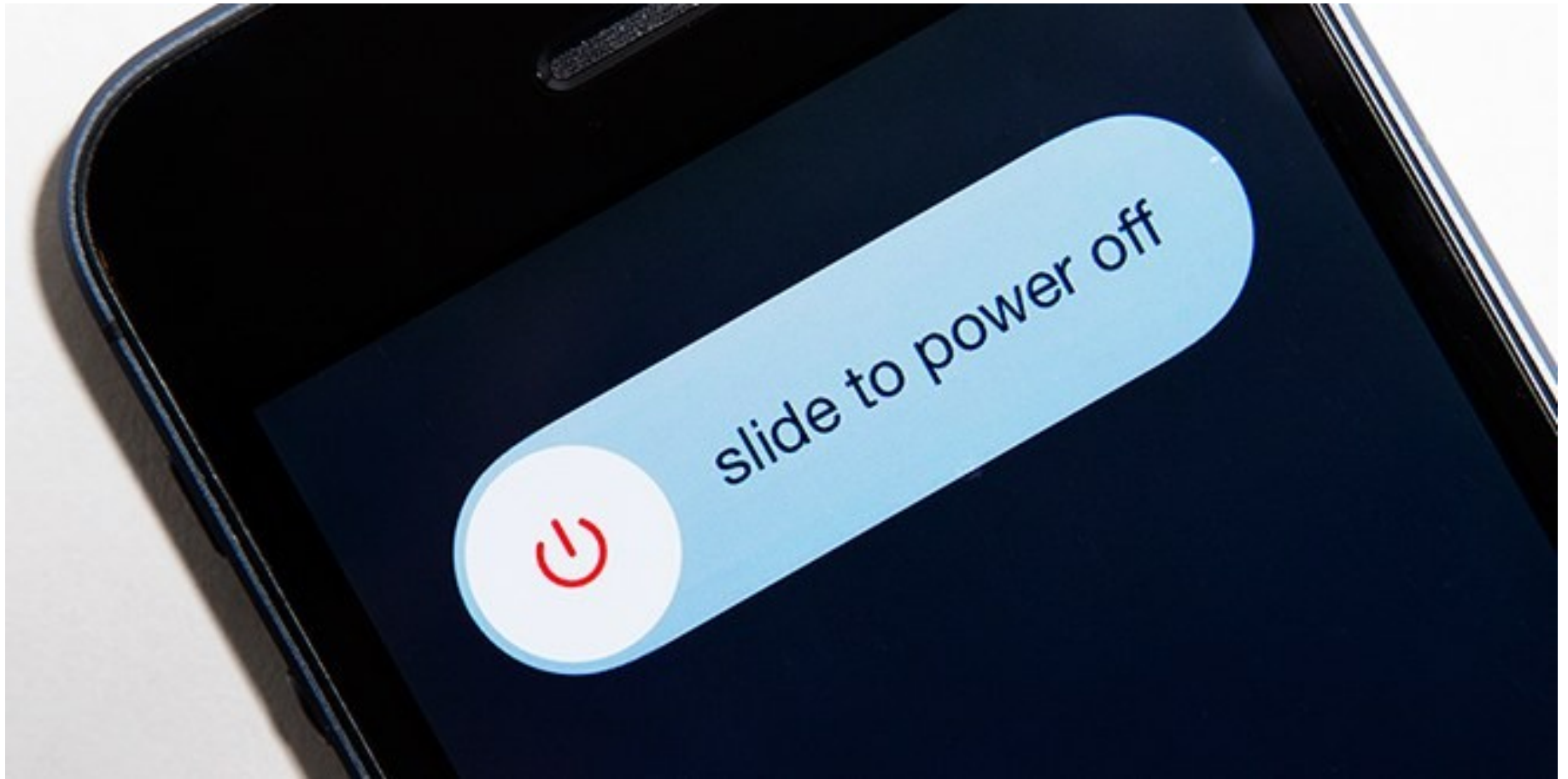
