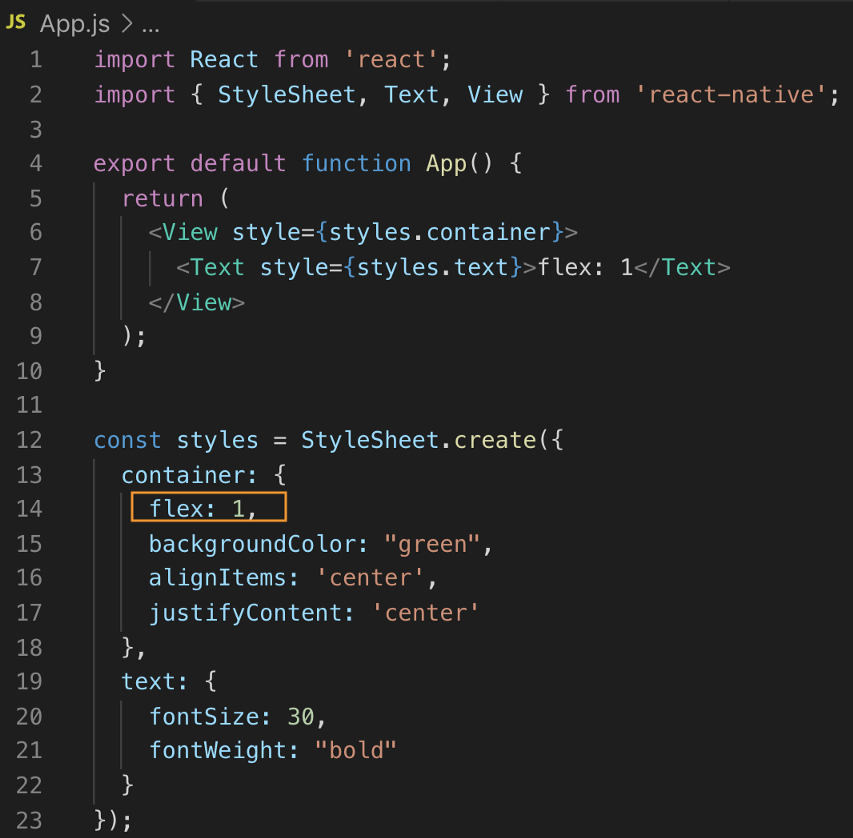
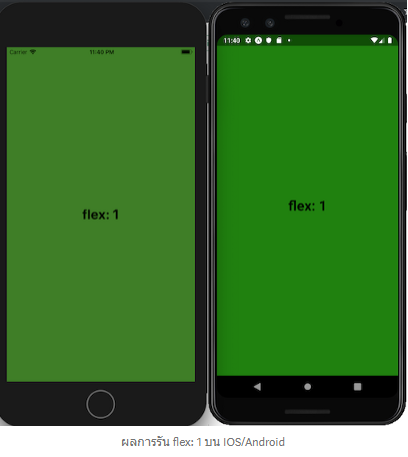
