



# STUDY JAMS



#  
STUDYJAMS



# AGENDA

---

**10 MARÇO**

START!

CONFIGURANDO AMBIENTE  
HELLO ANDROID!

**24 MARÇO**

LIÇÃO 02

FAZENDO UM APP INTERATIVO  
BUTTONS, VARIÁVEIS, DEBUG!

**07 ABRIL**

PROJETO!

FAZENDO UM APP INTERATIVO  
BUTTONS, VARIÁVEIS, DEBUG!

**14 ABRIL**

APRESENTAÇÕES  
DOS PROJETOS  
ENCERRAMENTO



LIÇÃO 01

CONSTRUINDO LAYOUT  
VIEWS, XML SYNTAX

**17 MARÇO**

LIÇÃO 03

ORIENTAÇÃO A OBJETOS  
INTENTS, LOCALIZATION, STYLES

**31 MARÇO**

# IN THE LAST EPISODE

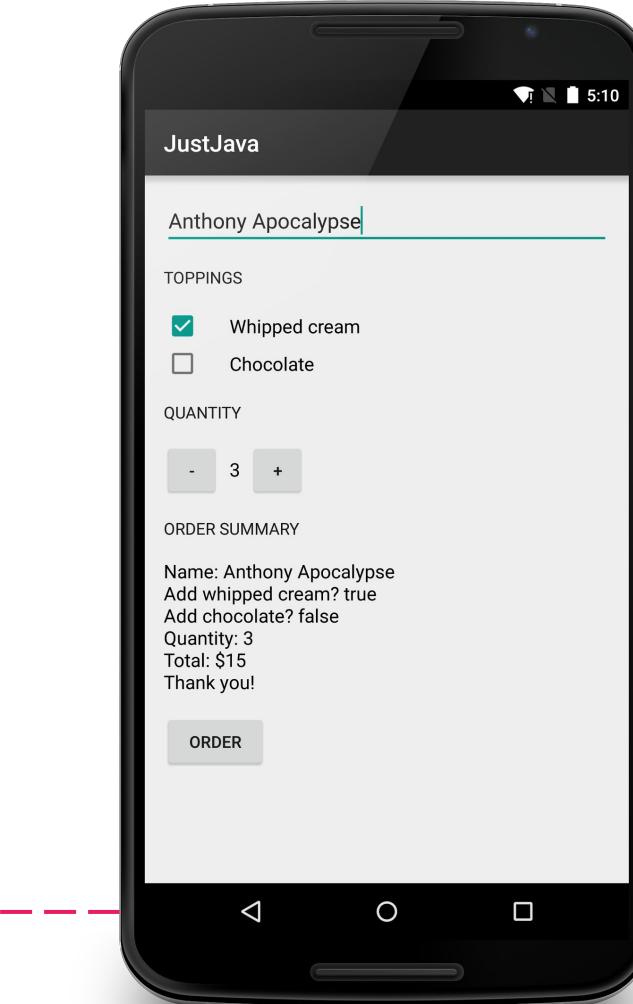
*LESSON 2 - REMEMBER*

## Lesson 2 - Making an App Interactive

---

- Button Click
- Methods
- Math Expressions
- Variables
- Debugging
- Nesting ViewGroups
- Variables Types
- String Concatenation

