<https://youtu.be/Yl69abRLGbY>

**React Native's Flexbox Implementation**

React Native implements flexbox for build layouts, but there are some key differences to keep in mind as you develop your applications. First, all containers in React Native are *flex containers* by default. Recall that in traditional CSS flexbox, you would normally define a flex container like so:

*/\*example.css\*/*

.container {

display: flex;

}

However, this is completely *unnecessary* in React Native! By default, everything is display: flex;. You can just use the defaults as they are, without adding different properties or writing extra code.

Another important distinction is how React Native handles flex-direction, a property that establishes the main axis (i.e., defining the direction in which flex items are placed). In web applications, items default to row. But since we're working on mobile devices, React Native sets the default to column, which lays out items *vertically*.

One more major difference you'll encounter is how the flex property is used. On the web, flexspecifies how a flex item grows or shrinks to manage the space around it (along the main axis). In React Native, flex is generally used with flex items that are on the same level, but hold different flexvalues. For example:

**import** React **from** 'react';

**import** { View } **from** 'react-native';

**const** FlexDemo = props => (

<View style={{flex: 1}}>

<View style={{flex: 1, backgroundColor: 'red'}} />

<View style={{flex: 2, backgroundColor: 'green'}} />

<View style={{flex: 3, backgroundColor: 'blue'}} />

</View>

);

**export** **default** FlexDemo;

Here, FlexDemo is a stateless functional component which renders <View> components with different flex values. Its outermost container is set to flex: 1, which will expand the full available width along the main axis (i.e., the entire screen in this example). Its children <View> components will fill the space accordingly, rendering a blue background color that takes up three times as much space as redtakes, and green that takes up exactly twice as much space as red takes.

**Other Differences**

In addition to the above, here is a list of defaults in other common CSS properties that React Native applies to components:

box-sizing: border-box;

position: relative;

align-items: stretch;

flex-shrink: 0;

align-content: flex-start;

border: 0 solid black;

margin: 0;

padding: 0;

min-width: 0;