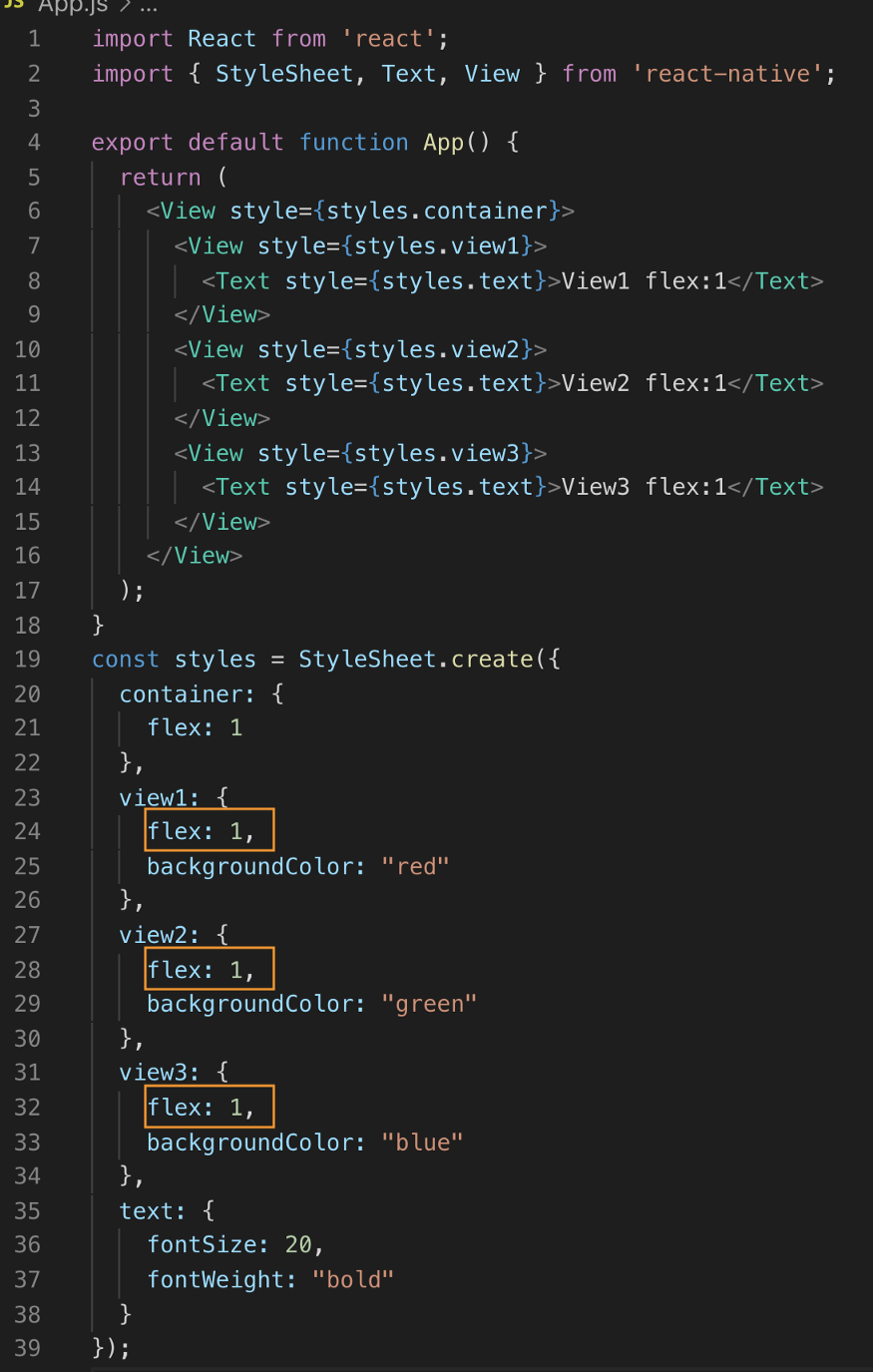
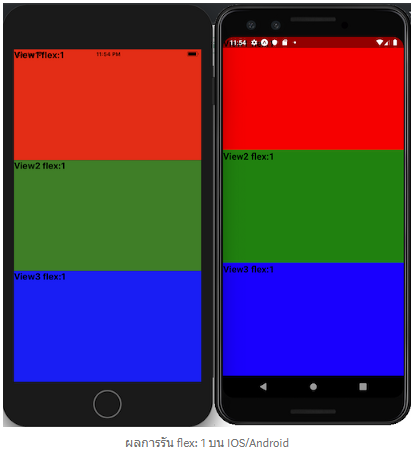
**การจัดหน้าจอด้วย Flexbox**