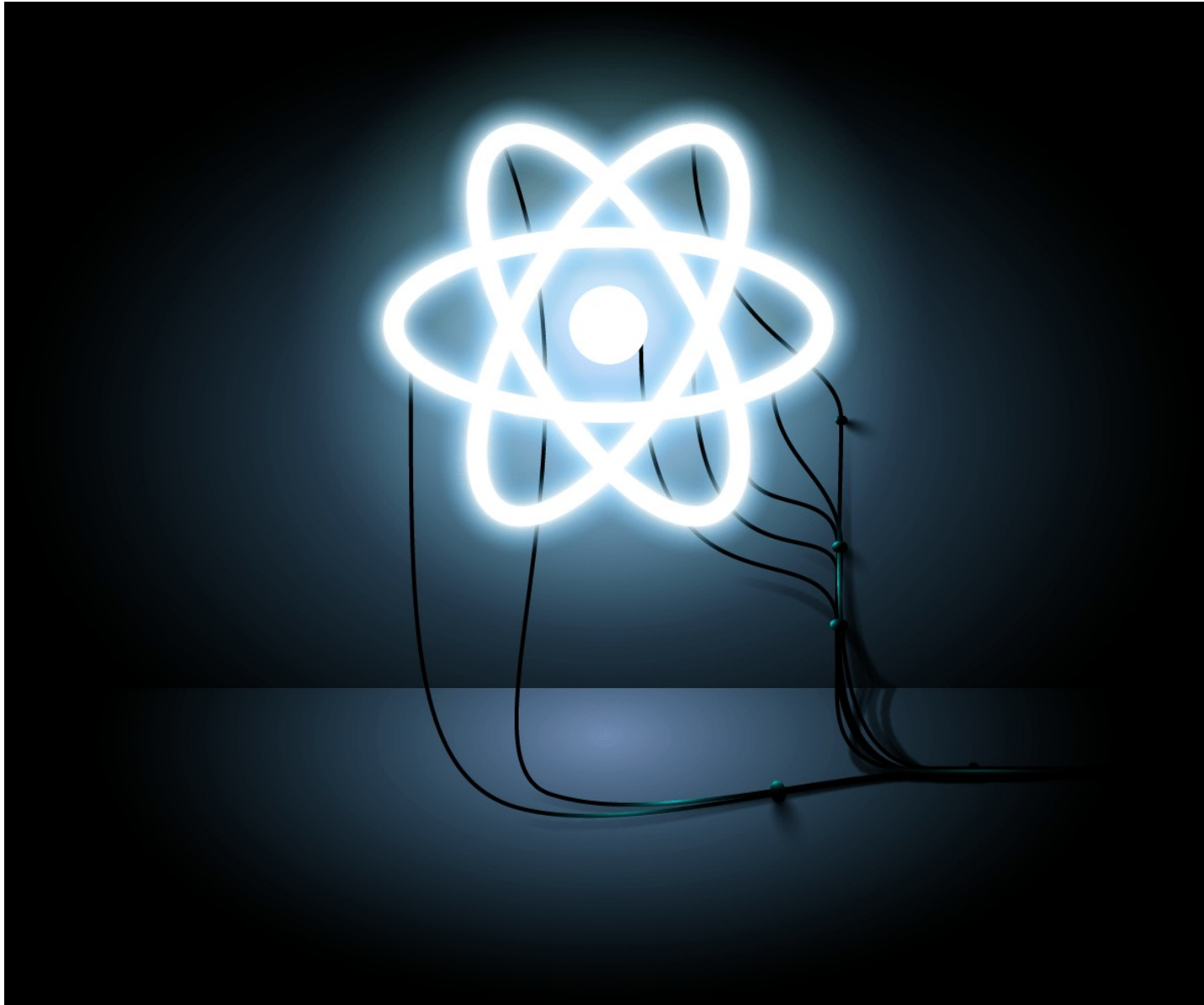


