# REACT NATIVE FOR WEB

@MIKE_PIDGEON

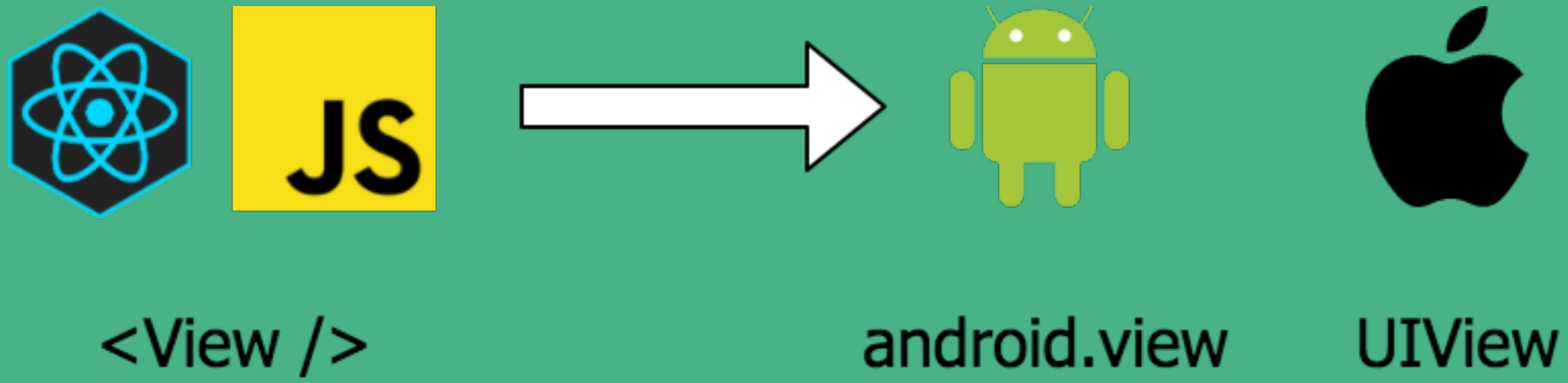
MICHAELPIDGEON.CO.UK

MRM

BRAND GLOBAL, MARKET LOCAL

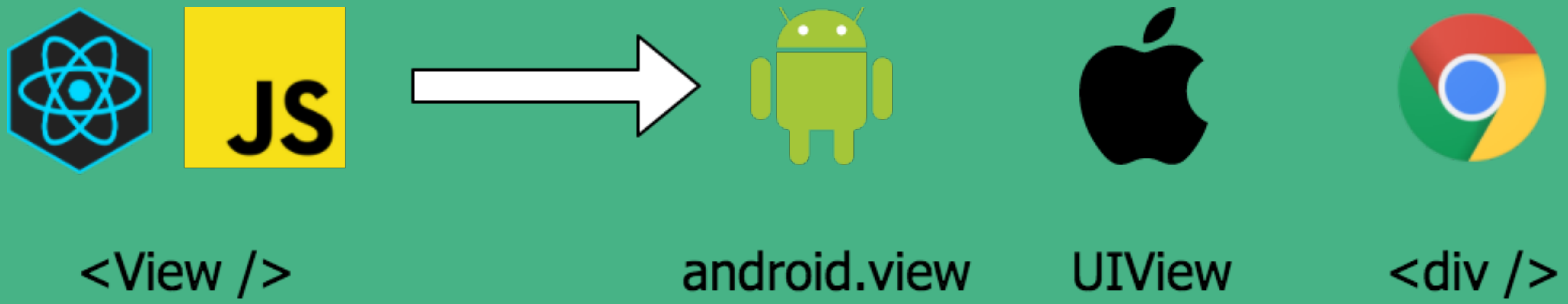
# REACT NATIVE

- Write JS/React with React Native JSX syntax (View, TouchableOpacity etc) -> compose to native components on iOS/Android



# REACT NATIVE FOR WEB

- Write JS/React with React Native JSX syntax (View, TouchableOpacity etc) -> compose to native components on iOS/Android and to web primitives (div, span etc)



<View />

android.view

UIView

<div />

# “SOUNDS SKETCHY - USING REACT NATIVE, TO THEN COMPOSE BACK TO THE WEB??”

- Team A

> “SOUNDS GREAT - WRITE ONCE, RUN ANYWHERE! REACT IS MEANT TO BE CROSS PLATFORM AFTER ALL.”

- Team B

# There are definitely benefits!

- Quickly prototype and start user testing.
- Scaffold your application end to end - allow developers to pick off different bits.

## But is it prod ready?

# Probably not

I couldn’t find much/any evidence of people using it in production.

