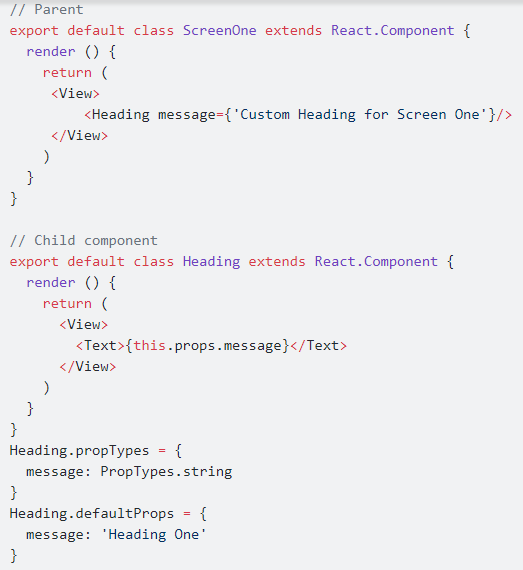
**Tổng quan về Component**

Props: props là viết tắt của Properties. Một điều mà bạn cần phải nhớ khi sử dụng props đó là không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu **immutable**.

Bạn có thể tạo ra component sử dụng props. Ý tưởng của props đó là việc trừu tượng hoá các component để có thể sử dụng được ở nhiều chỗ khác nhau trong app. Hãy xem một ví dụ đơn giản sau:



Trong ví dụ trên thì chúng ta có component Heading, với props là message. Class cha ScreenOnesẽ quyết định giá trị của message, và trong trường hợp trên thì là "Custom Heading for Screen One". Sở dĩ nói component Heading có thể được tái sử dụng ở nhiều nơi là do chúng ta có thể tạo ra một class khác, ScreenTwo chẳng hạn, và thay đổi giá trị của message để dùng lại được component Heading. Các class cha cũng hoàn toàn không cần thay đổi giá trị này nếu không cần thiết, vì bạn có thể set giá trị mặc định cho message (mặc định là "Heading One" trong đoạn code trên).

**State:** state thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi propslà dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu **mutable**.

Ví dụ nếu bạn cần phải điền một cái form với nhiều text input, thì mỗi text input sẽ giữ đoạn text mà bạn gõ vào trong state của nó. Nếu bạn gõ một đoạn text khác vào, state của text input đó sẽ thay đổi và kích hoạt việc render lại component, và text mới của bạn sẽ được vẽ lên.



Trong đoạn code trên thì bạn có thể thấy một class Form với state input. Nó có nhiệm vụ render ra một text input để nhận vào thông tin nhập từ user. Một khi người dùng bắt đầu gõ chữ vào, onChangeText sẽ được gọi đến và sau đó nó gọi setState đối với state input.

Vậy nếu bạn không muốn component render lại khi state thay đổi thì có cách nào không? Đây là lúc chúng ta cần biết về vòng đời của một component.

**Style, With, Hight, Flex trong React Native**

Trong phần này, chúng ta sẽ tìm hiểu sâu hơn về cách tạo và điều chỉnh giao diện trong React Native

**Style**: Thuộc tính style là một thuộc tính đã có từ lâu trong đối tượng JavaScript. Đây là cách đơn giản nhất và cũng là những thứ mà chúng ta sẽ dùng trong code ví dụ. Bạn động thời cũng có thể khai báo một mảng các style, style cuối cùng được khai báo trong mảng style sẽ được ưu tiên, như vậy bạn hoàn toàn có thể sử dụng style theo hình thức kế thừa.

Dường như một thành phần được tạo ra sẽ phức tạp. Vậy nên nó thường được làm rõ bằng cách sử dụng StyleSheet.create để định nghĩa các style và đặt chúng tại một nơi để dễ quản lý. Dưới đây là ví dụ



**Và kết quả**



**Height & Width:** Chiều rộng và dài của các thành phần sẽ xác định kích thước của các thành phần giao diện trên màn hình.

### **Kích thước cố định**

Cách đơn giản nhất để chỉ định kích thước của các thành phần là sử dụng các tham số kích thước cố định width và height trong style. Tất cả kích thước được sử dụng trong React Native đều tuân theo đơn vị pixel khi hiển thị lên trên màn hình.

## 

## height&width

### **Kích thước linh động**

Sử dụng flex trong style của các thành phần để thanh đổi vùng hiển thị một cách linh động dựa trên vùng hiển thị có thể. Thông thường, bạn sẽ sử dụng flex: 1 khi mà bạn muốn một thành phần trải rộng hết mức có thể so với các thành phần khác có chung một view cha. Đây là giá trị tỷ lệ cao nhất mà flex cung cấp. Với các tỷ lệ khác thì nó sẽ so sánh với các thành phần ngang hàng với nó để đưa ra tỷ lệ hiển thị trong cùng một view cha.

## 

## flex

## Flexbox

Một thành phần có thể chỉ định rõ bố cục cho các view con của nó bằng cách sử dụng thuật toán của flexbox. Flexbox được thiết kế để cung cấp giao diện tương đồng trên những màn hình có kích thước khác nhau.

Bạn sẽ thường phải sử dung với các thuộc tính flexDirection, alignItems, và justifyContent để tạo ra được một giao diện đúng ý muốn.

## *Flexbox làm việc với React Native tương tự với cách mà CSS làm việc trên Web. nhưng cũng có một vài ngoại lệ. Giá trị mặc định là khác nhau, ví dụ như đối với thuộc tính flexDirection giá trị mặc định của nó sẽ là column thay thế cho giá trịnh row của CSS trên bản web, và giá trị của alignItems có giá trị mặc định là stretch chứ không phải là flex-start và tham số flexchỉ hỗ trợ số nguyên.*