

# What's new in React 18?

A quick look into React 18 features

# Caution

Not an exhaustive list



# New features

createRoot API

Automatic Batching

Concurrent React: startTransition



@shrutikapoor08

# Alpha version



# Easy upgrade!



# Easy upgrade!



```
npm install react@alpha react-dom@alpha
```

# React Working Group



# createRoot API



# ReactDOM.render



~~ReactDOM.render~~

ReactDOM.createRoot



@shrutikapoor08



```
import ReactDOM from 'react-dom';
import App from 'App';

const container = document.getElementById('app');

ReactDOM.render(<App />, container);
```



```
import ReactDOM from 'react-dom';
import App from 'App';

const container = document.getElementById('app');

// create a root
const root = ReactDOM.createRoot(container);

//render app to root
root.render(<App />);
```

# createRootAPI

Root no longer opaque to the user.

Render using Root instead of HTML container.



@shrutikapoor08

# Root

Not a new feature but part of the upgrade.

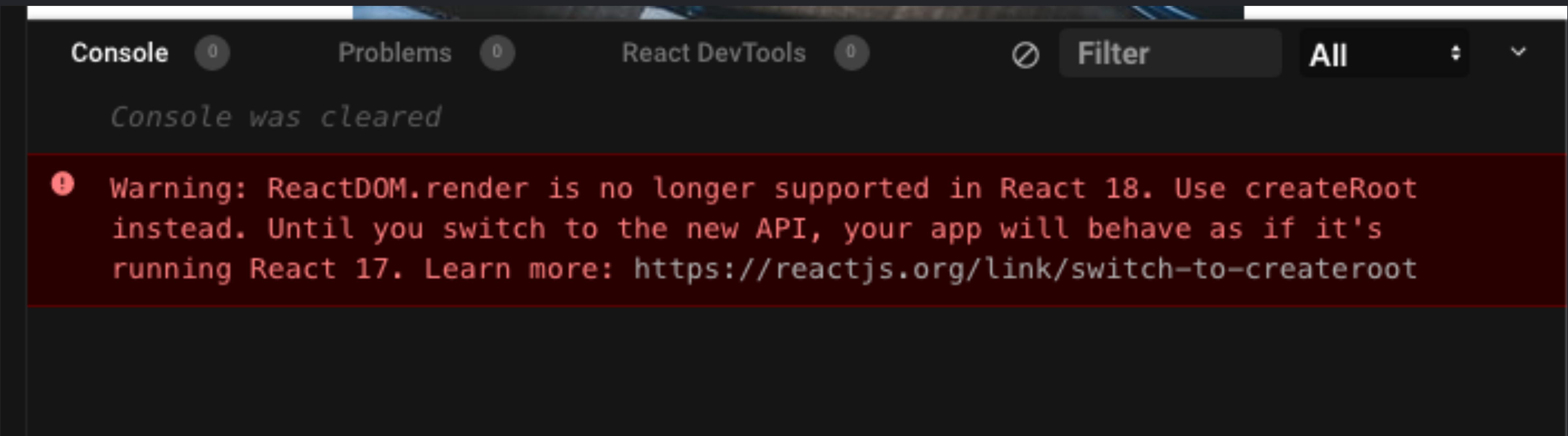
Enables Automatic Batching.

Makes all new features available.



@shrutikapoor08

Current way of rendering will still work but will be treated with a deprecated warning



# Automatic Batching



# Automatic Batching

“Batch” multiple state updates into one.



@shrutikapoor08



```
setIsFetching(false);  
setError(null);  
setFormStatus('success')
```

CODE



# Helps performance

- <https://codesandbox.io/s/clever-shape-doyy0?file=/src/index.js>



@shrutikapoor08

# Automatic Batching

With automatic batching, all updates, even within promises, setTimeouts, will be batched.

Helps to prevent multiple re-renders.



