CS47: Cross-Platform Mobile Development

Lecture 1A: Introductions and Syllabus

James Landay
Abdallah AbuHashem
Claire Rosenfeld
Ryan Chen

Winter 2021

https://cs47.stanford.edu

Coming soon!

Overview for today

- Logistics and syllabus
- Why React Native? - Introduction to cross-platform mobile development
- JavaScript basics
- Assignment 1 overview
- Exit ticket
- Looking Forward

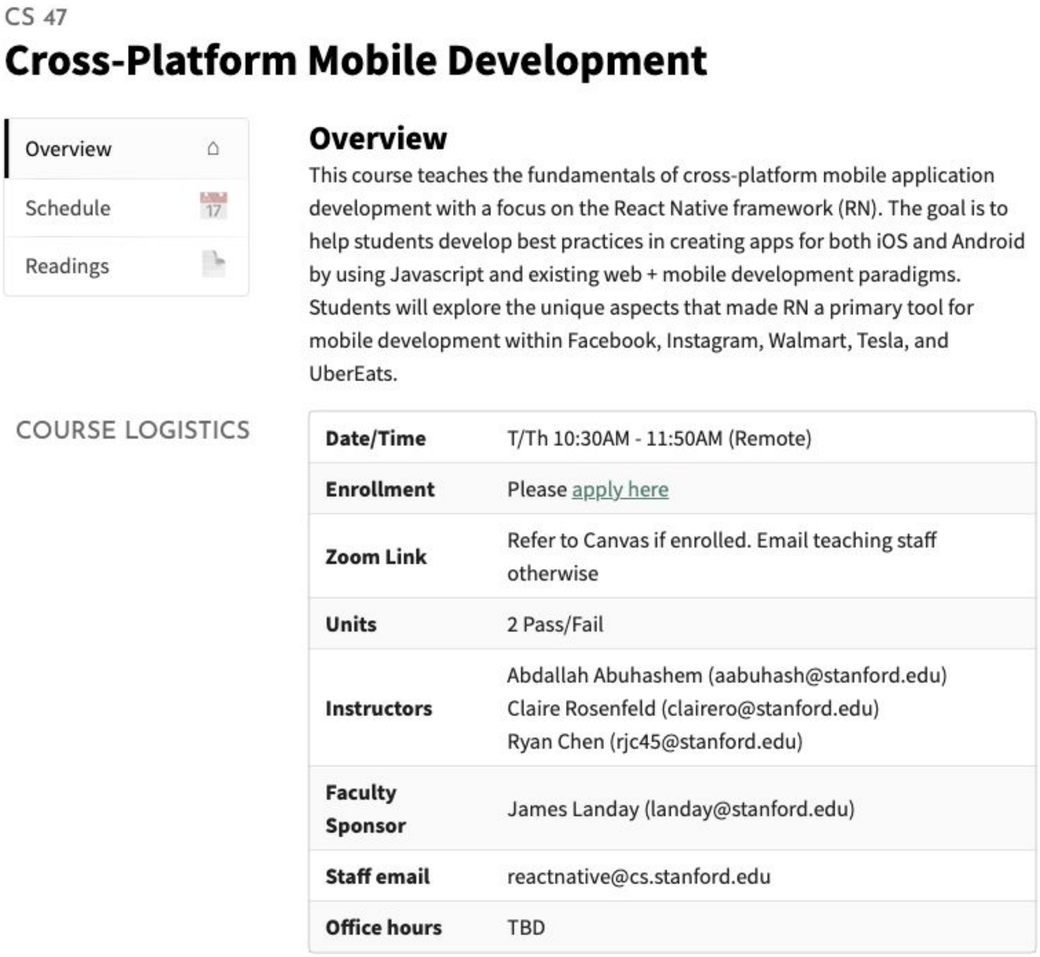
Who are we?

Prof. James Landay
Faculty Advisor

Abdallah AbuHashem
Co-instructor

Claire Rosenfeld
Co-instructor

Ryan Chen
Co-instructor



CS 47

Cross-Platform Mobile Development

Overview

This course teaches the fundamentals of cross-platform mobile application development with a focus on the React Native framework (RN). The goal is to help students develop best practices in creating apps for both iOS and Android by using Javascript and existing web + mobile development paradigms. Students will explore the unique aspects that made RN a primary tool for mobile development within Facebook, Instagram, Walmart, Tesla, and UberEats.

COURSE LOGISTICS

Date/Time	T/Th 10:30AM - 11:50AM (Remote)
Enrollment	Please apply here
Zoom Link	Refer to Canvas if enrolled. Email teaching staff otherwise
Units	2 Pass/Fail
Instructors	Abdallah Abuhashem (aabuhash@stanford.edu) Claire Rosenfeld (clairero@stanford.edu) Ryan Chen (rjc45@stanford.edu)
Faculty Sponsor	James Landay (landay@stanford.edu)
Staff email	reactnative@cs.stanford.edu
Office hours	TBD

https://cs47.stanford.edu

Logistics

- Class Time: TuTh, 10:30 - 11:50 am
- Credit/no credit only
- Unit count: 2. Expected workload: 4 - 7 hrs/wk
- Prerequisites
  - None!
- Attendance
  - Lectures will be recorded, and posted on canvas
  - Attending live is recommended if possible
  - If you choose to watch the recorded version, we expect you do so within 48 hours

Logistics

- Grading
  - All assignments have to be turned in to pass the class.
- Assignments
  - 5 Assignments. Starting from today, with setup, and finishing on week 7.
  - Assignments are designed to solidify each week material understanding within the 4-7 hrs/wk
- Final Project
  - Week 5. Project Idea Writeup. You will propose an idea for an application to build.
  - Weeks 6-9. You are required to build an app that employs the functionalities you will learn in the class.
  - We will help you with the idea for your final project, but you will have lots of input.
  - Week 10 will be presentations week, where you demo your final project.
  - **CS 147 students**: If you use React Native in your CS 147 final project, you are allowed to count it as your final project for this class.