# QUEM FEZ?



# OBJECT ORIENTED PROGRAMMING

*LESSON 3A*

# Lesson 3A

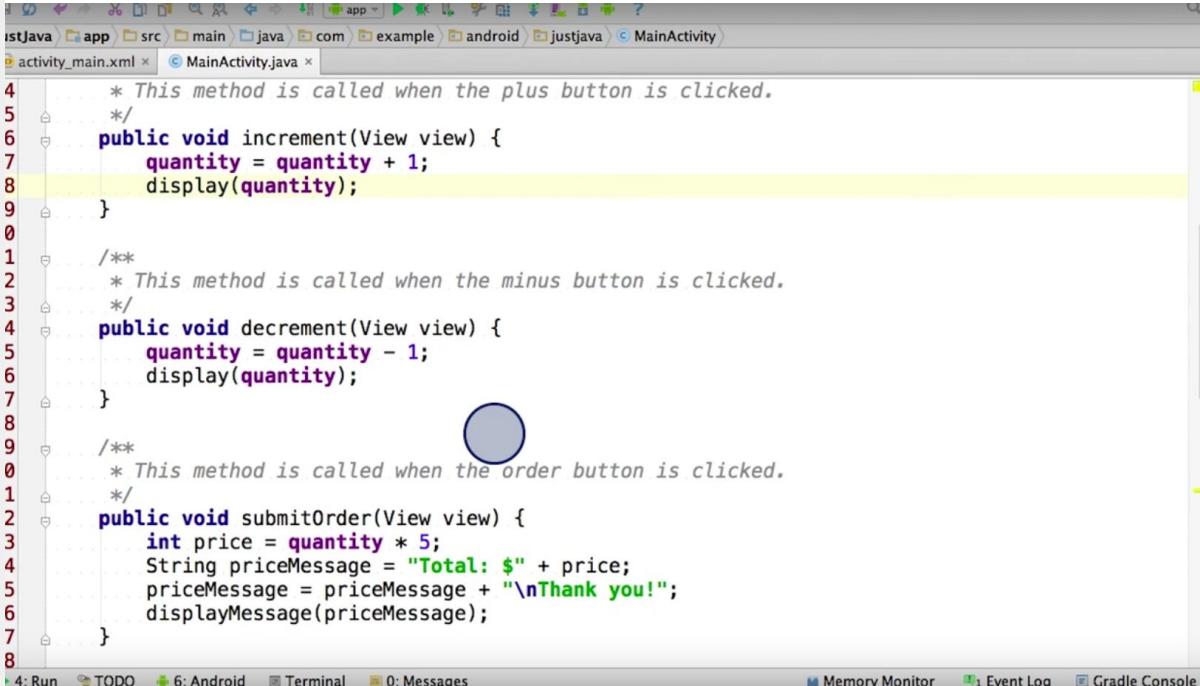
## Defining vs Calling a Method

Como definir e realizar uma chamada de um método

---

## Notes

- Go To > Declaration  
(Command/Control + B)



```
stsJava app src main java com.example.android.justjava c MainActivity
activity_main.xml x MainActivity.java x
4     * This method is called when the plus button is clicked.
5     */
6     public void increment(View view) {
7         quantity = quantity + 1;
8         display(quantity);
9     }
0
1     /**
2      * This method is called when the minus button is clicked.
3      */
4     public void decrement(View view) {
5         quantity = quantity - 1;
6         display(quantity);
7     }
8
9     /**
0      * This method is called when the order button is clicked.
1      */
2     public void submitOrder(View view) {
3         int price = quantity * 5;
4         String priceMessage = "Total: $" + price;
5         priceMessage = priceMessage + "\nThank you!";
6         displayMessage(priceMessage);
7     }
8
```

The screenshot shows the Android Studio interface with the code editor open to MainActivity.java. The file contains Java code defining three methods: increment, decrement, and submitOrder. The increment and decrement methods both call the display method with the quantity parameter. The submitOrder method calculates a total price and then calls display with the priceMessage parameter. A blue circle highlights the line 'display(quantity);' in the increment method. The code editor has a light gray background with syntax highlighting for keywords and comments.

# Lesson 3A

## Inputs & Outputs

Identificando os parâmetros de entrada e saída de métodos.

### INPUTS & OUTPUT OF A METHOD

submit Order(...);



calculate Price(...);

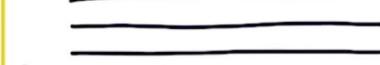


}

zero or more  
input parameters  
quantity

zero or one  
return value  
total price

calculate Price(...);



}

## Lesson 3A

# Define a Method

Descreve o comportamento e criação de um método.

---

Access  
modifier

# DEFINE A METHOD

Who has  
access  
to this  
method?



or

private



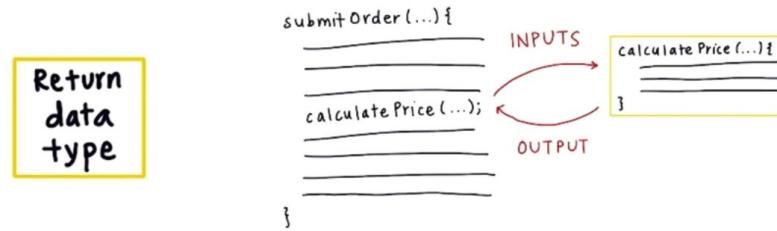
```
private int calculatePrice(int quantity) {
```

# Lesson 3A

## Define a Method

Descreve o comportamento e criação de um método.

# DEFINE A METHOD



```
private int calculatePrice(int quantity) {
```

## Lesson 3A

# Define a Method

Descreve o comportamento e criação de um método.

---

# DEFINE A METHOD

Method  
name



```
private int calculatePrice(int quantity)
```



# Lesson 3A

## Define a Method

Descreve o comportamento e criação de um método.

— — —

INPUTS

Method name  
↓  
`private String createCustomGreeting(String firstName, String lastName) {`  
Input Parameter 1 data type  
↓ variable name  
 `return "Welcome, " + firstName + " " + lastName + "!";`  
Input Parameter 2 data type  
↓ variable name



# Lesson 3A

## Define a Method

Descreve o comportamento e criação de um método.

# DEFINE A METHOD



```
private int calculatePrice(int quantity, int priceOfOneCup)
```



## Lesson 3A

# Use Return Value

Descreve a chamada de um método e a utilização do valor retornado por este.

---

## USE RETURN VALUE

### Pseudocode

In `submitOrder` method:

1. Call `calculatePrice` method and store output in integer variable called `price`.
2. Display `price` on screen.

### Java Code

In `submitOrder` method:

```
int price = 10 + calculatePrice();
```

```
displayPrice(price);
```

