**JavaScript**

**Array Dimensional Exemplo.**

Var lista\_compras = Array();

Var lista\_compras = [];

Var lista\_compras = [‘Arroz’, ‘Feijão’];

**Array Multidimensional Exemplo**

var automoveis = Array();

automoveis['carros'] = Array();

automoveis['motos'] = Array();

automoveis['carros'][1] = 'Fiat Palio';

automoveis['carros'][2] = 'Fiat Punto';

automoveis['carros'][3] = 'Honda Civic';

automoveis['carros'][4] = 'Honda Fit';

automoveis['motos'][1] = 'Yamaha Fazer';

automoveis['motos'][2] = 'Yamaha 2';

automoveis['motos'][3] = 'Suzuki 1';

automoveis['motos'][4] = 'Suzuki 2';

document.write(automoveis['carros'])

**ifElse**

if(2 === ‘2’) falso (=== compara o tipo também)

if(2 == ‘2’) verdadeiro

**var e let**

var = escopo de função

let = escopo de bloco

**Classe e Objetos Exemplo**

class Casa{

constructor(corAtributo, qtdQuartosAtributo){

this.cor = corAtributo;

this.qtdQuartos = qtdQuartosAtributo;

}

getInformacoes(){

document.write("Cor: " + this.cor + " Quatidade de quartos: " + this.qtdQuartos);

}

static abrirPortao(){

document.write("Abriu o portao");

}

}

//var casa = new Casa("Amarela", 20);

//casa.getInformacoes();

Casa.abrirPortao();

**Arrow Functions**

/\*

var soma = function(a, b){

return a+b;

}\*/

var soma = (a, b) => { return a + b };

document.write(soma(9, 7))

**Destruction**

[a, b, c, d] = [10, 15, 40, 75];

({nome, idade, sexo} = {nome: 'José', idade: 33, sexo: 'M'}) //Tem que ter o mesmo nome

document.write(a)

=======================================================================

**Exemplos Aulas:**

**Exemplo flex react**

//Import

import React from 'react';

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

//Formatações

const Estilos = {

principal: {

flex: 1, //1:1 1 ocupa de 1

backgroundColor: 'cornflowerblue',

},

topo: { //2:11

flex: 2,

backgroundColor: 'brown'

},

conteudo: { //8:11

flex: 8,

backgroundColor: 'yellowgreen'

},

rodape: { //1:11

flex: 1,

backgroundColor: 'orangered'

}

};

//Criar o componente

const App = () => {

const { principal, topo, conteudo, rodape } = Estilos;

return (

<View style={principal}>

<Text style={topo}>Topo</Text>

<Text style={conteudo}>conteudo</Text>

<Text style={rodape}>Rodape</Text>

</View>

);

};

//Renderizar para o dispositivo

AppRegistry.registerComponent('app02', () => App);

**Exemplo Button**

//Import

import React from 'react';

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

//Formatações

const Estilos = {

principal: {

paddingTop: 40

}

};

const botaoPressionado = () => {

alert('Botão Pressionado');

};

//Criar o componente

const App = () => {

const { principal } = Estilos;

return (

<View style={principal}>

<Button

onPress={botaoPressionado}

title="Clique aqui"

color="#841584"

accessibilityLabel="Clique para abrir as notícias!!"

/>

</View>

);

};

//Renderizar para o dispositivo

AppRegistry.registerComponent('app02', () => App);

**Exemplo Botão Customizado**

//Import

import React from 'react';

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

//Formatações

const Estilos = {

principal: {

paddingTop: 40

},

botao: {

backgroundColor: '#48BBEC',

padding: 10,

borderColor: '#1D8EB8',

borderWidth: 1,

borderRadius: 8

},

textoBotao: {

color: '#fff',

fontSize: 16,

fontWeight: 'bold',

alignSelf: 'center'

}

};

//Criar o componente

const App = () => {

const { principal, botao, textoBotao } = Estilos;

return (

<View style={principal}>

<TouchableOpacity style={botao}>

<Text style={textoBotao}>Clique Aqui</Text>

</TouchableOpacity>

</View>

);

};

//Renderizar para o dispositivo

AppRegistry.registerComponent('app02', () => App);

**Props e State**

**Props:** Valor enviado ao componente é fixo, ou seja, o valor recuperado no momento da execução é o mesmo.

**Exemplo props**

import React, { Component } from 'react';

import {

AppRegistry,

StyleSheet,

Text,

View

} from 'react-native';

class Componente extends Component {

render() {

return (

<View>

<Text>{this.props.setor}</Text>

<Text>{this.props.xyz}</Text>

<Text>{this.props.p}</Text>

</View>

);

