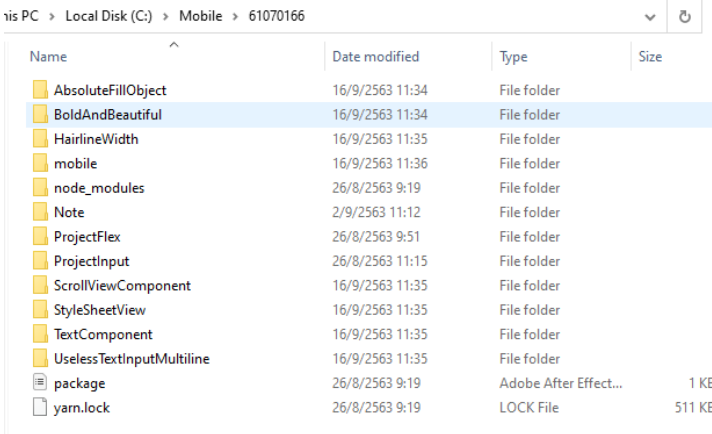


Lab 5 : Core Components : Basic Component

- ให้นักศึกษาทำการสร้าง New Project ใหม่ ทุกครั้งที่ทำการรันโปรแกรม ในข้อ 2-8 โดยให้ Folder มีดังนี้ Mobile\<รหัสนักศึกษา>\Project ตามชื่อ Component

Expo init <path\folder\StudentID\Project name>



Name	Date modified	Type	Size
AbsoluteFillObject	16/9/2563 11:34	File folder	
BoldAndBeautiful	16/9/2563 11:34	File folder	
HairlineWidth	16/9/2563 11:35	File folder	
mobile	16/9/2563 11:36	File folder	
node_modules	26/8/2563 9:19	File folder	
Note	2/9/2563 11:12	File folder	
ProjectFlex	26/8/2563 9:51	File folder	
ProjectInput	26/8/2563 11:15	File folder	
ScrollViewComponent	16/9/2563 11:35	File folder	
StyleSheetView	16/9/2563 11:35	File folder	
TextComponent	16/9/2563 11:35	File folder	
UselessTextInputMultiline	16/9/2563 11:35	File folder	
package	26/8/2563 9:19	Adobe After Effect...	1 KB
yarn.lock	26/8/2563 9:19	LOCK File	511 KB

- ให้นักศึกษานำที่ผลลัพธ์จากโปรแกรมข้างล่างนี้ (**Text Component**)

```

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

const ViewBoxesWithColorAndText = () => {
  return (
    <View
      style={{
        flexDirection: "row",
        height: 80,
        padding: 40
      }}
    >
      <View style={{ backgroundColor: "blue", flex: 0.3 }} />
      <View style={{ backgroundColor: "orange", flex: 0.5 }} />
      <Text>Hello World!</Text>
    </View>
  )
}
    
```

```

</View>

);

};

export default ViewBoxesWithColorAndText;
    
```

บันทึกผลการทดลอง :



ใช้ **View** ตัวบนสุด เป็น **flexDirection: "row"** เพื่อเรียง **Box View** เป็นแนวนอน ปรับ **flex** เป็น 1 จะได้เป็นกล่องแนวยาวลงมา และต่อมา สร้างกล่อง 2 กล่อง โดยใส่ **backgroundColor** แล้ว ใส่ **flex** เป็น 0.3 และ 0.5 ตามลำดับ จะได้กล่องขนาดต่างกัน ต่อมาใส่ **Text** ต่อจาก 2 กล่องที่ผ่านมา

3. ให้นักศึกษำบันทึกผลลัพท์จากโปรแกรมข้างล่างนี้ (**Nested Text : Text Component**)

```

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

const BoldAndBeautiful = () => {
  return (
    <Text style={styles.baseText}>
    
```

```

    I am bold
    <Text style={styles.innerText}> and red</Text>
  </Text>
);
};

const styles = StyleSheet.create({
  baseText: {
    fontWeight: 'bold'
  },
  innerText: {
    color: 'red'
  }
});

export default BoldAndBeautiful;

```

บันทึกผลการทดลอง :

I am bold and red

สร้าง **Text 2** ตัว โดยตัวแรกใส่ **fontWeight** เป็น **bold** ซึ่งจะได้ตัวหนา ตัวที่ 2 ใส่ **color red** จะได้ตัวอักษรสีแดง