# Lesson 3A

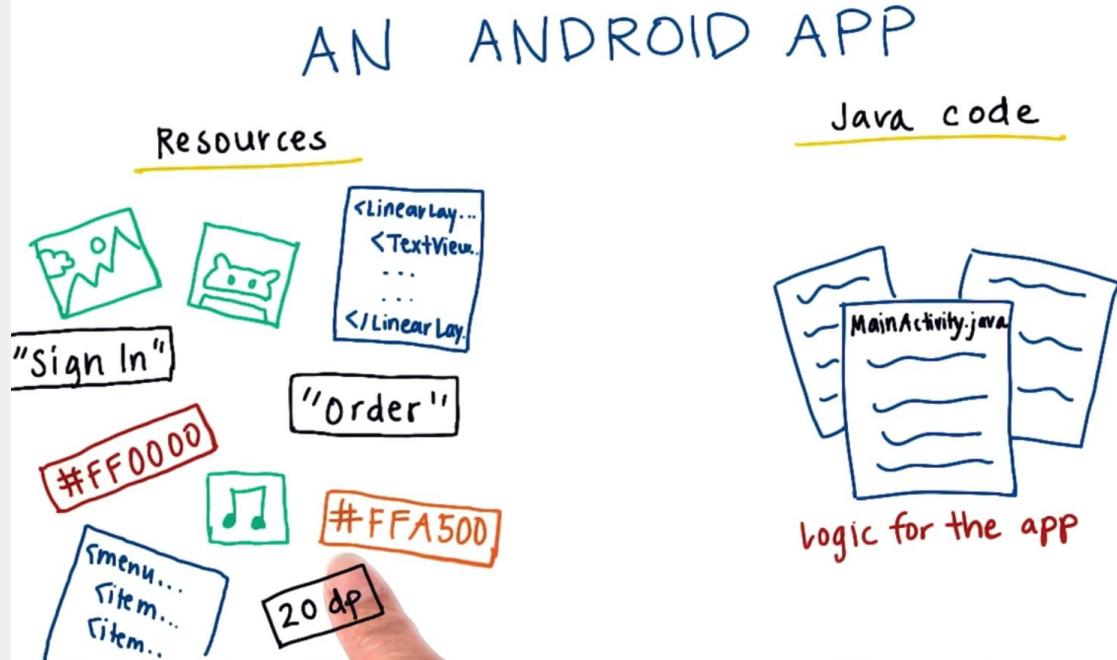
## Resources

Introduz o conceito dos resources e a sua utilização nas Activities.

### Vocabulário

- drawable
- layout
- menu
- strings
- styles
- colors

...



## Lesson 3A

# Resource IDs

Introduz o conceito dos resources e a sua utilização nas Activities.

## Vocabulário

- drawable
- layout
- menu
- strings
- styles
- colors

...

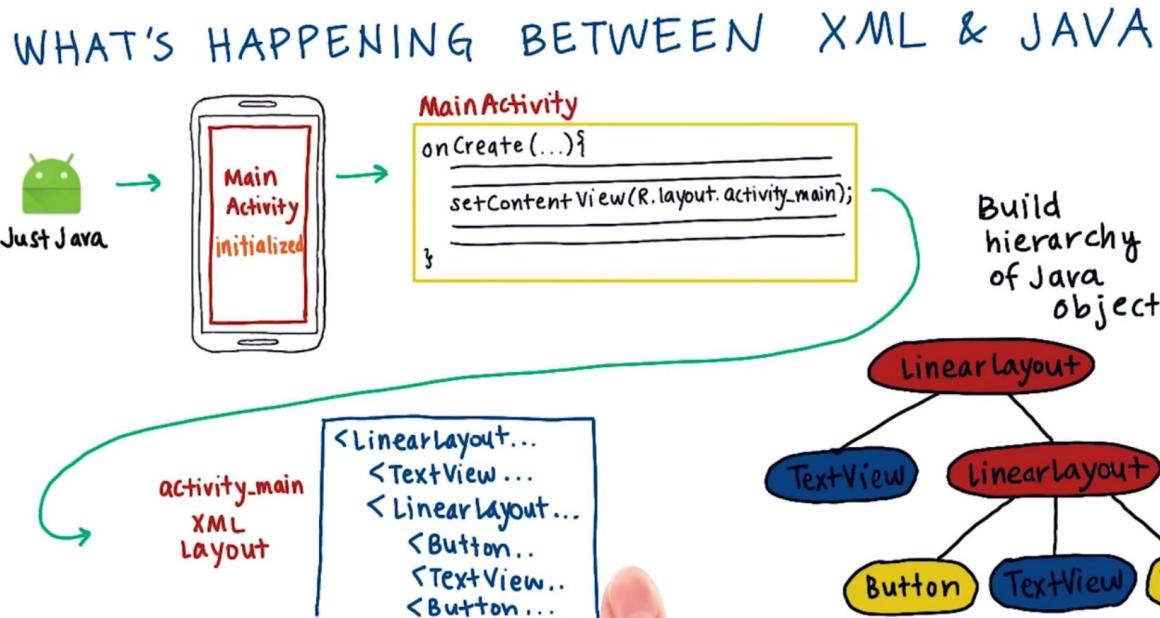
## ACCESS RESOURCES

Resource Type	In Java code	In XML
Image		R.drawable.photo
String	"Hello"	R.string.hello
Layout XML file		@layout/activity_main
ID		R.id.price_text_view
Color	#FF0000	R.color.red

