

React 5

CS571: Building User Interfaces

Cole Nelson

Logistics

- Cole's Office Hours will be on Wednesday, 9a-12p this week (on Zoom)!
- Midterm exam coming up...
 - Short review of F22 solution at end of class!
- Please fill out [the AEFIS Survey](#) by March 10th!

Midterm Exam

- Thursday, March 9th 5:45-7:15pm in the Chemistry Building Room S429. You will have **75 minutes** for 24 MC (8 pts), 5 SA (5 pts), and 1 LR (2 pts)
 - Be conscious of your use of time!
- **Single-sided** standard sheet of paper for notes.
 - Handwritten or typed is okay!
- Bring your Wiscard and a pencil!
- F22 midterm and solution on Canvas.

Memoization

Not memorization!

Memoization

Storing the result so you can use it next time instead of calculating the same thing again and again

[what the frik is: memoization](#)

`useCallback` to memoize functions

`useMemo` to memoize calculated values

`memo` to memoize components

useCallback Hook

Consider the following functional component...

```
function MyApp() {  
  const myComplicatedFunction = () => {  
    // ...  
  }  
  
  return <>  
    <button onClick={myComplicatedFunction}>Click Me</button>  
  </>  
}
```

How many times do we *create* the function
`myComplicatedFunction`? We do on *every render!*

useCallback Hook

useCallback is used to 'memoize' a callback function.

```
function MyApp() {
  const myComplicatedFunction = useCallback(() => {
    // ...
  }, []);
}

return <>
  <button onClick={myComplicatedFunction}>Click Me</button>
</>
}
```

Takes a callback function to 'memoize' and an optional list of dependencies (e.g. when to re-'memoize').

useMemo Hook

Same thing as `useCallback`, except memoizes the *value* of a *callback* rather than the *callback* itself.

```
function MyApp() {
  const myComplicatedValue = useMemo(() => { /* Some complex call */ }, []);

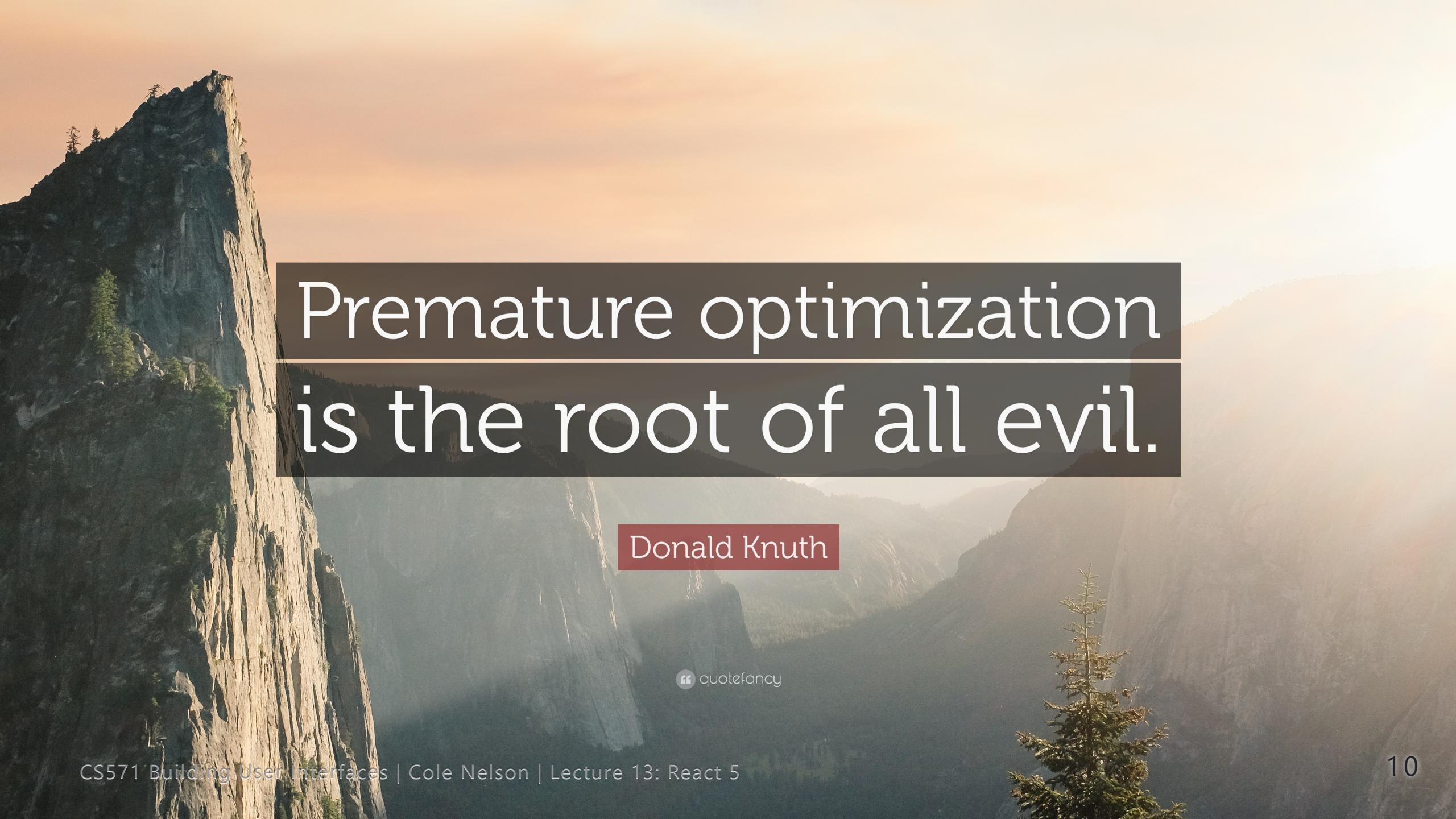
  return <>
    <p>{myComplicatedValue}</p>
  </>
}
```