CS385 Mobile Application Development (Lecture 22)



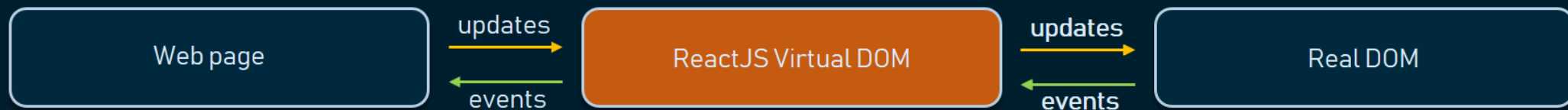
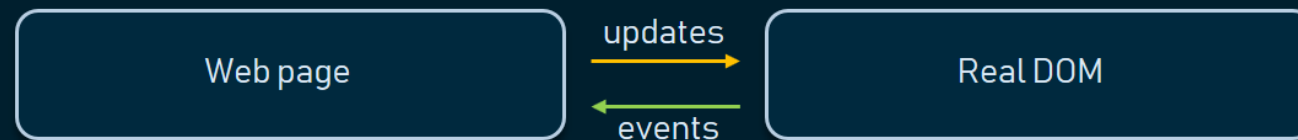
Why React in CS385?



Virtual DOM in ReactJS makes user experience better and developer's work faster

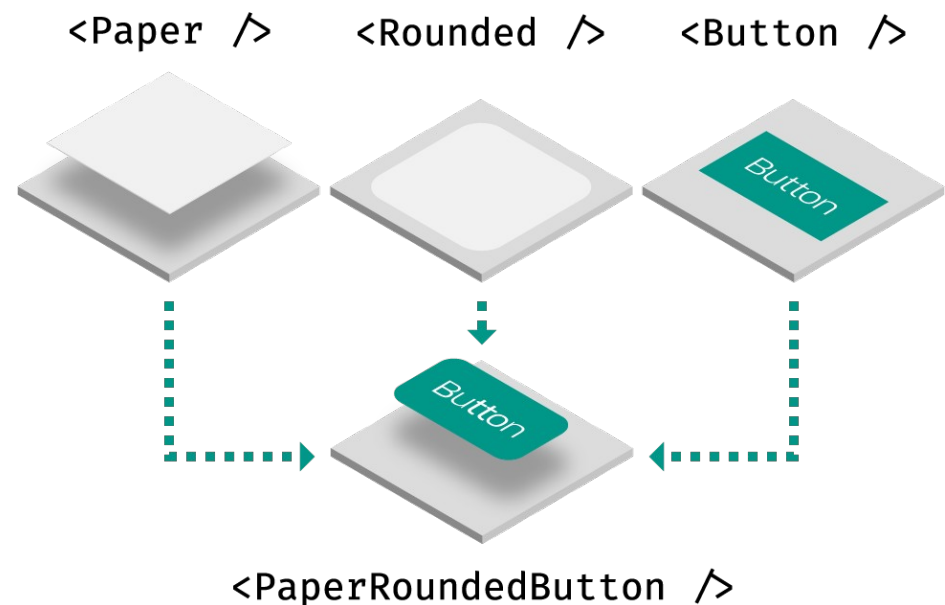
- Unlike other frameworks that work with the Real DOM, ReactJS uses its abstract copy – the Virtual DOM. This makes updates really quick, allowing for the building of a highly dynamic UI.

REAL and VIRTUAL DOMs



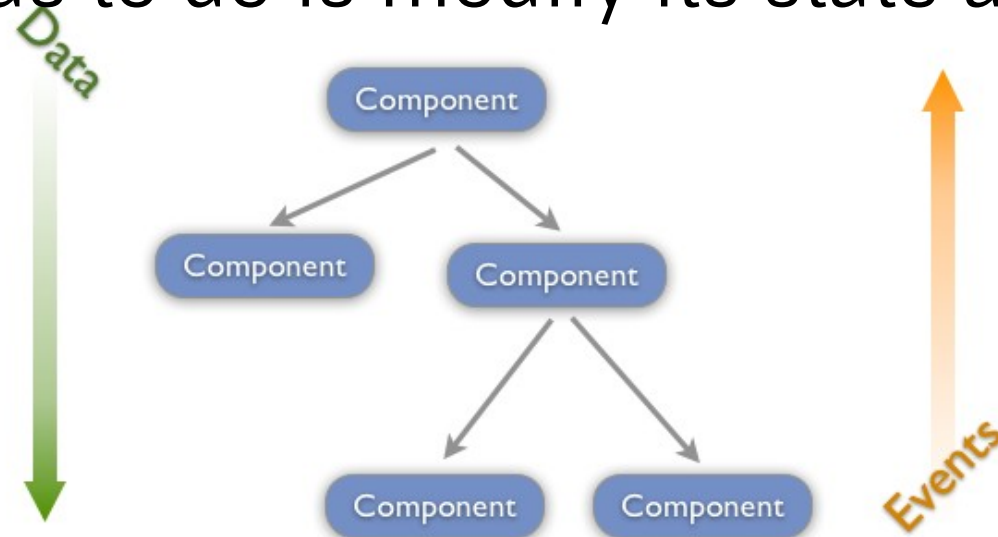
Permission to reuse React components significantly saves time

- React has the ability to reuse code components of a different level anytime, another meaningful time-saving effect.
- Managing updates is easy for developers because all ReactJS components are isolated and change in one doesn't affect others.