Logistics

- Enrollment
  - You must apply to stay in class/join.
  - Form is on the website.
  - Class caps at 50 students, with priority for CS147 students
  - Codes for enrollment will be sent over this weekend
  - Class will be open for auditing
- Late Days
  - Two 24-Hour Late Days
- Slack and Canvas - coming week 2
  - Email will be the main way of communication for this week
  - Ask in #general if the question isn’t personal
  - Create a staff group if you want to ask us privately

Contact Info

- Prof. James Landay, landay@stanford.edu
- Abdallah AbuHashem, aabuhash@stanford.edu
- Claire Rosenfeld, clairero@stanford.edu
- Ryan Chen, rjc45@stanford.edu

- You can also email us at reactnative@cs.stanford.edu

Sharing

- Class website: https://cs47.stanford.edu
- Assignments and lectures will also be shared through Slack and Canvas.

Syllabus

- **Part 1: React Native Basics**
  - Weeks 1-4
- **Part 2: Navigation**
  - Weeks 5-6
- **Part 3: From Prototype to App**
  - Weeks 7-10

- For more details, check cs47.stanford.edu

What is cross-platform mobile development?

What is cross-platform mobile development?

Approaches

1. Web apps
   - Pros: you just need a browser; pushing updates.
   - Cons: slow performance; limited capabilities.
   - Example: Progressive Web Apps by Google.

Approaches

1. Web apps
   - Pros: you just need a browser; pushing updates.
   - Cons: slow performance; limited capabilities.
   - Example: Progressive Web Apps by Google.
2. Cross-platform native apps
   - Pros: user does not notice any difference; capabilities similar to non-cross-platform apps.
   - Cons: performance better than web apps but worse than native apps; feature adoption is slower than native apps
   - Example: React Native by Facebook and Flutter by Google.

React Native

React Native

- Considered the next generation of ReactJS.

React Native

- Considered the next generation of ReactJS.
- A JavaScript code library developed by Facebook and Instagram. Released on Github in 2013.

React Native

- Considered the next generation of ReactJS.
- A JavaScript code library developed by Facebook and Instagram. Released on Github in 2013.
- Main idea: Engineers won't have to build the same app for iOS and for Android from scratch - reusing the code across each operating system.

React Native

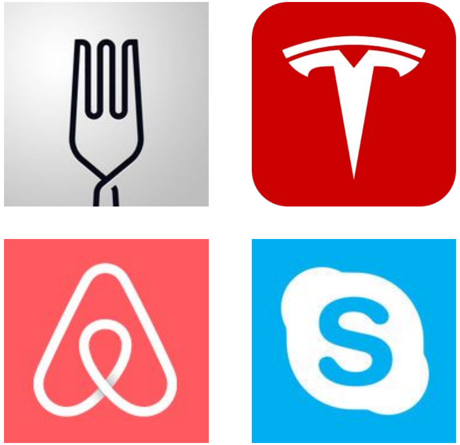
- Considered the next generation of ReactJS.
- A JavaScript code library developed by Facebook and Instagram. Released on Github in 2013.
- Main idea: Engineers won't have to build the same app for iOS and for Android from scratch - reusing the code across each operating system.
- Pros: The community; Cross-Platform teams; integration of React Native and native elements

React Native

- Considered the next generation of ReactJS.
- A JavaScript code library developed by Facebook and Instagram. Released on Github in 2013.
- Main idea: Engineers won't have to build the same app for iOS and for Android from scratch - reusing the code across each operating system.
- Pros: The community; Cross-Platform teams; integration of React Native and native elements
- Cons: It's still improving - rapidly changing/developing

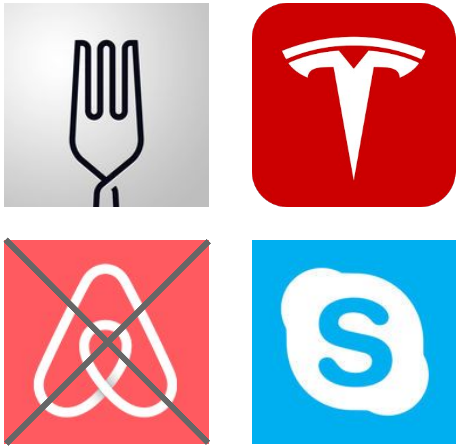
React Native: Why that much faith?

- Cross-platform saves the companies a lot of resources.
- The only threat is Facebook cutting off the project.
- But lots of companies and Facebook themselves heavily rely on it.

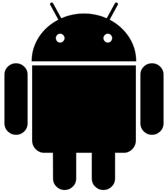


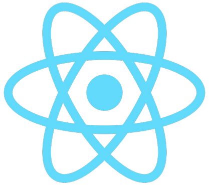
React Native: Why that much faith?

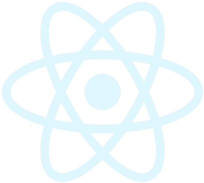
- Cross-platform saves the companies a lot of resources.
- The only threat is Facebook cutting off the project.
- But lots of companies and Facebook themselves heavily rely on it.

