

```
export default class FriendsList extends Component {
  state = {
    friends: friends
  }
  deleteFriend(id) {
    let frnds = this.state.friends.filter(f => f.id !== id);
    this.setState({
      friends: frnds
    })
  }
  // override to return JSX
  render() {
    return <div>
      <Filter />
      {
        this.state.friends.map(friend => <FriendRow
          person={friend}
          delEvent={(id) => this.deleteFriend(id)} />)
      }
    </div>
  }
}

export default class FriendRow extends Component {
  render() {
    let {id, firstName, lastName} = this.props.person;
    return <div className="row">
      {firstName} &nbsp; {lastName} &nbsp;
      <button type="button" onClick={() => this.props.delEvent(id)}>Delete</button>
    </div>
  }
}
```

Diagram illustrating the prop passing between `FriendsList` and `FriendRow`:

- `FriendsList` (left) has a `render` method that maps over `this.state.friends` and renders `<FriendRow>` components.
- `FriendRow` (right) receives props from `FriendsList`.
- Handwritten annotations: A red arrow points from the `person` prop in the `map` function of `FriendsList` to the `person` prop in `FriendRow`. A green arrow points from the `delEvent` prop in the `map` function of `FriendsList` to the `delEvent` prop in `FriendRow`. Handwritten letters 'A' and 'B' are present near the respective prop names.