# Lesson 3A

## From XML to Java

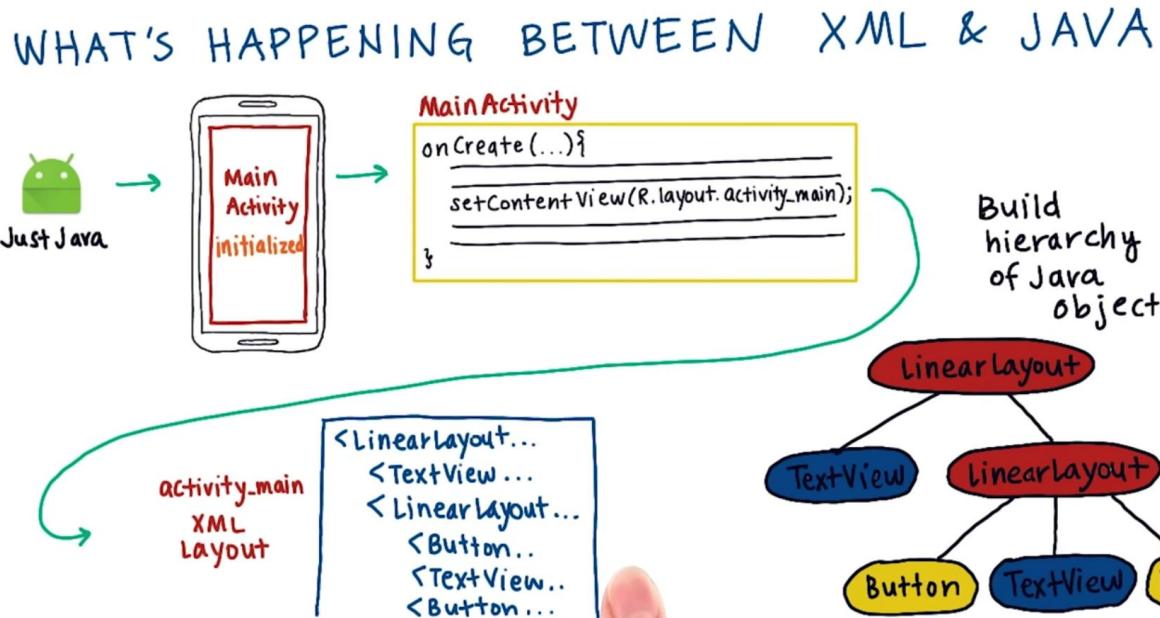
Como funciona um aplicativo  
Android? Do Java ao XML.



# Lesson 3A

## From XML to Java

Como funciona um aplicativo  
Android? Do Java ao XML.

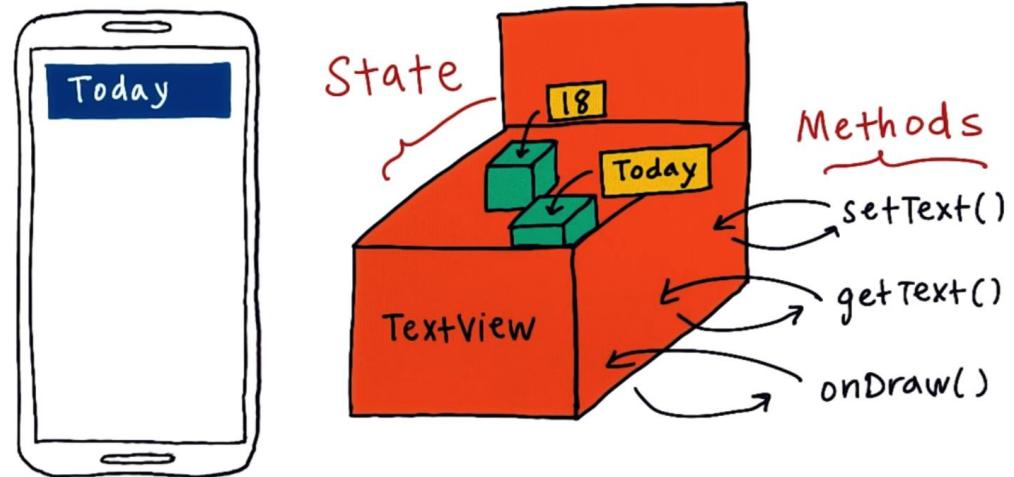


# Lesson 3A

## Java Object

Descreve um objeto Java e sua implementação.

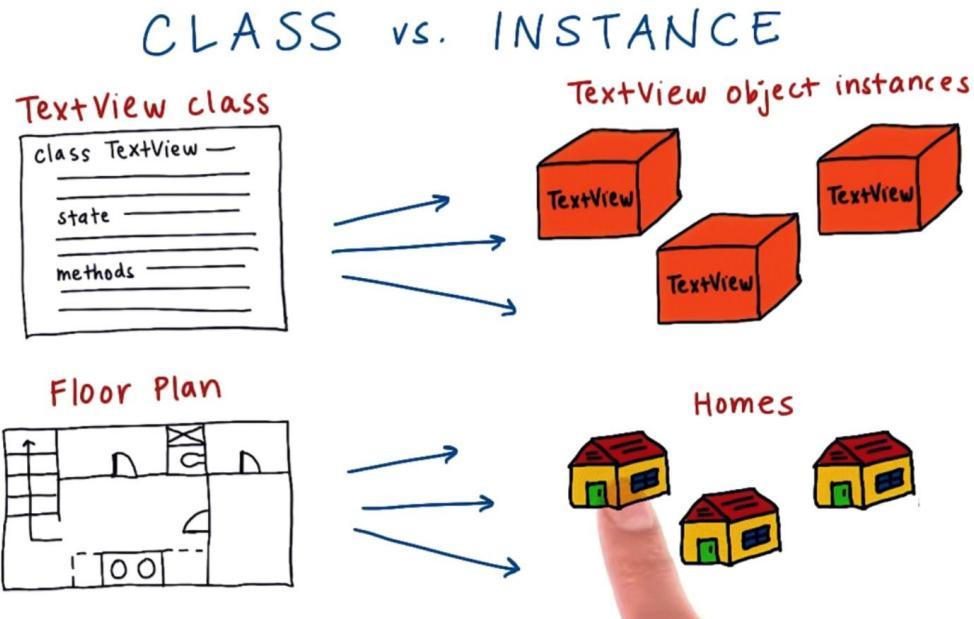
OBJECT HAS STATE & METHODS



# Lesson 3A

## Class vs. Instance

O que é uma classe? O que é uma instância de uma classe?