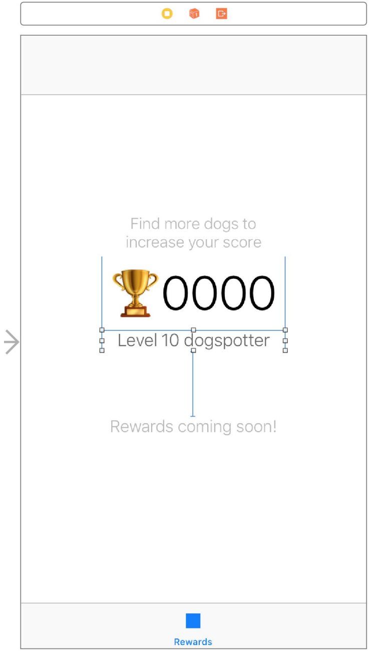


USEFUL LINKS / [AirBnB Sunsetting React Native Article]







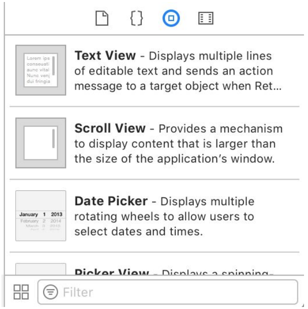


Find more dogs to increase your score

🏆0000

Level 10 dogspotter

Rewards coming soon!



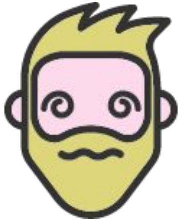
Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

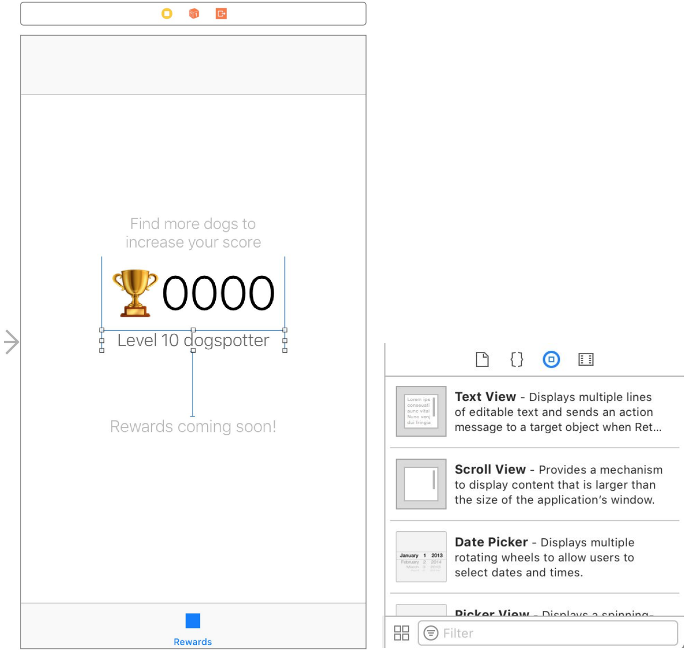
Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning-

Unfortunately you will have to deal with autolayout constraints





Find more dogs to increase your score

🏆0000

Level 10 dogspotter

Rewards coming soon!

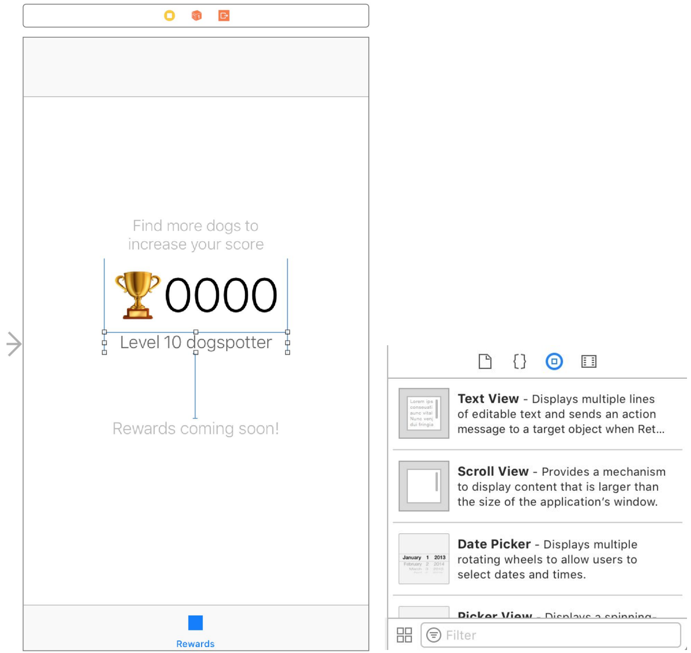
Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning...

```swift
let textLayer = CATextLayer()
textLayer.backgroundColor = color.cgColor
textLayer.foregroundColor = UIColor.white.cgColor
textLayer.frame = frame
textLayer.alignmentMode = kCAAlignmentLeft
textLayer.isWrapped = true
let font = CTFontCreateWithName("System" as CFString, 18.0, nil)
textLayer.font = font
textLayer.fontSize = 18.0
textLayer.contentsScale = UIScreen.main.scale
textLayer.string = label

self.view.layer.addSublayer(textLayer)
```



