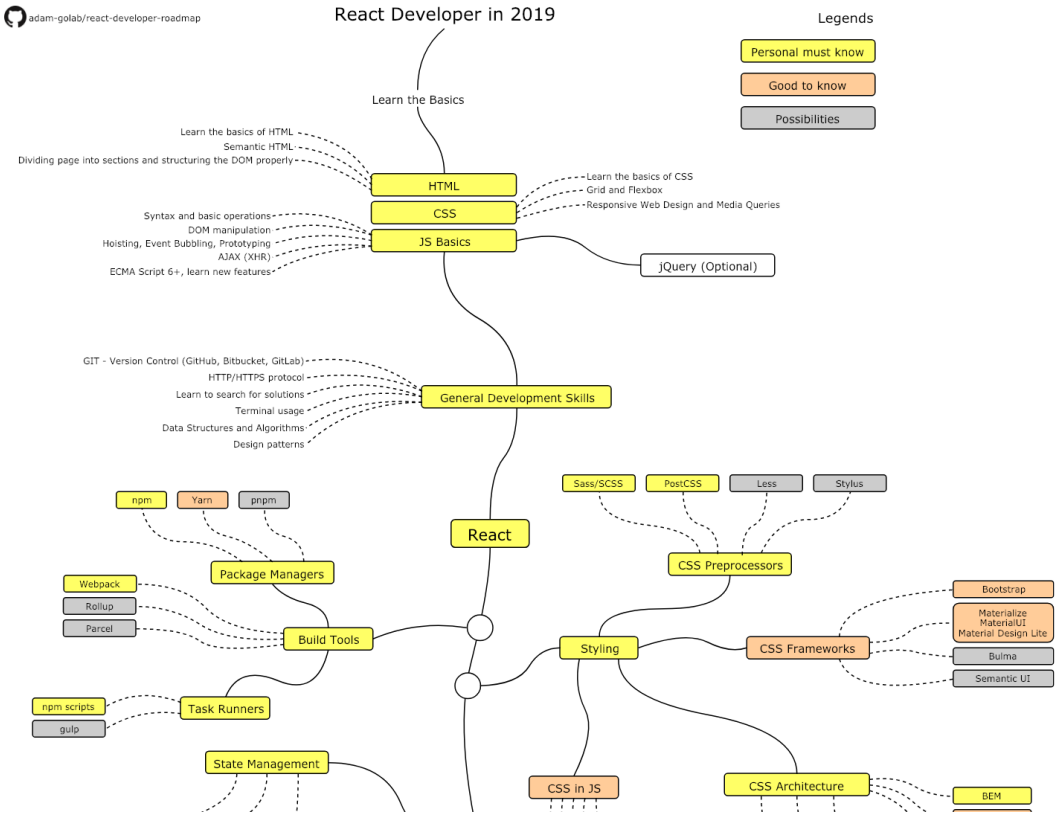
Learn technologies from the bottom up

- Javascript
  - Learn about null, nothing, truthy-ness, and shallow and deep equality
  - Learn about looping who knew it could be so complicated
  - Learn how closures work and pitfalls (memory),
  - Learn about Javascript OO (prototype inheritance) and the meaning of "this"
  - Learn about promises, await, and how event dispatching works
- Learn HTML basics (MDN is great resource)
- CSS - Learn CSS Box model
  - Grid, or Flex or both
  - How adjacent space is joined (affects spacing)
- Books are good.
- Web articles are OK.
- Video is often best due to visual nature
  - I like UDEMY & YouTube
- Use some AI search tool https://claude.ai/chats

Patrick@Steranka.com

New Big Picture

- **React Developer Roadmap**
  - https://github.com/adam-golab/react-developer-roadmap
  - **Lots more to learn than I realized**



Patrick@Steranka.com

What not to do

Learn technologies from the bottom up

- Don’t try to learn it all (like I did)
- My prior strategy for learning about technology from the ground up is no longer viable.
  - Too much to learn
  - Implementations change too frequently to benefit from deep study
    - React Render deep dive

Patrick@Steranka.com

What you should do

Learn technologies from the bottom up

- Pick one or two technologies to learn and try to use them everyday.
- Repetition is the key to learning!
- Try to find some podcast, article, or video to watch every day to re-inforce a new technology