@shrutikapoor08



```
function handleClick() {
  console.log("== click ==");
  setCount((c) => c + 1); // Does not re-render yet
  setFlag((f) => !f); // Does not re-render yet
  // React will only re-render once at the end (that's batching!)

  const timeoutCallback = () => {
    // Previously, batching didn't work inside timeouts, fetch, promises.
    // These two setStates caused re-render in React 17.
    // With React 18, these are now batched.
    setCount((c) => c + 1);
    setFlag((f) => !f);
  };

  setTimeout(timeoutCallback, 1000);
}
```

# Batching - In simple words

From Dan Abramov. React Working Group discussions

- Non batched = getting groceries on demand
- Batching = planned grocery trip



@shrutikapoor08

# Behind the scenes

Previously, React used to re-render immediately in this case.

With automated batching in React 18, React will wait until the end of the microtask



@shrutikapoor08

# Automatic Batching

Available out of the box

Can opt out using `flushSync`



@shrutikapoor08

# startTransition



# startTransition

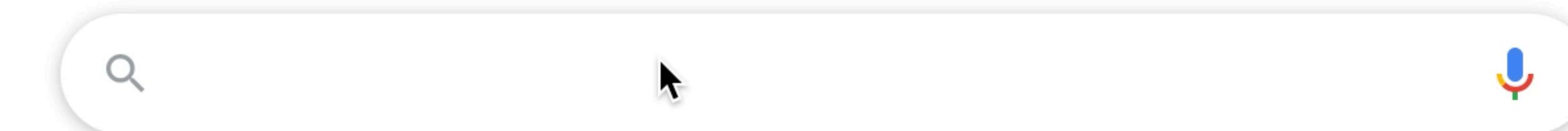
Prioritize updates

Mark non-urgent UI updates as “transitions”



@shrutikapoor08

# Google



Google Search

I'm Feeling Lucky

# Google

Google Search

I'm Feeling Lucky

# startTransition

- Marking non-urgent UI updates as “transitions”
- Helps in letting React “interrupt” when urgent updates come in.



@shrutikapoor08



```
import { startTransition } from 'react';

// Urgent: Show what was typed
setInputValue(input);

// Mark any state updates inside as transitions
startTransition(() => {
  // Transition: Show the results
  setSearchQuery(input);
});
```

What if we want to do something while  
transition is happening?



```
import { useTransition } from 'react';  
  
const [isPending, startTransition] = useTransition( );
```

# Examples



<https://react-fractals-git-react-18-swizec.vercel.app/>



@shrutikapoor08

<https://codesandbox.io/s/usetransition-tab-test-forked-dxukf?file=/src/App.js>



# Where to add startTransition?

# Where to use

Slow rendering: React needs to perform a lot of work.

Slow network: React is waiting for some data from the network. Needs Suspense



@shrutikapoor08

# Determining candidates for startTransitions

Urgent updates?

Non-urgent updates?

Which states can be “transitioned”?

What to do when we are “transitioning”?



@shrutikapoor08

CODE



# Suspense



# Suspense

- Puts into a suspend mode with a fallback until everything inside the suspense is ready to be rendered.
- Can be used to render a loading component



@shrutikapoor08



```
<Suspense fallback={<Spinner />}>
  <ProfilePage />
</Suspense>
```



@shrutikapoor08

# Suspense

- React 18 includes foundational work.
- Data fetching not included yet.