# Lesson 3A

## Java Class

Definindo uma classe Java e sua estrutura.

---

```
4  public class TextView extends View {  
5      // String value  
6      private String mText;  
7  
8      // Text color of the text  
9      private int mTextColor;  
10  
11     // Context of the app  
12     private Context mContext;   
13  
14     /**  
15      * Constructs a new TextView with initial values for text and text color.  
16      */  
17     public TextView(Context context) {  
18         mText = "";  
19         mTextColor = 0;  
20         mContext = context;  
21     }  
22  
23     /**  
24      * Sets the string value in the TextView.  
25      */  
26 }
```

## Lesson 3A

# Create an Object

Como instanciar/criar um objeto no Java?

---

CREATE OBJECT WITH CONSTRUCTOR



```
TextView priceTextView = new TextView(context);  
ImageView coffeeImageView = new ImageView(context);
```



## Lesson 3A

# Create an Object with Factory Methods

Como instanciar/criar um objeto no Java?

---

CREATE OBJECT  
WITH FACTORY METHODS



```
MediaPlayer player = MediaPlayer.create(context, R.raw.song);
```



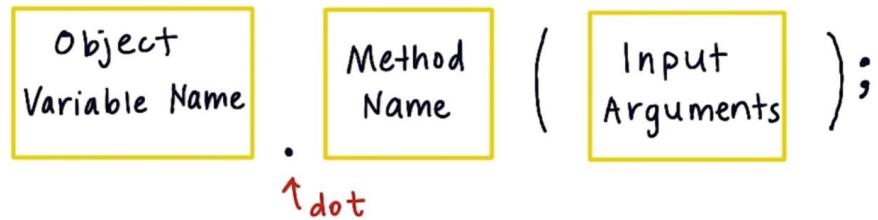
## Lesson 3A

# Call Object Methods

Descreve como realizar uma chamada de um método de um objeto.

---

### CALL METHOD ON OBJECT



```
titleTextView.setText("News");  
titleTextView.setTextSize(18);  
warningTextView.setTextColor(Color.RED);
```



# Lesson 3A

## Call Object Methods

Descreve como realizar uma chamada de um método de um objeto.

INSIDE A CLASS

TextView.java

```
class TextView {  
    _____  
    setText method  
    _____  
    some other method  
    _____  
    setText("Hello");  
}
```

Same as `this.setText("Hello");`

vs. OUTSIDE A CLASS

MainActivity.java

```
class MainActivity {  
    _____  
    displayMessage method  
    _____  
    textView.setText(  
        "Price: $10");  
}
```

# Lesson 3A

## Inheriting Behavior

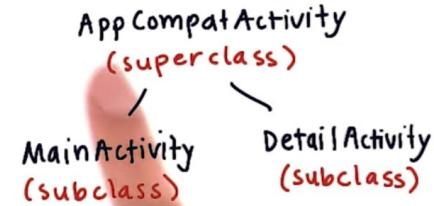
O que é uma herança e como funciona?

— — —

### CLASS INHERITANCE

```
class MainActivity extends  
    AppCompatActivity {  
  
    method1() {  
    }  
  
    method2() {  
    }  
  
    @Override  
    method from AppCompatActivity {  
    }  
}
```

### Class Hierarchy



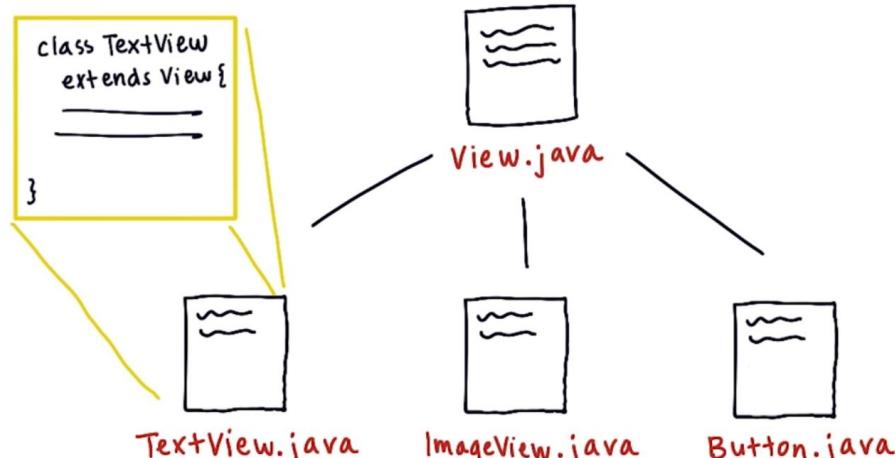
# Lesson 3A

## Inheriting Behavior

O que é uma herança e como funciona?