One-direction data flow in ReactJS provides a stable code

- ReactJS allows for direct work with components and uses downward data binding to ensure that changes of child structures don't affect their parents. That makes code stable.
- In such a structure, child elements cannot directly affect parent data. To change an object, all a developer needs to do is modify its state and apply updates.



Benefits from ReactJS being Open Source Software

- ReactJS was one of the first JavaScript-connected projects released as open source by Facebook.
- That means that ReactJS uses all advantages of free access – a lot of useful applications and additional tools from off-company developers.

Releases 96

17.0.2 (March 22, 2021) Latest
on 22 Mar

[+ 95 releases](#)

Packages

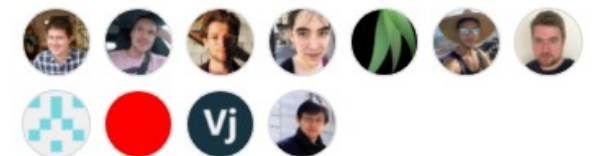
No packages published

Used by 8.2m



[+ 8,181,062](#)

Contributors 1,529



[+ 1,518 contributors](#)

React is 'easier' to teach as a language for new programmers

- React greatly simplifies how to develop interactive applications
- **React greatly simplifies JSON**
- **React greatly simplifies the concept of state**

```
class Messenger extends Component {  
  
  constructor(props)  
  {  
    super(props);  
  
    // declare the state variables for the Messenger component  
    this.state = {currentMessage: "",messageError: "",errors:true};  
  
    // bind methods where required.  
    this.handleMessageBoxChanged = this.handleMessageBoxChanged.bind(this);  
    this.handleClick = this.handleClick.bind(this);  
    this.deleteMsg = this.deleteMsg.bind(this);  
  }  
}
```

