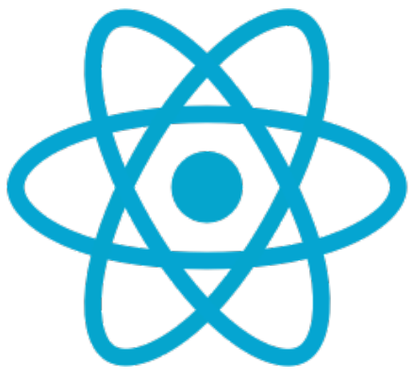
День 1

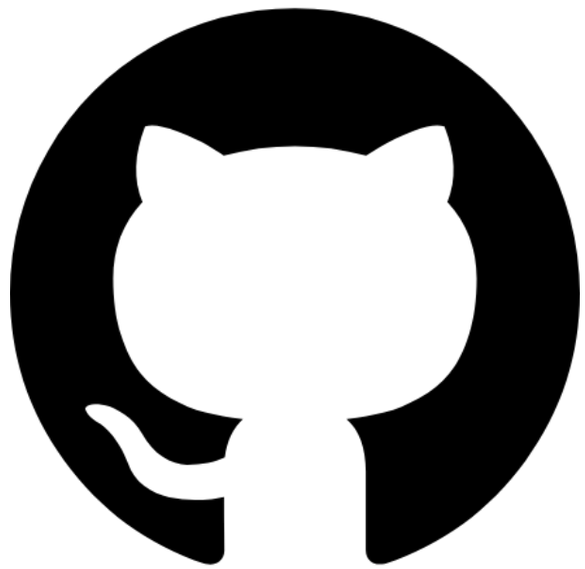
Знакомство с React Native

01↑



React Native

JavaScript



GitHub

Установка React Native

на Ubuntu

Установка Node.js

```
$ sudo apt-get update
$ sudo apt-get install nodejs
```

Установка npm - менеджера пакетов для Node.js

```
$ sudo apt-get install npm
```

Установка React Native CLI

```
$ npm install -g react-native-cli
```

Project configuration

🖥️ Visual Studio Code - редактор исходного кода, разработанный Microsoft.

✅ Поддержка отладки.

✅ Встроенный контроль Git.

✅ Подсветка синтаксиса.

✅ и т.д.

🖥️ ES Lint - инструмент для выявления и отчетности по шаблонам, найденным в коде ECMAScript / JavaScript.

✅ Последовательный код.

✅ Выявление ошибок.

🖥️ Prettier - инструмент для форматирования кода.

✅ Согласованный стиль.

✅ Собственные настраиваемые правила.

Document Object Model

📌 DOM (Document Object Model) - способ представления структурного документа с помощью объектов.

Недостатки DOM

❌ DOM не рассчитан для создания динамического пользовательского интерфейса.

❌ Низкая производительность.

Virtual DOM

📌 Virtual DOM - это работа с легковесной копией DOM.

✅ Изменения вносятся в копию DOM.

Преимущества Virtual DOM

📌 **Перерисовка DOM применяется после сравнения DOM с виртуальной копией, в том месте где было изменение.**

✅ **Эффективные алгоритмы сравнения.**

✅ **Обновление под-деревьев.**

✅ **Высокая производительность.**

JSX

✏️ `const element = <Text>Hello world!</Text>;`

❗ JSX это расширение синтаксиса для JavaScript.

❗ JSX позволяет писать язык разметки внутри кода.

✏️ `<Image source={pic} style={{width: 193, height: 110}}/>`

✏️ `<Greeting name='Hello world' />`

Стили

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

const StylesExample = () => {
  return (
    <View style={styles.container}>
      <View style={styles.card}>
        <Text style={styles.text}>Hello, NIS!</Text>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center'
  },
  card: {
    width: 200,
    height: 200,
    backgroundColor: 'gold',
    alignItems: 'center',
    justifyContent: 'center',
    borderRadius: 8
  },
  text: {
    color: 'white',
    fontWeight: '800',
    fontSize: 18
  }
});

