https://www.codecademy.com/learn/learn-react-native/modules/styling-react-native/cheatsheet

All Cheat Sheet  
  
C:\projects\react-native\war\app.json

{

  "expo": {

    "name": "jaisriram",

    "slug": "jaisriram",

    "version": "1.0.0",

    "orientation": "portrait",

    "icon": "./assets/icon.png",

    "userInterfaceStyle": "light",

    "splash": {

      "image": "./assets/splash.png",

      "resizeMode": "contain",

      "backgroundColor": "#ffffff"

    },

    "assetBundlePatterns": [

      "\*\*/\*"

    ],

    "ios": {

      "supportsTablet": true

    },

    "android": {

      "adaptiveIcon": {

        "foregroundImage": "./assets/adaptive-icon.png",

        "backgroundColor": "#ffffff"

      },

      "package": "com.pradeep786expo.jaisriram"

    },

    "web": {

      "favicon": "./assets/favicon.png"

    },

    "extra": {

      "eas": {

        "projectId": "0b09e580-81c1-498b-aa61-497468bef084"

      }

    },

    "owner": "pradeep786expo"

  },

  "extra": {

    "eas": {

      "projectId": "0b09e580-81c1-498b-aa61-497468bef084"

    }

  }

}

C:\projects\react-native\war\App.js

import \* as React from 'react';

import {NavigationContainer} from '@react-navigation/native';

import {createNativeStackNavigator} from '@react-navigation/native-stack';

import HomeScreen from './HomeScreen';

import Bat from './Bat';

import Cat from './Cat';

const Stack = createNativeStackNavigator();

const App = () => {

  return (

    <NavigationContainer>

      <Stack.Navigator>

        <Stack.Screen

          name="Home"

          component={HomeScreen}

          options={{title: 'Welcome'}}

        />

        <Stack.Screen name="Bat" component={Bat} />

        <Stack.Screen name="Cat" component={Cat} />

      </Stack.Navigator>

    </NavigationContainer>

  );

};

export default App;

C:\projects\react-native\war\HomeScreen.js

import \* as React from 'react';

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

const HomeScreen = ({navigation}) => {

  return (

<View>

    <Button

      title="Go to Cat's profile"

      onPress={() =>

        navigation.navigate('Cat', {name: 'Jane'})

      }

    />

    <Button

      title="Go to Bat's profile"

      onPress={() =>

        navigation.navigate('Bat', {name: 'Jane'})

      }

    />

    </View>

  );

};

const ProfileScreen = ({navigation, route}) => {

  return <Text>This is {route.params.name}'s profile</Text>;

};

export default HomeScreen;

C:\projects\react-native\war\Cat.js

import React, {useState} from 'react';

import { StyleSheet, TextInput, Text, View, Image, ScrollView } from 'react-native';

const Cat = () => {

  return (

    <ScrollView>

      <Text>Some text</Text>

      <View>

        <Text>Some more text</Text>

        <Image

          source={{

            uri: 'https://reactnative.dev/docs/assets/p\_cat2.png',

          }}

          style={{width: 200, height: 200}}

        />

      </View>

      <TextInput

        style={{

          height: 40,

          borderColor: 'gray',

          borderWidth: 1,

        }}

        defaultValue="You can type in me"

      />

    </ScrollView>

  );

}

export default Cat;

C:\projects\react-native\war\Bat.js

import React, {useState} from 'react';

import { StatusBar } from 'expo-status-bar';

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

const Bat = () => {

  const [ftext, setFtext] = useState('');

  const getFullName = (

    firstName,

    secondName,

    thirdName,

  ) => {

    return firstName + ' ' + secondName + ' ' + thirdName;

  };

  return (

    <View style={styles.container}>

      <Text>Jai</Text>

      <Text>Sri Ram</Text>

      <Text>Hello, I am {getFullName('Rum', 'Tum', 'Tugger')}!</Text>

      <Image source={require('./img/photo.jpg')} style={{width: 40, height: 40}} />

      <TextInput

        style={{height: 40,borderWidth:"1px", borderBlockColor:"red"}}

        placeholder="Type here to translate!"

        onChangeText={newText => setFtext(newText)}

        defaultValue={ftext}

      />

      <Text style={{padding: 10, fontSize: 42 }}>

        {ftext

          .split(' ')

