

PASSING DATA DOWN THE COMPONENT CHAIN

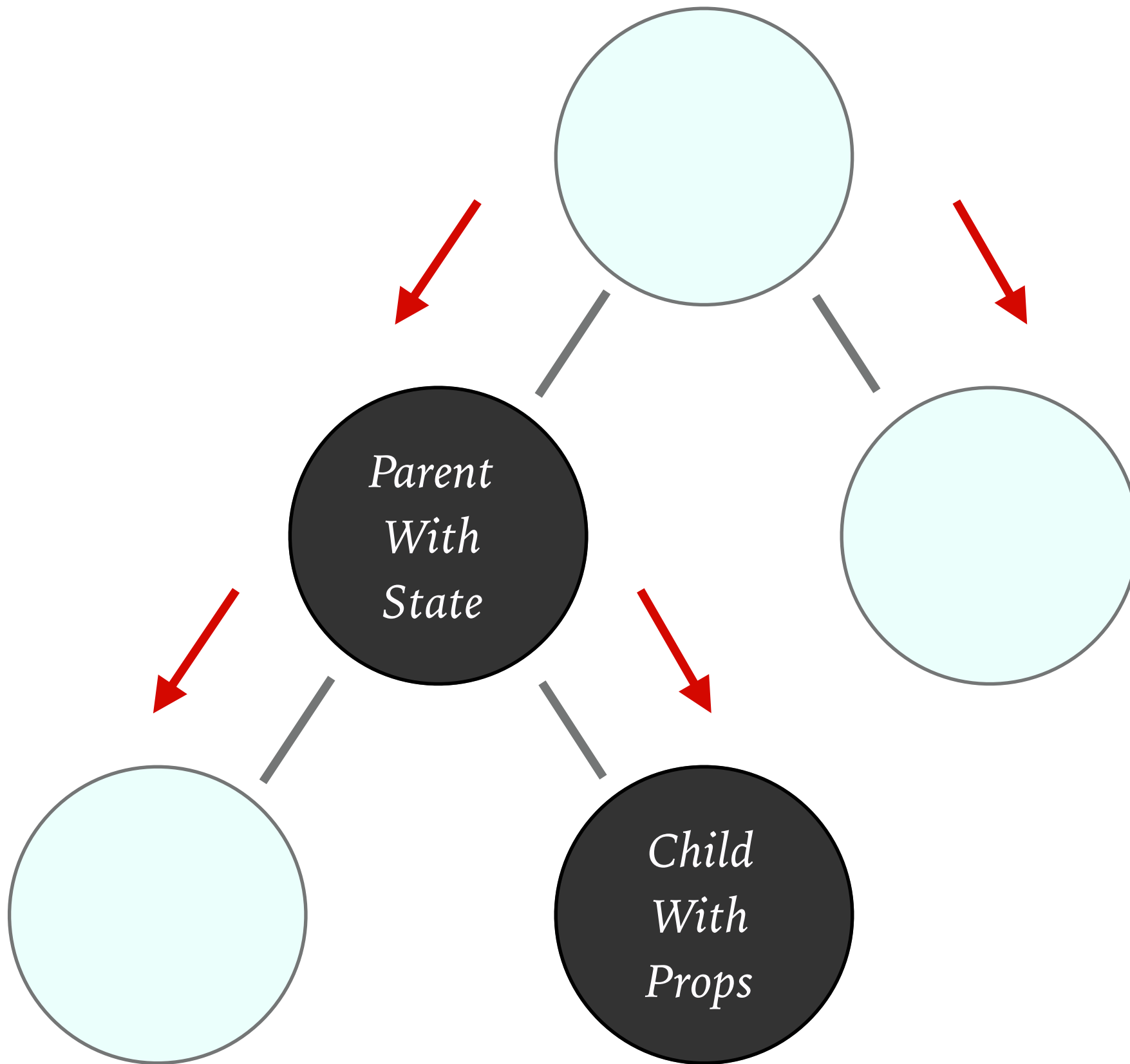


REACT DATA IS
UNIDIRECTIONAL

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';
```

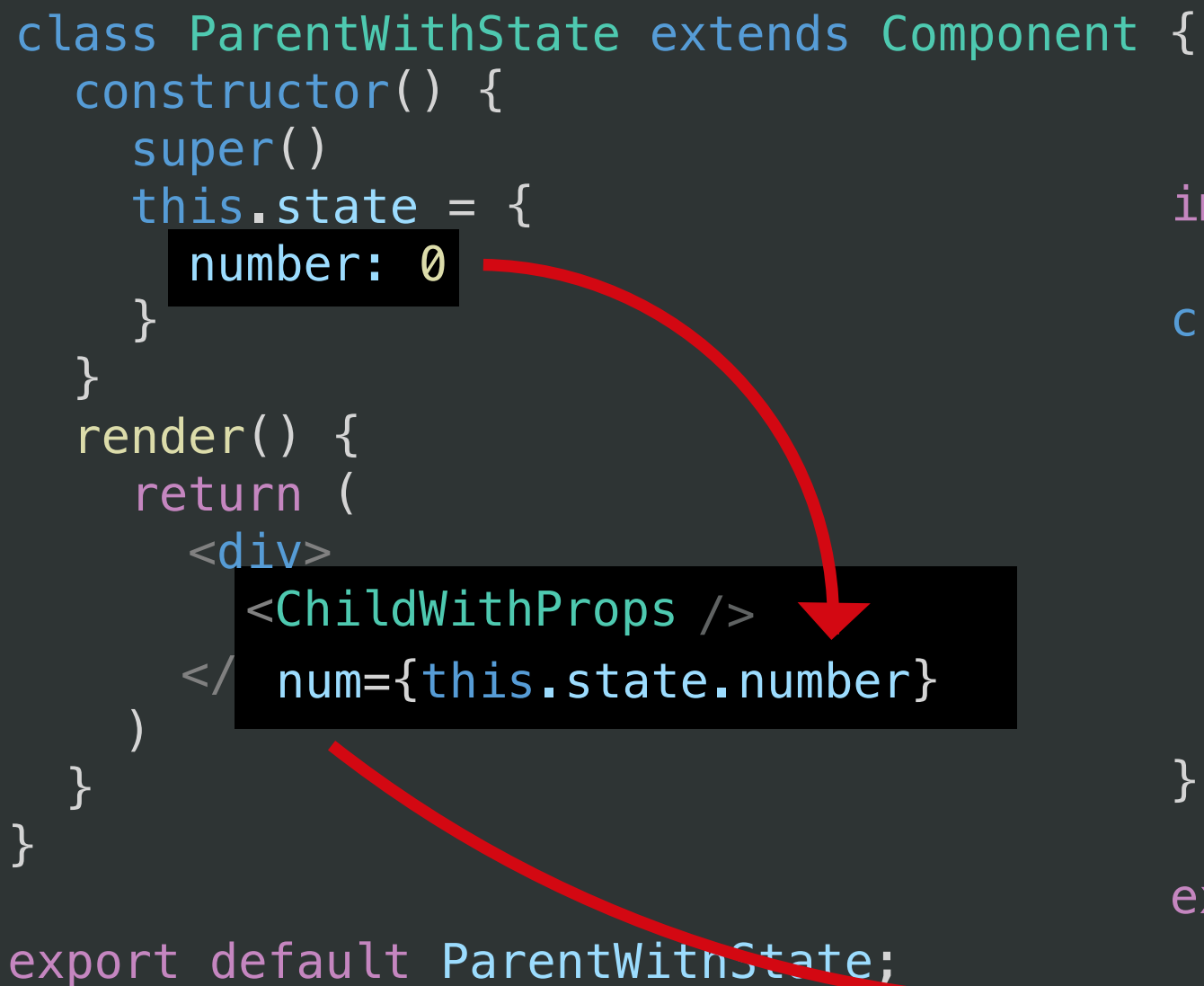
```