---

### CLASS INHERITANCE

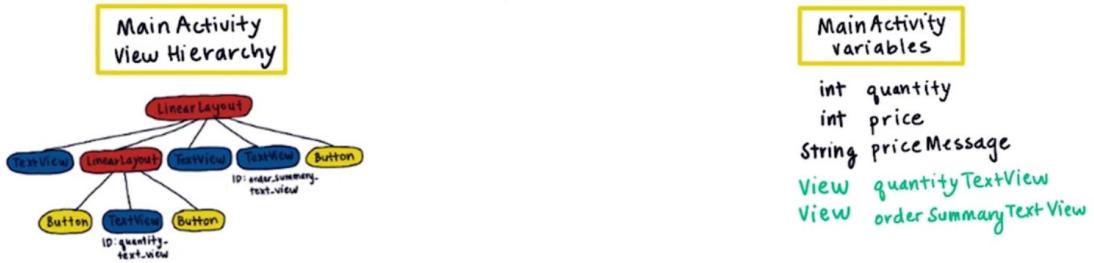


# Lesson 3A

## Find View By Id

Interagindo XML e Java, como utilizar os objetos do layout?

FIND VIEW IN VIEW HIERARCHY  
USING VIEW ID



Main Activity Java code

```
View orderSummaryTextView = findViewById(R.id.order_summary_text_view);  
  
View quantityTextView = findViewById(R.id.quantity_text_view);
```

# Lesson 3A

## Find View By Id

Interagindo XML e Java, como utilizar os objetos do layout?

FIND VIEW IN VIEW HIERARCHY  
USING VIEW ID



Main Activity Java code

```
View orderSummaryTextView = findViewById(R.id.order_summary_text_view);
```

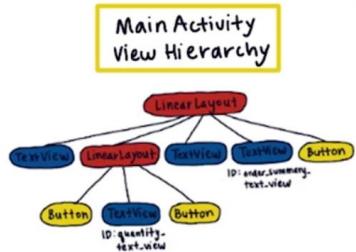
```
View quantityTextView = findViewById(R.id.quantity_text_view);
```

# Lesson 3A

## Find View By Id

Interagindo XML e Java, como utilizar os objetos do layout?

FIND VIEW IN VIEW HIERARCHY  
USING VIEW ID



MainActivity variables

```
int quantity
int price
String priceMessage
```

TextView quantityTextView  
TextView orderSummaryTextView

Main Activity Java code

```
TextView orderSummaryTextView = (TextView) findViewById(R.id.order_summary_text_view);  
  
TextView quantityTextView = (TextView) findViewById(R.id.quantity_text_view);
```

# OBJECT ORIENTED PROGRAMMING

*LESSON 3B*

## Lesson 3B CheckBox

Um novo componente!

Apresentando o CheckBox.

## Resposta

- TextView e CheckBox view;
- Inserir as novas views no topo do primeiro LinearLayout;
- Estilo do título igual aos outros e ajustar as margens do CheckBox;

CHECKBOX FOR WHIPPED CREAM

STEP 1: Select Views



STEP 2: Position Views



STEP 3: Style Views



Then add this to your app!

## Lesson 3B

# Boolean Data Type

Descreve um novo tipo de variável, o booleano.

---

## DECLARE BOOLEAN VARIABLE



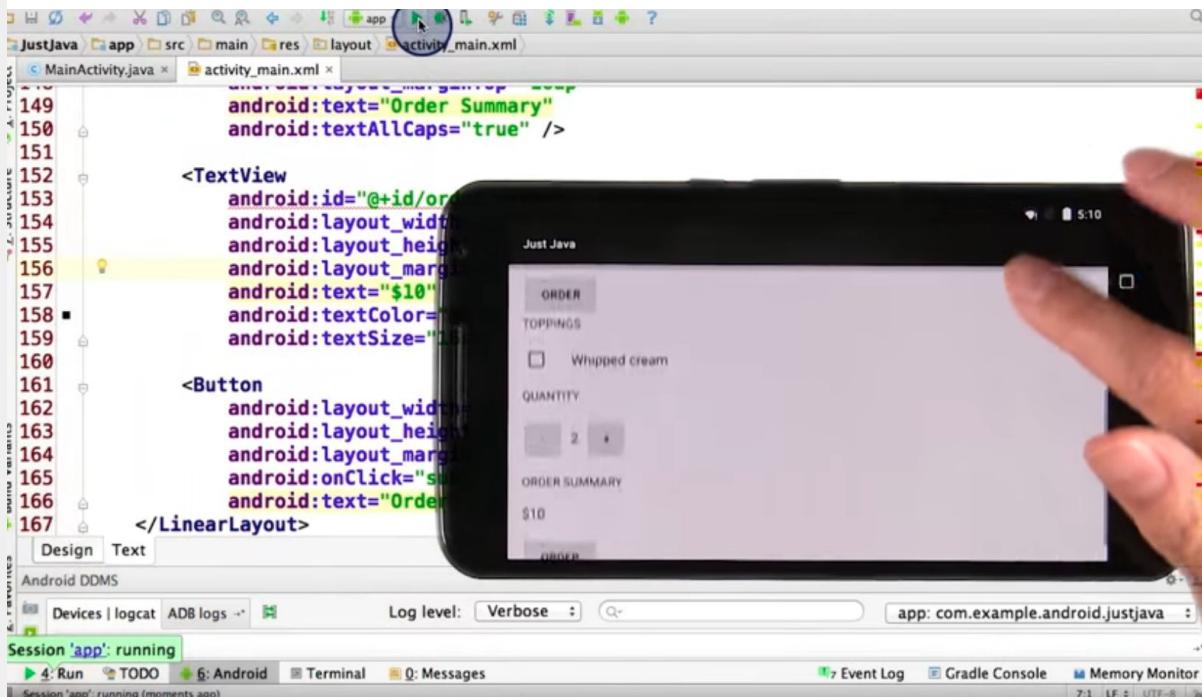
The diagram illustrates the syntax for declaring a Boolean variable. It shows three boxes: 'Data type' (containing 'boolean'), 'Variable name' (containing 'hasWhippedCream'), and 'Initial value' (containing 'true;'). These three components are connected by '=' and ';' symbols to form the full declaration statement.