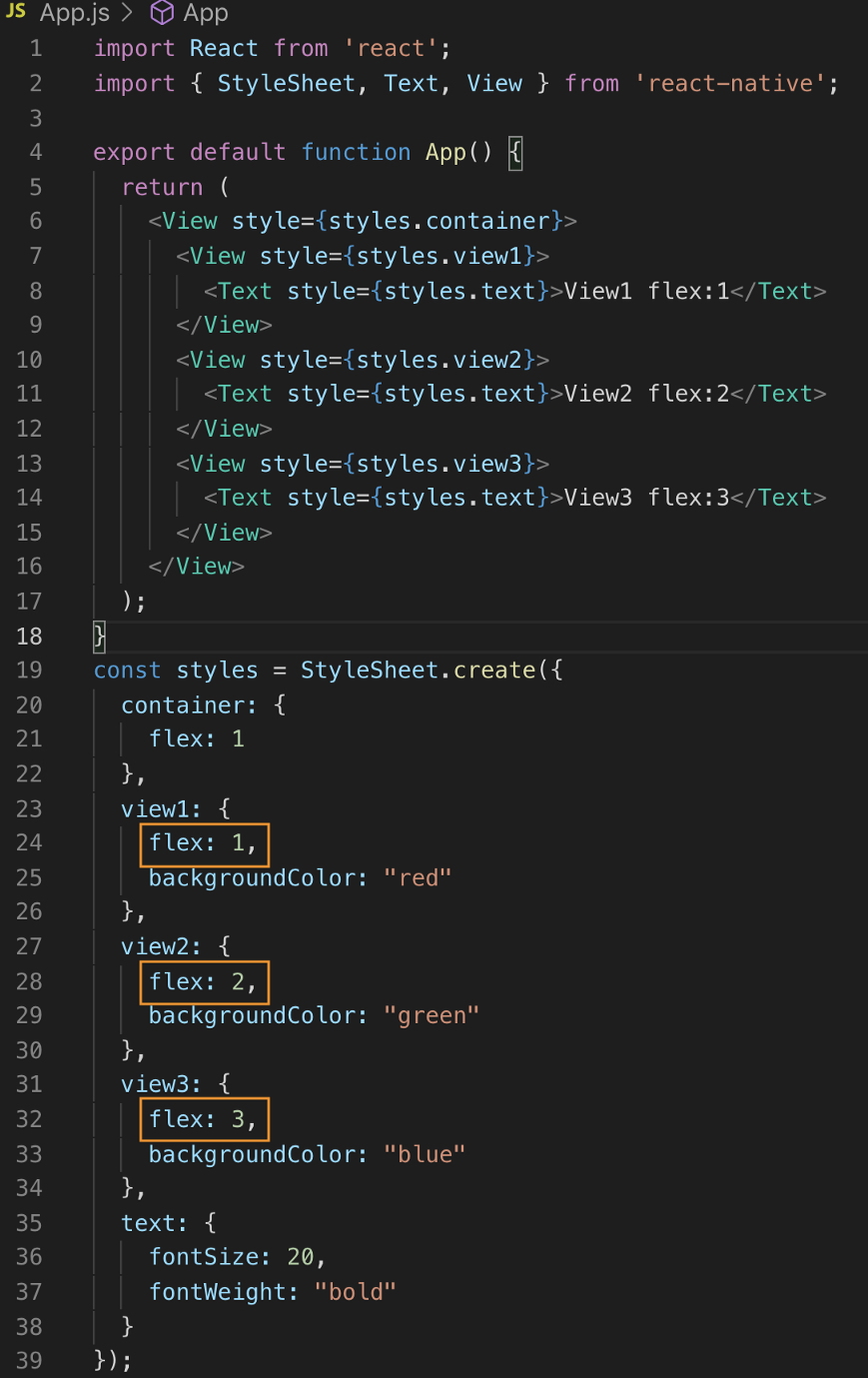
* Flexbox ถูกออกแบบมาเพื่อจัดหน้าจอของอุปกรณ์ขนาดเล็กอย่าง โทรศัพท์ หรือ tablet คนที่เคยทำเว็บน่าจะเคยได้ใช้ใน CSS มาบ้างแล้ว วันนี้เรามาเรียนรู้การใช้งาน flexbox ใน React Native
* เราสามารถจัดหน้าจอของโปรแกรมโดยใช้ Flexbox โดยมีรูปแบบการใช้งานดังนี้
* flex: 1 คือ ขยายเต็มพื้นที่
* ในตัวอย่างนี้จะสร้าง View ขึ้นมา 1 ตัว และกำหนดให้ flex = 1 ผลของมันจะทำให้ View นี้ขยายเต็มหน้าจอ และเราเรียก View ตัวนี้ว่า Flex container