Patrick@Steranka.com

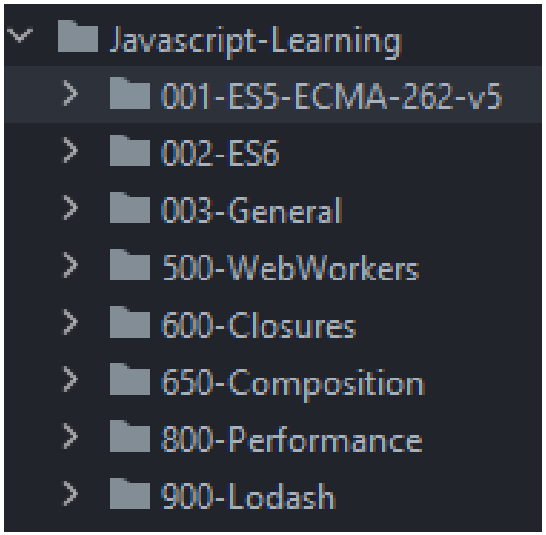
Tools and Recommendations

- Browser - pick one and get good at it
- git, npm, node, chrome dev tools – Must learn these tools
- Build Tools – tsc, webpack, rollup, parcel
- IDE - Webstorm, Vscode, Sublime4
  - pick one and get good at it

Patrick@Steranka.com

For each concept … create program

- Javascript, React
  - I created a stand alone app



Patrick@Steranka.com

How are modules initialized?

(Q) If a module is included by multiple other modules, is the module only loaded once?

(A) No. I didn't think so but wrote a test program to prove it.

To run this program run

```
webpack
node bundle.js
```

NOTE: The output is

```
file1 is stand alone
file3 is stand alone
file2-uses-file3 is being loaded now
file4-uses-file3 is being loaded now
Inside main
```

Because no line is printed more than once, this proves that initialization/loading of a file only occurs one time regardless of how many times it is *included*.

Patrick@Steranka.com

What not to do

Creating stand alone test programs

- Don't try to use NodeJS to test javascript
- Don't try to write javascript that will run in the browser and in NodeJS
- Don't try and figure out how to change require() (in NodeJS) to import statements (in JS).
- I did and the result was largely a waste of time googling for things that no one else would ever do! OK some others did these things.
- Realize that the solutions used for the browser don't always work in NodeJS and visa-versa.
  - Some utility libraries are exceptions (like lodash)
  - The way application building works between NodeJS and Javascript for the Browser is completely different.
  - The way threading (or not!) works in NodeJS is different from the browser.

Patrick@Steranka.com

What You should do
Creating stand alone test programs

- Use browser sandboxes to test out concepts
  - CodeSandbox.io
  - CodePen.io
  - BabelJS.io
  - NodeJS [CodeSandbox.io]
- You can use

Patrick@Steranka.com

New Big Picture

Tools
- Build Tools
- Package Management
- Dependency Management Tools
- CSS Transpilers
- Unit Testing (Jest, Mocha, etc)

React / Angular / Vue | Yahoo YUI | jQuery | DOM

Utility Libraries
lodash, moment, classnames, etc

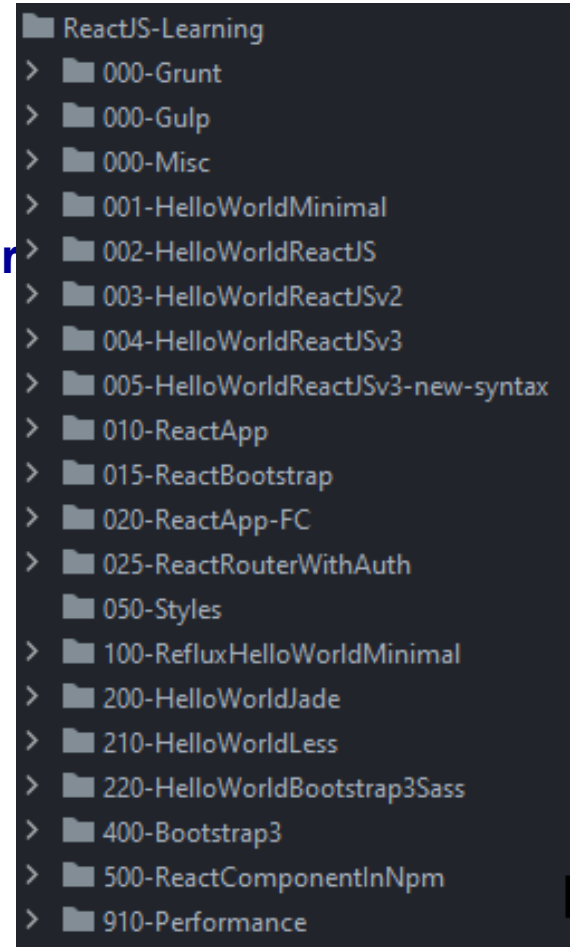
Javascript | HTML | CSS | NodeJS

Operating System

Patrick@Steranka.com

Learning React

- I created many examples
  - Examples with Classes
  - Examples with Functional Components
  - Sample components to use as cookie cutter
- Some examples used CDN
- Some used grunt to transpile/bundle
- Some used create-react-app



