

Hook Me UP



Carlos Mafla
carlosmafla.com



meetup.com/reactcali

About Me

- Web Developer (Backend, Frontend, DevOps, etc)
- 15+ years of experience
- @gigo6000 (Twitter, Instagram, Github)
- youtube.com/CarlosMafla

Proposal: Hooks

Use all React features without a class.
Reuse stateful logic between components.
Opt-in and 100% backwards-compatible.

reactjs.org/hooks



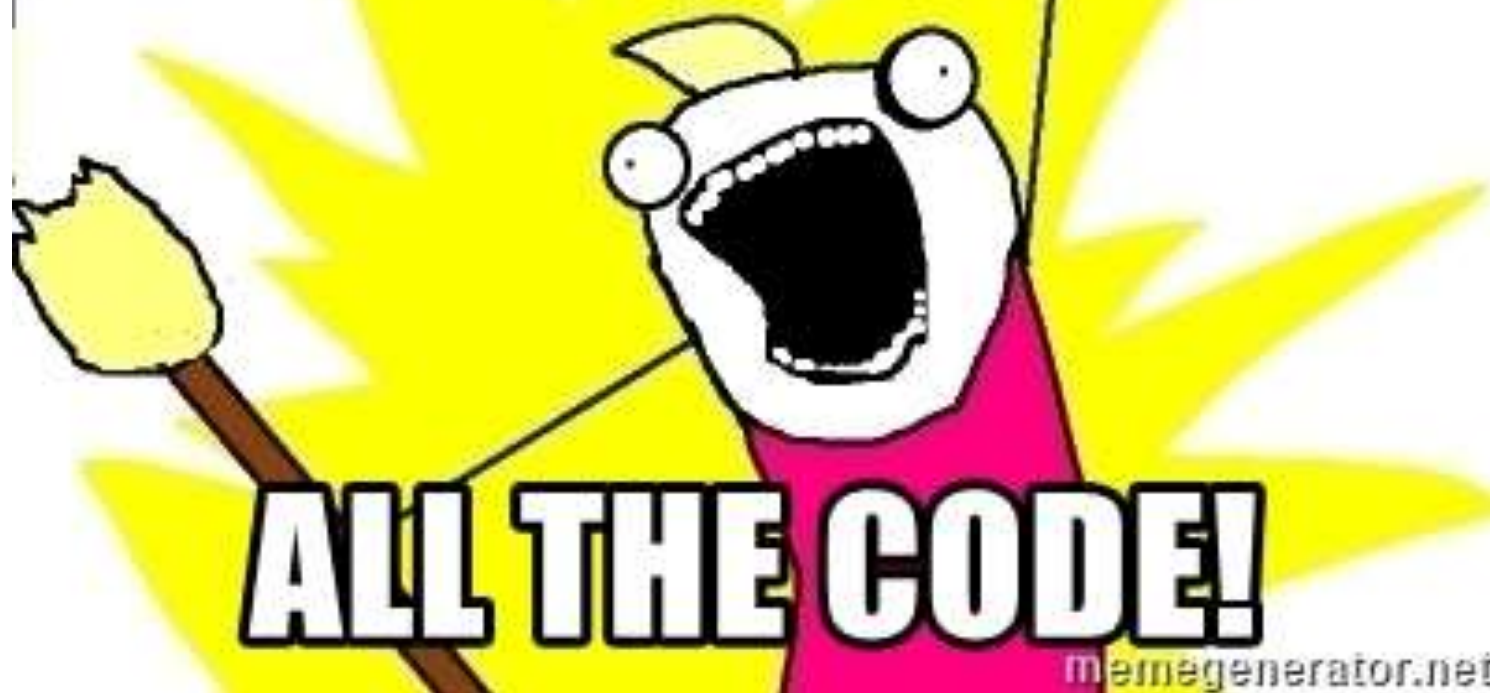
Propo

- Use all React fea
- Reuse stateful lo
- Opt-in and 100%

reactj



REWRITE



memegenerator.net



Hooks are still alpha



Dan Abramov

@dan_abramov

Following



Heads up: react@16.8.0-alpha.0 and react-dom@16.8.0-alpha.0 are out — for people who want the latest build with Hooks enabled. (⚠️ The warnings about using alphas apply.)

We're back from vacations and are focused on getting Hooks into a stable release as our biggest priority.

6:11 PM - 9 Jan 2019

322 Retweets 1,470 Likes



32



322



1.5K



What are Hooks?

- New React API
- Function components (Chao classes!)
- The future of React

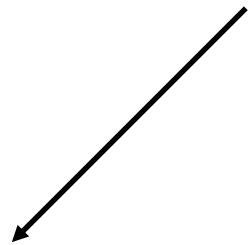
FUNCTIONS, FUNCTIONS

EVERYWHERE

makeameme.org

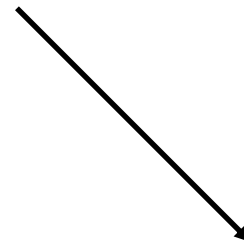
Component Has State?

Yes



Class Component

No



Functional Stateless
Component (FSC)

Hooks can be used side
by side with classes

Why use Hooks?