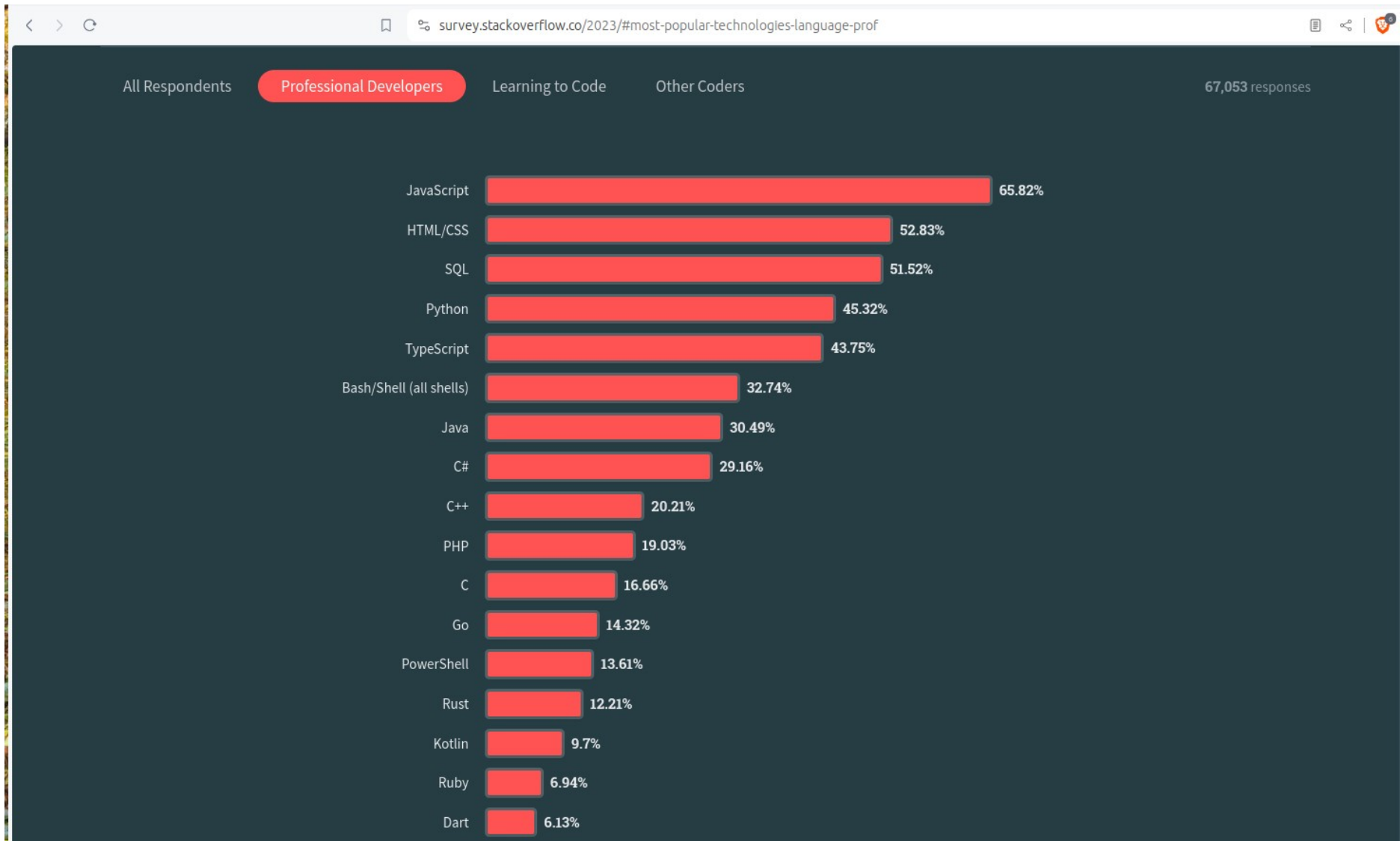
Pros and Cons

- At the end of the day – the aim of the module is to teach the concepts of Mobile Application Development.
- So, in some ways, the language and framework is NOT the most important factor.
- Learning concepts about state, event handling, communication, working with objects are important learning outcomes.
- Functional React brings you closer to industry (but could be a bit more difficulty for beginners to CS385)

JSX – Can be a barrier....

- **ReactJS uses JSX. It's a syntax extension that allows HTML with JavaScript mixed together.**
- **This approach has its own benefits, but some members of the development community consider JSX as a barrier, especially for new developers. Developers complain about its complexity in the learning curve.**

But once you have started with Javascript ... there are LOTS of options



Web frameworks and technologies



Node.js and React.js are the two most common web technologies used by all respondents.

Professional Developers use both fairly equally and those learning to code use Node.js more than React (52% vs. 48%). jQuery and Express are the next two popular web technologies for all respondents, and jQuery is used more by Professional Developers than those learning to code (24% vs 18%), whereas Express is used more by those learning than professionals (25% vs. 20%).

Next.js moved from 11th place in 2022 to 6th this year, likely driven by its popularity with those learning to code.

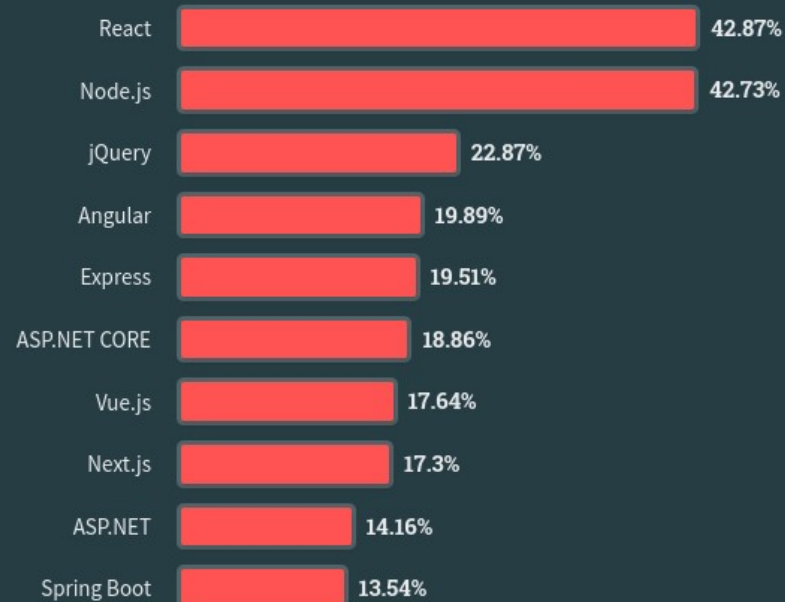
All Respondents

Professional Developers

Learning to Code

Other Coders

56,742 responses



1 Semester, 5 credits

- 22 lectures = 18.5 hours lectures
- 10 labs = 20 hours
- Total teaching time approx 40.
- **This is just a 5 day training course!**

Questions that I am asked when providing an IT job reference for a CS385 student

- In the position you have known the candidate do they require close supervision or were they **able to work on their own**?
- Could you comment on the candidate's ability to **work within a team environment**?
- Did the candidate demonstrate their ability to **meet deadlines under pressure**?

