

PASSING DATA DOWN THE COMPONENT CHAIN



REACT DATA IS
UNIDIRECTIONAL

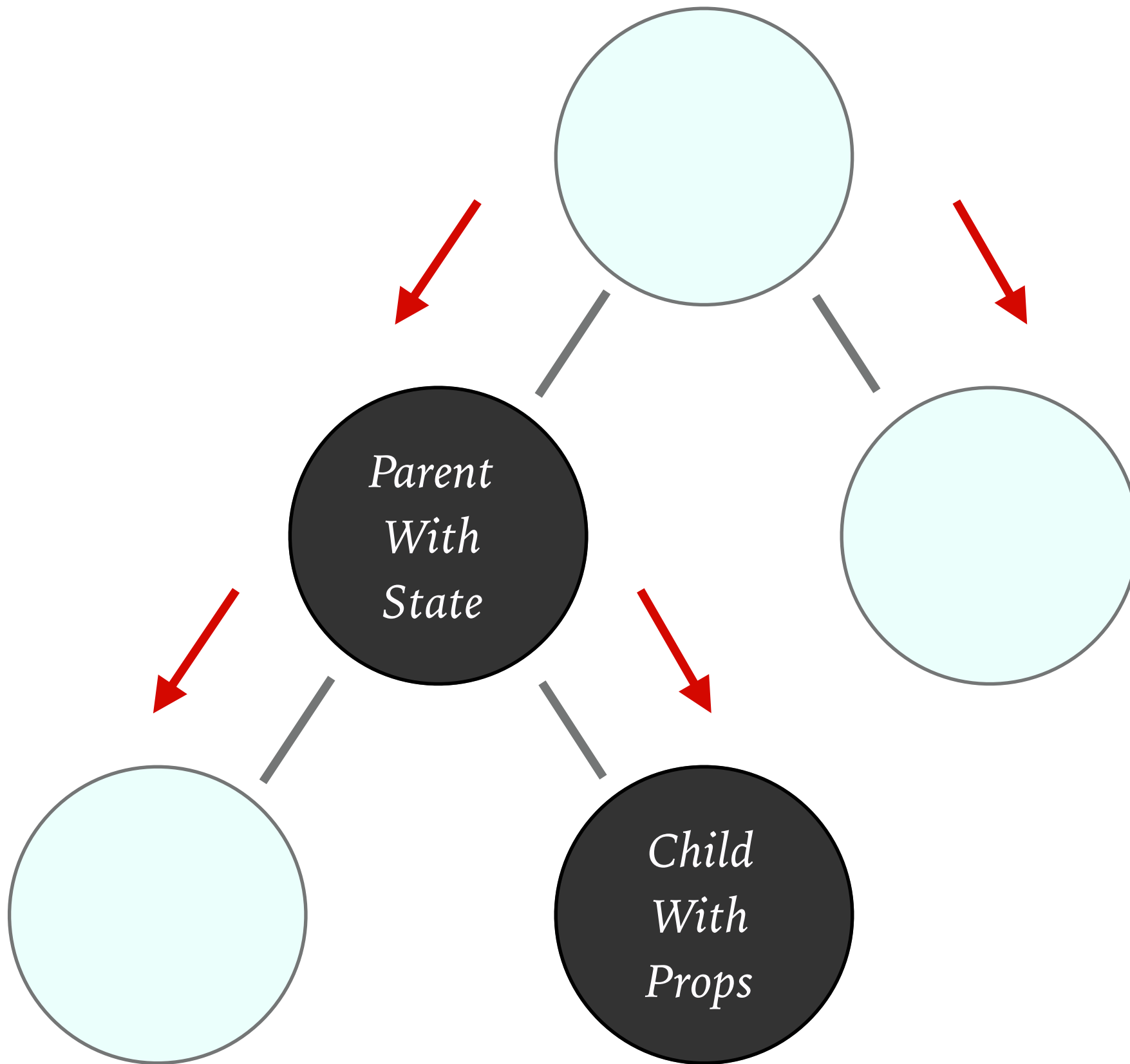
IT CAN ONLY BE PASSED FROM
PARENT

TO

CHILD

THROUGH.....

PROPS!