export default StylesExample;
```

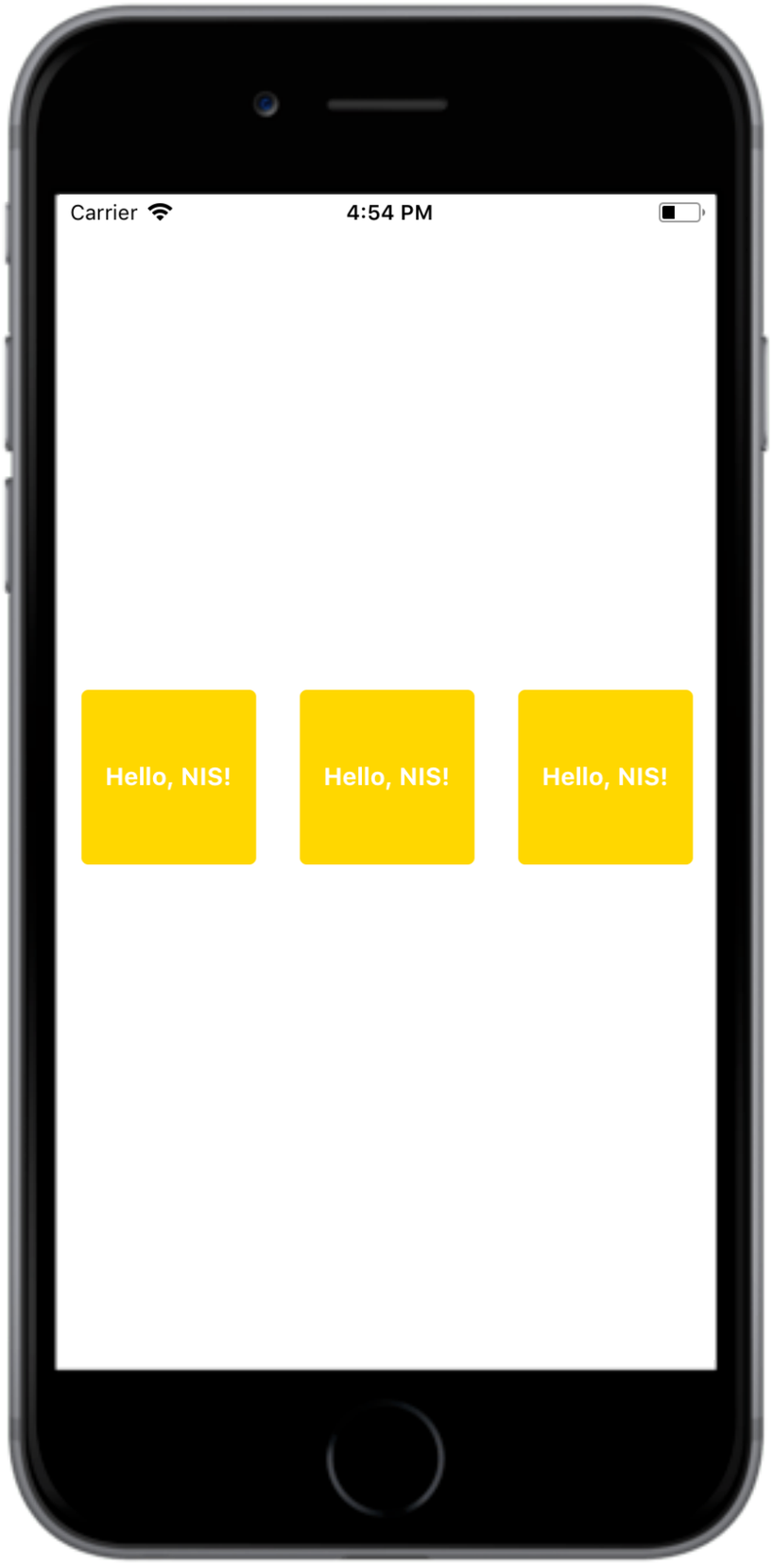
```jsx
import React from 'react';
import { View, Text, StyleSheet } from 'react-native';

const Card = () => {
  return (
    <View style={styles.card}>
      <Text style={styles.text}>Hello, NIS!</Text>
    </View>
  );
};
const FlexboxExample = () => {
  return (
    <View style={styles.container}>
      <Card />
      <Card />
      <Card />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row', // 'column' as default, 'row'
    justifyContent: 'space-around', // 'flex-start', 'center', 'flex-end',
                                    'space-between', 'space-around'

    alignItems: 'center' // 'flex-start', 'center', 'flex-end', 'stretch
  },
  card: {
    width: 100,
    height: 100,
    borderRadius: 4,
    margin: 5,
    backgroundColor: 'gold',
    alignItems: 'center',
    justifyContent: 'center'
  },
  text: {
    color: 'white',
    fontWeight: '800',
    fontSize: 14
  }
});

