ES6

Class

class Employee {

setName (name) {

this.name = name;

}

getName () {

return this.name;

}

setAge (age) {

this.age = age;

}

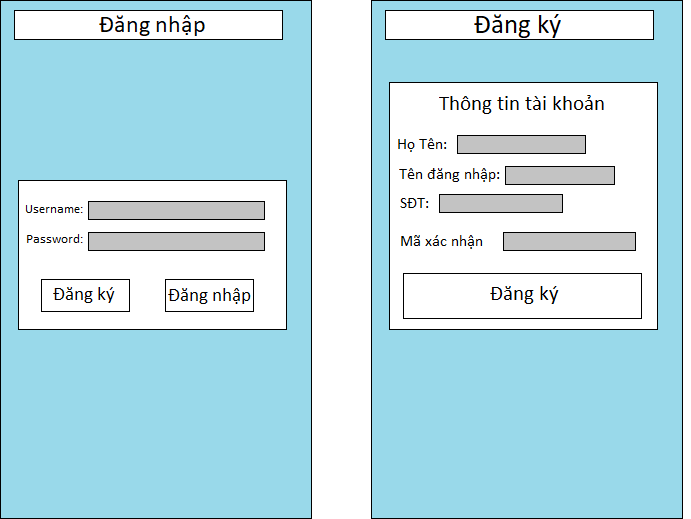
getAge () {

return this.age;

}

};

**Vẽ màn hình**



**Set**

Set như là một kiểu dữ liệu mảng một chiều.

Một Object Set chỉ có thể lưu mình keys, còn Map có thể lưu keys và values theo từng cặp.

Một số minh họa cho cách làm việc với set:

var set = new Set();

// Add key

set.add({x: 1});

set.add(2);

set.add("text");

console.log(set.has("text")); // kiếm tra xem có key == 'text' hay không.

set.delete(1); // xóa một key

var set\_1 = new Set([1, 2, 3, 4]); // khởi tạo một set bằng dữ liệu là một mảng

console.log(set\_1.size); // in ra size của 1 set "5"

var set\_2 = new Set(set\_1.values()); // sao chép các giá trị key từ một set khác.

**Reacte-Native**

**Styles**

Với React-Native hỗ trợ style cho tất cả các components . Nó đơn giản như chúng ta viết CSS cho Web. Tuy nhiên có một số thay đổi về quy định cách thức viết các selector, hay cách gọi css… chúng ta cần nắm rõ các quy định đó. Ví dụ trong CSS sử dụng cho HTML chúng ta viết background-color thì sang React-Native chúng ta sử dụng backgroundColor Thao tác với css cho đối tượng, tức là chúng ta sẽ thao tác trong class hiển thị ra phần view

import React, { Component } from 'react';

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

const styles = StyleSheet.create({

bigblue: {

color: 'blue',

fontWeight: 'bold',

fontSize: 30,

},

red: {

color: 'red',

},

});

export default class LotsOfStyles extends Component {

render() {

return (

<View>

<Text style={styles.red}>just red</Text>

<Text style={styles.bigblue}>just bigblue</Text>

<Text style={[styles.bigblue, styles.red]}>bigblue, then red</Text>

<Text style={[styles.red, styles.bigblue]}>red, then bigblue</Text>

</View>

);

}

}

Flexbox

Là một thuộc tính giúp căn chỉnh vị trí tương đối của các thành phần con trong thành phần cha. Cách hoạt động giống Flexbox của CSS cho HTM. Có thể sử dụng các thuộc tính flexDirection ,alignItems và justifyContent để thiết kế layout cho vị trí các component. Có 2 đối tượng chính trong Flexbox:

* Đối tượng bao quanh ( container ) chính là view cha
* Đối tượng con bên trong ( items) là các view con

import React, { Component } from 'react';

import { AppRegistry, View } from 'react-native';

export default class FlexDirectionBasics extends Component {

render() {

return (

// Try setting `flexDirection` to `column`.

<View style={{flex: 1, flexDirection: 'row'}}>

<View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />

<View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />

<View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />

</View>

);

}

};

// skip this line if using Create React Native App

AppRegistry.registerComponent('AwesomeProject', () => FlexDirectionBasics);