

# Lecture 4: Lists, User Input

Jordan Hayashi

# Previous Lecture

- React Native
- Style
- Event Handling
- Stateless Functional Components
- Components
- Lifecycle Methods
- Expo
- Import/Export
- PropTypes

contacts.js

# Lists

- In web, browsers will automatically become scrollable for content with heights taller than the window
- In mobile, we need to do that manually
  - `ScrollView`
  - `ListView` (deprecated)
  - `FlatList`, `SectionList`

# ScrollView

- The most basic scrolling view
- Will render all of its children before appearing
- To render an array of data, use `.map()`
  - Components in an array need a unique key prop

<https://facebook.github.io/react-native/docs/scrollview.html>

# FlatList

- A performant scrolling view for rendering data
- “Virtualized:” only renders what’s needed at a time
  - Only the visible rows are rendered in first cycle
  - Rows are recycled, and rows that leave visibility may be unmounted
- Pass an array of data and a renderItem function as props
- Only updates if props are changed
  - Immutability is important

<https://facebook.github.io/react-native/docs/flatlist.html>

# SectionList

- Like `FlatList` with additional support for sections
- Instead of `data` prop, define sections
  - Each section has its own data array
  - Each section can override the `renderItem` function with their own custom renderer
- Pass a `renderSectionHeader` function for section headers

<https://facebook.github.io/react-native/docs/sectionlist.html>

# User Input

- Controlled vs uncontrolled components
  - Where is the source of truth for the value of an input?
- React recommends always using controlled components
- Pass `value` and `onChangeText` props

<https://facebook.github.io/react-native/docs/textinput.html>