A PARENT COMPONENT WITH A CHILD COMPONENT

```
import React, { Component } from 'react';
import ChildWithProps from './ChildWithProps.js';
```

```
class ParentWithState extends Component {
  constructor() {
    super()
    this.state = {
      number: 0
    }
  }
  render() {
    return (
      <div>
        <ChildWithProps />
      </div>
    )
  }
}

export default ParentWithState;
```

```
import React, { Component } from 'react';

class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <h1>Number</h1>
      </div>
    )
  }
}

export default ChildWithProps;
```

Number

A PARENT COMPONENT WITH A CHILD COMPONENT

```
import React, { Component } from 'react';
import ChildWithProps from './ChildWithProps.js';
```

```
class ParentWithState extends Component {
  constructor() {
    super()
    this.state = {
      number: 0
    }
  }
  render() {
    return (
      <div>
        <ChildWithProps />
      </div>
    )
  }
}
```

```
export default ParentWithState;
```

```
import React, { Component } from 'react';
```

```
class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <h1>Number</h1>
      </div>
    )
  }
}
```

```
export default ChildWithProps;
```

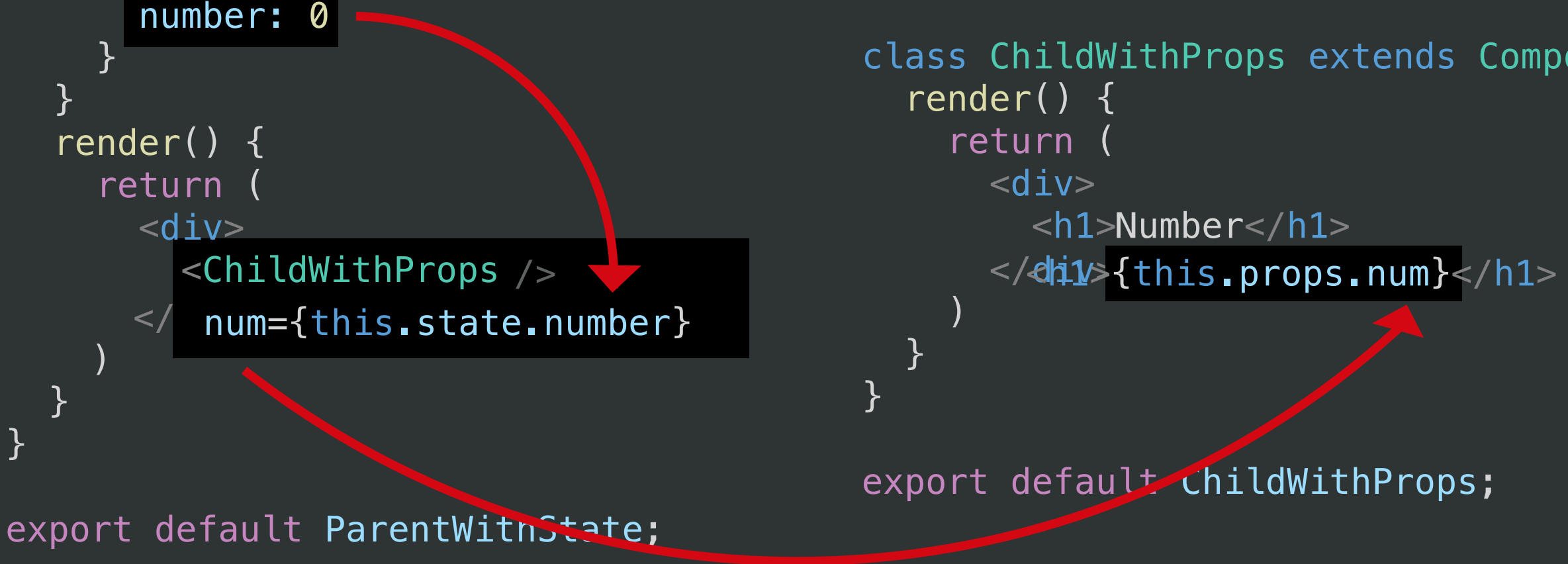
A PARENT COMPONENT WITH A CHILD COMPONENT

