



Minicurso de Android Para Iniciantes

onedaycode.com

Conheça o Curso Completo



Esse E-book faz parte do
“Curso de Android Para Iniciantes” da [One Day Code](https://onedaycode.com)

Para conhecer o Curso Online Completo em Vídeo, que
Recebe atualizações, possu quizzes e novos exercícios:



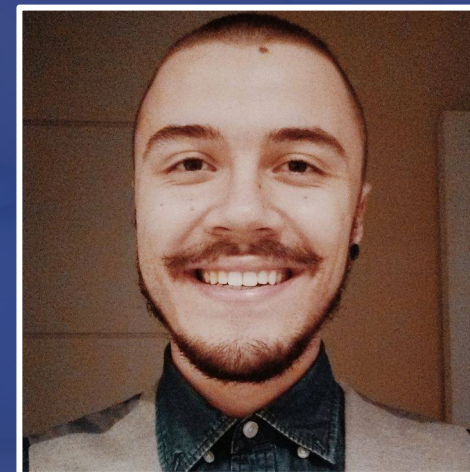
CLIQUE NO BOTÃO ABAIXO:

**CLIQUE PARA CONHECER
O CURSO COMPLETO**

onedaycode.com

Sobre o Autor

- João Victor Santos Pinho Teixeira
- Técnico em Tecnologia da Informação pelo TECPUCPR.
- Estudante de Sistemas da Informação da UTFPR.
- Contato: joao@onedaycode.com
- Dono dos Canais de Tutoriais [Games Parati](#) e [One Day Code](#).
- Desenvolvedor dos apps [Vestfoco](#) e [Save My Night](#).



**CLIQUE PARA CONHECER
O CURSO COMPLETO**

onedaycode.com

Introdução

- Para Instalar e configurar o Android Studio com Debug USB [Clique Aqui](#).
- Para instalar e Configurar um Emulador Rápido para Android [Clique Aqui](#).
- Para Gerar Arquivos Instaláveis para Android [Clique Aqui](#).
- Para Adquirir os Módulos do curso de Android [Clique Aqui](#).
- O sistema Android é desenvolvido pela Google e baseado em Unix.
- 82.2% dos dispositivos móveis usam Android (Mais de um bilhão).
- Programa utilizado para criação de projetos: Android Studio.



**CLIQUE PARA CONHECER
O CURSO COMPLETO**

onedaycode.com

Sumário

1 - Conceitos Básicos

- 1 - [Acelerando o Android Studio](#)
- 2 - [Criando um Novo Projeto](#)
- 3 - [Configurando Emulador](#)
- 4 - [Para usar Acentos](#)
- 5 - [A Interface do Android Studio](#)
- 6 - [Java Básico – Tipos de Variáveis](#)
- 7 - [Log.i – Imprimindo Valores](#)
- 8 - [Erros – Soluções](#)
- 9 - [Comentários](#)
- 10 - [Strings.xml – Frases Pré Definidas](#)
- 11 - [TextViews – Campos de Texto](#)
- 12 - [ImageViews – Campos de Imagem](#)
- 13 - [Button - Botões](#)

2 - Funções

- 14 - [Funções](#)
- 15 - [Disponibilidade de Funções \(Public, Protected e Private\)](#)
- 16 - [Void - Função que retorna Vazio](#)
- 17 - [Função chamada por Button](#)
- 18 - [String - Função que retorna String](#)
- 19 - [Integer - Função que retorna Integer](#)
- 20 - [TextView – Alterando com Código](#)
- 21 - [ImageView – Alterando com Código](#)
- 22 - [ImageView – Animando Imagens](#)
- 23 - [Button – Alterando com Código](#)
- 24 - [Toasts – Alertas na Tela](#)

3 - Formulários

- 25 - [EditTexts](#)
- 26 - [EditTexts – Recebendo String](#)
- 27 - [EditTexts – Recebendo Integer](#)

Extra

- 28 - [TextViews com Fontes Próprias](#)

**CLIQUE E CONHEÇA O
CURSO COMPLETO**

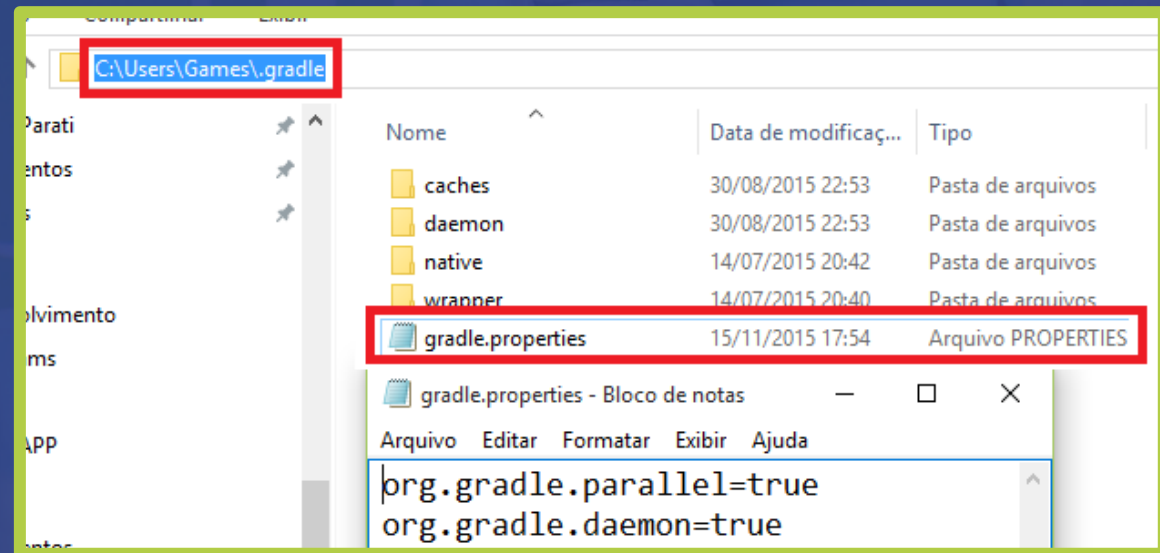


Acelerando o Android Studio no Windows

onedaycode.com

Acelerando o Android Studio

- O Gradle é o responsável por dar build nos projetos do Android, inicialmente ele pode ser bem lento, mas podemos acelerá-lo.
- Em seu pc, vá até a pasta: “**C:\Users\NomeDoSeuUsuário\gradle**”.
- Crie um arquivo chamado “**gradle.properties**” com as linhas:
“**org.gradle.parallel = true**”
“**org.gradle.daemon=true**”

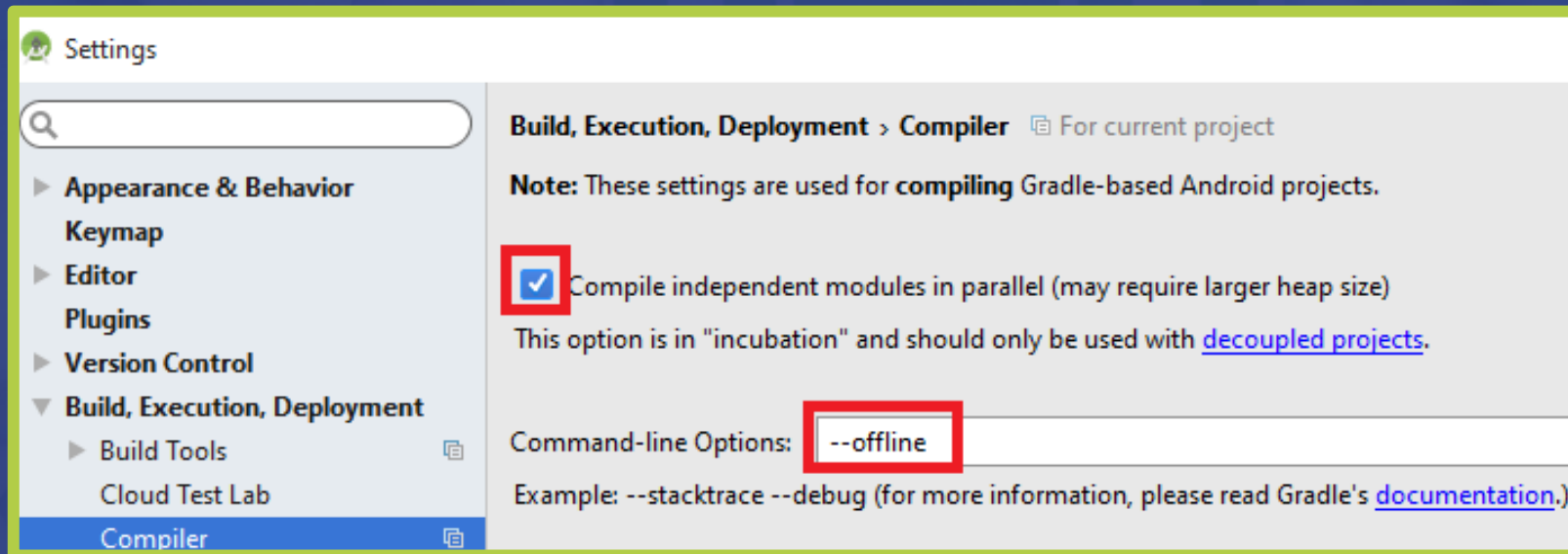


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Acelerando o Android Studio

- Depois disso, abra o Android Studio e pressione CTRL+ALT+S para abrir Settings.
- Em “**Build, Execution, Deployment**” clique em **Compiler**.
- Marque a opção “**Compile independent modules in parallel**”.
- E coloque o texto “**--offline**” em “**Command-line Options:**”.



