INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCHERS
ISSN: 3030-332X Impact factor: 8,293 Volume 9, issue 1, December 2024
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

INTRODUCTION TO VIEW, TEXTVIEW, IMAGE, STYLES IN REACT NATIVE PROGRAMMING LANGUAGE

Rajabov Azizbek Ravshan o'g'li

Teacher of the Department of "General Technical Sciences", International University of Asia

Annotation: The rapid evolution of mobile development has brought forward frameworks like React Native, which have revolutionized the way developers build mobile applications. Developed by Facebook, React Native allows developers to write code in JavaScript while building cross-platform applications for both Android and iOS. By leveraging reusable components, developers can craft user interfaces (UI) with greater efficiency and consistency. In this article, we will delve into some of the foundational components in React Native — **View**, **Text**, **Image**, and **Styles**, with practical examples to get started.

Keywords: React Native, View, Text, Image, Style, mobile development, cross-platform programming, JavaScript, UI components.

Introduction

Overview of React Native

React Native is an open-source framework designed for building mobile applications using JavaScript and React. It focuses on creating interactive and efficient user interfaces by utilizing a component-based architecture. React Native's ability to share code across platforms reduces development time and costs, making it a popular choice for developers.

React Native applications are built using components that function like building blocks. Each component has specific properties and styles, which allows for the creation of modular and reusable UIs.

The View Component

The **View** component in React Native serves as a container for other components, such as text, images, or even other views. It is comparable to the tag in HTML and is used to group UI elements.

Key Properties of View:

- **style**: Defines the appearance of the container.
- **onLayout**: Retrieves information about the size and position of the component.

Example:

javascript

Copy code

ISSN: 3030-332X Impact factor: 8,293 Volume 9, issue 1, December 2024
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

```
import React from 'react';import { View, Text } from 'react-native';

const App = () => {
  return (
    <View style={{ backgroundColor: '#f0f0f0', padding: 20 }}>
      <Text>This is a Text component inside a View container.</Text>
    </View>
  );
};
export default App;
```

Here, the **View** component acts as a wrapper for the **Text** component. By using the style property, we customize the background color and padding.

The Text Component

The **Text** component is used to display text content in React Native applications. Similar to <p> or in HTML, it allows for custom styling and interactive features like touch handling.

Key Properties of Text:

- **numberOfLines**: Restricts the text to a specified number of lines.
- **onPress**: Executes a function when the text is clicked.

Example:

javascript

Copy code

```
import React from 'react';import { Text, View } from 'react-native';

const App = () => {
  return (
    <View>
      <Text style={{ fontSize: 24, fontWeight: 'bold' }}>
        Welcome to React Native!
      </Text>
```

ISSN: 3030-332X Impact factor: 8,293 Volume 9, issue 1, December 2024
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

```
    <Text style={{ color: 'gray' }}>
      This is an example of using the Text component.
    </Text>
  </View>
 );
};
export default App;
```

In this example, two **Text** components are used with different styles applied. The style property is used to define font size, weight, and color.

The Image Component

React Native's **Image** component is essential for displaying images in mobile applications. Images can be sourced from the web or local directories.

Key Properties of Image:

- **source**: Specifies the location of the image.
- **resizeMode**: Defines how the image fits into its container (cover, contain, etc.).

Example:

javascript

Copy code

```
import React from 'react';import { Image, View } from 'react-native';
const App = () => {
 return (
  <View>
   <Image
    source={{ uri: 'https://reactnative.dev/img/tiny_logo.png' }}
    style={{ width: 100, height: 100 }}
   />
  </View>
```

ISSN: 3030-332X Impact factor: 8,293 Volume 9, issue 1, December 2024
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

```
  );
};
export default App;
```

In the example, the **Image** component displays a remote image using the source property. The style property sets the dimensions of the image.

Working with Styles

React Native uses JavaScript objects for styling instead of traditional CSS. Developers can either use inline styles or the **StyleSheet** API to define reusable styles.

Example (Using StyleSheet):

javascript

Copy code

```
import React from 'react';import { StyleSheet, Text, View } from 'react-native';
const App = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Styled with StyleSheet</Text>
    </View>
  );
};
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#eaeaea', },
  title: {
```

ISSN: 3030-332X Impact factor: 8,293 Volume 9, issue 1, December 2024
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
Index: google scholar, research gate, research bib, zenodo, open aire.
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

```
    fontSize: 22,
    fontWeight: 'bold',
    color: 'blue',
  },
});
export default App;
```

Here, the StyleSheet.create() method defines a centralized object for styles, which is then referenced in the style attribute of components.

Advantages of React Native Components

1. **Code Reusability**: The same codebase works for multiple platforms, reducing effort and time.
2. **Dynamic UIs**: Components like **Text** and **Image** enable easy customization and interactivity.
3. **Faster Development**: The **Hot Reloading** feature lets developers see changes in real-time without restarting the app.
4. **Modular Architecture**: Components are self-contained, which simplifies maintenance and testing.

Conclusion

React Native provides a robust framework for building cross-platform mobile applications efficiently. Components like **View**, **Text**, **Image**, and **Style** form the foundation of any application's UI. By understanding and utilizing these components effectively, developers can create dynamic, responsive, and visually appealing applications. This article provides a starting point for mastering React Native, and with continued practice, developers can unlock its full potential.

References:

1 React Native Documentation

2 "React Native in Action" by Nader Dabit

3 Udemy Courses: "React Native for Beginners"

4 GitHub: React Native Examples

5 JavaScript.info: Comprehensive Guide to JavaScript