# WHY NOT?

- Styles are flaky and don’t support media queries/ pseudo selectors
- Compatibility issues with npm modules
- Have to accept sub optimal design, without a lot of code like this:

```
const styles = StyleSheet.create({
  height: (Platform.OS === ‘web’) ? 200 : 100,
});
```

# WHY NOT?

» Styles are flaky and don’t support media queries/ pseudo selectors

» Compatibility issues with npm modules

» Have to accept sub optimal design, without a lot of code like this:

```
const styles = StyleSheet.create({
  height: (Platform.OS === ‘web’) ? 200 : 100,
});
```

Also, it’s really difficult to Google error messages…

# A Better Way Part 1

- Re-write components into plain HTML. It’s easy and you’ll get a better design.

- Can piggyback on React Native cross platform extensions for this:

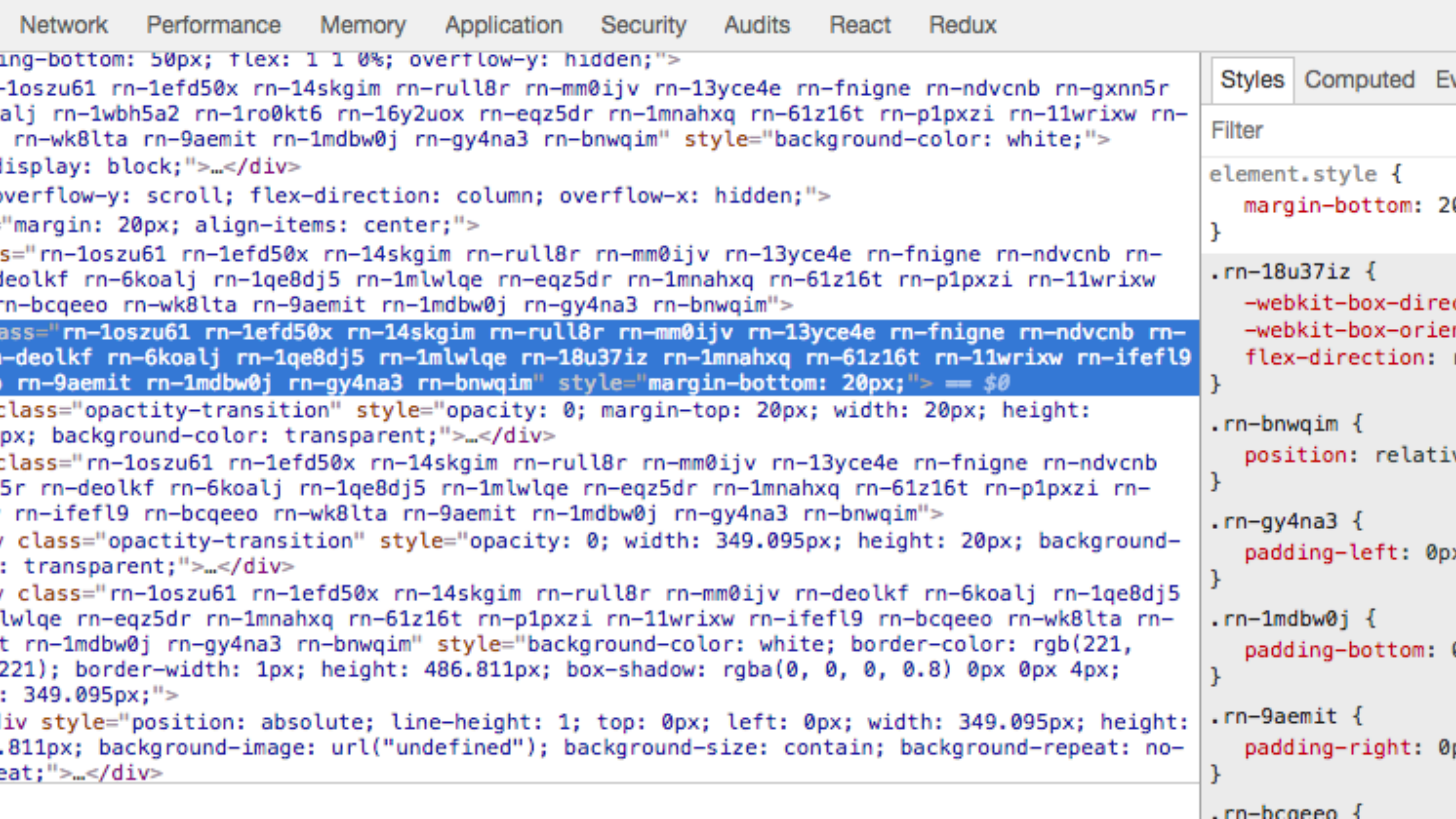
```
button.ios.js // iOS: React Native knows to use
button.android.js // Android: React Native knows to use
button.js // Web: nothing clever going on here, webpack just falls back to the default when on web
```

# A Better Way Part 2

- Plan for for cross platform ~~compatibility~~ migration from the start of the project.
- Extract as much application logic as possible from components into something else.
- If using RN for Web for prototyping, make sure you plan to deprecate it completely.

