Example 1: for class component

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

import React,{useState} from 'react'

import FoodList from './screens/FoodList'

import FunctionalComp from './screens/FunctionalComp'

const Index = () => {

    const [extraVal, setExtraVal] = useState<string>('emp')

  return (

    <View>

      <Text>{extraVal}</Text>

      <FoodList nm={'Kareem'} sal={234567}/>

      <hr/>

      <FunctionalComp nm={'Kareem'} sal={234567} />

    </View>

  )

}

export default Index

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

import React, { Component } from 'react'

interface ComingVal {

    nm: string,

    sal:number

}

export class FoodList extends Component<ComingVal> {

  render() {

    return (

      <View>

        <Text>{this.props.nm}</Text>

        <Text>{this.props.sal}</Text>

      </View>

    )

  }

}

export default FoodList

example if functional Component

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

import React from 'react'

interface ComingVal {

    nm: string,

    sal:number

}

const FunctionalComp = ({nm, sal}:ComingVal) => {

  return (

    <View>

      <Text>FunctionalComp</Text>

      <Text>{nm} - <Text>{sal}</Text></Text>

    </View>

  )

}

export default FunctionalComp

IF you want to make it more strict with type script the define as

const FunctionalComp:FunctionComponent<ComingVal> = ({nm, sal}:ComingVal):ReactElement => {

  return (

    <View>

      <Text>FunctionalComp</Text>

      <Text>{nm} - <Text>{sal}</Text></Text>

    </View>

  )

}