Data type	Variable name	=	Initial value ;
boolean	hasWhippedCream	=	true;
boolean	hasWhippedCream	=	false;
boolean	isRegistered	=	false;
boolean	isOrderForPickup	=	true;

# Lesson 3B

## ScrollView

Mais ViewGroups. E se o layout da tela é muito grande e começa a cortar?

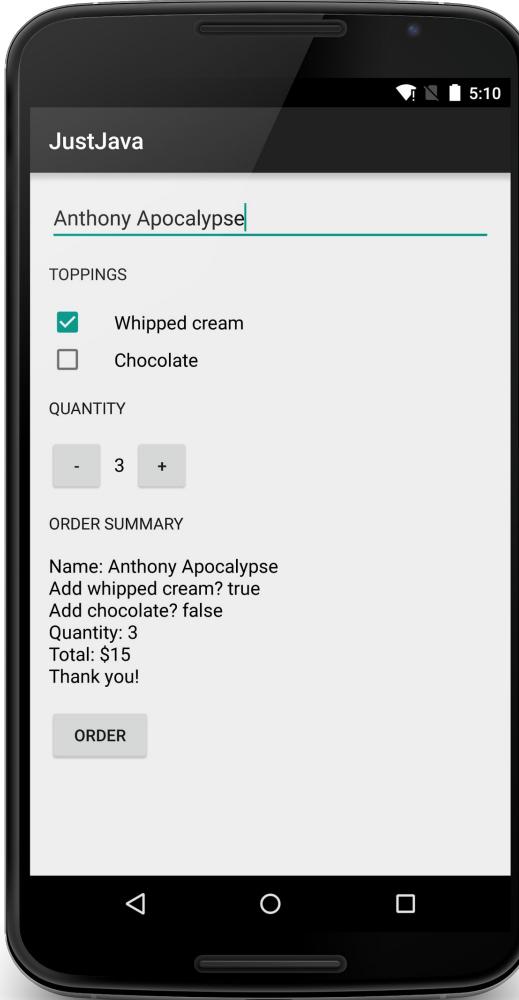


## Lesson 3B

# EditText

Adicionando o nome do cliente ao pedido? Sendo proativo! :D

---



# Lesson 3B

## Conditional Code

Introduz *if/else* e descreve como e quando condicionar um código.



The screenshot shows a GitHub repository page for "Android for Beginners : If/Else Weather Sample Quiz". The main content is a code snippet titled "Code snippet in WeatherActivity.java". The code is as follows:

```
1 boolean isRaining = true;
2 if (isRaining) {
3     Log.v("WeatherActivity", "It's raining, better bring an umbrella.");
4 } else {
5     Log.v("WeatherActivity", "It's unlikely to rain.");
6 }
7 Log.v("WeatherActivity", "Thank you for using the WhetherWeather App.");
```

A blue circle highlights the line of code: `Log.v("WeatherActivity", "It's raining, better bring an umbrella.");`

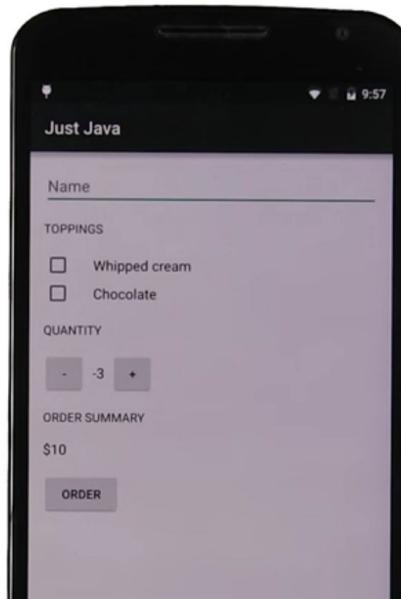
At the bottom of the code block, there is a green button with the text "Sign up for free" and a message: "to join this conversation on GitHub. Already have an account? Sign in to comment".

## Lesson 3B

# Conditional Code

Introduz *if/else* e descreve como é quando condicionar um código.