memo -ized Components

Used for creating *purely functional* components. Given the same props, the function renders the same output.

```
//           v--- Name of functional component!
export default memo(GroceryList, (prevProps, nextProps) => {
  return prevProps.apples === nextProps.apples &&
  prevProps.bananas === nextProps.bananas &&
  prevProps.coconuts === nextProps.coconuts;
})
```

See StackBlitz for `useCallback`, `useMemo`, and `memo`



Premature optimization
is the root of all evil.

Donald Knuth

“ quotefancy

A Plea for Lean Software

Niklaus Wirth
ETH Zürich

Memory requirements of today's workstations typically jump substantially—from several to many megabytes—whenever there's a new software release. When demand surpasses capacity, it's time to buy add-on memory. When the system has no more extensibility, it's time to buy a new, more powerful workstation. Do increased performance and functionality keep pace with the increased demand for resources? Mostly the answer is no.

About 25 years ago, an interactive text editor could be designed with as little as 8,000 bytes of storage. (Modern program editors request 100 times that much!) An operating system had to manage with 8,000 bytes, and a compiler had to fit into 32 Kbytes, whereas their modern descendants require megabytes. Has all this inflated software become any faster? On the contrary. Were it not for a thousand times faster hardware, modern software would be utterly unusable.

Finding a Balance

1. Given the same input, renders the same output.
2. Is rendered often.
3. Does not change often.
4. Is of substantial size.

Dmitri Pavlutin Blog Post



Heuristics whether a React component should be wrapped in React.memo()

01

Pure functional component

Your <Component> is functional and given the same props, always renders the same output.

02

Renders often

Your <Component> renders often.

03

Re-renders with the same props

Your <Component> is usually provided with the same props during re-rendering.

04

Medium to big size

Your <Component> contains a decent amount of UI elements to reason props equality check.

Your turn!

Expand on our ticket tracking app from last week...

1. Replace "ENTER_YOUR_BID" with your Badger ID.
2. Don't allow the user to post a message if it contains a bad word like "frik".
3. Memoize! Use `useCallback` , `useMemo` , and `memo`

[Clone from here.](#)

What will we learn today?

- How are legacy React apps made?
- How can we write re-usable logic?
- What is the bigger picture for React Apps?
- Preparing for React Native...

Legacy React Apps

Legacy React apps were class-based and had "lifecycle methods". We won't go into any more detail.

```
class Welcome extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { counter: 0 };  
    this.handleClick = this.handleClick.bind(this);  
  }  
  componentDidMount() { /* ... */ }  
  handleClick() { /* ... */ }  
  render() {  
    return <h1>Hello World!</h1>;  
  }  
}
```

How Can We Reuse Logic?