* ถ้ามี View มากกว่า 1 กล่องทุกๆกล่องจะแบ่งอัตราส่วนตามค่า flex ที่กำหนด ตัวอย่างนี้เราจะสร้าง View 3 กล่องให้แต่ละกล่องเป็น flex: 1 เราเรียก view1, view2, view3 ว่า Flex Items

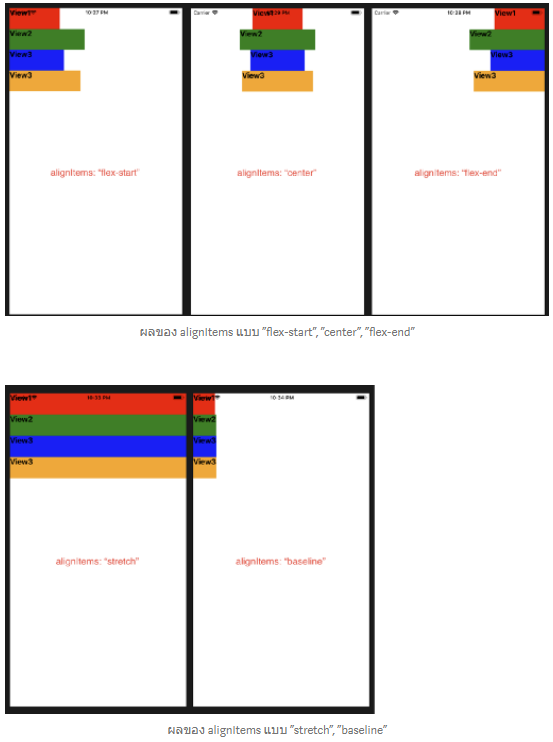
* เราสามารถกำหนด Flex Item ให้มีขนาดไม่เท่ากัน โดยกำหนดค่า flex แต่ละกล่องไม่เท่ากัน มันจะถูกแบ่งตามอัตราส่วน ตัวอย่างนี้ view1, view2, view3 มีค่า flex 1,2,3 ตามรูป ลองกำหนดค่า flex เป็นค่าอื่นๆ แล้วดูผล

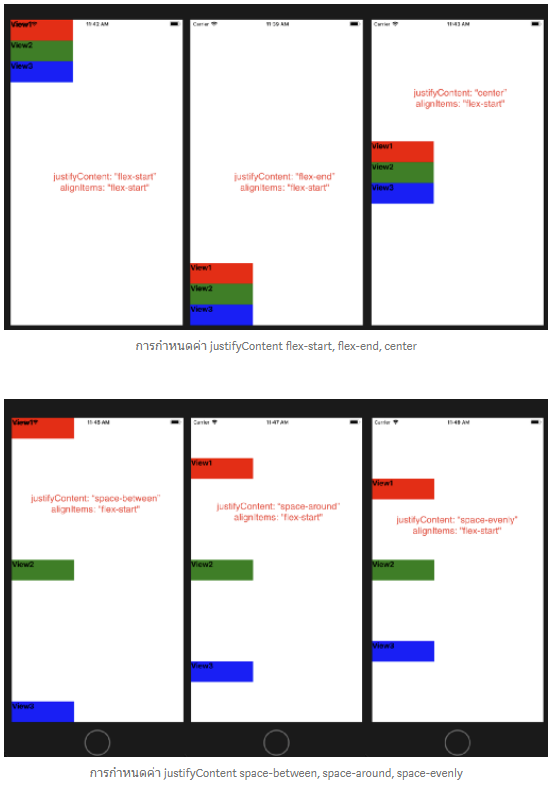
* ถ้า Flex Items มีขนาดตายตัว(มีการกำหนด width, height) เราสามารถกำหนดตำแหน่งมันด้วย properties ในส่วนของ justifyContent และ alignItems