Questions that I am asked when providing an IT job reference for a CS385 student (2)

- Could you please comment on the **candidate's interpersonal skills**?
- Were there ever **lateness or absenteeism problems**?
- **Would you re-hire** if given the opportunity?
- What question would you ask the candidate in an interview?



Things I cannot change about CS385 in Maynooth.....

- **The timetable** (I cannot change labs or lectures timetable(s) ... sorry)
- **The classroom venue** (I have no control over the venues)
- **Selection:** I have no influence regarding whether the module is compulsory in your programme or not

Student Survey

[Optional Participation]

Enter the event ID
and URL when it is
displayed (in a
moment)

- All questions are answered anonymously (so you can be truthful and honest!)

Join at:
vevox.app

ID:
175-293-446

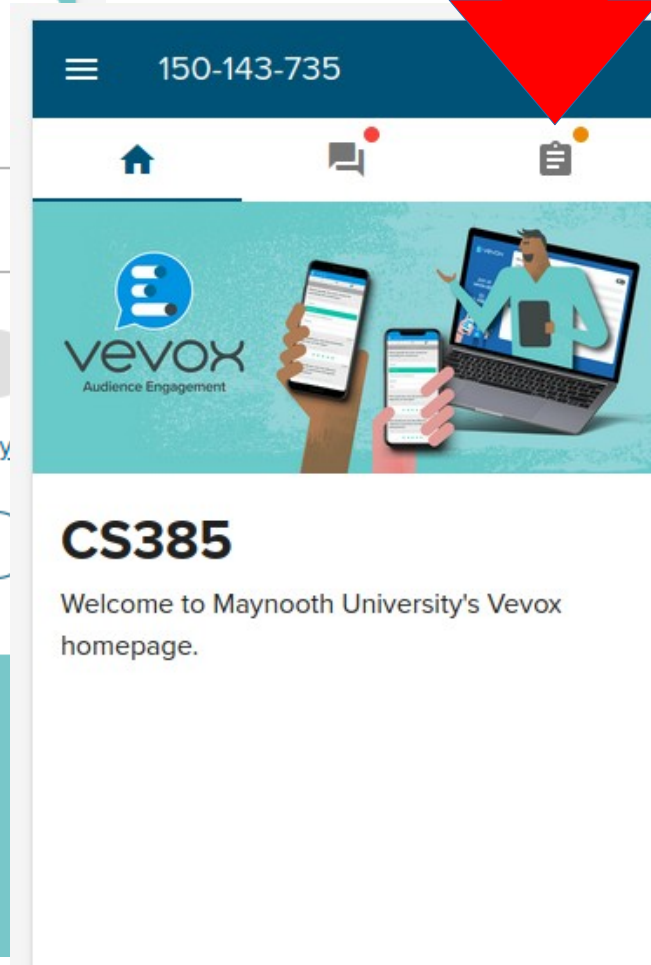


Student Survey

[Optional Participation]

Join at:
vevox.app

ID:
175-293-446

A screenshot of the Vevox app's session entry screen. It features the Vevox logo at the top, followed by a text input field labeled 'Enter Session ID'. Below the field is a grey 'JOIN SESSION' button. At the bottom, there is a link to 'Terms & Privacy Policy' and an 'ABOUT VEVOX' button. A red arrow points from the ID '175-293-446' to the 'Enter Session ID' field.

Topic 11 – in class quiz

- **9 questions – similar standard and style to Lab Exam 3**
- This popup quiz only provides questions with NEW content (different from Lab Exam 2 and Lab Exam 1)
- Short explainer of answers after the quiz.