Practicing Don't Repeat Yourself (DRY)

Custom React Hooks

You can write your own custom hooks! These are just JavaScript functions that can use React's features!

- We use **custom components** to re-use **UI elements**.
- We use **custom hooks** to re-use **business logic**.

JSConf Talk

Let's Write a Custom Hook!

Writing reusable logic for persisting data.

[StackBlitz Solution](#) | [Inspitation from WDS](#)

Congrats!

You are now a React Devloper! 🎉🎊

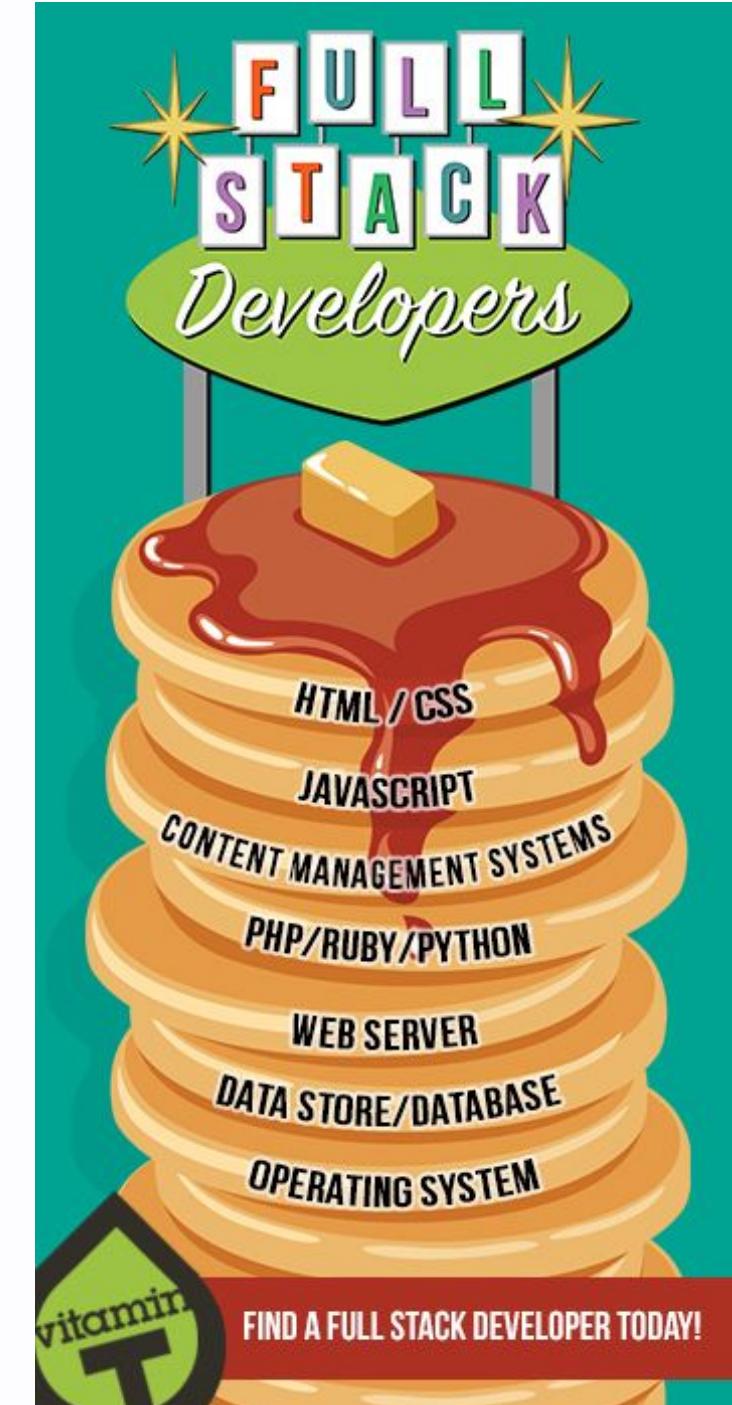
Questions You Should Ask Yourself...

