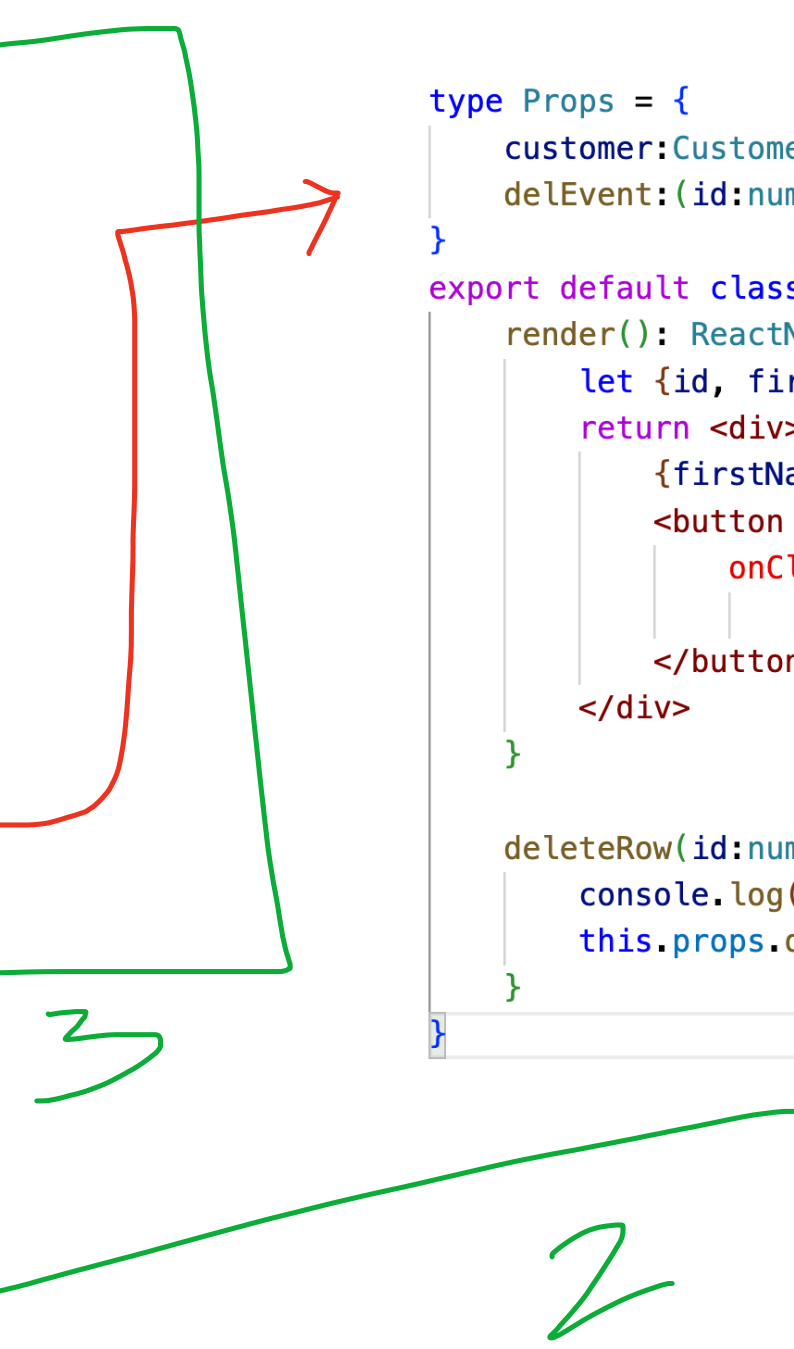


## CustomerList

```
deleteCustomer(id:number):void {  
  // custs will all customers except the one which has id  
  let custs = this.state.customers.filter(c => c.id !== id)  
  //don't do below code --> doesn't enforce reconcillation  
  // this.state.customers = custs;  
  this.setState({  
    "customers": custs  
  }, () => console.log("updated!!!")); // updates state asyn  
  //console.log("updated !!!" , this.state.customers);  
}  
  
render(): ReactNode {  
  return <div>  
    <Filter />  
    {  
      this.state.customers.map(c => <CustomerRow  
        delEvent={this.deleteCustomer.bind(this)}  
        customer={c}  
        key={c.id}/>)  
    }  
  </div>  
}
```

The image shows a code snippet for a React component. A green arrow points from the top right to the `deleteCustomer` method. Another green arrow points from the `deleteCustomer` method to the `delEvent` prop in the `CustomerRow` component within the `render` method. A red bracket highlights the `delEvent` and `customer` props. A green arrow points from the `delEvent` prop to the `deleteCustomer` method. A green arrow points from the `customer` prop to the `customer` object in the `map` function. A green arrow points from the `key` prop to the `c.id` expression in the `map` function. A green arrow points from the `render` method to the `delEvent` prop. A green arrow points from the `customer` prop to the `customer` object in the `map` function. A green arrow points from the `key` prop to the `c.id` expression in the `map` function.

Whenever state or props change



A hand-drawn diagram on the left side of the page. It features a large green bracket spanning from the top to the bottom. A red arrow points from the top of this bracket towards the code. Below the bracket, the number '3' is written in green. At the bottom, the number '2' is written in green. A green line extends from the bottom of the bracket, passing under the number '2'.

```
type Props = {
  customer: Customer,
  delEvent: (id: number) => void
}
export default class CustomerRow extends Component<Props, {}> {
  render(): ReactNode {
    let {id, firstName, lastName} = this.props.customer;
    return <div>
      {firstName} &nbsp; {lastName} &nbsp;
      <button type="button"
        onClick={() => this.deleteRow(id)}>
        Delete
      </button>
    </div>
  }

  deleteRow(id: number) : void {
    console.log("Delete Row ", id);
    this.props.delEvent(id);
  }
}
```

es --> reconciliation happens



—  
—