CLIQUE E CONHEÇA O
CURSO COMPLETO

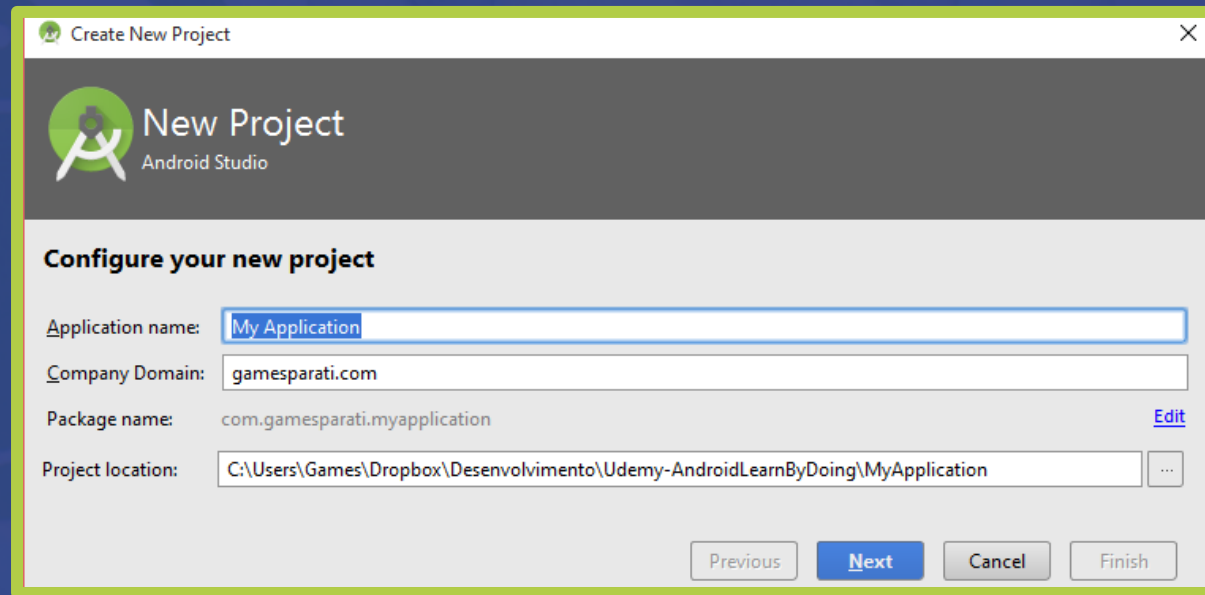
onedaycode.com



Criando um Novo Projeto no Android Studio

Criando um novo Projeto no Android Studio

- File -> New -> **New Project** .
- Application Name: **Nome do Projeto**.
- Company Domain: **Site da Empresa**.
- Project Location: **Local de Salvamento**.



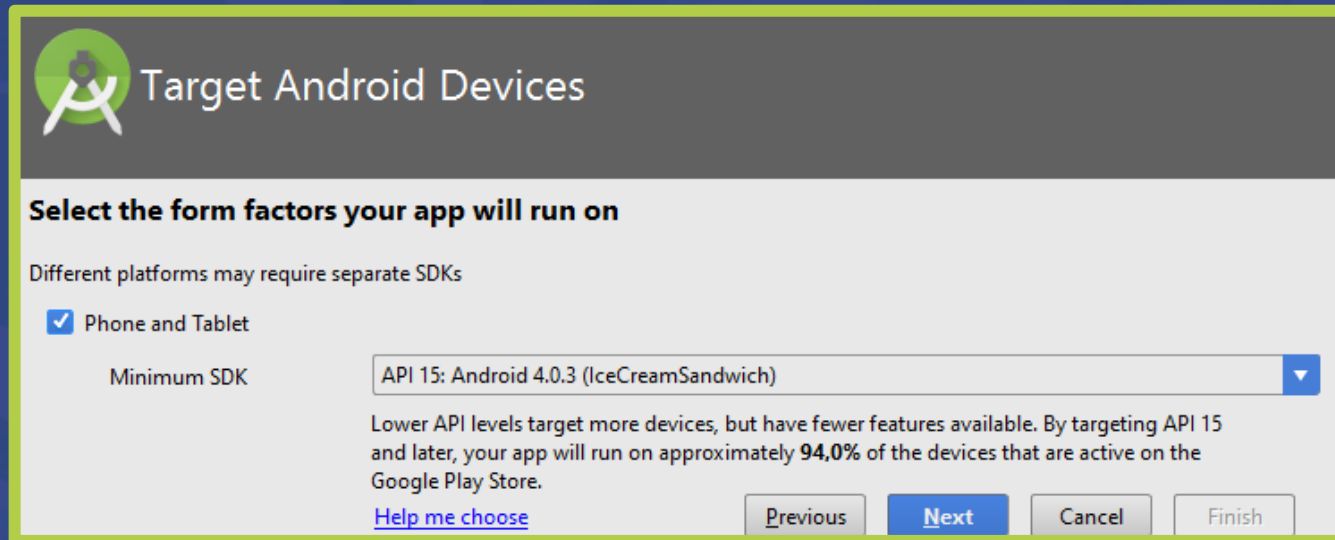
The screenshot shows the 'Create New Project' dialog box in Android Studio. The dialog has a title bar 'Create New Project' and a close button. Below the title bar is the Android Studio logo and the text 'New Project Android Studio'. The main section is titled 'Configure your new project'. It contains four input fields: 'Application name' with the text 'My Application', 'Company Domain' with the text 'gamesparati.com', 'Package name' with the text 'com.gamesparati.myapplication' and an 'Edit' link, and 'Project location' with the text 'C:\Users\Games\Dropbox\Desenvolvimento\Udemy-AndroidLearnByDoing\MyApplication' and a browse button. At the bottom right are four buttons: 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Finish'.


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando um novo Projeto no Android Studio

- **Minimum SDK:** Versão mínima de Android que será compatível.
- (Geralmente usa-se a API 15).



 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94,0%** of the devices that are active on the Google Play Store.

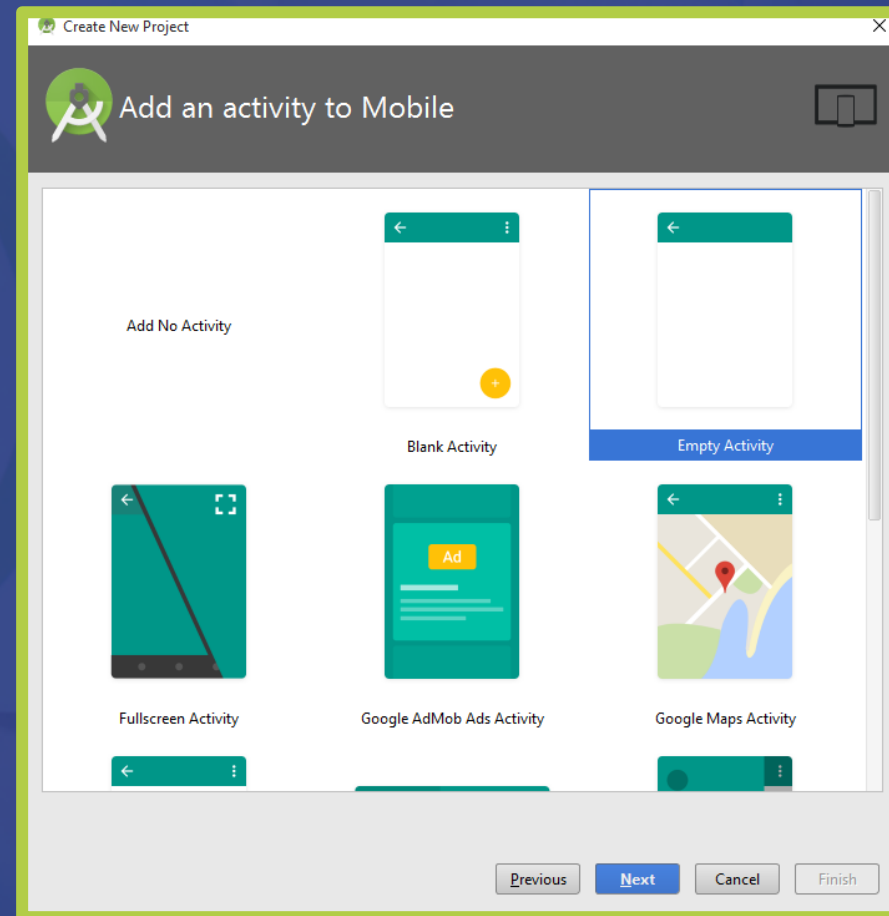
[Help me choose](#) [Previous](#) [Next](#) [Cancel](#) [Finish](#)

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando um novo Projeto no Android Studio

- Escolha o modelo da sua primeira tela, no caso usaremos “**Empty Activity**”.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Para usar Acentos