- What even is "frontend development"?
- Where does this fit in to the software stack?
- How can I get my web app out in front of customers?
- What concerns should I have about my web app?
- What else can I do with these skills?
- How much of a raise should I ask for? 

Software Stack

Think of software like a stack of pancakes...

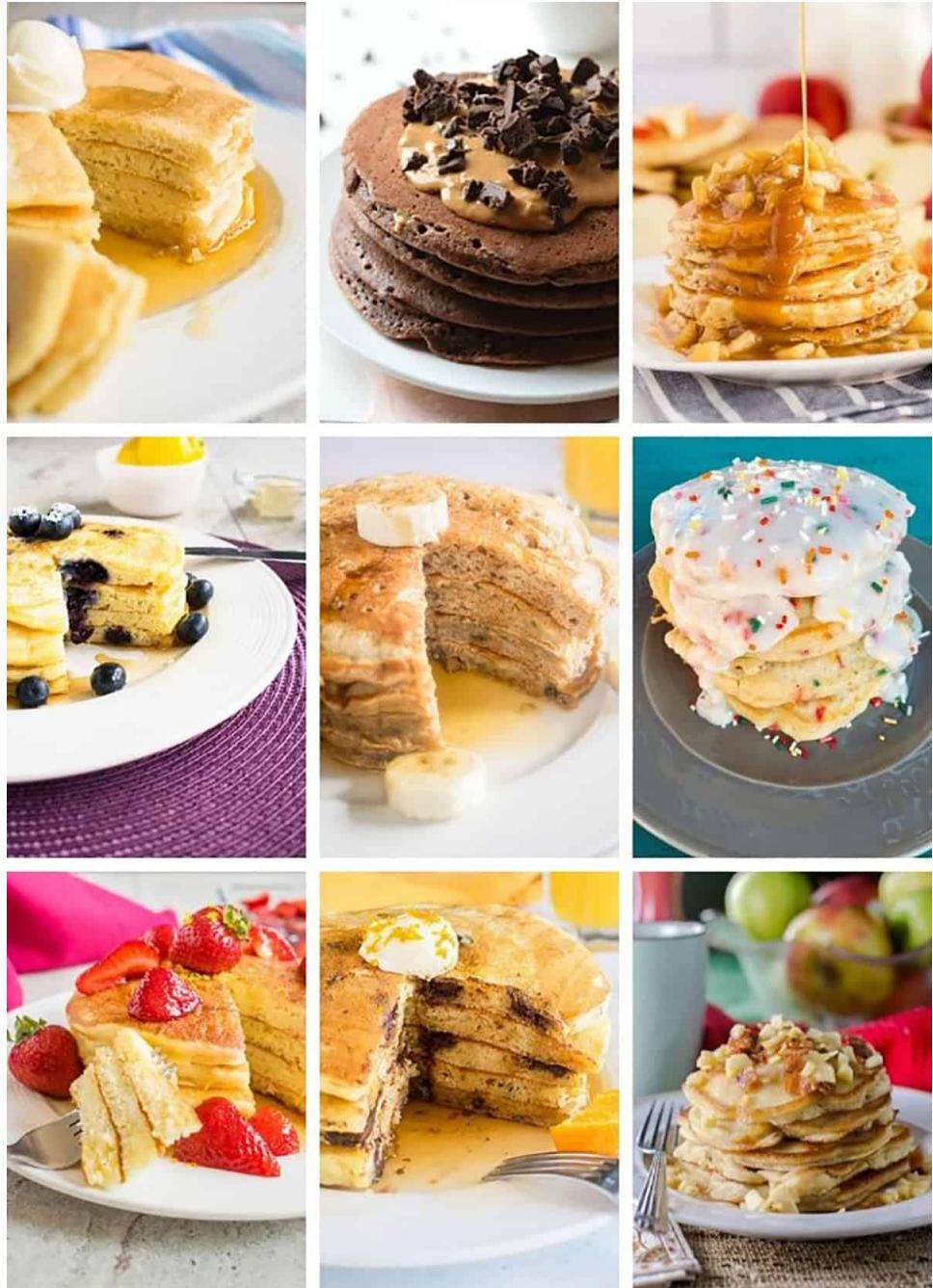
[Image Source](#)



Software Stack

... where each pancake
can be its own flavor...