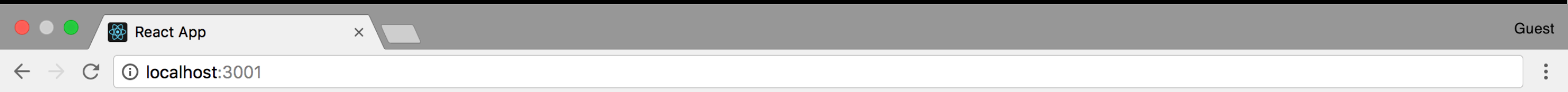
```
import React, { Component } from 'react';
import ChildWithProps from './ChildWithProps.js';
```

```
class ParentWithState extends Component {
  constructor() {
    super()
    this.state = {
      number: 0
    }
  }
  render() {
    return (
      <div>
        <ChildWithProps />
        num={this.state.number}
      </div>
    )
  }
}
export default ParentWithState;
```

```
import React, { Component } from 'react';

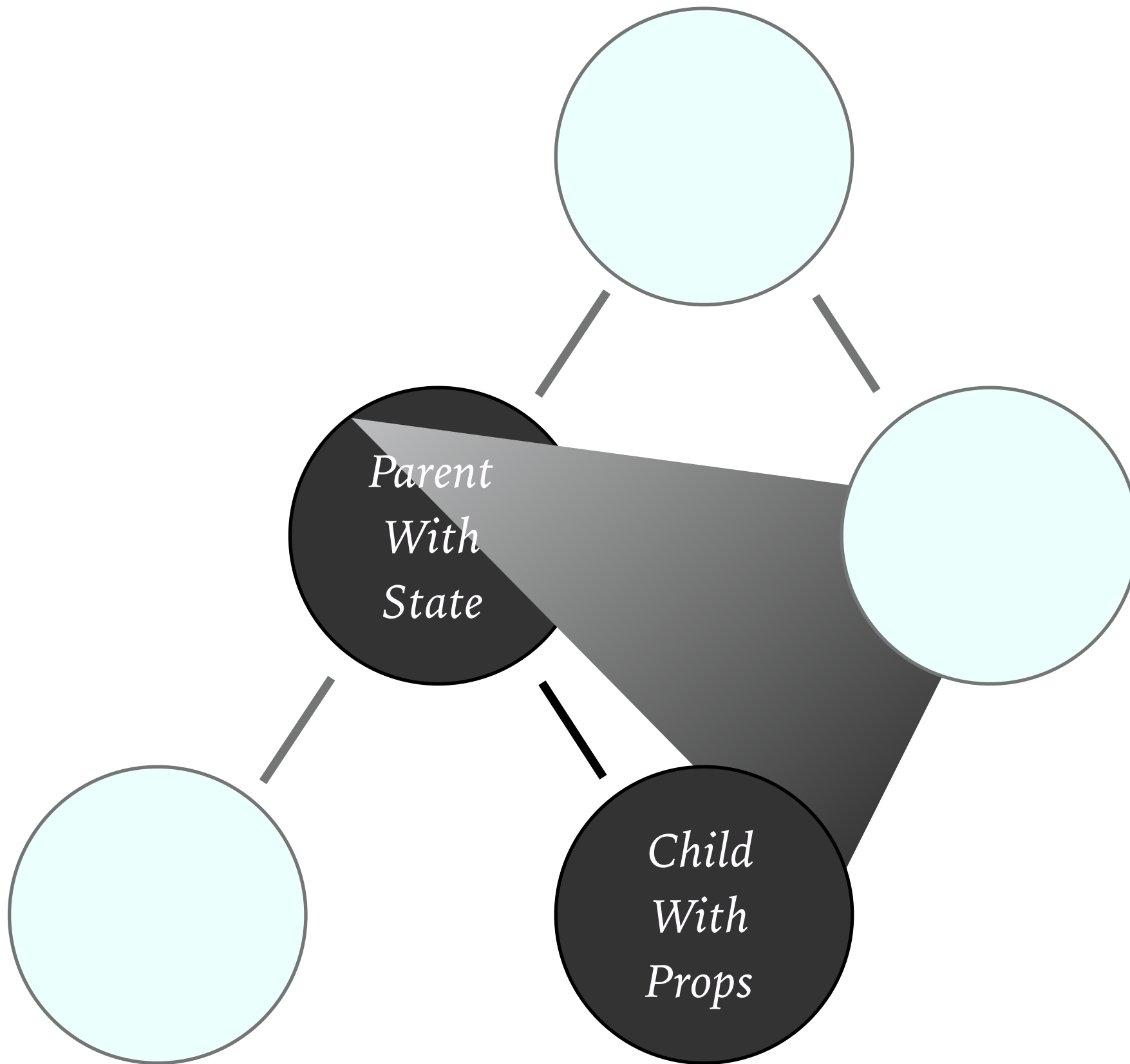
class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <h1>Number</h1>
        </div>{this.props.num}</h1>
      )
    }
  }
}
export default ChildWithProps;
```