@shrutikapoor08

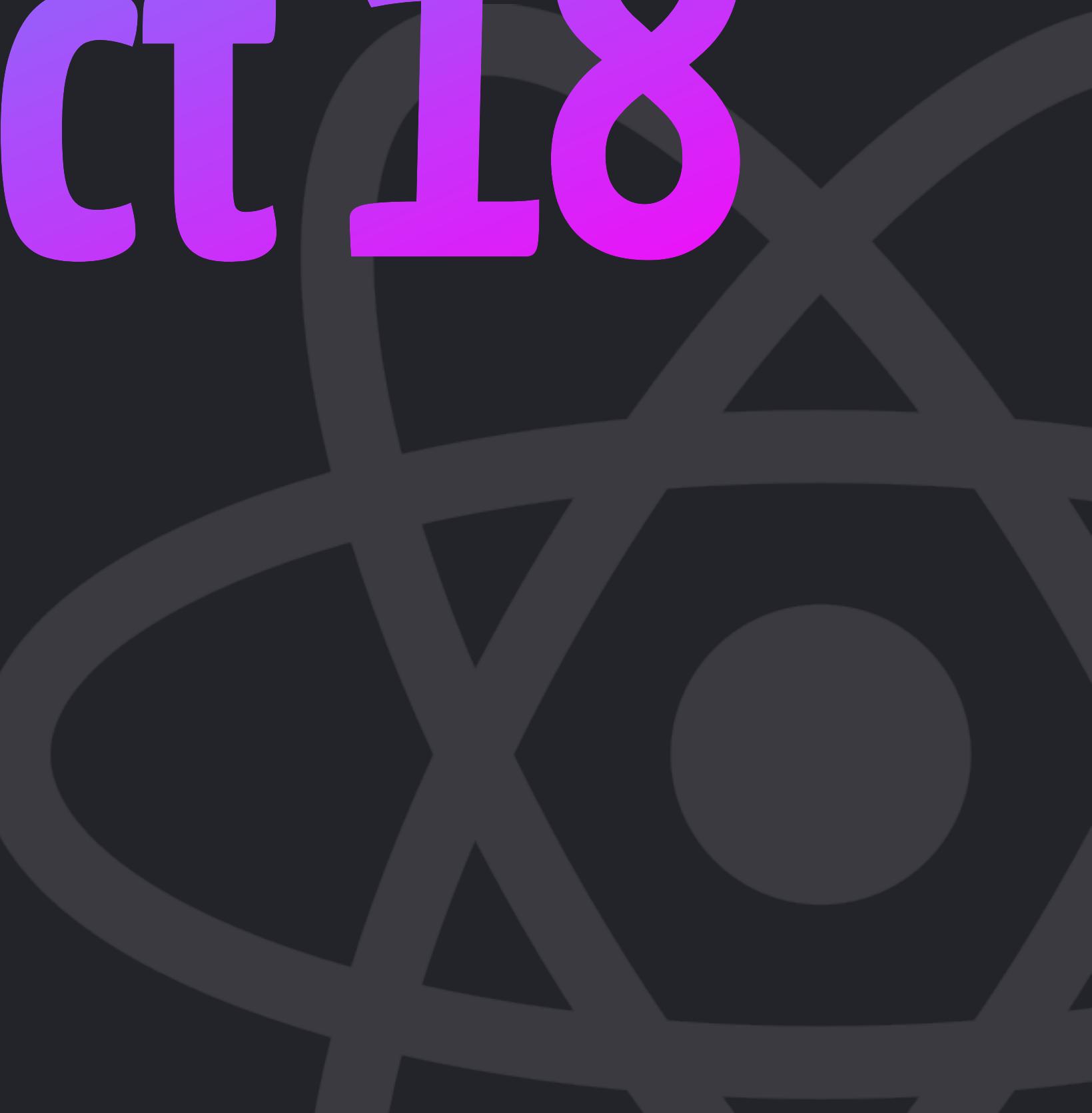
# Big Takeaway

## Concurrent

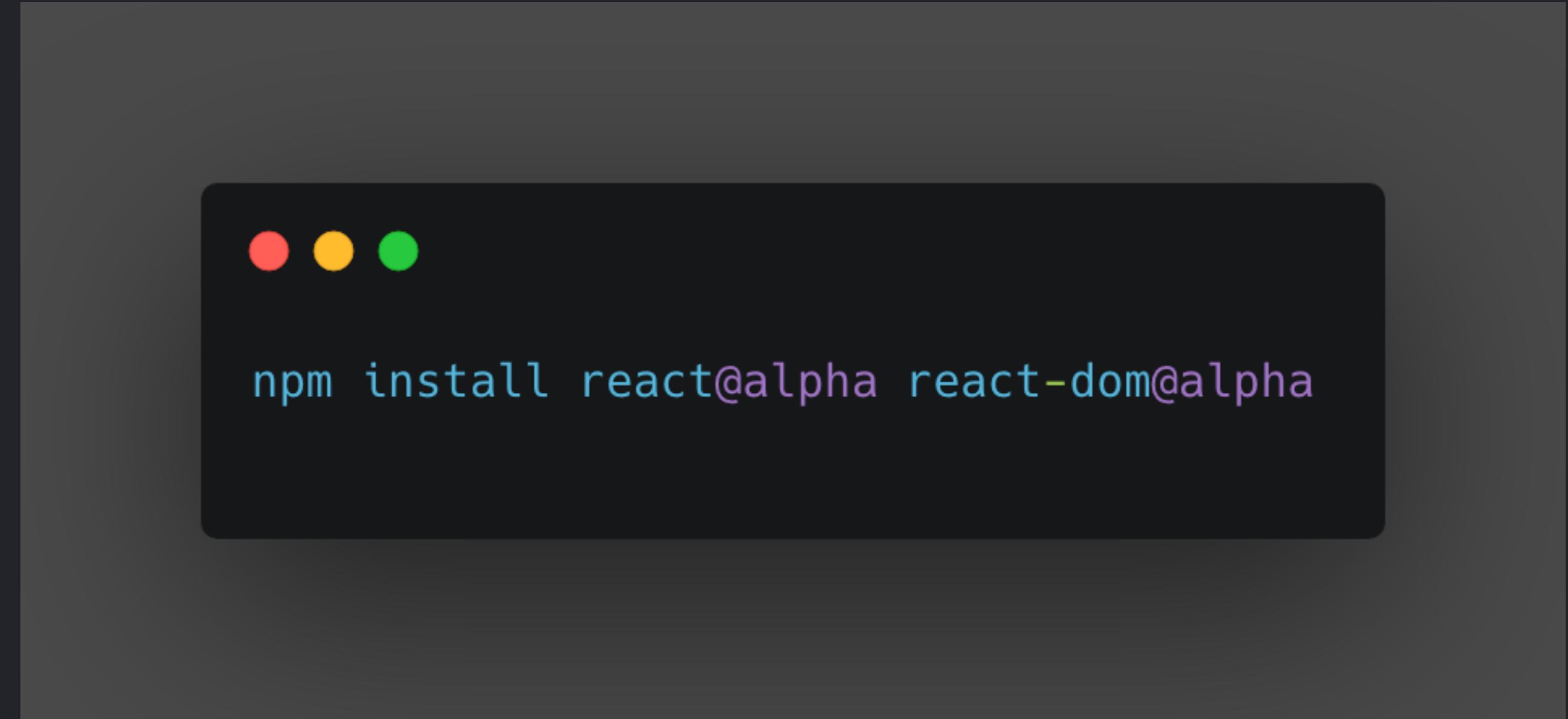
### React



# Moving to React 18



# How to move to React 18?



@shrutikapoor08

# How to move to React 18?



```
npm install react@alpha react-dom@alpha
```



```
const rootElement = document.getElementById("root");
ReactDOM.createRoot(<App />, rootElement).render(<App />);
```

# Noteworthy discussion threads

# Glossary + Explain Like I'm Five #46

sylwiavargas started this conversation in Q&A



sylwiavargas on Jun 9

Collaborator

edited ▾ ⋮

**tl;dr This thread contains plain English and explain-like-I'm-five definitions of the terms and concepts used throughout other discussions. Scroll to comments to find (or add) a term and explanations connected to it ✨**

First of all, thank you for amazing discussions and articles shared here so far. It has been an amazing reading adventure ❤️

As has been pointed out in another thread, complicated technical terminology often poses an obstacle for folks to learn new technology. I'd like to invite you all to **co-create a glossary of terms that are crucial for this release** but that are not commonly-used in the beginner/intermediate courses and tutorials.