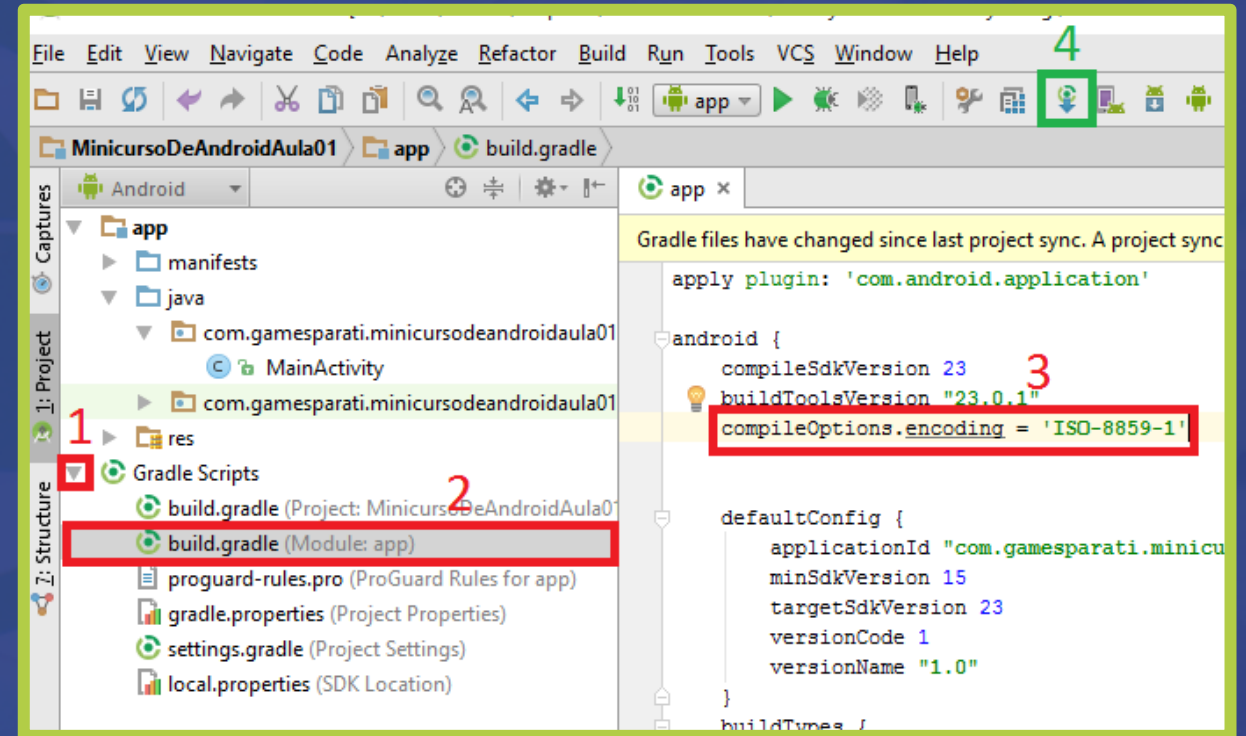
- Clique em Gradle Scripts.
- Dois cliques em build.gradle(Module:app).

- Adicione o código:

`compileOptions.encoding = "ISO-8859-1"`

Abaixo de `buildToolsVersion` .

- Clique no  para sincronizar.

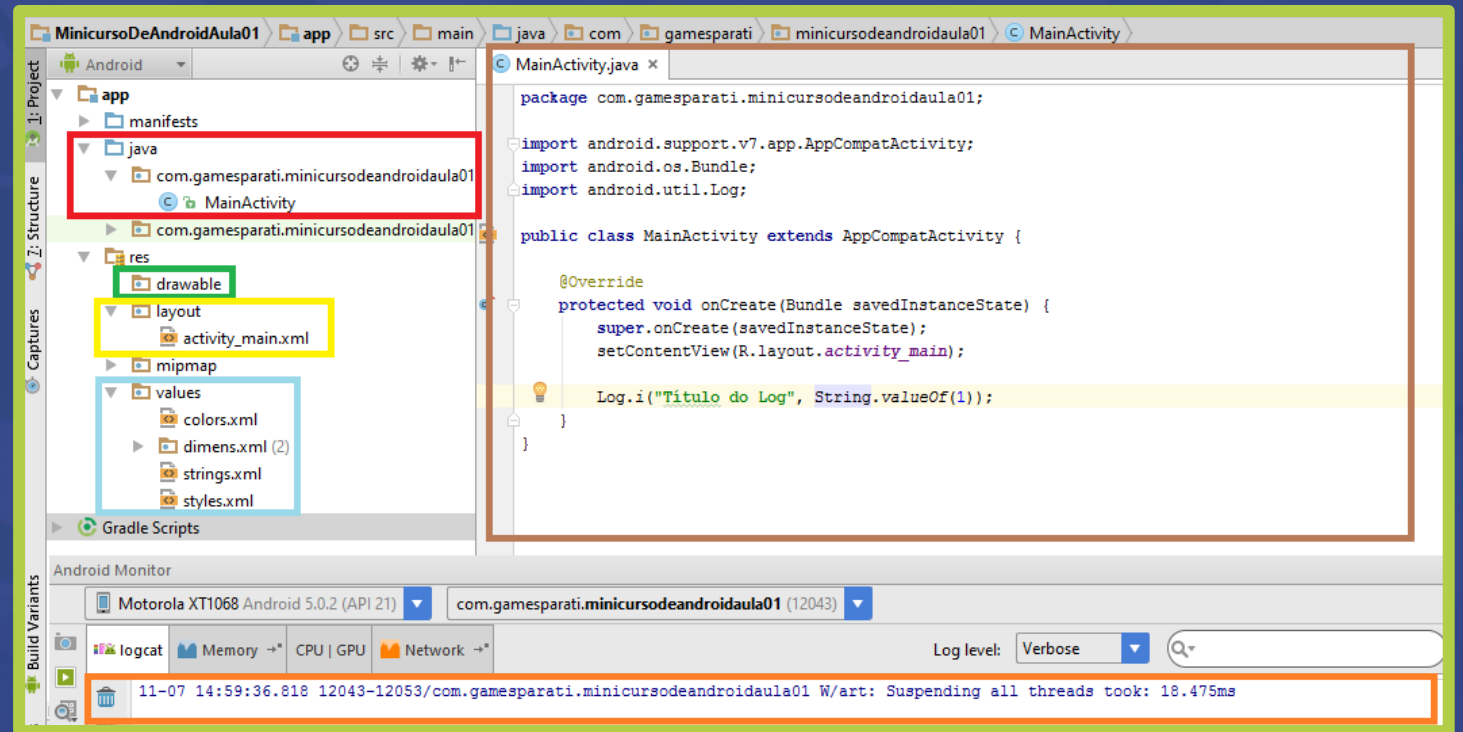


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

A Interface

- Classes ■
- Imagens e Logo ■
- Layouts das Telas ■
- Valores de Cores, Frases e Estilos ■
- Edição de Códigos ■
- LogCat com informações ■



CLIQUE E CONHEÇA O
CURSO COMPLETO

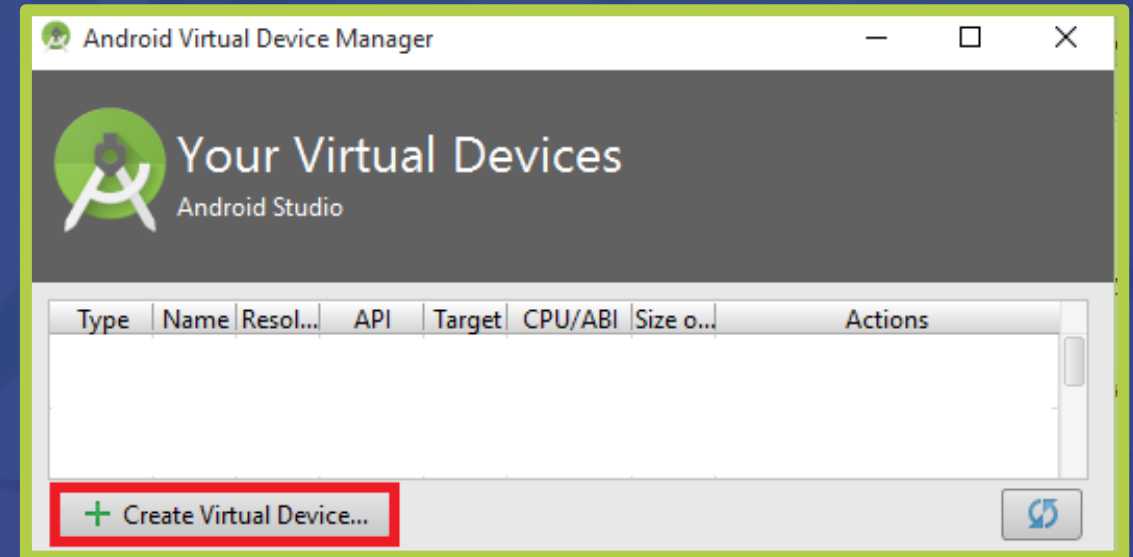
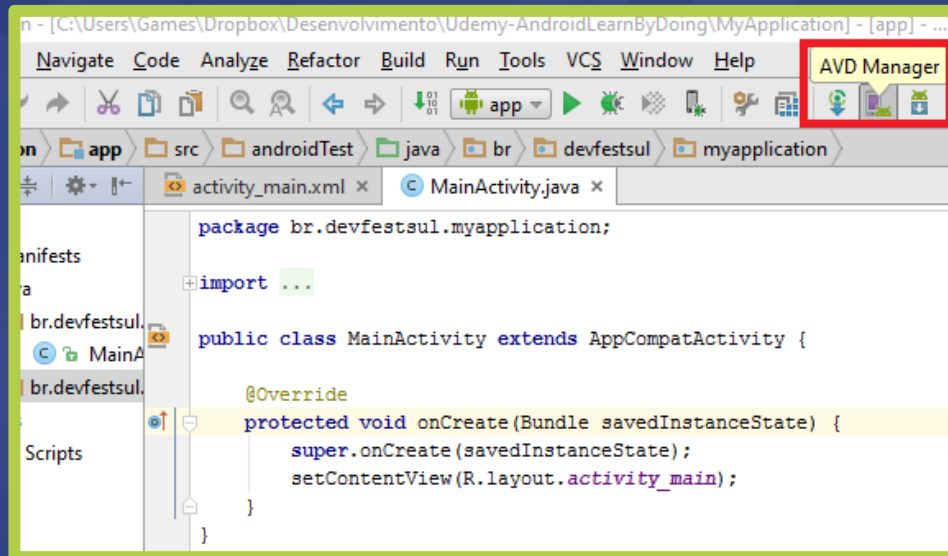
onedaycode.com