4. ให้นักศึกษานำบันทึกผลจากโปรแกรมข้างล่างนี้ (**Multiline : TextInput Component**)

```

import React, { Component } from 'react';
import { View, TextInput } from 'react-native';

const UselessTextInput = (props) => {
  return (
    <TextInput
      {...props} // Inherit any props passed to it; e.g., multiline, numberOfLines below
      editable
      maxLength={40}
    />
  );
};

```

```
    />
  );
}

const UselessTextInputMultiline = () => {
  const [value, onChangeText] = React.useState('พิมพ์ข้อความที่นี่');

  // If you type something in the text box that is a color, the background will change to that
  // color.
  return (
    <View
      style={{
        backgroundColor: value,
        borderBottomColor: '#000000',
        borderBottomWidth: 1,
      }}>
      <UselessTextInput
        multiline
        numberOfLines={4}
        onChangeText={text => onChangeText(text)}
        value={value}
      />
    </View>
  );
}

export default UselessTextInputMultiline;
```

บันทึกผลการทดลอง :



สร้างกล่อง `TextInput` ขึ้นมา จากนั้นทำการสร้าง `state` ที่ใช้สำหรับเปลี่ยนค่า `value` ใน `TextInput` และ เปลี่ยน `backgroundColor` ของพื้นหลัง ที่อ้างอิงจากตัวแปร `value` ที่มาจากการสร้าง `state` ใส่ event `onChangeText` เพื่อทำงานตอนใส่ค่าใส่ `TextInput` ใน `onChangeText` มีฟังก์ชันการทำงานเป็น `onChangeText` (ของ `state`) เพื่อ set ค่า `value` ใหม่ ถ้าค่า `value` เปลี่ยนแล้วได้ค่าเป็นสี สีพื้นหลังก็จะเปลี่ยนตาม

5. ให้นักศึกษำบันทึกผลลัพธ์จากโปรแกรมข้างล่างนี้ (**ScrollView Component**)

```
import React from 'react';
import { StyleSheet, Text, SafeAreaView, ScrollView } from 'react-native';
import Constants from 'expo-constants';
```

```
const App = () => {
  return (
    <SafeAreaView style={styles.container}>
      <ScrollView style={styles.scrollView}>
        <Text style={styles.text}>
```

วิชาเหมือนสินค้า อันมีค่าอยู่เมืองไกล
 ต้องยากลำบากไป จึงจะได้สินค้ามา
 จงตั้งเอากายเจ้า เป็นสำเภาอันโสกา
 ความเพียรเป็นโยธา แขนซ้ายขวาเป็นเสาใบ
 นิ้วเป็นสายระยาง สองเท้าต่างสมอใหญ่

ปากเป็นนางงานไป อัจฉาสัยเป็นเสบียง
 สติเป็นหางเสือ ถือท้ายเรือไว้ให้เที่ยง
 ถือไว้อย่าให้เอียง ตัดแล่นเสียงข้ามคงคา
 ปัญญาเป็นกลองแก้ว ส่องดูแถวแนวหินผา
 เจ้าจงเอาหุตา เป็นล้าดำฟุ้งดูลม
 ชีก็เยจคือปลาร้าย จะทำลายให้เรือจม
 เอาใจเป็นปืนคม ยิ่งระดมให้จมไป
 จึงจะได้สินค้ำมา คือวิชาอันพิสมัยจงหมั่นมั่นหมายใจ อย่าได้คร้านการวิชา

```

    </Text>
  </ScrollView>
</SafeAreaView>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    marginTop: Constants.statusBarHeight,
  },
  scrollView: {
    backgroundColor: 'pink',
    marginHorizontal: 20,
  },
  text: {
    fontSize: 42,
  },
});

```

export default App;

บันทึกผลการทดลอง :



สร้าง **SafeAreaView** เพื่อใส่ตัว **ScrollView** จากนั้นใน **style** ใส่ **marginTop: Constants.statusBarHeight** เพื่อเว้นระยะขอบด้านบนเพื่อไม่ให้ทับกับ **status bar** ของหน้าจอ ใน **ScrollView** ใส่ **marginHorizontal: 20** เป็นการเว้นระยะขอบในทางซ้ายขวา จากนั้นก็ใส่ **Text** กำหนด **style** ได้ปกติ แต่เพิ่มการเลื่อนหน้าจอได้