[Image Source](#)



Software Stack

... and can be cooked its own way...

Image Source

PANCAKES RECIPE

1. EGGS
2. FLOUR
3. MILK
4. SUGAR
5. COOKING OIL
6. SALT
7. BUTTER
8. FRYING PAN
9. WHISK
10. SPATULA
11. BOWL
12. CUP
13. SPOON



Software Stack

... with as many or as few as we want!

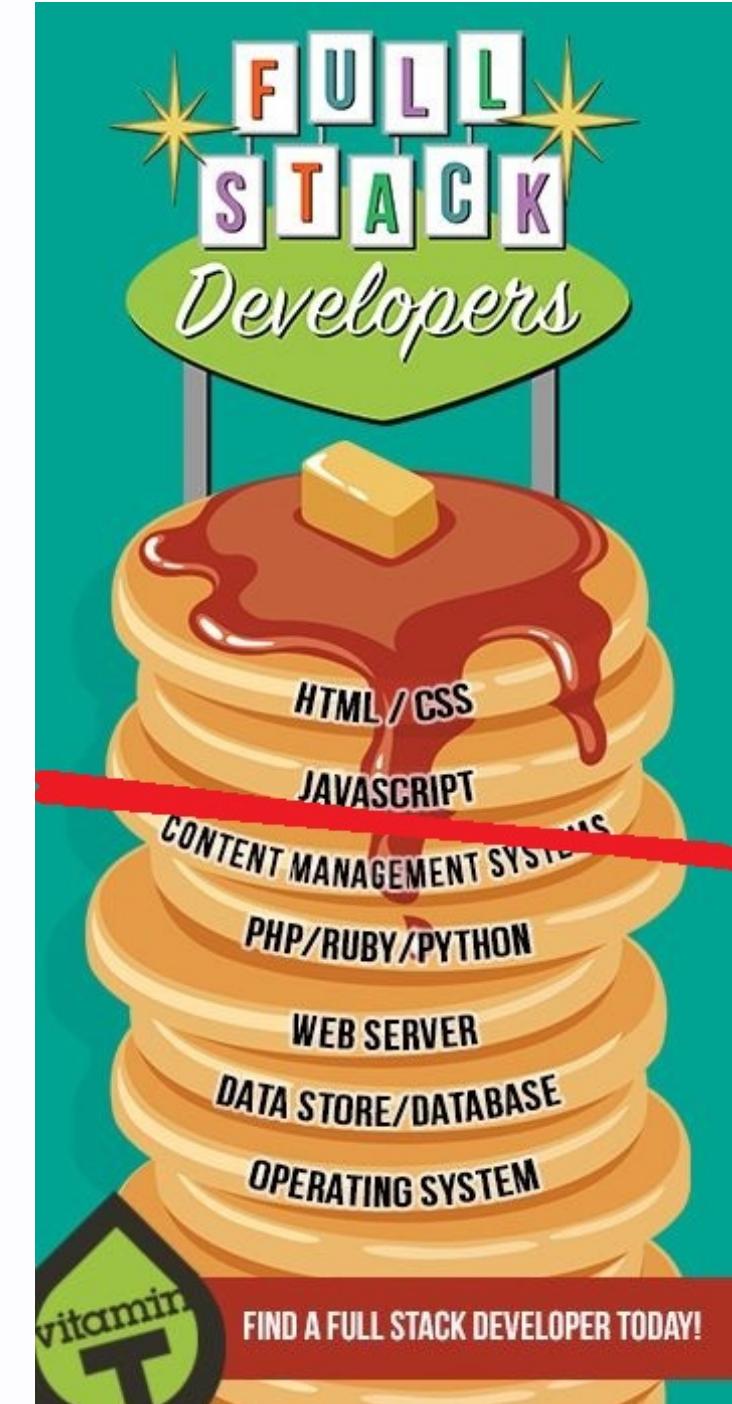
[Image Source](#)



Software Stack

We typically refer to the "frontend" as the content that gets delivered to the user...