Configurando Emulador

Criando Máquina Virtual

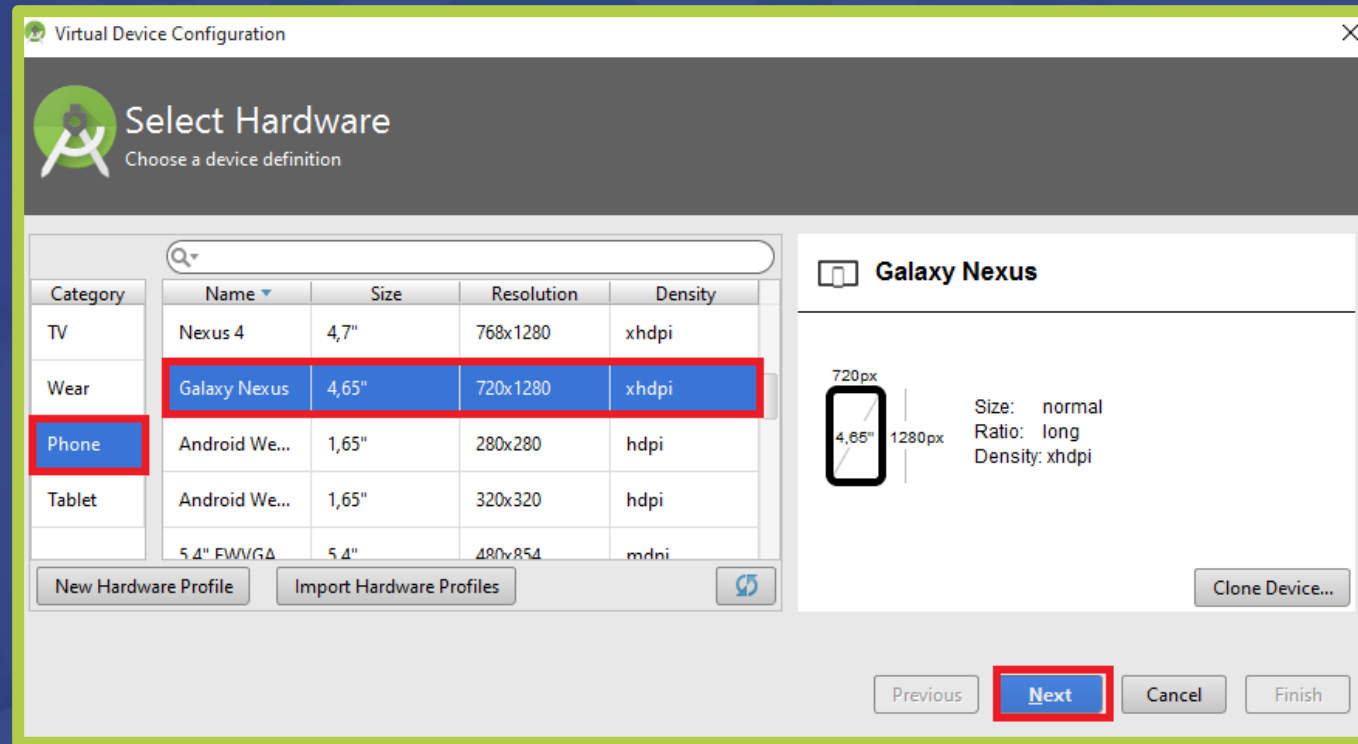
- No Menu superior, clique em “**AVD Manager**”.
- Clique em “**Create Virtual Device**”.



CLIQUE E CONHEÇA O
CURSO COMPLETO

Criando Máquina Virtual

- Escolha o modelo desejado e clique em “Next”.

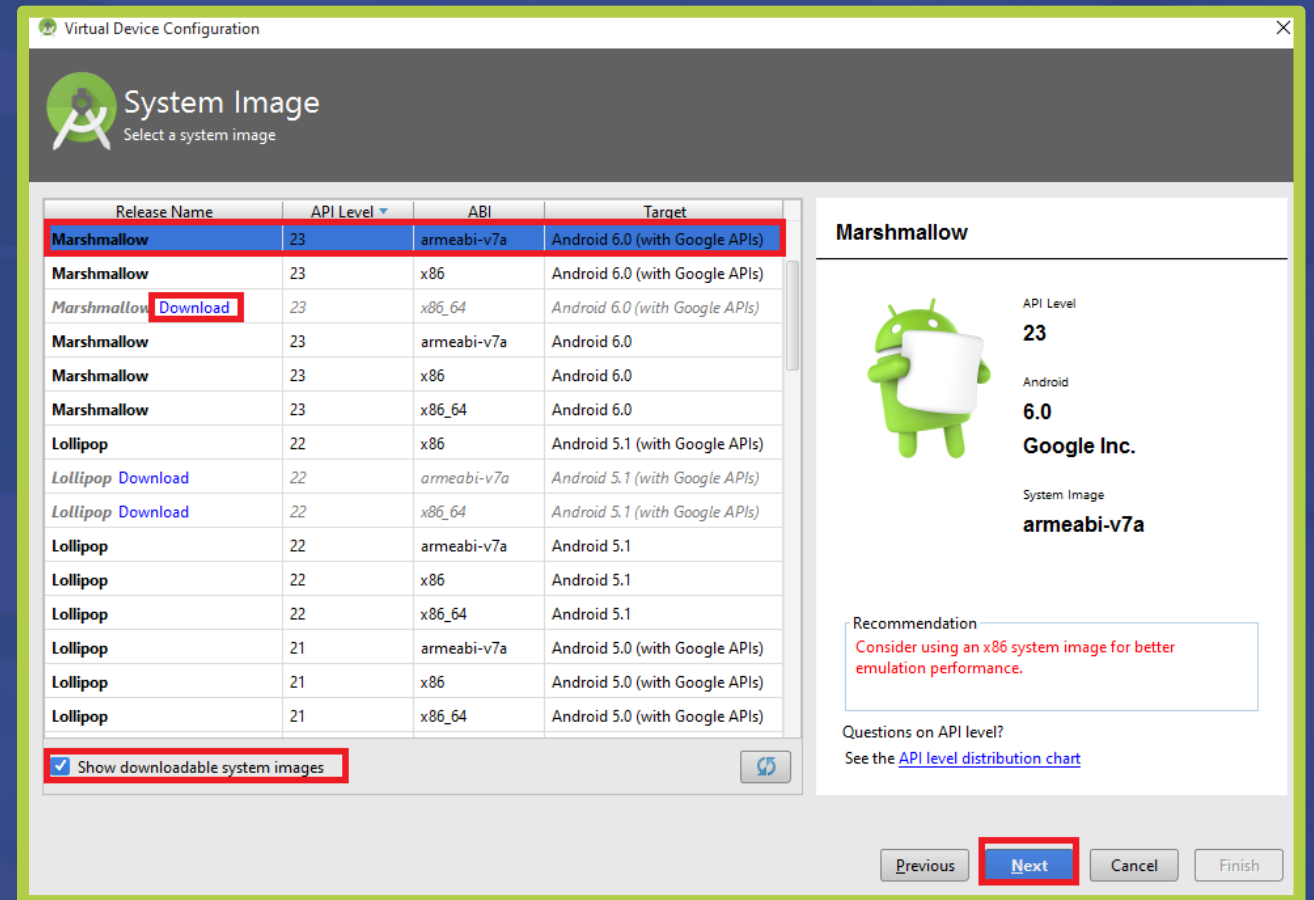


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando Máquina Virtual

- Marque a caixa:
“**Show downloadable system images**”.
- Escolha uma versão e clique em
“**Download**”.
- Nesse curso , usaremos a
“**Marshmallow, API 23, armeabi-v7a,
Android 6.0 with Google APIs**”.