ให้นักศึกษานำที่ผลลัพธ์จากโปรแกรมข้างล่างนี้ (**Compose : StyleSheet View**)

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

const App = () => (
  <View style={container}>
    <Text style={text}>ไอที สจล</Text>
  </View>
);

const page = StyleSheet.create({
  container: {
    flex: 1,
    padding: 24,
    backgroundColor: 'orange',
  },
});
```

```

    },
    text: {
      fontSize: 48,
      color: '#000'
    },
  });

const lists = StyleSheet.create({
  listContainer: {
    flex: 1,
    backgroundColor: '#61dafb',
  },
  listItem: {
    fontStyle: 'italic',
    fontWeight: 'bold'
  },
});

const container = StyleSheet.compose(page.container, lists.listContainer);
const text = StyleSheet.compose(page.text, lists.listItem);

export default App;
    
```

บันทึกผลการทดลอง :



สร้างกล่อง View ที่มี Text ปกติ แล้วสร้าง StyleSheet มา 2 ตัว ชื่อ page กับ lists ถัดมาสร้างตัวแปรสองตัว ชื่อ container และ Text โดยมาจากการทำ StyleSheet compose นำ StyleSheet 2 ตัวแรกมารวมกัน โดยจะถ้าซ้ำจะยึดตามตัวหลัง แล้วเอาไปใส่ style ของ View และ Text ตามลำดับ

6. ให้นักศึกษานำที่ผลลัพธ์จากโปรแกรมข้างล่างนี้ (**absoluteFillObject : StyleSheet Component**)

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

```
const App = () => (
  <View style={styles.container}>
    <View style={styles.box1}>
      <Text style={styles.text}>ก</Text>
    </View>
    <View style={styles.box2}>
      <Text style={styles.text}>ข</Text>
    </View>
    <View style={styles.box3}>
      <Text style={styles.text}>ค</Text>
    </View>
  </View>
);
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1
  },
  box1: {
    position: 'absolute',
    top: 40,
    left: 40,
    width: 100,
    height: 100,
    backgroundColor: 'red'
  },
  box2: {
    ...StyleSheet.absoluteFill,
    top: 120,
    left: 50,
    width: 100,
    height: 100,
    backgroundColor: 'blue'
  }
});
```

```

    },
    box3: {
      ...StyleSheet.absoluteFillObject,
      top: 120,
      left: 120,
      width: 100,
      height: 100,
      backgroundColor: 'green'
    },
    text: {
      color: '#FFF',
      fontSize: 80
    }
  });

export default App;

```

บันทึกผลการทดลอง :



สร้างกล่อง 3 กล่อง โดยจะมีการตั้ง **position** เป็น **absolute** ที่จะย้ายไปไหนมาไหนก็ได้ภายในหน้าจอแล้วจะอยู่คงที่ที่หน้าจอจนตลอด แล้วต่อมากล่องที่ 2 ใช้ **object StyleSheet.absoluteFillObject** ที่จะเป็นตัวตั้งต้นสำหรับ **View** ที่จะซ้อนกับกล่องก่อนหน้า ซึ่ง **absoluteFillObject** ที่ก็มีหลักการไม่ต่างกัน

7. ให้นักศึกษานำบันทึกผลลัพธ์จากโปรแกรมข้างล่างนี้ (**hairlineWidth : StyleSheet Component**)

```

import React from "react";
import { StyleSheet, Text, View } from "react-native";

const App = () => (
  <View style={styles.container}>
    <Text style={styles.row}>คณะเทคโนโลยีสารสนเทศ</Text>
    <Text style={styles.row}>สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง</Text>
  </View>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 24
  },
  row: {
    padding: 4,
    borderBottomColor: "red",
    borderBottomWidth: StyleSheet.hairlineWidth
  }
});

export default App;
    
```

บันทึกผลการทดลอง :



11:27 น. ๒ ...

๗๓

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง



ใน style ของ Text ทุกตัว มีการใส่เส้นขอบ `borderBottomWidth` ที่หมายถึงขอบด้านล่าง ตั้งเป็น `StyleSheet.hairlineWidth` ซึ่งหมายถึงการขีดเส้นใต้