Binding


```

import React from 'react'
import { Card, Row, Input, Text } from '../components'
import ThemeContext from '../ThemeContext'

export default class Greetings extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      name: 'Mary',
      surname: 'Popins',
      width: window.innerWidth
    }

    this.handleChange = this.handleChange.bind(this)
    this.handleSurnameChange = this.handleSurnameChange.bind(this)
    this.handleResize = this.handleResize.bind(this)
  }

  componentDidMount() {
    window.addEventListener('resize', this.handleResize)
    document.title = this.state.name + ' ' + this.state.surname
  }

  componentDidUpdate() {
    document.title = this.state.name + ' ' + this.state.surname
  }

  componentWillUnmount() {
    window.removeEventListener('resize', this.handleResize)
  }

  handleChange(event) {
    this.setState({ name: event.target.value })
  }

  handleSurnameChange(event) {
    this.setState({ surname: event.target.value })
  }

  handleResize() {
    this.setState({ width: window.innerWidth })
  }

  render() {
    const { name, surname, width } = this.state

    return (
      <ThemeContext.Consumer>
        {theme => (
          <Card theme={theme}>
            <Row label="Name">
              <Input value={name} onChange={this.handleChange} />
            </Row>
            <Row label="Surname">
              <Input value={surname} onChange={this.handleSurnameChange} />
            </Row>
            <Row label="Width">
              <Text>{width}</Text>
            </Row>
          </Card>
        )}
      </ThemeContext.Consumer>
    )
  }
}

```

Wrapper Hell



```
▼ <Unknown>
  ▼ <t debug={false} errorMessage="">
    ▼ <o>
      ▼ <t>
        ▼ <t>
          ▼ <Router>
            ▼ <RouterContext>
              ▼ <Apollo(Connect(Apollo(n)))>
                ▼ <t fetchPolicy="network-only" errorPolicy="ignore" ssr={false} displayName="Apollo(Connect(Apollo(n)))"
                  skip={false} warnUnhandledError={true}>
                    ▼ <Connect(Apollo(n)) authLoading={false} isAuthenticated={2539615}>
                      ▼ <Apollo(n) authLoading={false} isAuthenticated={2539615}>
                        ▼ <t errorPolicy="ignore" ssr={false} displayName="Apollo(n)" skip={false} warnUnhandledError={true}>
                          ▼ <n authLoading={false} isAuthenticated={2539615} userLoading={false}>
                            ▼ <Connect(Apollo(t)) authLoading={false} isAuthenticated={2539615} userLoading={false}>
                              ► <Apollo(t) authLoading={false} isAuthenticated={2539615} userLoading={false} isMobile={false}
                                } isOnline={true} lang="id" popUp={false} searchModalOpen={false} sessionId={2539615} xdevice
                                ="">...</Apollo(t)> == $r
                              </Connect(Apollo(t))>
                            </n>
                          </t>
                        </Apollo(n)>
                      </Connect(Apollo(n))>
                    </t>
                  </Apollo(Connect(Apollo(n)))>
                </RouterContext>
              </Router>
            </t>
          </o>
        </t>
      </Unknown>
```

useState

State Variable

Setter Function

Initial Value



```
// State with Hooks  
const [count, setCount] = useState(0);
```




```
// Without Destructuring  
const countStateArray = useState(0);  
  
const count = countStateArray[0];  
const setCount = countStateArray[1];
```



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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

~~this.setState()~~ ~~this.state.count~~



```
// State can be of any type and defined multiple times  
const [position, setPosition] = useState({ left: 0, top: 0 });  
const [count, setCount] = useState(0);  
const [query, setQuery] = useState('');
```

useEffect

- Interact with APIs
- Subscribe to events

useEffect

- `componentDidMount()`
- `componentDidUpdate()`
- `componentWillUnmount()`



ComponentDidUpdate
ComponentDidMount

```
useEffect(() => {  
  const timer = setInterval( () => tick(), 1000 );  
  
  return function cleanup() {  
    clearInterval(timer);  
  };  
});
```

ComponentWillUnmount

Rules of Hooks

Only call hooks at the Top Level

- No loops
- No conditions



```
const [query, setQuery] = useState('');  
const [loading, setLoading] = useState(true);  
useEffect(function updateTitle() {  
  document.title = 'Results for: ' + query;  
});
```



```
// 🖐️ Don't do this!  
if (query !== '') {  
  useEffect(function updateTitle() {  
    document.title = 'Results for: ' + query;  
  });  
}
```

| Pos | First Render | Second Render |
|-----|------------------------|-------------------------------|
| 0 | useState("") | useState('Rick') |
| 1 | useState(true) | useState(true) |
| 2 | useEffect(updateTitle) | useEffect(updateTitle) |



```
// 👍 Much better!  
useEffect(function updateTitle() {  
  if (query !== '') {  
    document.title = 'Results for: ' + query;  
  }  
});
```

Only call hooks from
React Functions

Custom Hook names
should start with “use”

Let's see Hooks in action



More Hooks

- `useReducer`
- `useContext`
- `useRef`
- `useMemo`
- ...

Resources

- Official Docs: <https://reactjs.org/docs/hooks-intro.html>
- React Hooks Course: <https://courses.carlosmafla.com/courses/react-hooks>
- Ryan Florence (React Conf 2018): <https://youtu.be/wXLf18DsV-I>
- Dave Ceddia: <https://www.youtube.com/watch?v=CpPCJigsPNY>

Gracias