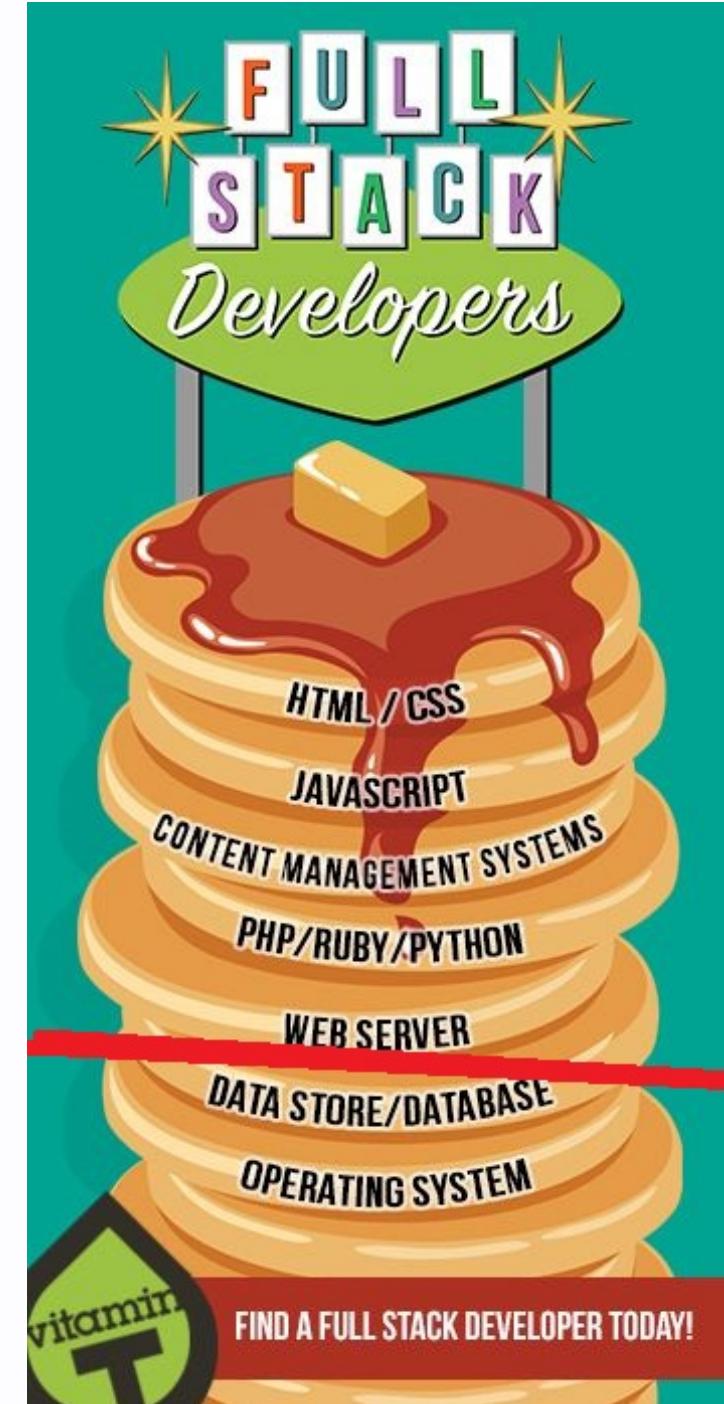
[Image Source](#)



Software Stack

...but this can change
based on your
perspective!