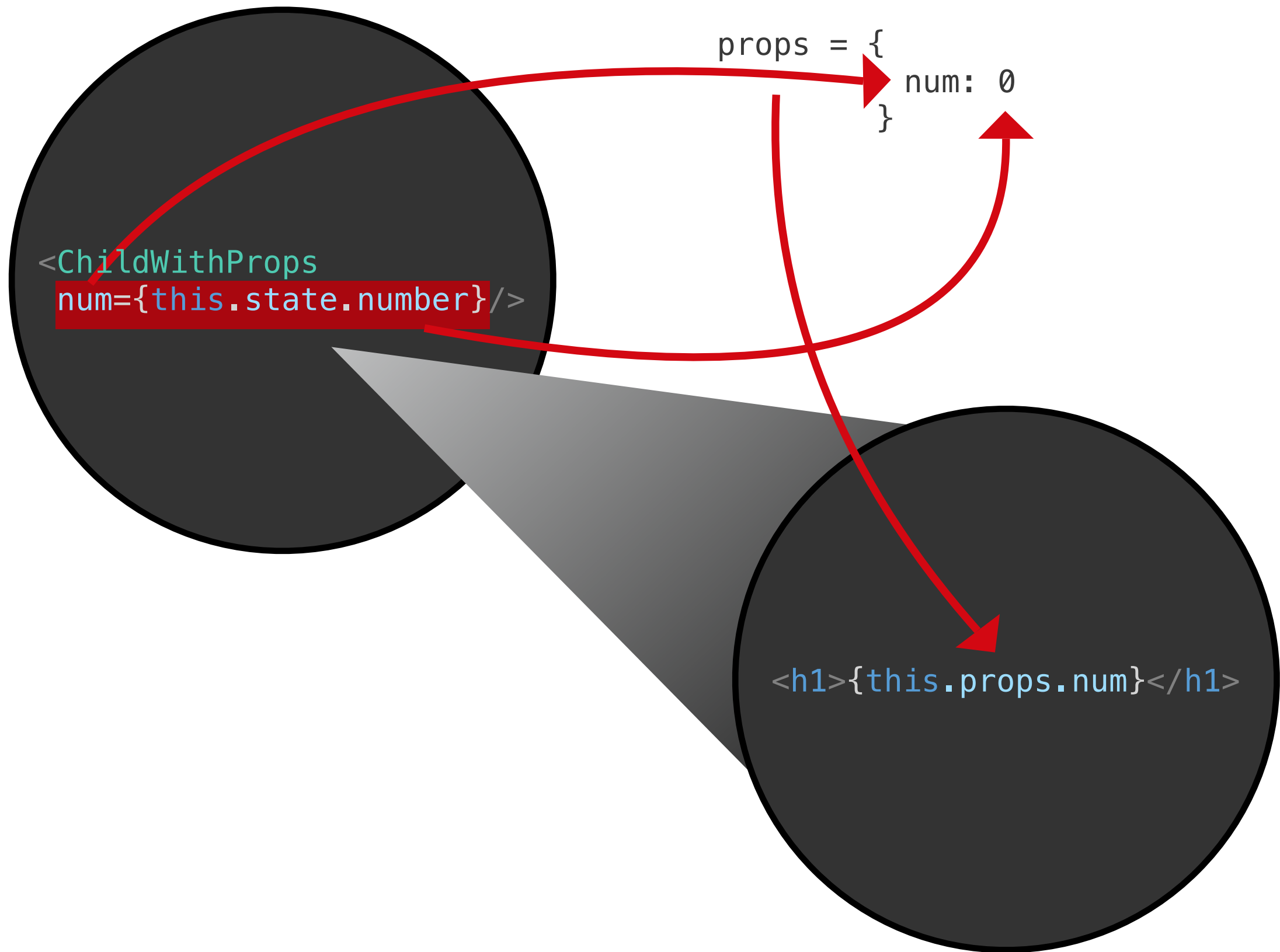




Number

0





PASSING
FUNCTIONS
THROUGH
PROPS

A PARENT COMPONENT WITH A CHILD COMPONENT

```
class ParentWithState extends Component {
  constructor() {
    super()
    this.state = {
      number: 0
    }
    this.addToNum =
      this.addToNum.bind(this)
  }
  addToNum() {
    this.setState({
      number: this.state.number + 1
    })
  }
  render() {
    return (
      <div>
        <h1>{this.state.number}</h1>
        <ChildWithProps
          add={this.addToNum} />
      </div>
    )
  }
}
export default ParentWithState;
```

```
import React, { Component } from 'react';

class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <button onClick={this.props.add}>
          Add</button>
      </div>
    )
  }
}

export default ChildWithProps;
```