**In other words, let's help the new React developers out there get as excited about this release as we are!**

## Structure

Here's a structure I propose:

- Each of the terms will be located in its own comment section below, which will allow keeping explanations of the same terms

# Real world example: adding startTransition for slow renders #65

rickhanlonii announced in Deep Dive



rickhanlonii on Jun 22

Maintainer

edited by gaaron ▾ ⋮

In React 18 we announced a new `startTransition` API and shared a [high-level overview](#) of the problem it solves. In this post, we're going to dive into a real-world example of speeding up a slow update with `startTransition` using a large-scale open source React app.

## tl;dr

If you don't feel like reading the whole page, here's a few videos to show the difference on a fast and a slow computer.

- ▶ Fast computer: No `startTransition`
- ▶ Fast computer: With `startTransition`
- ▶ Slow computer: No `startTransition`
- ▶ Slow computer: With `startTransition`

## Overview

<https://github.com/reactwg/react-18/discussions/65>

We covered

Features of React 18 - createRoot API, Automatic Batching,  
startTransition

Demo - Using these features in your app

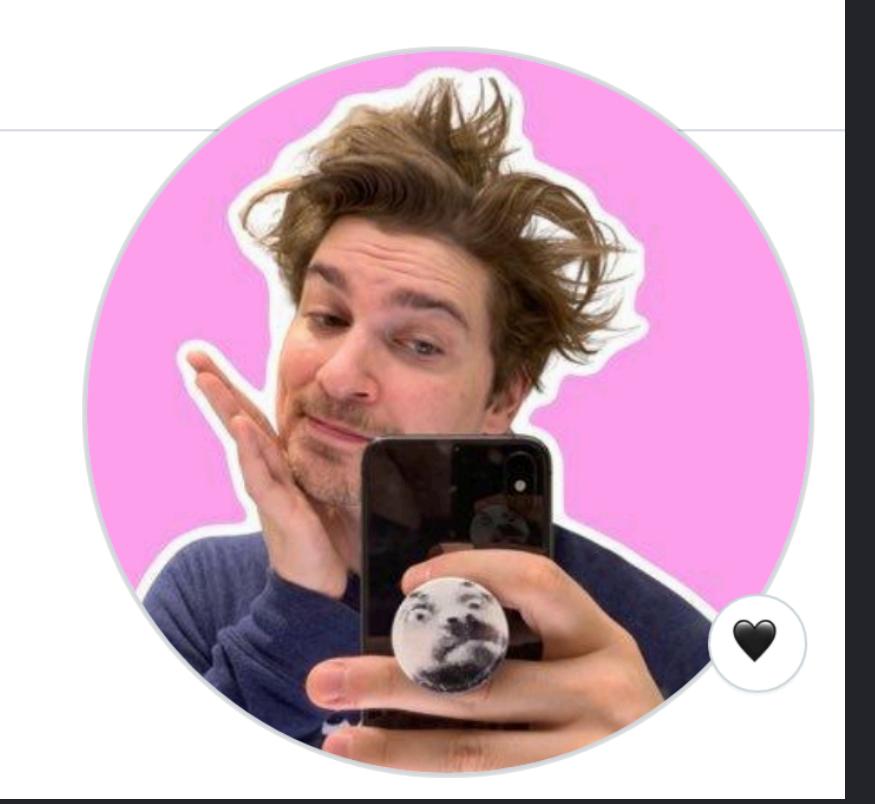


@shrutikapoor08

# Acknowledgements



@shrutikapoor08



# Acknowledgements

## Rick Hanlonii



@shrutikapoor08



# Thank You



@shrutikapoor08