# HOW NOT TO DO IT

Fasten those seatbelts - we’re about to look at some code



```
ing-bottom: 50px; flex: 1 1 0%; overflow-y: hidden;">
-1oszu61 rn-1efd50x rn-14skgim rn-rull8r rn-mm0ijv rn-13yce4e rn-fnigne rn-ndvcnb rn-gxnn5r
alj rn-1wbh5a2 rn-1ro0kt6 rn-16y2uox rn-eqz5dr rn-1mnahxq rn-61z16t rn-p1pxzi rn-11wrixw rn-
 rn-wk8lta rn-9aemit rn-1mdbw0j rn-gy4na3 rn-bnwqim" style="background-color: white;">
isplay: block;">…</div>
verflow-y: scroll; flex-direction: column; overflow-x: hidden;">
"margin: 20px; align-items: center;">
s="rn-1oszu61 rn-1efd50x rn-14skgim rn-rull8r rn-mm0ijv rn-13yce4e rn-fnigne rn-ndvcnb rn-
deolkf rn-6koalj rn-1qe8dj5 rn-1mlwlqe rn-eqz5dr rn-1mnahxq rn-61z16t rn-p1pxzi rn-11wrixw
rn-bcqeeo rn-wk8lta rn-9aemit rn-1mdbw0j rn-gy4na3 rn-bnwqim">
ass="rn-1oszu61 rn-1efd50x rn-14skgim rn-rull8r rn-mm0ijv rn-13yce4e rn-fnigne rn-ndvcnb rn-
-deolkf rn-6koalj rn-1qe8dj5 rn-1mlwlqe rn-18u37iz rn-1mnahxq rn-61z16t rn-11wrixw rn-ifefl9
 rn-9aemit rn-1mdbw0j rn-gy4na3 rn-bnwqim" style="margin-bottom: 20px;"> == $0
class="opactity-transition" style="opacity: 0; margin-top: 20px; width: 20px; height:
px; background-color: transparent;">…</div>
class="rn-1oszu61 rn-1efd50x rn-14skgim rn-rull8r rn-mm0ijv rn-13yce4e rn-fnigne rn-ndvcnb
5r rn-deolkf rn-6koalj rn-1qe8dj5 rn-1mlwlqe rn-eqz5dr rn-1mnahxq rn-61z16t rn-p1pxzi rn-
 rn-ifefl9 rn-bcqeeo rn-wk8lta rn-9aemit rn-1mdbw0j rn-gy4na3 rn-bnwqim">
 class="opactity-transition" style="opacity: 0; width: 349.095px; height: 20px; background-
: transparent;">…</div>
 class="rn-1oszu61 rn-1efd50x rn-14skgim rn-rull8r rn-mm0ijv rn-deolkf rn-6koalj rn-1qe8dj5
lwlqe rn-eqz5dr rn-1mnahxq rn-61z16t rn-p1pxzi rn-11wrixw rn-ifefl9 rn-bcqeeo rn-wk8lta rn-
t rn-1mdbw0j rn-gy4na3 rn-bnwqim" style="background-color: white; border-color: rgb(221,
221); border-width: 1px; height: 486.811px; box-shadow: rgba(0, 0, 0, 0.8) 0px 0px 4px;
: 349.095px;">
iv style="position: absolute; line-height: 1; top: 0px; left: 0px; width: 349.095px; height:
.811px; background-image: url("undefined"); background-size: contain; background-repeat: no-
eat;">…</div>
```

```
element.style {
  margin-bottom: 20
}
.rn-18u37iz {
  -webkit-box-direc
  -webkit-box-orien
  flex-direction: r
}
.rn-bnwqim {
  position: relativ
}
.rn-gy4na3 {
  padding-left: 0px
}
.rn-1mdbw0j {
  padding-bottom: 0
}
.rn-9aemit {
  padding-right: 0p
}
.rn-bcqeeo {
```

# SUMMARY

- RN for Web is a great tool for prototyping and early stage UI testing.
- Write once, ~~run~~ debug anywhere - a hackneyed concept and a well worn gag.
- Even if a good idea in principle, maybe UI layer is not best place for it.
- In the meantime, let's make sure we're separating our application logic from our components!