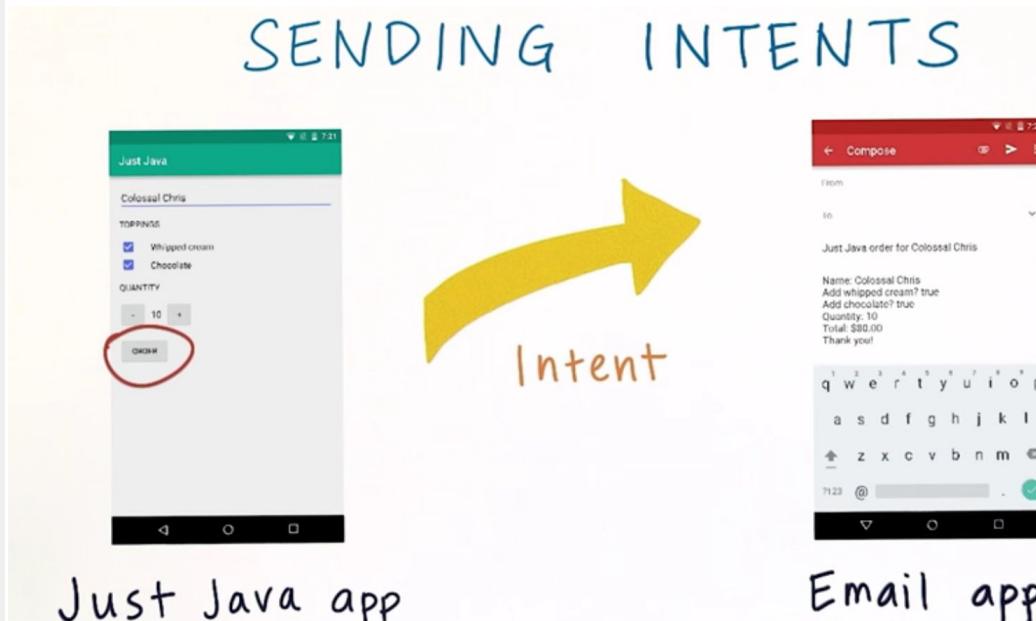
---



## Lesson 3B

# Intents

O que são intents?  
Para que servem?



## Lesson 3B

# Intents

O que são intents?  
Para que servem?

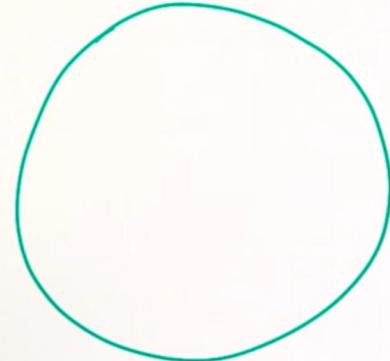
---

## Vocabulary

- Common Intents

## WHAT'S INSIDE AN INTENT?

Example dial intent



- Action ACTION\_DIAL
- Data URI tel:2125551212
- Category
- Component
- Extras

# PROVIDE RESOURCES

## Lesson 3B Localization

Aprendendo um pouco mais sobre resources. Como suportar diferentes idiomas?

### Notes

- values-[es-pt-en-fr]/

Default Resources



Alternate Resources



Different physical sizes

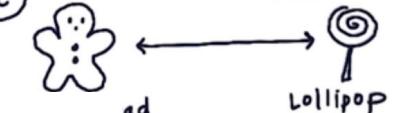
Different screen resolutions

low # of pixels ← → high # pixels  
(very sharp screen)

Different language settings



Different versions of Android



Spanish, /values-es/values.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">Mi Aplicación</string>
    <string name="hello_world">Hola Mundo!</string>
</resources>
```

```
// Get a string resource from your app's Resources
String hello = getResources() .getString(R.string.hello_world);

// Or supply a string resource to a method that requires a string
TextView textView = new TextView(this);
textView.setText(R.string.hello_world);
```

## Lesson 3B

# Style and Themes

Descreve o conceito do style e como reutilizar os estilos e padronização dos seus componentes.

---

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CodeFont" parent="@android:style/TextAppearance.Medium">
        <item name="android:layout_width">fill_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:textColor">#00FF00</item>
        <item name="android:typeface">monospace</item>
    </style>
</resources>
```

```
<TextView
    style="@style/CodeFont"
    android:text="@string/hello" />
```

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="#00FF00"
    android:typeface="monospace"
    android:text="@string/hello" />
```

**GO BEYOND**

## More!

---

- Build for tablets;
- Create lists;
- Create cards;
- Load up things from the internet;
- Store data on the device;
- Post notifications;
- Do background operations;
- Use location;
- Animations;
- Many possibilities!

# FINAL PROJECT

*5 THINGS*

# Overview

---

- Educational App that teaches 5 new things to a user;
- Criterion groups:
  - Idea Definition;
  - Solution Design and README;
  - Code Functionality;
  - Code Readability;

# What I Will Learn?

---

- Planning your app design before coding;
- Taking an app layout from drawing to XML code;
- Creating, positioning, and styling views;
- Creating interactivity through button clicks and Java code;
- Commenting and documenting your code;

# How Do I Complete this Project?

---

- Brainstorm about Your Target User;
- Gather Information;
- Pick an App Idea;
- Design a Solution;
- Read the Project Rubric;
- Write Code to Build Your App;
- Iterate;
- Write a README file.

# Share or Publish your Project

---



Google play

# References

---

- <https://goo.gl/P9Zxiu> ← essa apresentação
- <https://goo.gl/ROI1kl> (Avaliação do Projeto)
- <http://www.gdgbh.org>
- <gdgbh.slack.com> (Private Channel: **#android-study-jam**)
- <https://goo.gl/dhwwY5> (Android Development for Beginners)
- <https://developers.google.com>

# JORDAN SILVA

Mobile Engineer in the free time  
MSc. Computer Science at UFMG\*  
Code for Food



@jordansilva



+jordansilvabr



silva.jordan@gmail.com



jordansilva

\* Sim, já trabalhei no mercado por 9 anos.

# FELIPE ARIMATÉIA

Engenheiro de Software Mobile na CI&T  
Swat Team  
I Love Code



@twitterdoari



+FelipeArimateia



felipearimateia@gmail.com



felipearimateia

— — —