0

Add

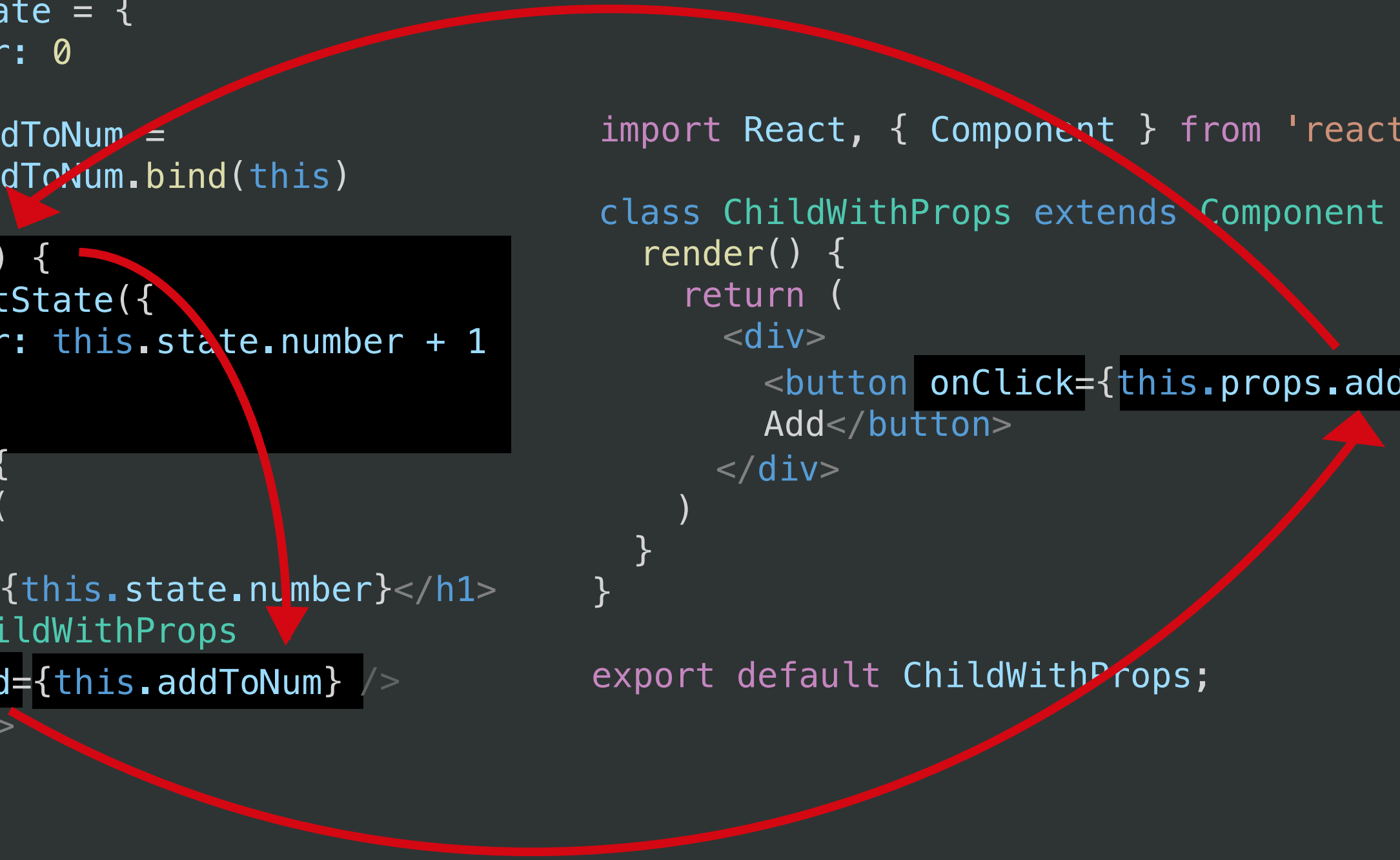
A PARENT COMPONENT WITH A CHILD COMPONENT

```
class ParentWithState extends Component {
  constructor() {
    super()
    this.state = {
      number: 0
    }
    this.addToNum =
      this.addToNum.bind(this)
  }
  addToNum() {
    this.setState({
      number: this.state.number + 1
    })
  }
  render() {
    return (
      <div>
        <h1>{this.state.number}</h1>
        <ChildWithProps
          add={this.addToNum} />
      </div>
    )
  }
}
export default ParentWithState;
```

```
import React, { Component } from 'react';

class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <button onClick={this.props.add}>
          Add</button>
      </div>
    )
  }
}

export default ChildWithProps;
```