Patrick@Steranka.com

What you should do

Learning React

- Pick a version of React and try to stick with it.
- Document the version of React, Node and NPM you are using in your README.md so that if someone clones the code, they know what version of node/npm should work!
- Commit your package-lock.json files!
- When watching tutorials, avoid tutorials where the author doesn't include the versions of the packages installed!
  - When you watch it 6 months later, your code might not compile/build/install
- Find a version of create-react-app you like and use the same one for all of your sample apps. (I wish I did this!)
  - Or spend time learning something you didn't plan on learning when what worked on one project no longer works the same way in this project.
- Learn client side react before trying to learn server side!

Patrick@Steranka.com

What I did wrong

Learning React

- I thought React was javascript.
- I thought React was like jQuery but with a buffer to update DOM off screen
- I thought I could make changes to the DOM
  - I thought I could change arrays and data would update on the screen
  - I changed this.state without realizing that should never be done.
- I tried to take existing code and tweak it to work in React
- I didn't realize it but at the time I learned React it was on the bleeding edge.
  - No package-lock.json
  - No way of easily managing dependencies
  - Every 6 months my examples stopped working
- When packages changed I tried to keep up with them instead of staying put.

Patrick@Steranka.com

Learning webpack

- **React eject**

  npx createreactapp mywebpacktest

  cd mywebpacktest

  npm start

  npmrun eject

- **Study webpack build scripts**
  - **My logic: Since create-react-app is giving a sample react app, then studying how it's build system works with webpack will help me learn webpack**
  - **BUZZ! AAAHHH! Nope**
    - **Danger Will Robinson**
    - **Pain, run, run**

Patrick@Steranka.com

What not to do
Learning WebPack

- Don't try and do this! It's painful
- The scripts are written using advanced techniques which may be difficult for a beginning Javascript/React developer to understand. And there's no one who is going to explain them!
- Webpack does an awful lot of things, many of which are way beyond what you need to do in order to configure webpack and understand how it works
  - bundling, hot module reload, source maps, and much more)
- The version of webpack has changed over the years with create-react-app.

Patrick@Steranka.com

What You should do
Learning WebPack

- Search for one of the many good webpack tutorials
- Webpack does an awful lot of things, many of which are way beyond what you need to do in order to configure webpack and understand how it works
  - Something I put together
  - Google search for webpack tutorial
  - https://www.youtube.com/watch?v=MpGLUVbqoYQ (Not yet verified)
  - My answer on StackOverflow to answer the question:
    Invalid configuration object. Webpack has been initialised using a configuration object that does not match the API schema

Patrick@Steranka.com

Other things I've done

- Tried understanding how promises work in browser and NodeJS and trying to figure out why things were working slightly differently.
- Trying to understand why Chrome DevTools wasn't allowing me to set breakpoints in React Code that was (in theory) compiled for development mode.
  - I learned all about sourcemaps and why this happens...
  - See Debug Webpack App Browser

Patrick@Steranka.com

Patrick@Steranka.com

How I now learn technologies

- Understand Big Picture
- Get a task
  - Learn what you need to get the task done
  - Ask other developers the tools they use
  - Learn LINGO so you can search for answers
  - Find an AI search tool to use when google and stackoverflow.com don't have the answers
- Repeat performing tasks
  - Create mental models of how things work and validate them with senior devs

Patrick@Steranka.com

How I now learn technologies (2)

- For technologies I do the following
  - Buy and take a $10 UDEMY course
  - Find YouTube video presentations
  - Create bookmark of youtube videos I can listen to while working out
  - Use online website instead of trying to create stand alone tasks

Patrick@Steranka.com

Questions?

Patrick@Steranka.com

Summary of Changes

Version / Date	Description
V1.0 2023-07-30	Initial version for CharmCityJS August 2023 meeting
V1.1 2023-08-02	Tweeks from reviewing

Patrick@Steranka.com