          .map(word => word && '🍕')

          .join(' ')}

      </Text>

      <StatusBar style="auto" />

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});

export default Bat;

API Call  
C:\projects\react-native\war\component\MovieList.js  
  
import React, {useEffect, useState} from 'react';

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

const MovieList = () => {

  const [isLoading, setLoading] = useState(true);

  const [data, setData] = useState([]);

  const getMovies = async () => {

    try {

      const response = await fetch('https://reactnative.dev/movies.json');

      const json = await response.json();

      setData(json.movies);

    } catch (error) {

      console.error(error);

    } finally {

      setLoading(false);

    }

  };

  useEffect(() => {

    getMovies();

  }, []);

  return (

    <View style={{flex: 1, padding: 24}}>

      {isLoading ? (

        <ActivityIndicator />

      ) : (

        <FlatList

          data={data}

          keyExtractor={({id}) => id}

          renderItem={({item}) => (

            <Text>

              {item.title}, {item.releaseYear}

            </Text>

          )}

        />

      )}

    </View>

  );

};

export default MovieList;

<https://www.codecademy.com/learn/learn-react-native/modules/styling-react-native/cheatsheet>

**style Property**

Components can be styled using the style={} property, which accepts objects as inline-styling, style definitions created by StyleSheet, or an array of objects/definitions to compose styling.

<Text style={styles.paragraph} />

<Text style={{ fontSize: 16 }} />

<Text style={[styles.paragraph, { c

### Using StyleSheet Definitions

Splitting styling properties from the render method using StyleSheet definitions makes the rendering method more readable.

const styles = StyleSheet.create({

box: {

width: 100,

height: 100,

backgroundColor: 'red'

},

});

### Flex in React Native

Layouts are defined with Flex-like rules to account for a wide variety of screen sizes. The major difference between Flex on web and Flex in React Native is that a parent element with display: flex is not required.

<View style={{ flexDirection: 'row' }}>

<View style={{ flex: 1 }} />

<View style={{ flex: 1 }} />

<View style={{ flex: 1 }} />

</View>

D

<View style={{ flexDirection: 'row' , justifyContent: 'flex-start' }}>

<View style={{ flex: 1 }} />

<View style={{ flex: 1 }} />

<View style={{ flex: 1 }} />

</View>

D

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

S Navigation

import { NavigationContainer } from '@react-navigation/native';

const App = () => (

<NavigationContainer>

{ /\* Insert your navigators and content here \*/ }

</NavigationContainer>

);

### createStackNavigator Factory Method

In the react-navigation library, the stack navigator is created by the createStackNavigator factory method.

const Stack = createStackNavigator();

<Stack.Navigator>

<Stack.Screen name="Feed" component={FeedScreen} />

<Stack.Screen name="Catalog" component={CatalogScreen} />

</Stack.Navigator>

### createBottomTabNavigator Factory Method

In the react-navigation library, the bottom tab navigator is created by the createBottomTabNavigator factory method.

const Tab = createBottomTabNavigator();

<Tab.Navigator>

<Tab.Screen name="Feed" component={FeedScreen} />

<Tab.Screen name="Catalog" component={CatalogScreen} />

</Tab.Navigator>

### useNavigation Hook

In the react-navigation library, the useNavigation hook provides access to the navigation

/ Using properties, only available in screen components

const FeedScreen = (props) => {

const nav = props.navigation;

return (

<Button

title="Go to home"

onPress={() => nav.navigate('Home')}

/>

);

};

// Using the hook, available in all components

const HomeButton = () => {

const nav = useNavigation();

return (

<Button

title="Go to home"

onPress={() => nav.navigate('Home')}

/>

);

};

### Factory Methods

In the react-navigation library, all navigators are created by factory methods that use a naming pattern like create<type>Navigator(),

const Stack = createStackNavigator();

const Tab = createBottomTabNavigator();

// You can replace Tab with any other factory method result.

<Tab.Navigator>

<Tab.Screen name="Feed" component={FeedScreen} />

<Tab.Screen name="Catalog" component={CatalogScreen} />

</Tab.Navigator>

### Drawer Navigation

In the drawer navigation pattern, a user uses a pane that can be opened by either swiping or tapping a button, which provides a menu where users can switch between screens.