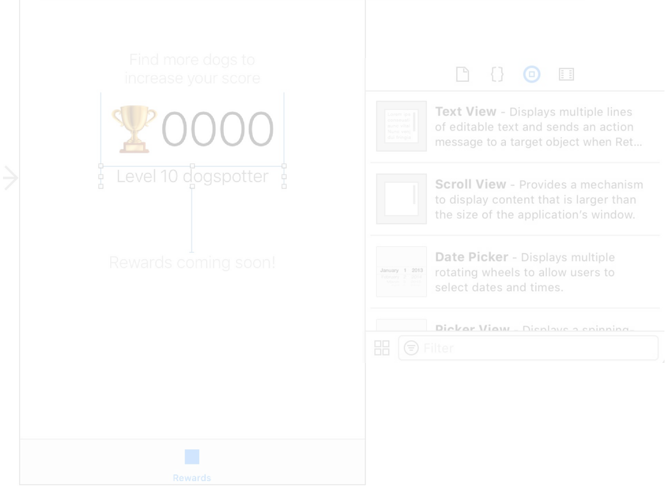
Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning...

```swift
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```


Find more dogs to increase your score

🏆0000

Level 10 dogspotter

Rewards coming soon!

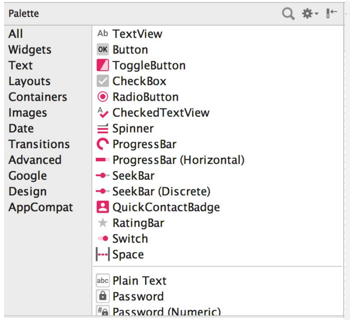
Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

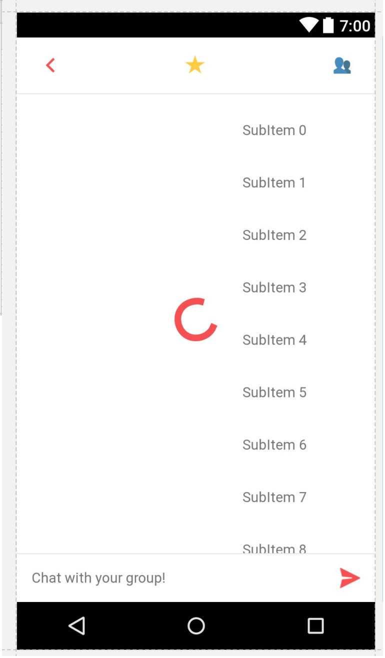
Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

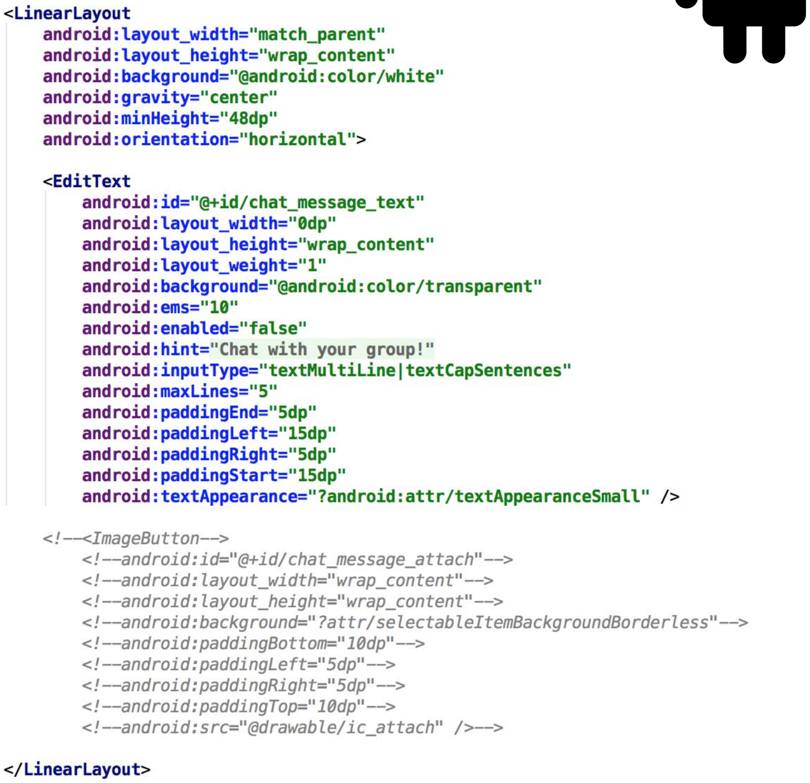
Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning...

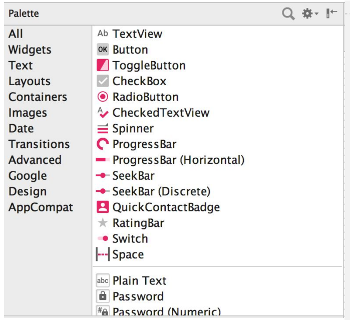
```
LinearLayout myLayout = findViewById(R.id.main);

Button myButton = new Button(this);
myButton.setLayoutParams(new LinearLayout.LayoutParams(
LinearLayout.LayoutParams.MATCH_PARENT,
LinearLayout.LayoutParams.MATCH_PARENT));

myLayout.addView(myButton);
```

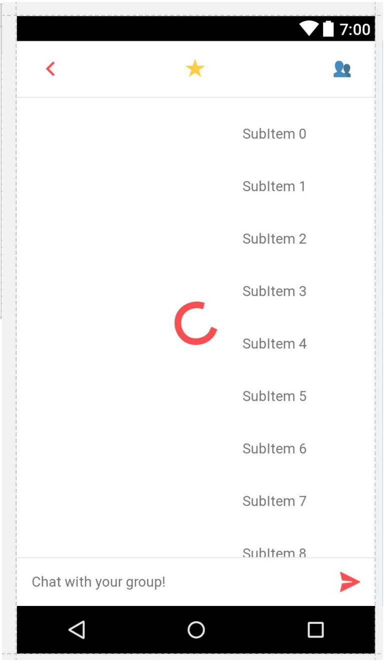




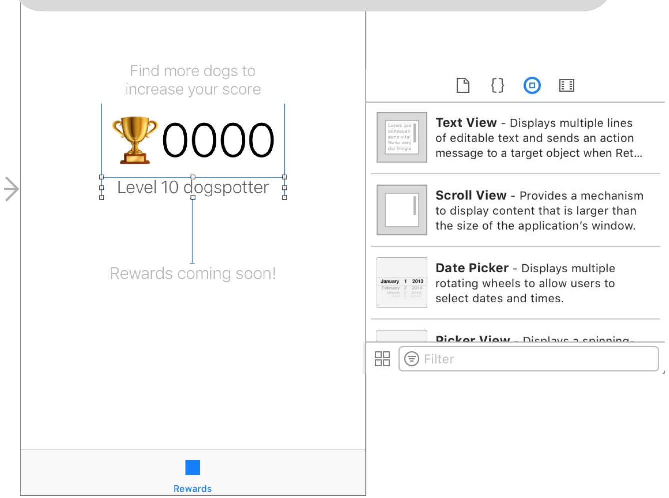