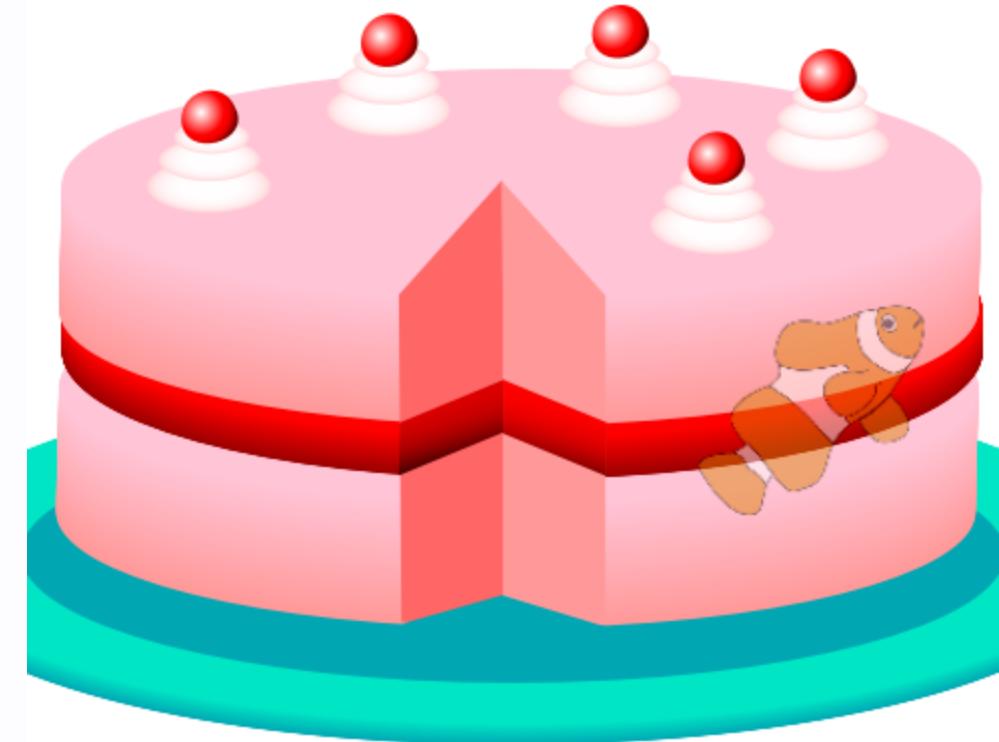
Image Source



The Browser

However, we are constrained to what the browser can interpret...

- HTML
- CSS
- JS



So... is React an Exception?

Facebook is influential, but not *that* influential!

Reminder: JSX

This React component displays Hello World on the webpage using JSX.

```
function Welcome() {  
  return <h1>Hello World!</h1>;  
}
```

Babel transpiles JSX into JS, CSS, and HTML.

Delivery of React App

We don't deliver our JSX code, we deliver HTML, CSS, and JS generated via `npm run build` !

Also, specify a home page (absolute or relative)...

```
{  
  "name": "hw3",  
  "version": "0.1.0",  
  "private": true,  
  "homepage": "https://coletnelson.us/mycoolapp/",  
  "dependencies": {  
    ...  
  }  
}
```

What does this do?

Creates our "build bundle"...

cs571-git > homework > solutions > hw3-solution > build

Name	Date modified	Type	Size
static	10/17/2022 4:53 PM	File folder	
asset-manifest.json	10/17/2022 4:53 PM	JSON File	1 KB
favicon.ico	9/12/2022 8:49 PM	ICO File	4 KB
index.html	10/17/2022 4:53 PM	Chrome HTML Do...	1 KB
logo192.png	9/12/2022 8:49 PM	PNG File	6 KB
logo512.png	9/12/2022 8:49 PM	PNG File	10 KB
manifest.json	9/12/2022 8:49 PM	JSON File	1 KB
robots.txt	9/12/2022 8:49 PM	Text Document	1 KB

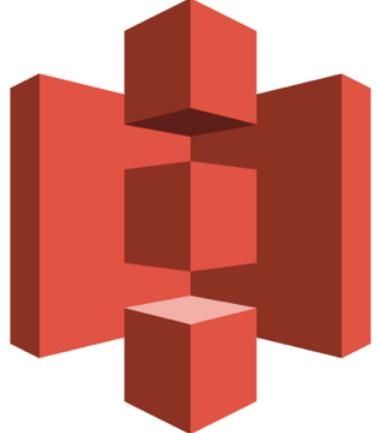
```
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="./favicon.ico" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description" content="Web site created using create-react-app" />
    <link rel="apple-touch-icon" href="./logo192.png" />
    <link rel="manifest" href="./manifest.json" />
    <title>React App</title>
    <script defer="defer" src="./static/js/main.aae268c3.js"></script>
    <link href="./static/css/main.ace4cd11.css" rel="stylesheet">
  </head>
  <body><noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