Lab Exam 3 – IN CLASS DEMO

- PASSWORD **final**
-
- You can work together (there are two attempts).
- There is no shuffling of questions or options – so everyone will see the questions in the same order (this is to make discussing our answers easier)
- 25 minutes

Discussion of Lab Exam 3 – IN CLASS DEMO – answers



Q1

Version 1 (latest)

Question 1

Answer saved

Marked out of
1.00

GetOwnPropertyNames: What is the FINAL line rendered by the application below - specifically by line 16 - for the array shown?

```
3 function App() {
4   let arrObjects = [
5     { name: "Luton", pos: 17, std: "Kennilworth Road" },
6     { name: "Leeds United", pos: 18, std: "Elland Road" },
7     { name: "Crystal Palace", pos: 13, std: "Selhurst Park" },
8   ];
9
10  return (
11    <>
12    {arrObjects.map((x, i) => (
13      <p key={i}>
14        {Object.getOwnPropertyNames(x).map((y, i) => (
15          <p key={i}>
16            {y} {x[y]}
17          </p>
18        ))}
19      </p>
20    ))}
21    </>
22  );
23 }
```

- ☒ A. std Selhurst Park

Which of the following options below indicate that is rendered in the FINAL iteration of the map function shown below?

```
4  function App() {
5    const a1 = [
6      { id: 1, name: "John", ex1: 25, ex2: 3, ex3: 10 },
7      { id: 2, name: "Jane", ex1: 30, ex2: 10, ex3: 25 },
8      { id: 3, name: "Colm", ex1: 28, ex2: 15, ex3: 25 },
9    ];
10
11   const a2 = a1.map((el) => ({
12     id: el.id,
13     name: el.name,
14     x: el.ex1 + el.ex2 + el.ex3,
15   }));
16
17   return (
18     <>
19     <ul>
20       {a2.map((obj, index) => (
21         <li key={index}>{JSON.stringify(obj)}</li>
22       ))}
23     </ul>
24   </>
25   );
26 }
```

- ☐ A. • `{"id":3,"name":"Colm","ex1":28,"ex2":15,"ex3":25,"x":68}`
- ☒ B. • `{"id":3,"name":"Colm","x":68}`

Which ONE of the options below correspond to the output generated by the fully working React Application code below?

```
4  function App() {
5    const originalArray = [
6      { type: "can", weight: 500, factor: 1 },
7      { type: "van", weight: 800, factor: 2 },
8      { type: "truck", weight: 1500, factor: 3 },
9    ];
10   const newArray = originalArray.map((element) => ({
11     ...element,
12     factorised: element.weight * element.factor,
13   }));
14
15   return (
16     <>
17       <ul>
18         {newArray.map((obj, index) => (
19           <li key={index}>{JSON.stringify(obj)}</li>
20         ))}
21       </ul>
22     </>
23   );
24 }
```

☒ A.

- {"type":"can","weight":500,"factor":1,"factorised":500}
- {"type":"van","weight":800,"factor":2,"factorised":1600}
- {"type":"truck","weight":1500,"factor":3,"factorised":4500}

What is rendered by the final iteration of the map function show below?

```
3 function App() {
4   let planes = [
5     { make: "Airbus A321-c", seats: 236, owner: "Aer Lingus" },
6     { make: "Airbus A321-d", seats: 242, owner: "United" },
7     { make: "Boeing B737-7-a", seats: 126, owner: "Emirates" },
8     { make: " Boeing B737-8-c", seats: 189, Owner: "CS385 Airlines" },
9   ];
10  let extraPlanes = [...planes, ...planes];
11  return (
12    <>
13    {extraPlanes.map((c, i) => (
14      <li key={i}>
15        ..... {c.owner} === undefined && <b>{c.make},Unknown</b>}
16        ..... {c.owner !== undefined && (
17        ..... <b>
18        ..... {c.make},{c.owner}
19        ..... </b>
20        ..... )}
21      </li>
22    )]}
23    </>
24  );
25 }
```

- ☐ A. Boeing B737-8-c,Undefined
- ☒ B. Boeing B737-8-c,Unknown
- ☐ C. Boeing B737-7-a,Emirates
- ☐ D. Boeing B737-7-a,Emirates
- ☐ E. Boeing B737-8-c,CS385 Airlines