class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <h1>Number</h1>
      </div>
    )
  }
}
```

```
export default ChildWithProps;
```

A PARENT COMPONENT WITH A CHILD COMPONENT

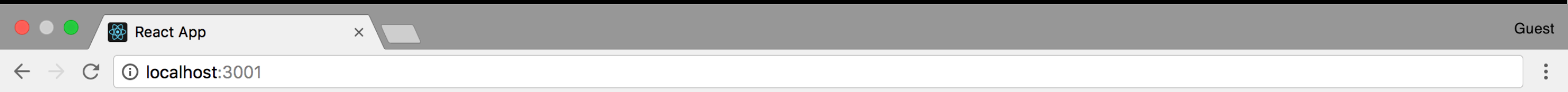
```
import React, { Component } from 'react';
import ChildWithProps from './ChildWithProps.js';
```

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



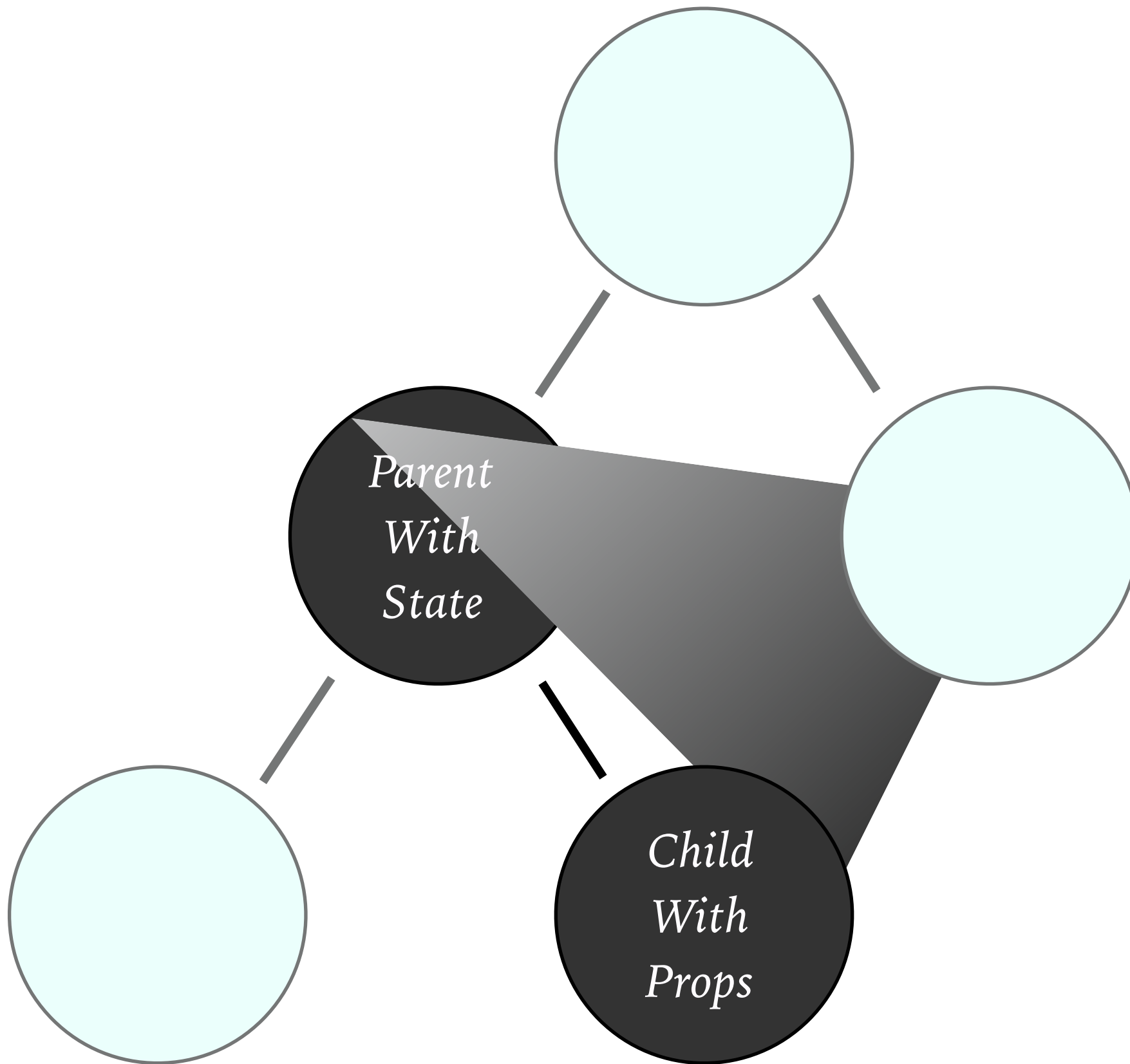
```
import React, { Component } from 'react';