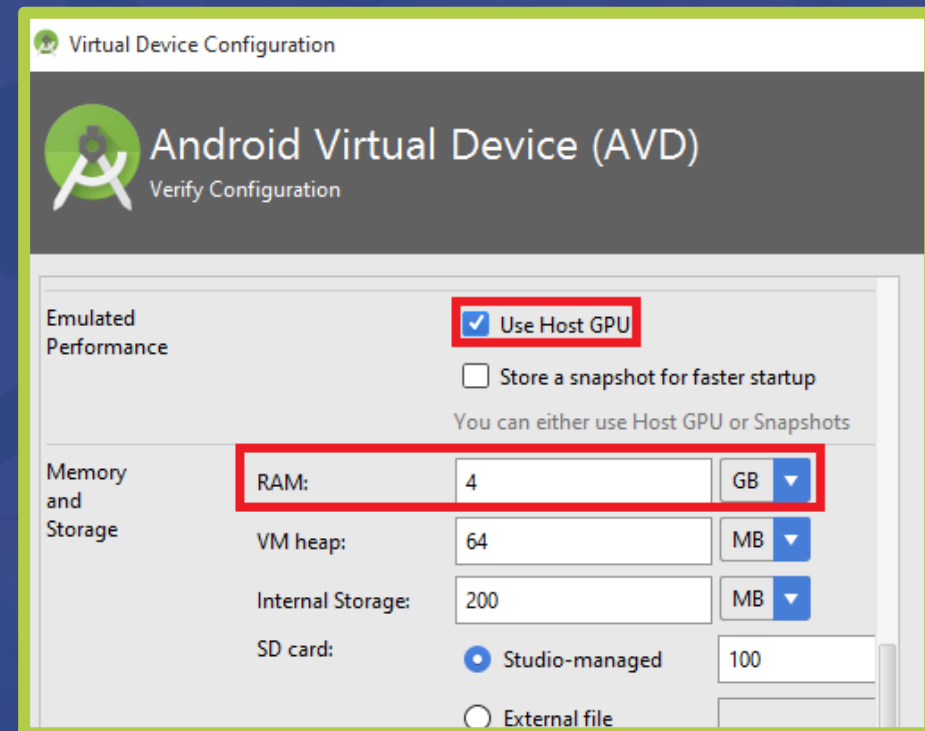
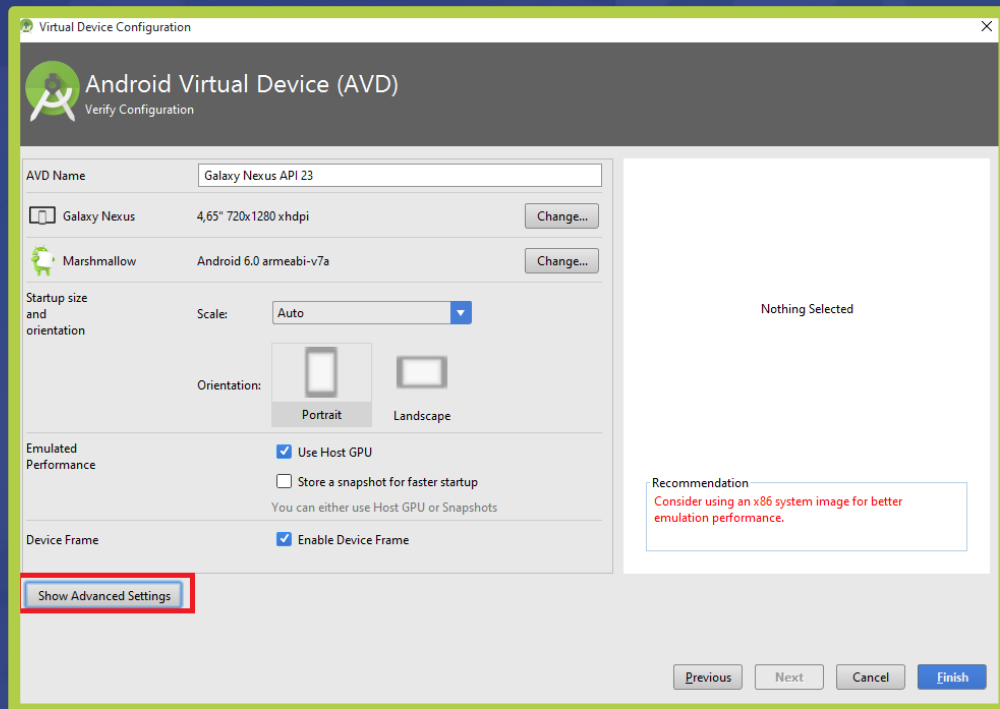


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando Máquina Virtual

- Clique em “**Show Advanced Settings**”.
- Deixe o campo RAM com **4 GB** e marque “**Use Host GPU**”, depois clique em “**Finish**”.

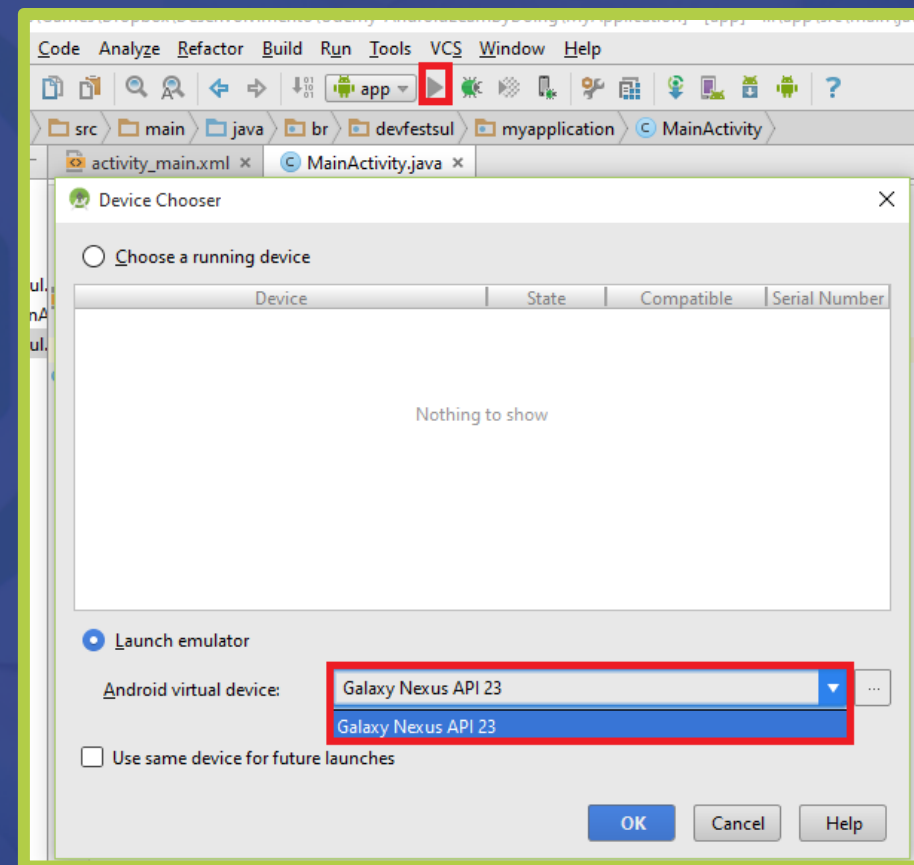


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando Máquina Virtual

- Agora quando você der “**Play**” no seu projeto, você verá a opção de iniciar com sua máquina virtual.