What is rendered on the SECOND to last/final iteration of the React Javascript code below?

```
3 function App() {
4   let places = [
5     { Country: "France", urban: ["Paris", "Nice", "Lyon"] },
6     {
7       Country: "China",
8       urban: ["Chongqing", "Shanghai", "Beijing", "Chengdu", "Guangzhou"],
9     },
10  ];
11  return (
12    <>
13      <ul>
14        {places.map((p, index) => (
15          <li key={index}>
16            <ul>
17              {p.urban.map((u, tindex) => (
18                <li key={tindex}>
19                .....{p.Country},{tindex + 1},{u}
20                </li>
21              )})}
22            </ul>
23          </li>
24        )})}
25      </ul>
26    </>
27  );
28 }
```

- ☒ A. • China,4,Chengdu

What is rendered to the screen by the application below?

```
2  function App() {  
3    let original = [  
4      { x: 4, y: 5 },  
5      { x: 102, y: 125 },  
6      { x: 99, y: 125 },  
7      { x: 97, y: 125 },  
8      { x: 298, y: 1000 },  
9    ];  
10  
11    function fuzzyFindObject(needle) {  
12      return function (haystack) {  
13        return (  
14          Math.abs(haystack.x - needle.x) < 5 &&  
15          Math.abs(haystack.y - needle.y) < 5  
16        );  
17      };  
18    }  
19    let toFind = { x: 100, y: 121 };  
20    let n = original.filter(fuzzyFindObject(toFind));  
21  
22    return (  
23      <>  
24      |   <h1>{n.length}</h1>  
25      | </>  
26    );  
27  }
```

What is rendered to the screen by Line 17 in the application below? Please note that this question combines **reduce()** and **getOwnPropertyNames()** in the same question. There will be an explanation in the automated feedback for the question.

```
3 function App() {
4   let coordinates = [
5     { x: 4, y: 3, i: 0 },
6     { x: 7, y: 4, z: 3, i: 1 },
7     { x: 7, y: 14, z: 13, t: 4, i: 2 },
8     { x: -4, y: 30, i: 3 },
9   ];
10
11   function coordFunction(acc, obj) {
12     let c = Object.getOwnPropertyNames(obj).length;
13     return acc + c + obj.i;
14   }
15   return (
16     <>
17     <h1>{coordinates.reduce(coordFunction, 0)}</h1>
18     </>
19   );
20 }
```

- ☐ A. 4
- ☐ B. 10
- ☐ C. 17
- ☒ D. 21
- ☐ E. 0

- In this code a reduce function is applied to the coordinates array. This means we iterate over the coordinates array and we start an accumulator variable with the value of zero.
- For each object in the coordinates array we check access the properties using `getOwnPropertyNames`. So for object 0 { x: 4, y: 3, i: 0 } this is $c = 3$ (as there are three properties), for the object 1 { x: 7, y: 4, z: 3, i: 1 } this is $c = 4$ as there are four properties and so on. Each time we access the object we also access the property `i` in the object. So the full calculation is as follows
- Object 0 $c = 3, i = 0, acc = 3$
- Object 1 $c = 4, i = 1, acc = 8$ ($3 + 1 + 4$)
- Object 2 $c = 5, i = 2, acc = 15$ ($8 + 5 + 2$)
- Object 3 $c = 3, i = 3, acc = 21$ ($15 + 6$)

Lab Exam 3

- 8 questions on NEW content from Topic 10
- 4 questions from previous Lab Exam 2 and Lab Exam 1 (the most frequently incorrect questions from both lab exams)
- Very slight differences with demo lab exam 3
 - for example may not use conditional rendering for a question or may not use useState.

Lab Exam 3 - CONTENTS

- **Object deconstruction**
- **Nested map functions** (including one with a `reduce0` function in it - lab exam 3 demo)
- **fuzzy filter** question
- **Object equality**
- **GetOwnPropertyNames** - what is the final render or second to final render output?
- **Undefined properties** and **conditional rendering**.
- **GetOwnPropertyNames** and conditional rendering.
- **4 questions from previous lab exams** - look at the questions you have got wrong.

Demo Lab Exam 3

- Available now until Friday 15th December at 11:00
- Contains 10 questions (excludes the four additional questions from the previous lab exams)
- Unlimited time available (until Lab Exam 2) – two attempts

**Thanks for all your work and
attention over the past 12 weeks.
Best wishes for the remainder of your
time here in Maynooth University**



**MAY
THE
FORCE
BE WITH
YOU**

"It's all going to
be alright in the
end, and if it's
not alright, it's
not the end"

(by Ol Parker, via Wittertainment)