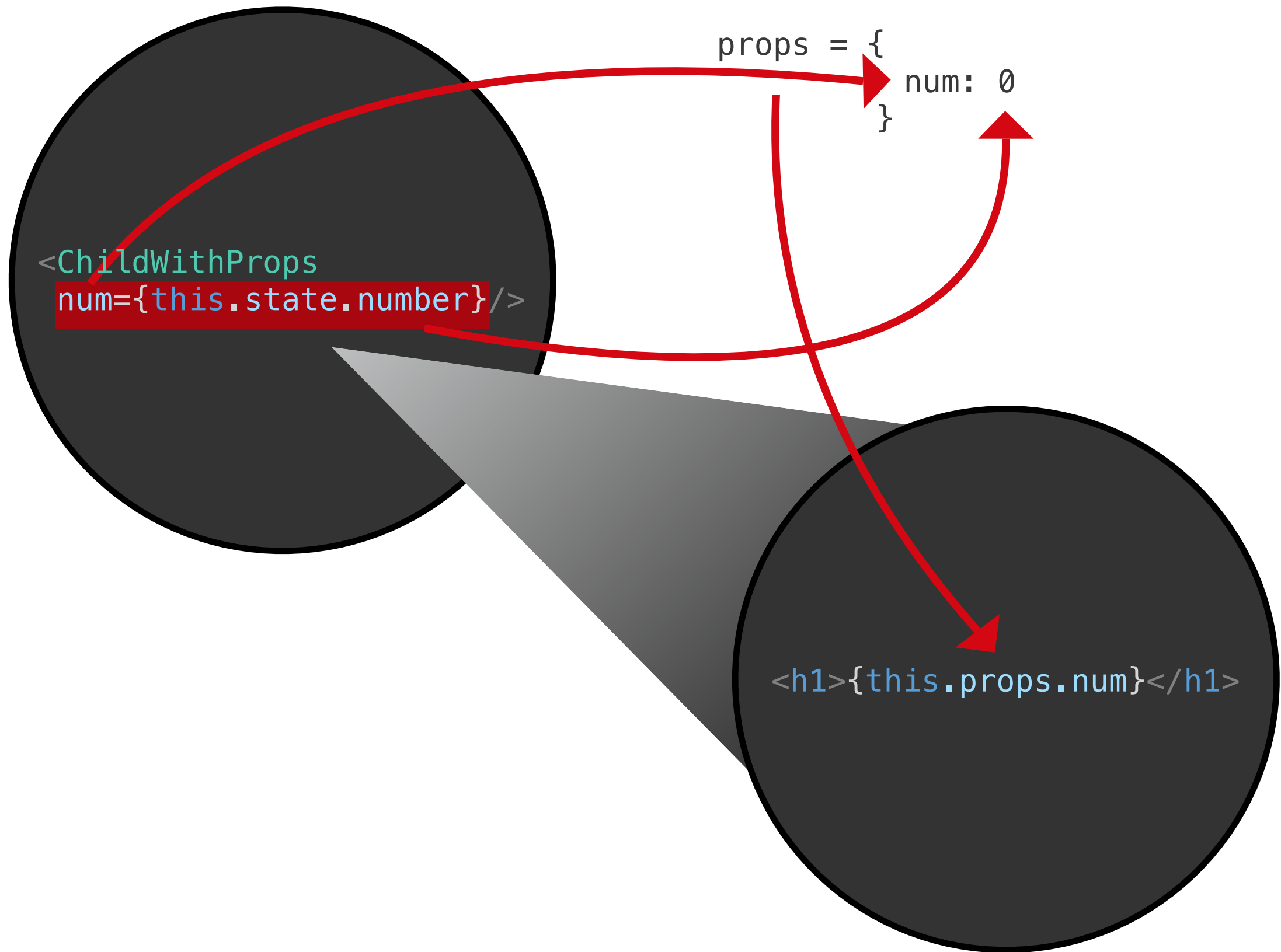
class ChildWithProps extends Component {
  render() {
    return (
      <div>
        <h1>Number</h1>
        </div>
        {this.props.num}
      </div>
    )
  }
}