- Determine what data in the Parent Component needs to be passed to the Child Component

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component
- Render the Child Component tag in the Parent's JSX

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component
- Render the Child Component tag in the Parent's JSX
- Add a prop to the Child Component tag

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component
- Render the Child Component tag in the Parent's JSX
- Add a prop to the Child Component tag
- Set the prop equal to the data that needs to be passed

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component
- Render the Child Component tag in the Parent's JSX
- Add a prop to the Child Component tag
- Set the prop equal to the data that needs to be passed
- Go to the Child Component's file

- Determine what data in the Parent Component needs to be passed to the Child Component
- Import the Child Component
- Render the Child Component tag in the Parent's JSX
- Add a prop to the Child Component tag
- Set the prop equal to the data that needs to be passed
- Go to the Child Component's file
- Render the data from the Parent Component by using dot notation on the props object
====> {this.props.PROP_NAME}

REACT COMPONENTS

A CLASS COMPONENT WITH NO STATE

```
import React, { Component } from 'react';

class NoState extends Component {
  render() {
    return (
      <div>
        // stuff .....
      </div>
    )
  }
}

export default NoState;
```


A CLASS COMPONENT WITH STATE

```
import React, { Component } from 'react';

class WithState extends Component {
  constructor() {
    super()
    this.state = {...}
  }
  render() {
    return (
      <div>
        // stuff .....
      </div>
    )
  }
}

export default WithState;
```

ANOTHER WAY TO MAKE
A COMPONENT
WITH
NO STATE

A FUNCTIONAL COMPONENT WITH NO STATE

```
import React from 'react';

function Functional() {
  return (
    <div>
      // stuff ...
    </div>
  )
}
```

```
export default Functional;
```

What is the same?

We still import React.

We still export default.

We still return JSX.

What is different?

This is a function, not a class.

There is no render method.

We do not import Component.

IMPORTING A **FUNCTIONAL COMPONENT**

A FUNCTIONAL COMPONENT WITH NO STATE

```
import React from 'react';
```

```
class WithState extends Component {  
  constructor() {  
    super()  
    this.state = {...}  
  }  
  render() {  
    return (  
      <div>  
        <Functional />  
      </div>  
    )  
  }  
}
```

```
export default WithState;
```

```
import React from 'react';  
  
function Functional() {  
  return (  
    <div>  
      // stuff ...  
    </div>  
  )  
}  
  
export default Functional;
```

CLASS COMPONENT WITH STATE

.....

CLASS COMPONENT WITH STATE

.....

- Import React and
Component

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX

CLASS COMPONENT WITH STATE

.....

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

CLASS COMPONENT WITH STATE

.....

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH STATE

.....

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

.....

CLASS COMPONENT WITH STATE

.....

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

.....

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

FUNCTIONAL COMPONENT

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

FUNCTIONAL COMPONENT

- Import React

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

FUNCTIONAL COMPONENT

- Import React
- function *Name*() {...}

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

FUNCTIONAL COMPONENT

- Import React
- function *Name*() {...}
- return JSX

CLASS COMPONENT WITH STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*
- Constructor method with super invoked and this.state = {...}

CLASS COMPONENT WITH NO STATE

- Import React and Component
- class *Name* extends Component {...}
- render method with a return that returns JSX
- export default *Name*

FUNCTIONAL COMPONENT

- Import React
- function *Name*() {...}
- return JSX
- export default *Name*

WHEN WOULD YOU USE
**A STATELESS FUNCTIONAL
COMPONENT**

OVER
A STATELESS CLASS COMPONENT?

LIFECYCLE HOOKS

WHAT ARE
LIFECYCLE HOOKS?

THERE ARE MULTIPLE LIFECYCLE HOOKS

WE WILL BE DISCUSSING ONE
**COMPONENT WILL RECEIVE
PROPS**