* การกำหนดค่า alignItems แบบต่างๆ “flex-start”, “flex-end”, “center”, “stretch” (ค่า default), “baseline”



* การกำหนดค่า justifyContent แบบต่างๆ “flex-start”(ค่า default), “flex-end”, “center”, “space-betwen”, “space-around”, “space-evenly”

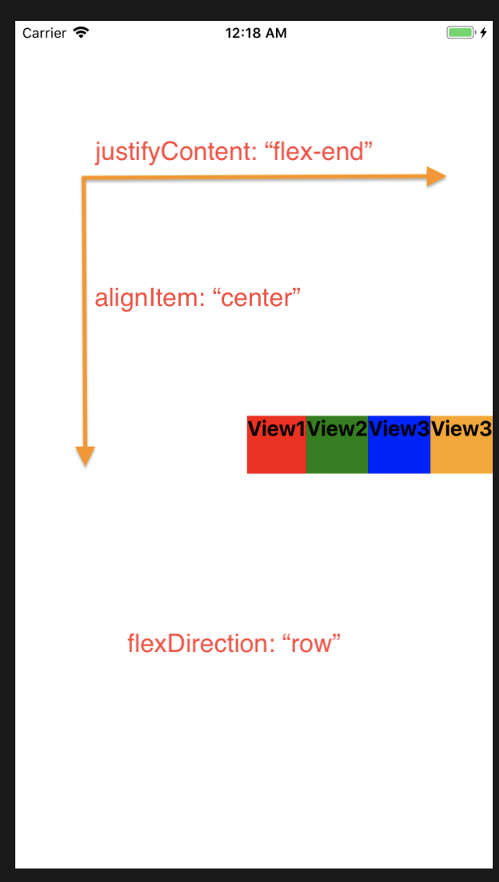


ในกรณีที่เราต้องการใช้ alignItems แบบ “stretch” หรือ “baseline” เราต้องไม่กำหนดค่า width เพราะมันจะเชื่อค่า width ที่เรากำหนด

* จากตัวอย่างที่ผ่านมาการจัด layout มันเป็นแบบ column หรือไล่จากบนลงล่าง แต่ถ้าเราต้องการ layout แบบ row คือเรียงจากซ้ายไปขวา เราสามารถกำหนดได้ด้วย flexDirection
* flexDirection ใช้สำหรับกำหนดว่า Flex Items ของเราจะเรียงยังไง ซึ่งเราสามารถกำหนดค่าได้ 4 แบบ “column”, “column-reverse”, “row”, “row-reverse”
* เรามาดูแบบ “column” (ค่า default) และ “row” ก่อนว่าต่างกันยังไง



* ตัวอย่างนี้เป็นการกำหนด flexDirection แบบ “column” ซึ่งจะได้ผลการรันเหมือนตัวอย่างที่ผ่านมา เนื่องจากเป็นค่า default ตั้งแต่แรกให้ทดลองเปลี่ยน flexDirection เป็น “row” ผลของมันจะทำให้ layout เป็นแบบเรียงจากซ้ายไปขวา
* เมื่อเปลี่ยน flexDirection เป็น “row” จะส่งผลให้ justifyContent มีผลทาง row ส่วน alignItems จะมีผลตามแนว column ดังตัวอย่างนี้

ดังนั้นเราสามารถสรุปได้ว่า justifyContent จะมีผลตามค่าที่กำหนดใน flexDirection ส่วน alignItems จะมีผลในแกนตรงข้ามกับ justifyContent