export default ChildWithProps;
```

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

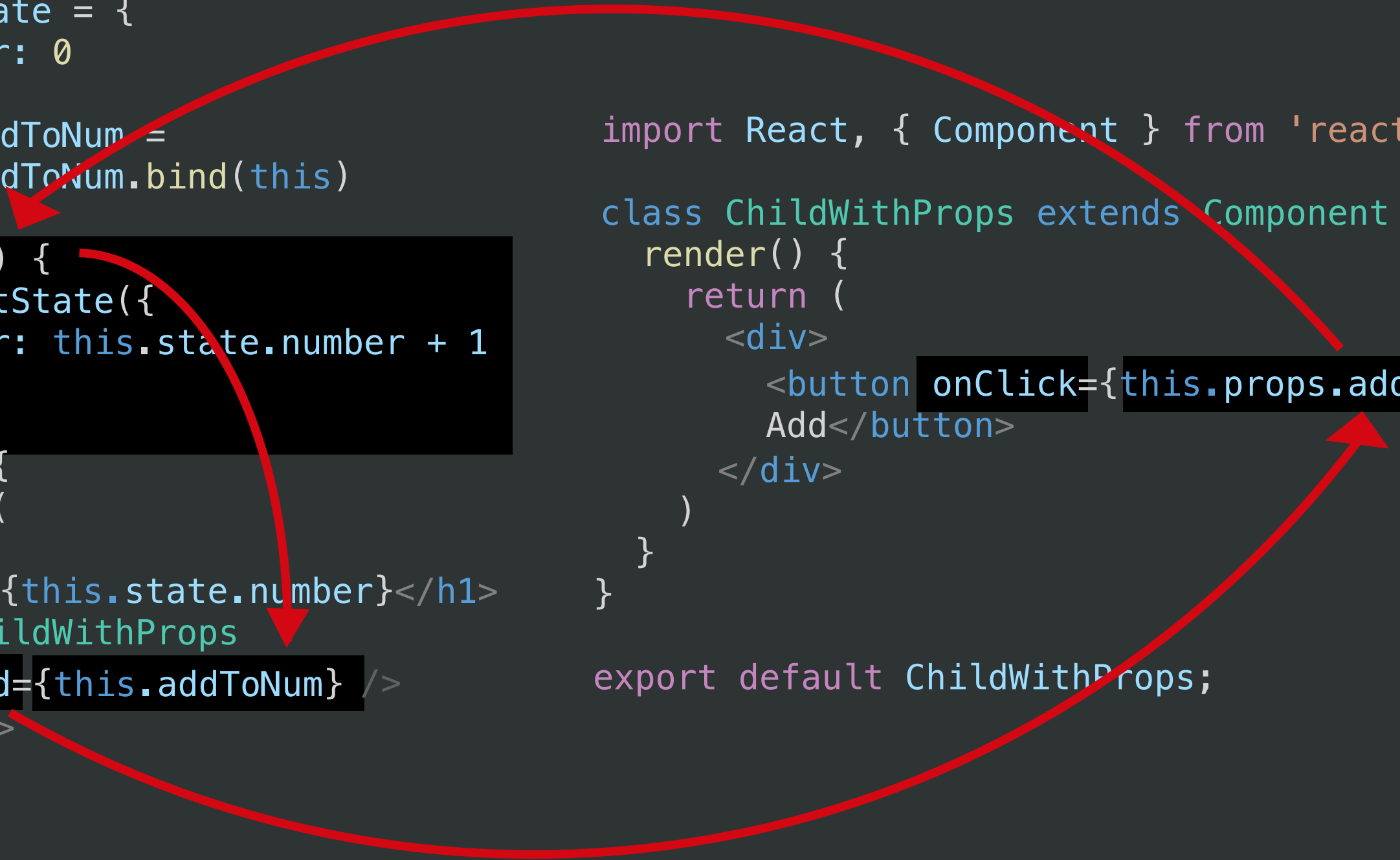
A PARENT COMPONENT WITH A CHILD COMPONENT

```
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}