}

}

class app03 extends Component {

render() {

return (

<Componente

setor='texto1'

xyz='texto2'

p='text' />

);

}

}

AppRegistry.registerComponent('app03', () => app03);

**State:** Valores alterados

import React, { Component } from 'react';

import {

AppRegistry,

StyleSheet,

Button,

Text,

View

} from 'react-native';

class MeuComponente extends Component {

render() {

return (

<View>

<Text>{this.props.teste}</Text>

</View>

);

}

}

class app03 extends Component {

constructor(props) {

super(props);

this.state = { texto: 'Texto teste 2' };

}

alteraTexto(){

this.setState({ texto: 'Outra coisa' })

}

render() {

return (

<View>

<MeuComponente teste={this.state.texto}/>

<Button

title='Botão'

onPress={() => {this.alteraTexto()}}

/>

</View>

);

}

}

AppRegistry.registerComponent('app03', () => app03);

Método carregado antes de renderizar o componente com o render

componentWillMount() {

justifyContent vertical

alignItem horizontal

**Firebase – Salvar, excluir e atualizar dados**

// import firebase from 'firebase';

import \* as firebase from 'firebase/app';

import 'firebase/database';

import React, { Component } from 'react';

import {

  AppRegistry,

  Text,

  Button,

  View

} from 'react-native';

export default class firebaseTest extends Component {

  constructor(props) {

    super(props);

    this.state = { pontuacao: '' };

  }

  componentWillMount() {

    const config = {

      apiKey: 'AIzaSyAKf3V\_4K8oNersvFvvvw6iBXbun7MIFgM',

      authDomain: 'configuracaofirebasereac-464d1.firebaseapp.com',

      databaseURL: 'https://configuracaofirebasereac-464d1.firebaseio.com',

      projectId: 'configuracaofirebasereac-464d1',

      storageBucket: '',

      messagingSenderId: '336663632054',

      appId: '1:336663632054:web:66544621c8ec0a1ac06e17'

    };

    firebase.initializeApp(config);

  }

  salvarDados() {

    // const database = firebase.database();

    // database.ref('pontucao').set('100');

    const funcionarios = firebase.database().ref('funcionarios');

    // funcionarios.child('001').child('nome').set('Alisson');

    // funcionarios.child('002').child('nome').set('João');

    // funcionarios.child('003').child('nome').set('Pedro');

    // funcionarios.push().child('nome').set('Pedro'); //Definir id unico

    funcionarios.push().set({

      nome: 'Alisson teste',

      altura: '1,75',

      peso: '60Kg'

    });

  }

  listarDados() {

    const pontuacao = firebase.database().ref('pontuacao');

    pontuacao.on('value', (snapshot) => { //Listner ouvinte chama a função toda vez que é alterado

      // eslint-disable-next-line no-undef

      this.setState({ pontuacao: snapshot.val() });

    });

  }

  render() {

    return (

      <View>

        <Button

          onPress={() => { this.salvarDados(); }}

          title="Salvar dados"

          color="#841584"

          accessibilityLabel="Salvar dados"

        />

        <Button

          onPress={() => { this.listarDados(); }}

          title="Listar dados"

          color="#841584"

          accessibilityLabel="Listar dados"

        />

        <Text>{this.state.pontuacao}</Text>

      </View>

    );

  }

}

AppRegistry.registerComponent('firebaseTest', () => firebaseTest);

**Exemplo Redux**

//CONTADOR

//Reducers

function contador(state=0, action){

switch(action.type){

case 'INCREMENTO':

return state + 1;

case 'DECREMENTO':

return state - 1;

case 'ZERAR\_CONTADOR':

return 0;

}

return state;

}

//Criar a store

let store = Redux.createStore(contador);

//Recuperar o estado

document.write(store.getState() + '<br/>') ;

store.dispatch({ type: 'INCREMENTO' })

store.dispatch({ type: 'INCREMENTO' })

store.dispatch({ type: 'INCREMENTO' })

store.dispatch({ type: 'INCREMENTO' })

document.write(store.getState() + '<br/>') ;

store.dispatch({ type: 'DECREMENTO' })

document.write(store.getState() + '<br/>') ;

store.dispatch({ type: 'ZERAR\_CONTADOR' })

document.write(store.getState() + '<br/>') ;

**Exemplo Entry point**

**Como não manipula estados dos componentes, pode usar dessa forma**

import React from 'react';

import { AppRegistry } from 'react-native';

import App from './src/app';

const app08 = props => (

   <App />

);

AppRegistry.registerComponent('app08', () => app08);

export default props => (

    <View>

        <Text>Oi</Text>

    </View>

);