```

/*! For license information please see main.aae268c3.js.LICENSE.txt */
!function(){var e={694:function(e,t){var n;!function(){ "use strict";var r={};
hasOwnProperty;function l(){for(var e=[],t=0;t<arguments.length;t++) {var n=arguments[t];
if(n){var a=typeof n;if("string"===a||"number"===a)e.push(n);else if(Array.isArray(n)){
if(n.length){var o=l.apply(null,n);o&&e.push(o)}else if("object"===a)if(n.toString===
Object.prototype.toString)for(var u in n)r.call(n,u)&&n[u]&&e.push(u);else e.push(n.
toString())}}return e.join(" ") }e.exports?(l.default=l,e.exports=l):void 0===(n=function
() {return l}.apply(t,[]))|| (e.exports=n)}()},618:function(e,t,n){var r;!function(){ "use
strict";var l=!("undefined"==typeof window||!window.document||!window.document.
createElement),a={canUseDOM:l,canUseWorkers:"undefined"!=typeof Worker,
canUseEventListeners:l&&!(!window.addEventListener&&!window.attachEvent),canUseViewport:
l&&!window.screen};void 0===(r=function(){return a}.call(t,n,t,e))|| (e.exports=r)}(),888:
function(e,t,n){ "use strict";var r=n(47);function l(){ }function a(){ }a.
resetWarningCache=l,e.exports=function(){function e(e,t,n,l,a,o){if(o!==r){var u=new
Error("Calling PropTypes validators directly is not supported by the `prop-types` `package. Use PropTypes.checkPropTypes() to call them. Read more at
http://fb.me/use-check-prop-types");throw u.name="Invariant Violation",u}}function t(){
return e}e.isRequired=e;var n={array:e,bigint:e,bool:e,func:e,number:e,object:e,string:e
,symbol:e,any:e,arrayOf:t,element:e,elementType:e,instanceOf:t,node:e,objectOf:t,oneOf:t
,oneOfType:t,shape:t,exact:t,checkPropTypes:a,resetWarningCache:l};return n.PropTypes=n,
n}},7:function(e,t,n){e.exports=n(888)()},47:function(e){ "use strict";e.exports=
"SECRET_DO_NOT_PASS_THIS_OR_YOU_WILL_BE_FIRED"},463:function(e,t,n){ "use strict";var r=n
(791),l=n(296);function a(e){for(var t=

```

Build Bundle Deployment



Amazon S3



Concerns in Production...

- Reliability
- Performance
- Monitoring
- Business Value of Delivery
- Search Engine Optimization (SEO)

Reliability

Does our code work?

- Manual testing
- Automated testing
 - [Jest](#)
 - [React Testing Library](#)
- Static analysis
 - [TypeScript](#): Used for type-checking
 - [ESLint](#): Used for following best practice

Performance

Does our code work well?

- Be aware of the "bundle size"!
- Our code specifically...
 - [Perf](#)
 - [Profiler](#)
- Our code broadly...
 - [Google Lighthouse](#)
 - [Chrome User Experience Report \(CrUX\)](#)

Monitoring

Does our code *continue to work well?*

- Logging
 - Not `console.log`
 - [Sentry](#)
 - [DataDog](#)
- Cloud Tools
 - Cloud Monitoring Tools
 - [DownDetector](#)

Business Value of Delivery

- Core Questions
 - Are we making money? 
 - Are users making use of new features?
- Analysis Methods
 - A/B Testing
 - Customer Surveys
- Commercial Tools
 - Pendo

Search Engine Optimization (SEO)

The generated HTML simply says...

You need to enable JavaScript to run this app.

What is a search engine crawler supposed to do?

Option: Server Side Rendering [next.js](#)

Your Considerations?

Up Next: React Native

React Native in 100 seconds

F22 Midterm Review

Questions? Gotchas?

What did we learn today?

- How are legacy React apps made?
- How can we write re-usable logic?
- What is the bigger picture for React Apps?
- Preparing for React Native...

Quick Reminder

Students must incorporate at least one additional, meaningful third-party library into at least one of their submissions and provide a short description about its use. **Please submit the separate Canvas assignment.**

You may go *back* and use a library!

Quick Reminder

Please complete the [AEFIS Mid-Semester Survey](#) by March 10th! 😊

This is optional and not required but encouraged!

Feel free submit the Canvas assignment acknowledging that you *can* (but don't have to) complete it.

Questions?