export default FlexboxExample;
```

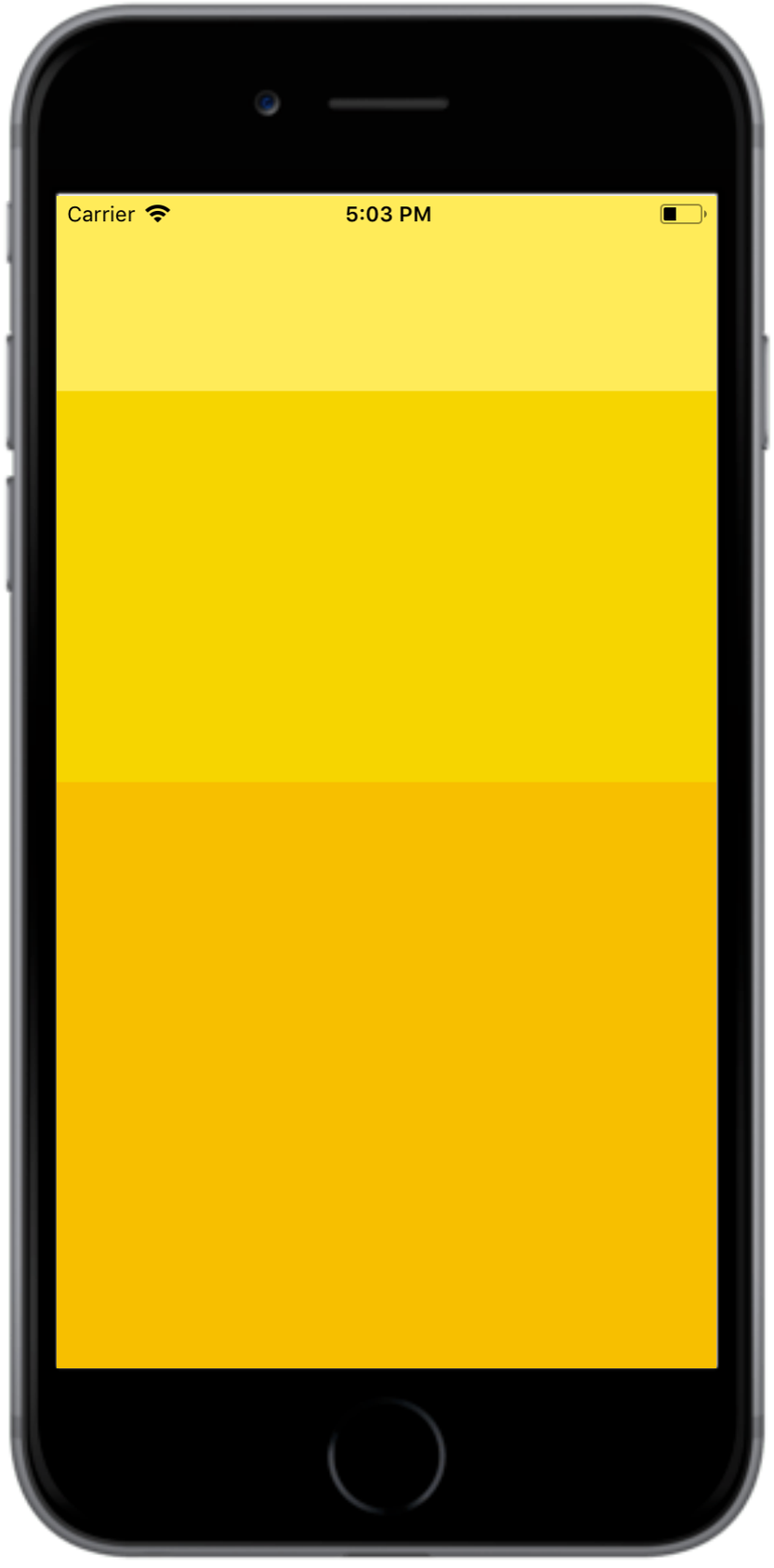


```jsx
import React from 'react';
import { View, StyleSheet } from 'react-native';

const FlexboxExample = () => {
  return (
    <View style={styles.container}>
      <View style={{ flex: 1, backgroundColor: '#FFEB59' }} />
      <View style={{ flex: 2, backgroundColor: '#F6D500' }} />
      <View style={{ flex: 3, backgroundColor: '#F7BF00' }} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1
  }
});

export default FlexboxExample;
```



Жизненный цикл компонента

📌 В процессе работы компонент проходит через ряд этапов жизненного цикла.

📌 На каждом из этапов вызывается определенная функция, в которой мы можем определить какие-либо действия.

1️⃣ constructor(props): конструктор, в котором происходит начальная инициализация компонента.

2️⃣ componentWillMount(): вызывается непосредственно перед рисовкой компонента.

3️⃣ render(): рендеринг компонента.

4️⃣ componentDidMount(): вызывается после рендеринга компонента. Здесь можно выполнять запросы к удаленным ресурсам.

5️⃣ componentWillUnmount(): вызывается перед удалением компонента из DOM.

npm
(Node Package Manager)

npm - менеджер пакетов, используемый Node.js-приложениями

npmjs.com **- сайт для поиска пакетов**

Установка пакета:

```
npm install <packagename> --save
```

Удаление пакета:

```
npm uninstall <packagename> --save
```

Список установленных пакетов: package.json

Homework

github.com/021NIS/Baspana

*Дополнительно

Установка React Native на Windows

Скачать NodeJS

https://nodejs.org/en/download/current/

Скачать Python 2.7

https://www.python.org/downloads/

Скачать Android Studio

https://developer.android.com/studio/index.html