```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:color/white"
    android:gravity="center"
    android:minHeight="48dp"
    android:orientation="horizontal">

    <EditText
        android:id="@+id/chat_message_text"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:background="@android:color/transparent"
        android:ems="10"
        android:enabled="false"
        android:hint="Chat with your group!"
        android:inputType="textMultiLine|textCapSentences"
        android:maxLines="5"
        android:paddingEnd="5dp"
        android:paddingLeft="15dp"
        android:paddingRight="5dp"
        android:paddingStart="15dp"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <!--<ImageButton-->
        <!--android:id="@+id/chat_message_attach"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:background="?attr/selectableItemBackgroundBorderless"-->
        <!--android:paddingBottom="10dp"-->
        <!--android:paddingLeft="5dp"-->
        <!--android:paddingRight="5dp"-->
        <!--android:paddingTop="10dp"-->
        <!--android:src="@drawable/ic_attach" />-->

</LinearLayout>
```





```
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```



Find more dogs to increase your score

🏆0000

Level 10 dogspotter

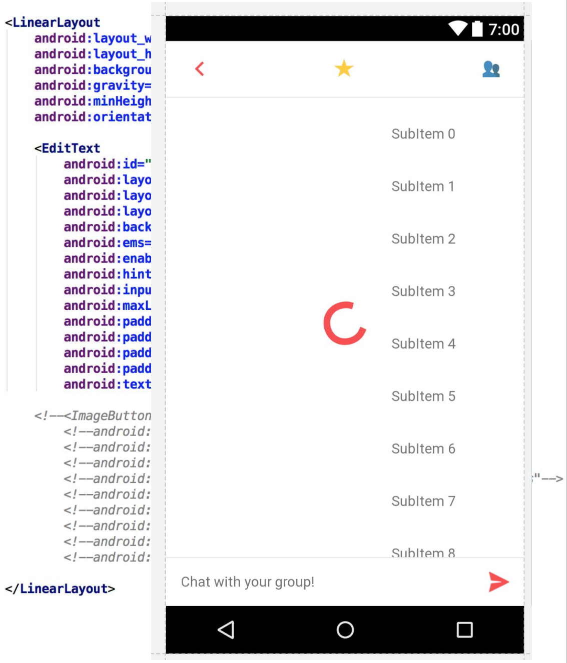
Rewards coming soon!

Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application’s window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning-



```
<LinearLayout
    android:layout_w
    android:layout_h
    android:backgrou
    android:gravity=
    android:minHeigh
    android:orientat

    <EditText
        android:id="
        android:layo
        android:layo
        android:layo
        android:back
        android:ems=
        android:enab
        android:hint
        android:inpu
        android:maxL
        android:padd
        android:padd
        android:padd
        android:padd
        android:text

    <!--<ImageButton
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:            "-->

</LinearLayout>
```

SubItem 0

SubItem 1

SubItem 2

SubItem 3

SubItem 4

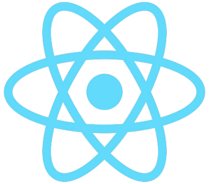
SubItem 5

SubItem 6

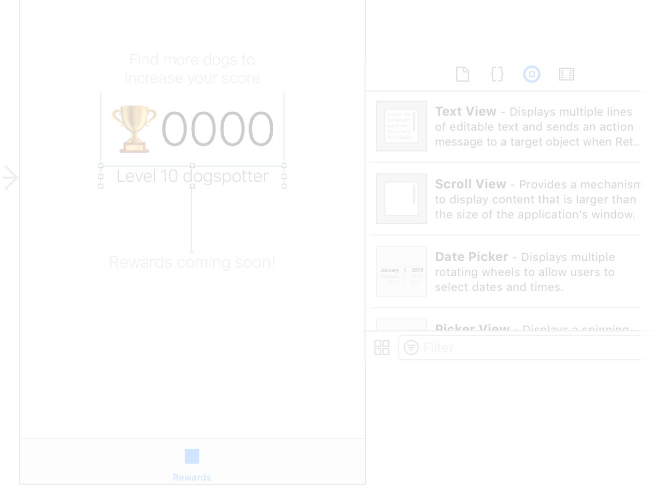
SubItem 7

SubItem 8

Chat with your group!



```
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```

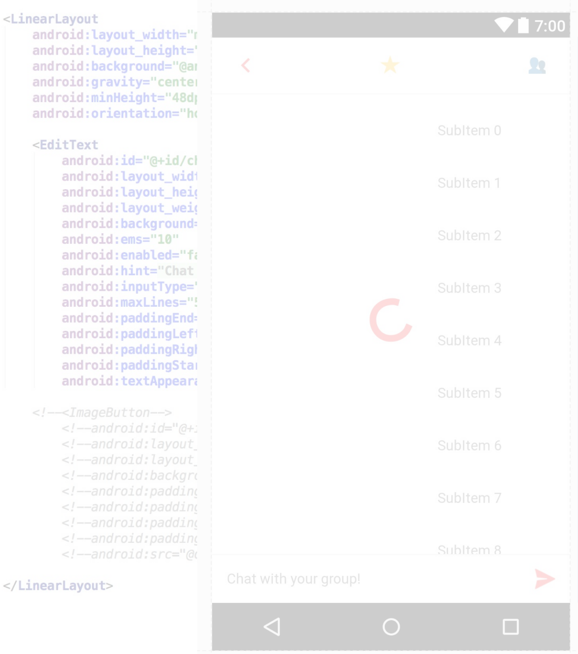


Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret..

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

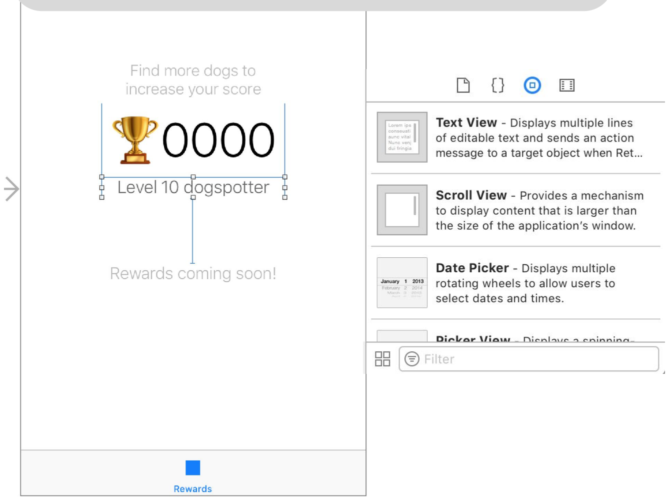
Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning...



```
<LinearLayout
    android:layout_width="m
    android:layout_height="
    android:background="@an
    android:gravity="center
    android:minHeight="48dp
    android:orientation="ho

    <EditText
        android:id="@+id/ch
        android:layout_widt
        android:layout_heig
        android:layout_weig
        android:background=
        android:ems="10"
        android:enabled="fa
        android:hint="Chat
        android:inputType="
        android:maxLines="5
        android:paddingEnd=
        android:paddingLeft
        android:paddingRigh
        android:paddingStar
        android:textAppeara

    <!--<ImageButton-->
        <!--android:id="@+i
        <!--android:layout_
        <!--android:layout_
        <!--android:backgro
        <!--android:padding
        <!--android:padding
        <!--android:padding
        <!--android:padding
        <!--android:src="@d

</LinearLayout>
```

```
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```

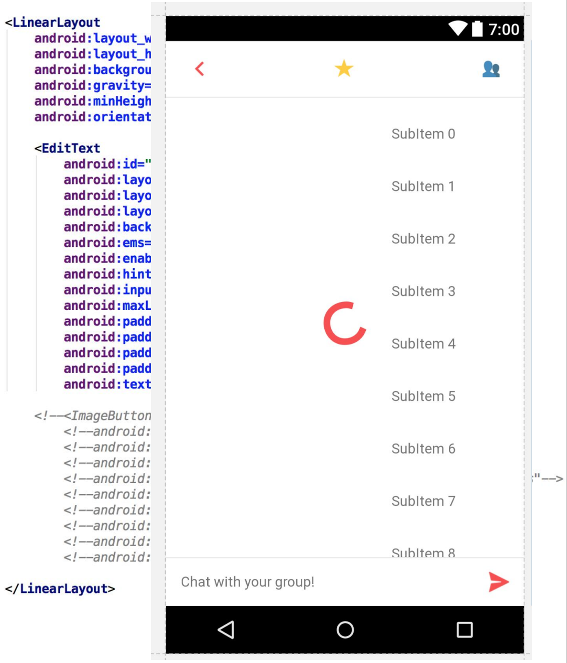


Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning-



```
<LinearLayout
    android:layout_w
    android:layout_h
    android:backgrou
    android:gravity=
    android:minHeigh
    android:orientat

    <EditText
        android:id="
        android:layo
        android:layo
        android:layo
        android:back
        android:ems=
        android:enab
        android:hint
        android:inpu
        android:maxL
        android:padd
        android:padd
        android:padd
        android:padd
        android:text

    <!--<ImageButton
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:

</LinearLayout>
```

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

Quick JavaScript/JSX Detour

JS: Variables

In ES6, variables:

- Don’t have types
- But must be declared before using them

- Global variables

```
var x = 1;

if (x === 1) {
  var x = 2;
  console.log(x);
  // expected output: 2
}

console.log(x);
// expected output: 2
```

JS: Variables

In ES6, variables:

- Don’t have types
- But must be declared before using them

- Local variables

```
let x = 1;

if (x === 1) {
  let x = 2;
  console.log(x);
  // expected output: 2
}

console.log(x);
// expected output: 1
```

JS: Variables

In ES6, variables:

- Don’t have types
- But must be declared before using them

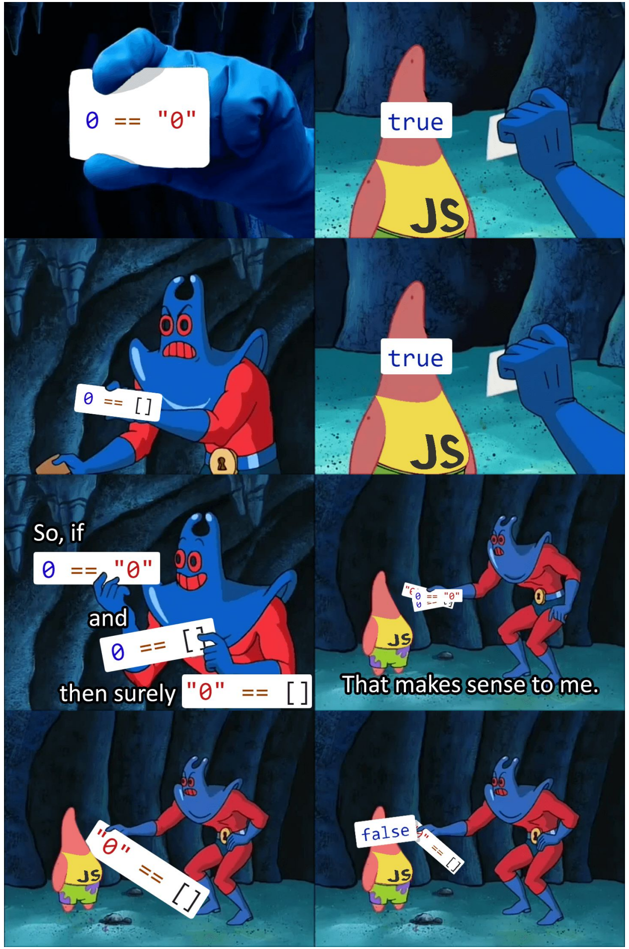
- Const variables

```
const x = 1;

if (x === 1) {
  x = 2;
  //Error
  console.log(x);
  // expected output: 2
}

console.log(x);
// expected output: 1
```

JS: Variables

	Global Variables	Local Variables	Constant Variables
Use	The scope here is the function in which it’s declared	The scope here is the block in which it is declared	The scope here is the block, but it cannot be changed in value
Syntax	`var x = 10;`	`let x = 10;`	`const x = 10;`

JS: If statements



ES6

JS: If statements

JS needs special care with equality.

If in doubt use `===` and `!==` (or use them always)
As opposed to `==` and `!=`

- Example

```
if (a > 0) {
    return "positive";
} else if (a === 0) {
    return "It’s a zero";
} else {
    return "NOT positive";
}
```

JS: Loops

You have many options for loops in JS

- For loops

```
for (var i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
```

- For of loops

```
for (var element of arr) {
  console.log(arr);
}
```

ES6

JS: Loops

You have many options for loops in JS

- For each loops

```
arr.foreach(function(element) {
  console.log(element);
});
```

- While loops

```
While (true) {
  console.log(“You can’t stop me”);
}
```

JS: Functions

- Functions in JS are declared in the following way

```
function addition(a, b = 10) {
  return a + b;
}
```

JS: Functions

- Functions in JS are declared in the following way

```
function addition(a, b = 10) {
  return a + b;
}
```

- Another way is as follows

```
var addition = (a, b = 10) => {
  return a + b;
}
```

JS: Functions

- Functions in JS can turn passed in arguments to arrays

```
function addition(a, ...b) {
  return a + b.length;
}
console.log(addition(2,1,7,5));
// 2 + 3 = 5
```

- On the opposite side, we can do

```
function addition(a, b, c) {
  return a + b + c;
}
console.log(addition(...[1,2,3]));
// 1 + 2 + 3 = 6
```

JS: Objects

- Objects are similar to dictionaries and/or maps in other languages

```
let obj = {
    name: ‘John’,
    age: 17,
};
console.log(obj.name + ‘ ‘ + obj[‘age’]);
```

- Other ways of representing properties

```
let obj = {
    [‘full’ + ‘name’]: ‘John Doe’,
    //same as age: age
    age,
    lorem() {
        return “ipsum”;
    },
};
```

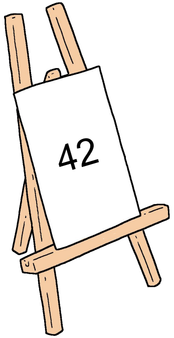
ES6

JSX

An extension to JavaScript that you will use to build your UI interfaces.

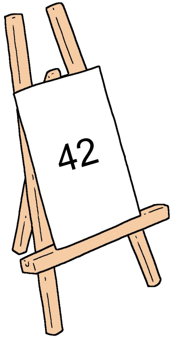
Without JSX

```
var ourNestedView = React.createElement(
  View,
  {
    foo: 'bar'},
  React.createElement(
    Text,
    null,
    '42'
  )
);
```



With JSX

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```



JSX

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

No JSX

```
var ourNestedView = React.createElement(
  View,
  {
    foo: 'bar'},
  React.createElement(
    Text,
    null,
    '42'
  )
);
```

JSX is a shortcut for using the React.createElement() API

JSX Benefits

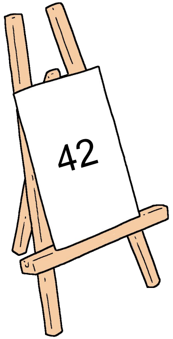
```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

- UI has a clear hierarchical structure. What you see in code mirrors what you will get.

JSX Benefits

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

- UI has a clear hierarchical structure. What you see in code mirrors what you will get.



JSX Benefits

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

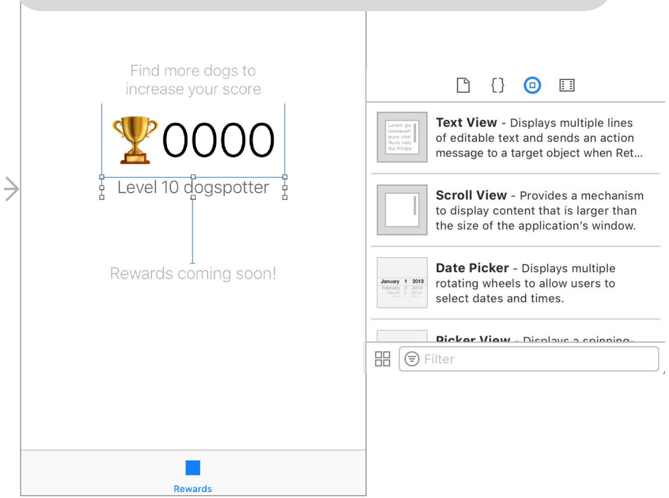
- UI has a clear hierarchical structure. What you see in code mirrors what you will get.
- This makes it easier for designers to contribute to code.

JSX Benefits

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

- UI has a clear hierarchical structure. What you see in code mirrors what you will get.
- This makes it easier for designers to contribute to code.
- You get the accessibility of templates AND the power of JS.

```swift
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```



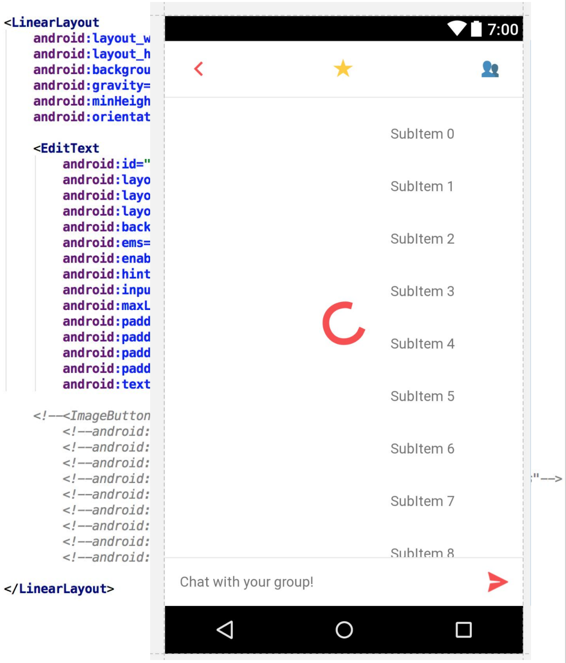
Find more dogs to increase your score

Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

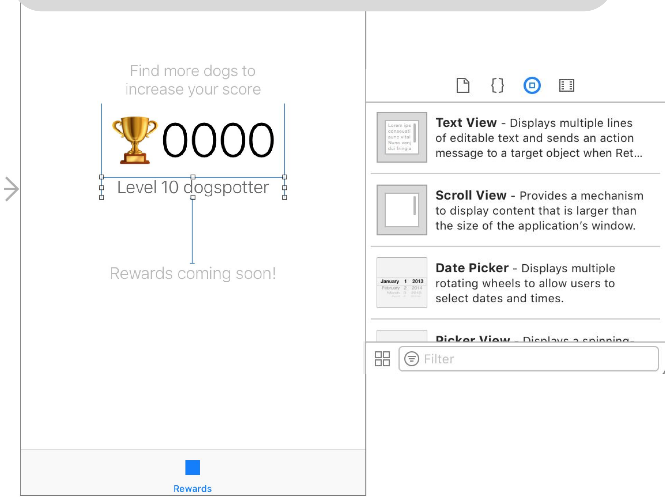
Picker View - Displays a spinning-



```xml
<LinearLayout
    android:layout_w
    android:layout_h
    android:backgrou
    android:gravity=
    android:minHeigh
    android:orientat

    <EditText
        android:id="
        android:layo
        android:layo
        android:layo
        android:back
        android:ems=
        android:enab
        android:hint
        android:inpu
        android:maxL
        android:padd
        android:padd
        android:padd
        android:padd
        android:text

    <!--<ImageButton
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:

</LinearLayout>
```

```jsx
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

```
let textLayer = CATextLayer()
...

self.view.layer.addSublayer(textLayer)
```



Find more dogs to increase your score

🏆0000

Level 10 dogspotter

Rewards coming soon!

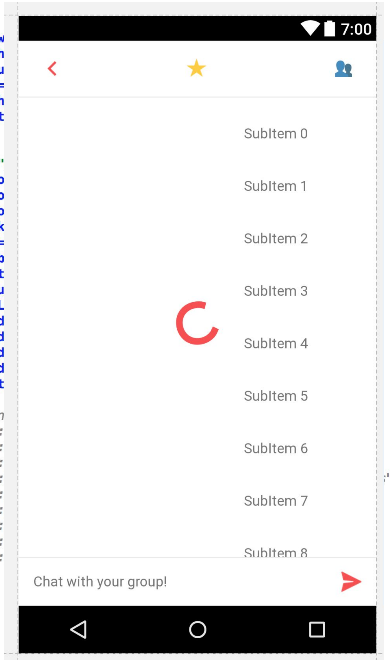
Text View - Displays multiple lines of editable text and sends an action message to a target object when Ret...

Scroll View - Provides a mechanism to display content that is larger than the size of the application's window.

Date Picker - Displays multiple rotating wheels to allow users to select dates and times.

Picker View - Displays a spinning-

```
<LinearLayout
    android:layout_w
    android:layout_h
    android:backgrou
    android:gravity=
    android:minHeigh
    android:orientat

    <EditText
        android:id="
        android:layo
        android:layo
        android:layo
        android:back
        android:ems=
        android:enab
        android:hint
        android:inpu
        android:maxL
        android:padd
        android:padd
        android:padd
        android:padd
        android:text

    <!--<ImageButton
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
        <!--android:
                                                          "-->

</LinearLayout>
```



SubItem 0
SubItem 1
SubItem 2
SubItem 3
SubItem 4
SubItem 5
SubItem 6
SubItem 7
SubItem 8

Chat with your group!

```
const ourNestedView = (
  <View
    foo='bar'>
    <Text>42</Text>
  </View>
)
```

Hot Reloading

Summary - Why React Native

- Cross-platform mobile apps

Summary - Why React Native

- Cross-platform mobile apps
- Companies depend on React Native already

Summary - Why React Native

- Cross-platform mobile apps
- Companies depend on React Native already
- Approachable for beginners and those with web development experience

Summary - Why React Native

- Cross-platform mobile apps
- Companies depend on React Native already
- Approachable for beginners and those with web development experience
- Power of JavaScript

Looking Ahead to Thursday

- Continued JavaScript basics
- Understanding JavaScript, JSX, React, React Native
- Build and test your first React Native app!

Assignment 1

- Released today.
- Due next Tuesday 11:59PM.
- Submission: Send us a screenshot of the running simulator, or show up on Monday OH with your bugs.

Exit Ticket

- Fill out this google form: What are you excited to learn in this class?

- Fill out the application form for this class if you haven’t filled it out already
- Also linked on cs47.stanford.edu

Office Hours

OH start on week 2. Email us directly if you’re not available at these times

Abdallah AbuHashem
TBD
By appointment

Claire Rosenfeld
TBD
By appointment

Ryan Chen
TBD
By appointment

CS47: Cross-Platform Mobile Development

Lecture 1A: Introductions and Syllabus

James Landay
Abdallah AbuHashem
Claire Rosenfeld
Ryan Chen

https://cs47.stanford.edu

Slack coming soon!

Winter 2020