CLIQUE E CONHEÇA O
CURSO COMPLETO

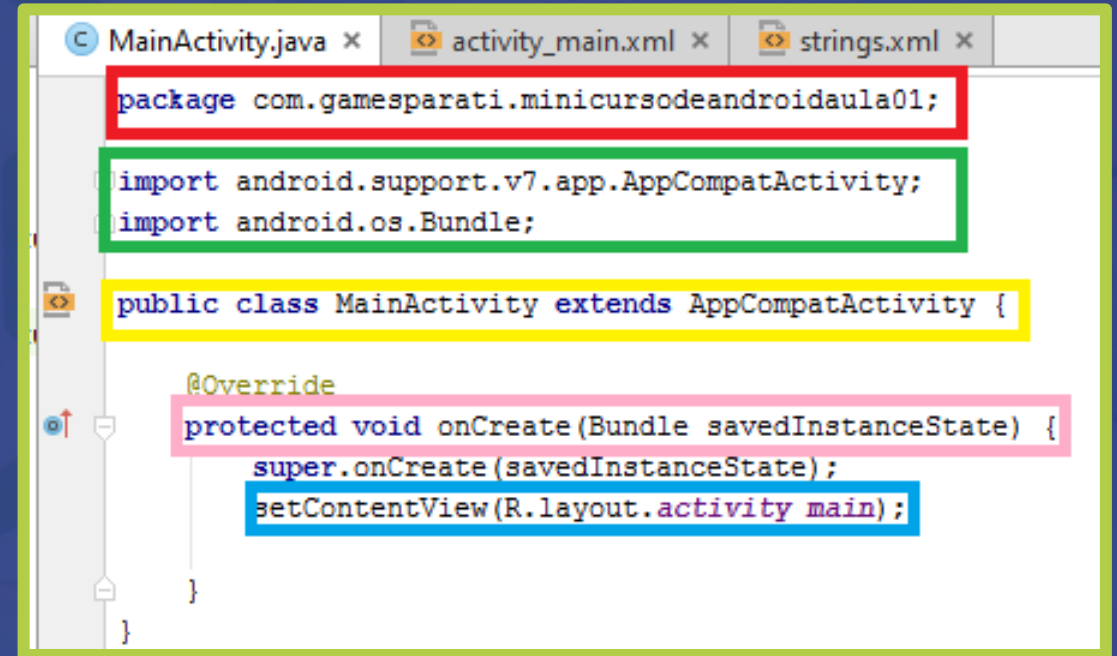
onedaycode.com



Java Básico , Importando Pacotes e Imprimindo Valores

MainActivity.java

- Nome do pacote do projeto ■
- Pacotes importados ■
- Função Principal (Extendendo uma classe de compatibilidade para Barra Superior) ■
- Função acionada após iniciar o app ■
- Infla com o layout escolhido ■



```
package com.gamesparati.minicursodeandroidaula01;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

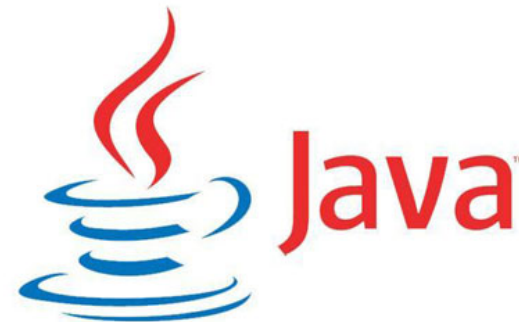
The screenshot shows the MainActivity.java file in an IDE. The code is color-coded: the package name is in red, imports are in green, the class declaration is in yellow, the @Override annotation is in pink, and the onCreate method body is in blue. The IDE tabs at the top show MainActivity.java, activity_main.xml, and strings.xml.

**CLIQUE E CONHEÇA O
CURSO COMPLETO**

onedaycode.com

Java Básico para Android

- No Android, as Classes podem ser utilizadas para gerenciar comportamentos das telas “**Activities**” armazenando métodos e ações.
- As variáveis que iremos utilizar possuem tipos:
- **String** (para Textos).
- **Integer** (para Números Inteiros).
- **Double** (para Números Quebrados).
- **Boolean** (Verdadeiro ou Falso).



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

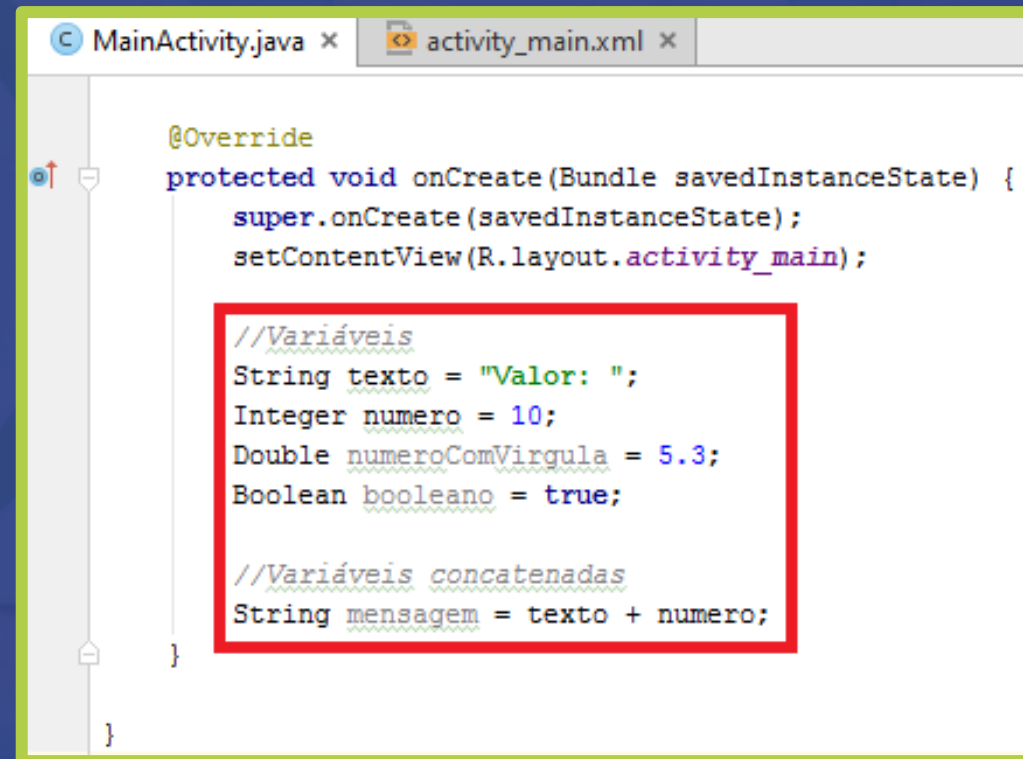
Definindo e Concatenado Variáveis

Definindo:

- **String** nomeDaVariavel = " Empolgante " ;
- **Integer** nomeDaVariavel = 69 ;
- **Double** nomeDaVariavel = 5.3 ;
- **Boolean** nomeDaVariavel = true ;

Concatenando String e Integer:

- **String** texto = " Valor: " ;
- **Integer** valorNumerico = 10 ;
- **String** mensagem = texto + valorNumerico .
- Resultado: mensagem = " Valor: 10 " .



```
MainActivity.java x activity_main.xml x

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Variáveis
    String texto = "Valor: ";
    Integer numero = 10;
    Double numeroComVirgula = 5.3;
    Boolean booleano = true;

    //Variáveis concatenadas
    String mensagem = texto + numero;
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

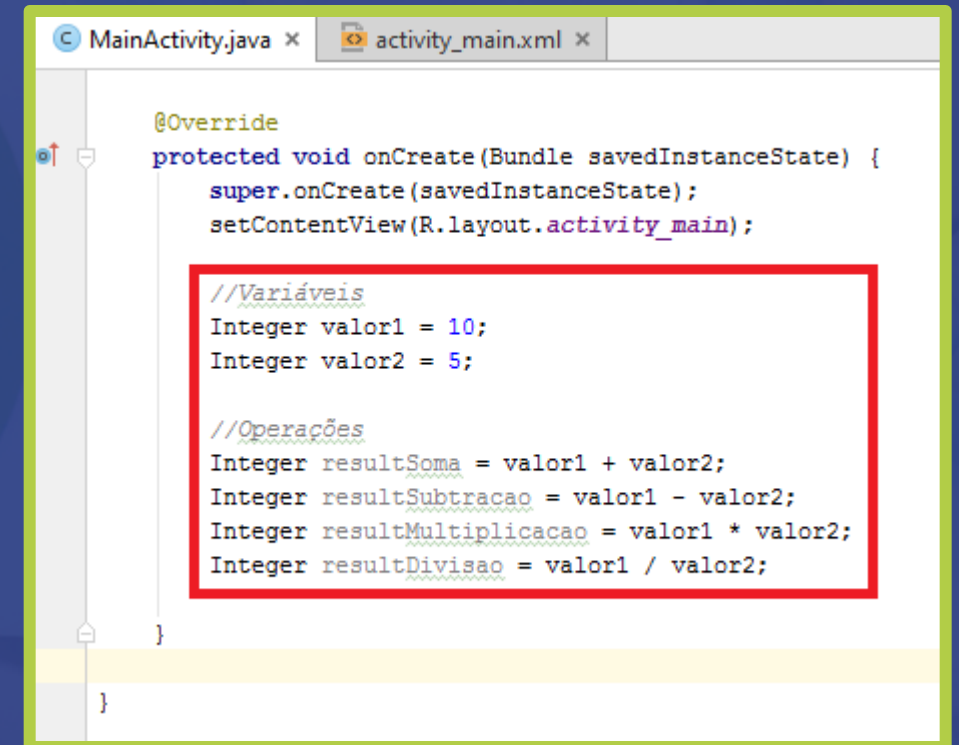
Operações Matemáticas

Definindo Variáveis

- **Integer** valor1 = 10 ;
- **Integer** valor2 = 5 ;

Operações:

- Soma: **Integer** result = valor1 + valor2 ;
- Subtração: **Integer** result = valor1 - valor2 ;
- Multiplicação: **Integer** result = valor1 * valor2 ;
- Divisão: **Integer** result = valor1 / valor2 ;



```
MainActivity.java x activity_main.xml x

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Variáveis
    Integer valor1 = 10;
    Integer valor2 = 5;

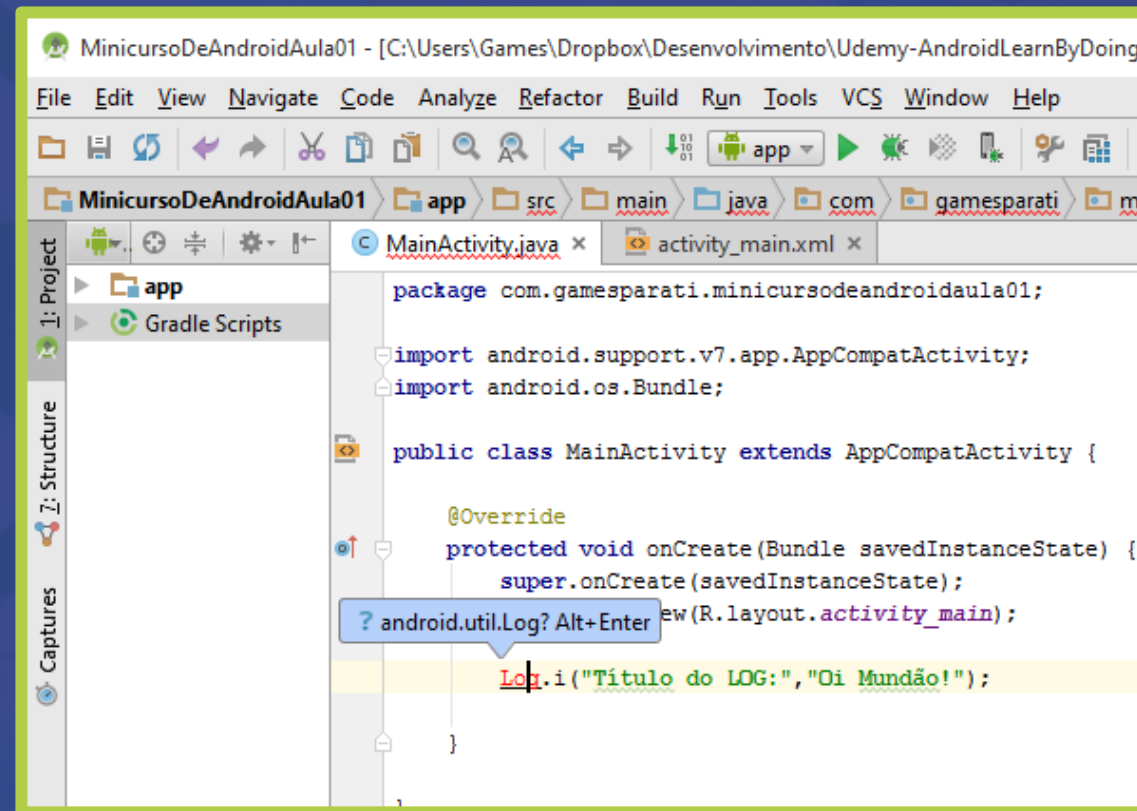
    //Operações
    Integer resultSoma = valor1 + valor2;
    Integer resultSubtracao = valor1 - valor2;
    Integer resultMultiplicacao = valor1 * valor2;
    Integer resultDivisao = valor1 / valor2;
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Imprimindo Valores e Importando Classes

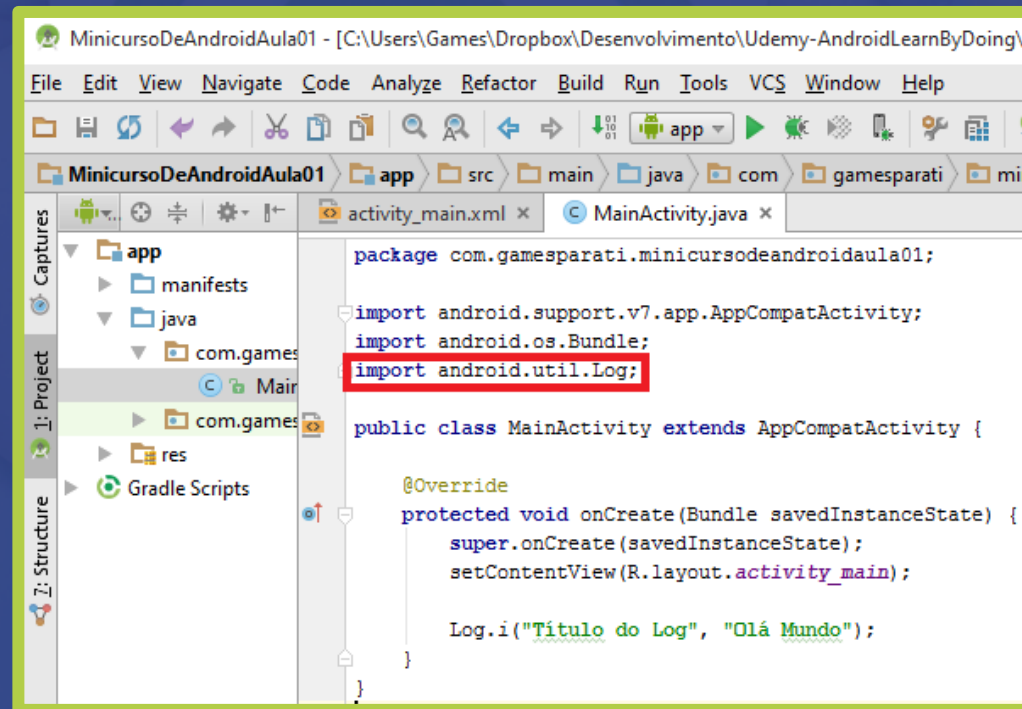
- O Logcat exibe o que o sistema está executando.
- No Android, para imprimir valores no Logcat, utilizamos o `Log.i(" String ", " String ")`, ou seja, esse comando utiliza como parâmetros dois Textos(**Strings**).
- Sempre que o texto ficar em **vermelho** com um balão azul, significa que precisamos importar uma classe.
- Para isso usamos ALT + ENTER.



CLIQUE E CONHEÇA O
CURSO COMPLETO

Imprimindo Valores e Importando Classes

- Agora o pacote necessário foi automaticamente importado.

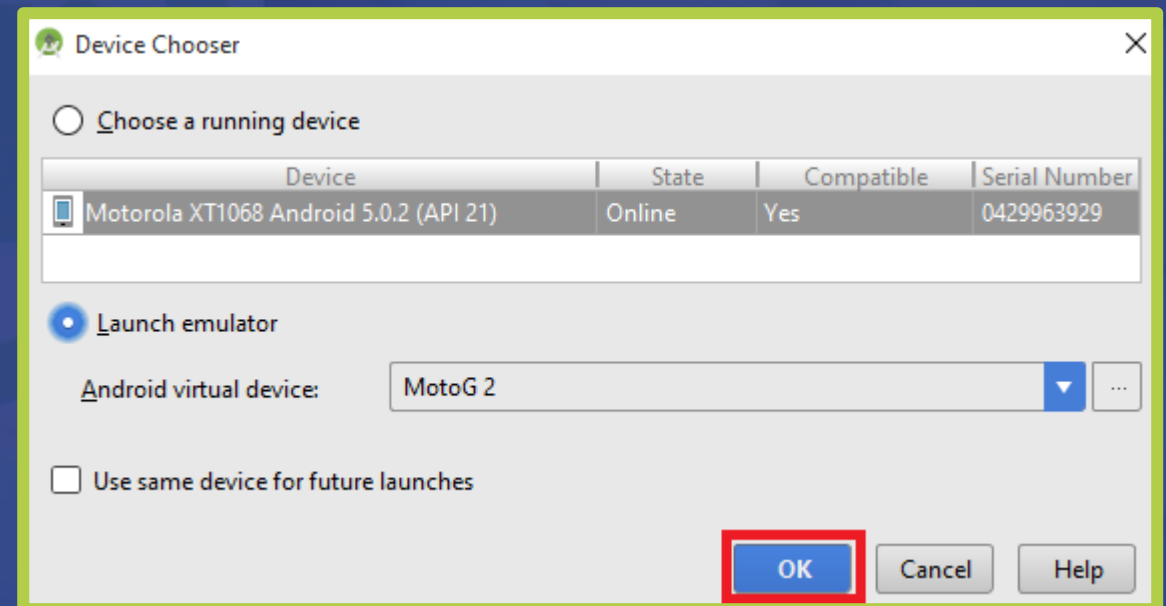
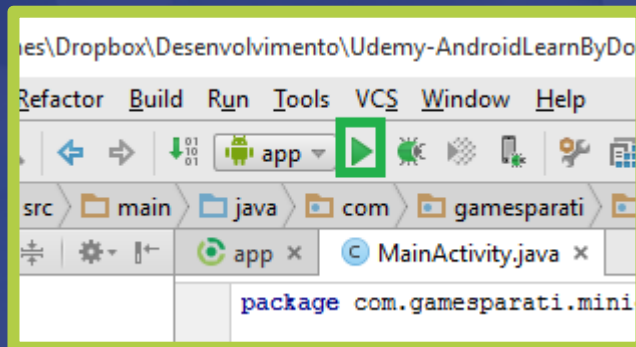


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Rodando o Projeto

- Clique no “**Play**” para iniciar o projeto.
- Escolha o “**Dispositivo**” ou “**Emulador**” e Clique em “**OK**”.

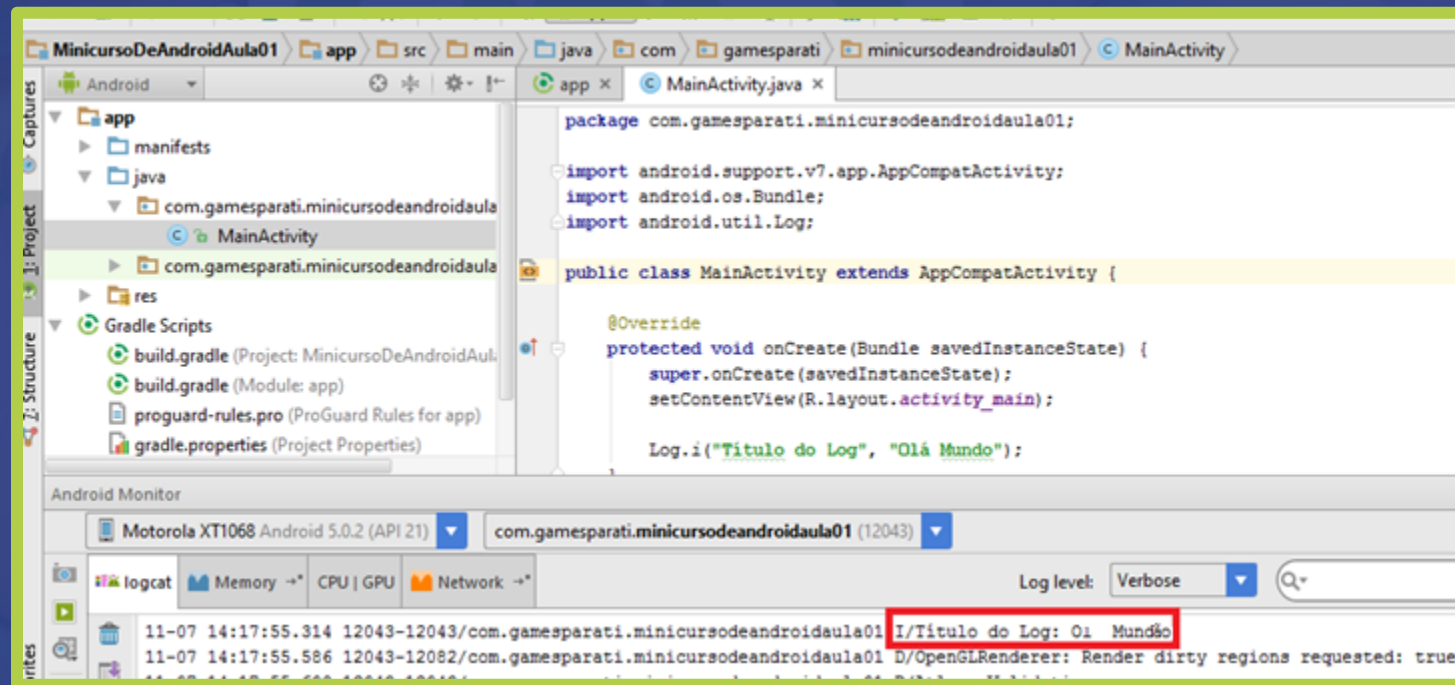


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Resultado

- Podemos observar no canto inferior no Logcat, os **Logs** e ver que a mensagem foi exibida.



CLIQUE E CONHEÇA O
CURSO COMPLETO

Desafio

Crie um projeto que exiba uma variável do tipo **String** utilizando o **Log.i** .

Slides usados
Definir String
Exibir Log.i


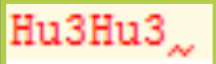



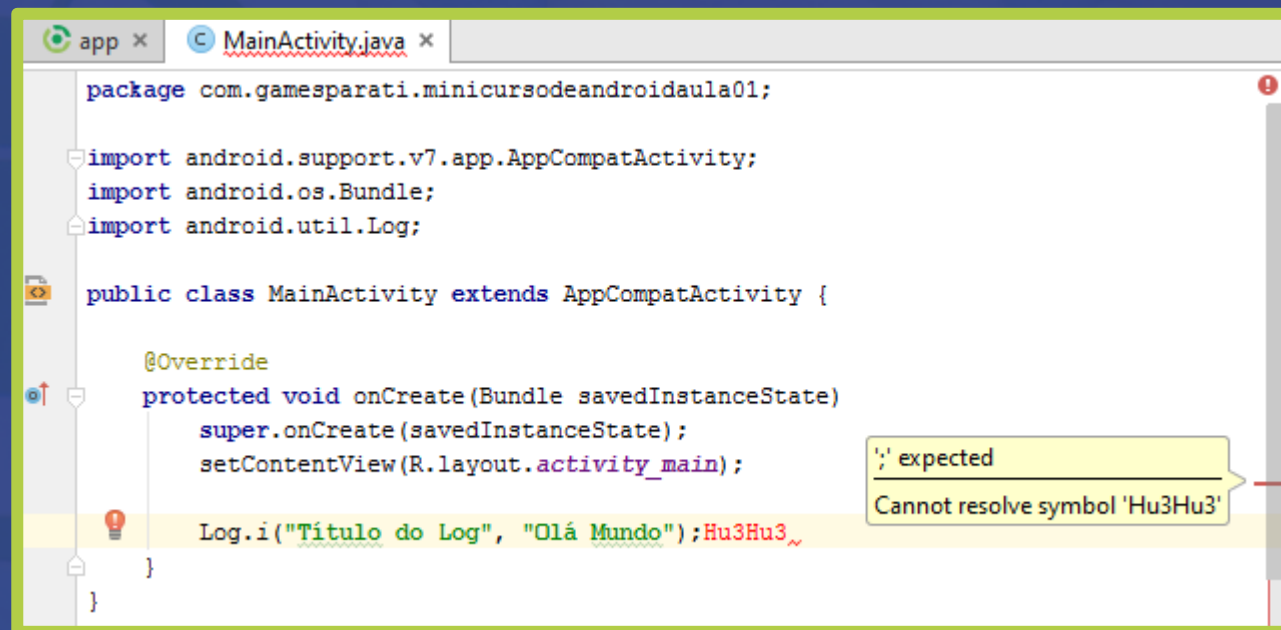
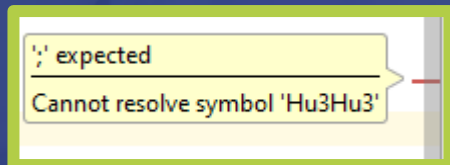
Resposta:
Exercício01



Erros e Soluções


Indicadores de Erros

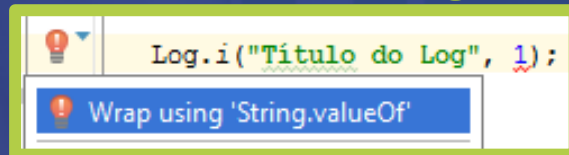
- O sinal  fica na linha do erro.
- O texto com erro fica vermelho:

- Na barra de rolagem há um: 
- E uma mancha:



CLIQUE E CONHEÇA O
CURSO COMPLETO


Solução

- Ao clicar no  o software sugere correções para o erro.
- Exemplo: Para usar o **Log.i**, os parâmetros são **Log.i(String, String)** que são (Título, Mensagem) , Se eu colocar um **Integer** como o número 1 como Mensagem, a solução proposta será:



- **Wrap using "String.valueOf"** , ou seja, converter o **Integer** para **String**.

- Ao clicar na opção, o novo código fica:



```
Log.i("Título do Log", String.valueOf(1));
```

- GGWP, problema resolvido!

CLIQUE E CONHEÇA O
CURSO COMPLETO

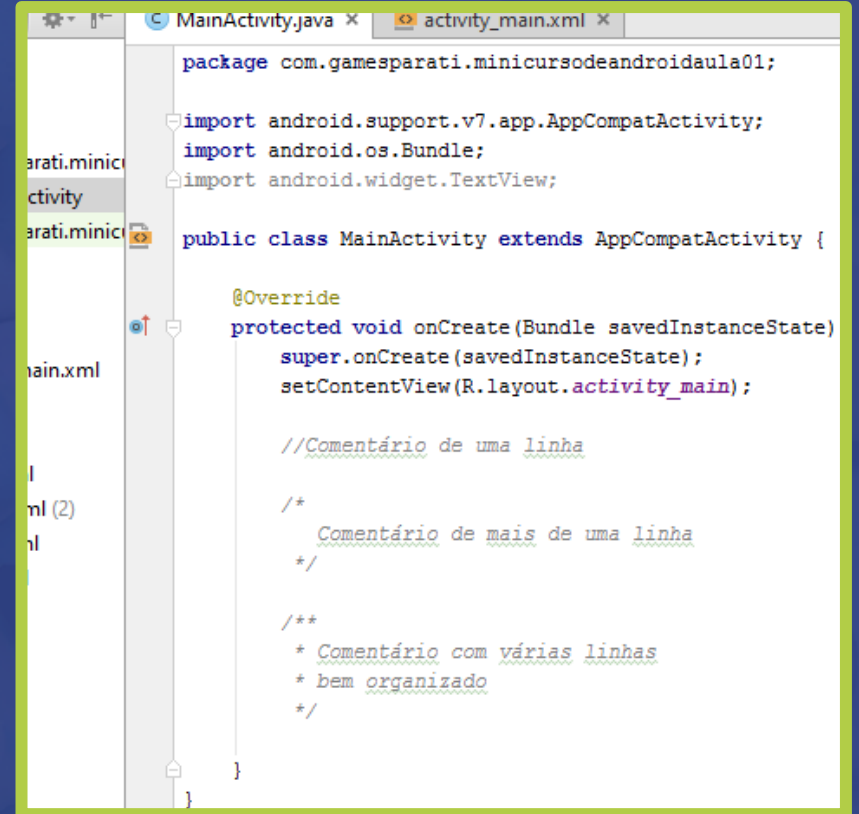
onedaycode.com



Comentários

Comentários

- Servem para organizar o código com descrições.
- Tudo que for comentado não será lido pelo app.
- Para comentar uma linha, use `//`.
- Para comentar mais de uma linha use `/*` e termine com `*/`.
- Para criar um bloco use `/**` pressione ENTER e termine com `*/`.



```
package com.gamesparati.minicursodeandroidaula01;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Comentário de uma linha

        /*
         Comentário de mais de uma linha
        */

        /**
         * Comentário com várias linhas
         * bem organizado
         */
    }
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

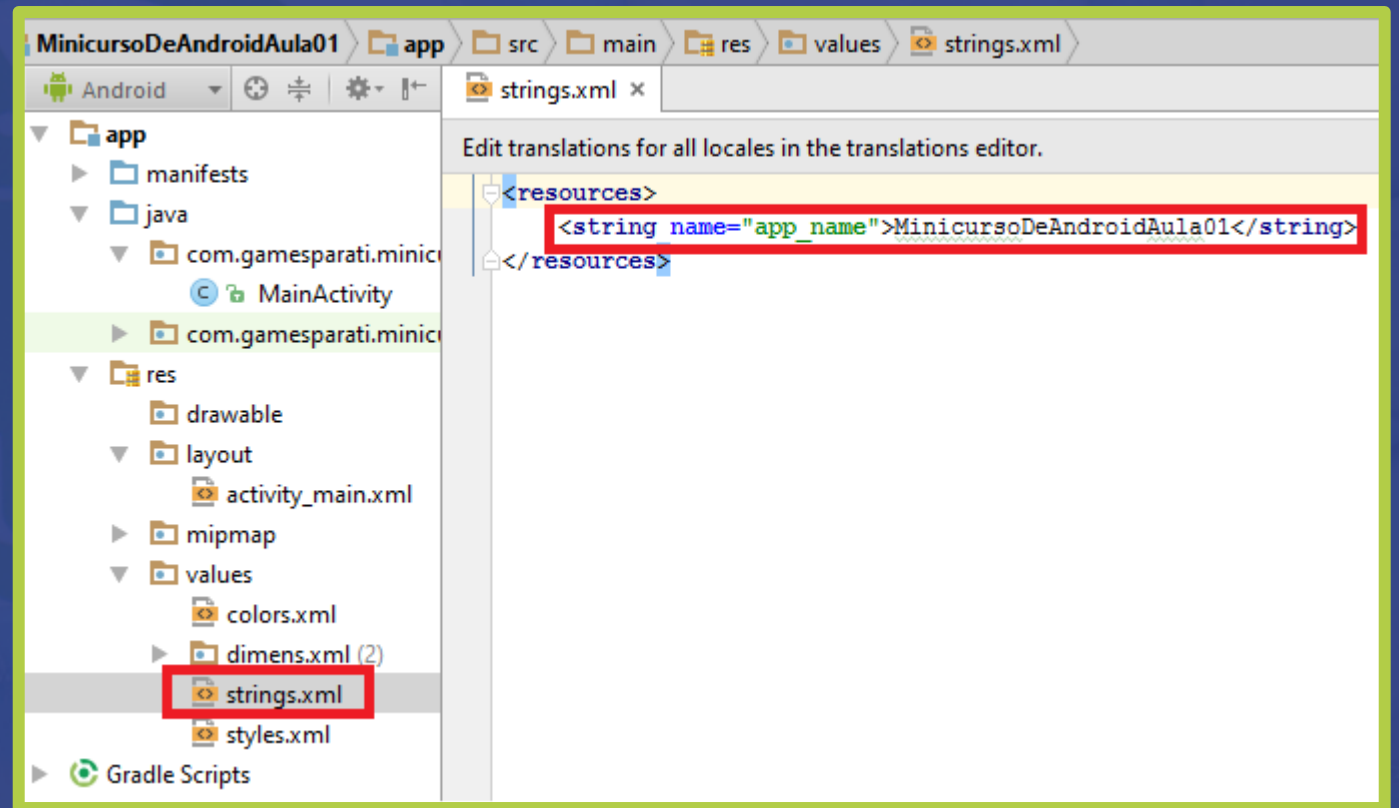
onedaycode.com



String Values

Strings Pré-Definidas

- Em values/strings.xml podemos deixar pré-definidas as frases que serão usadas muitas vezes no app.
- Facilitam quando precisarmos modificar algo que aparece com frequência no app.
- Ajuda caso use outras línguas, pois facilita a organização.

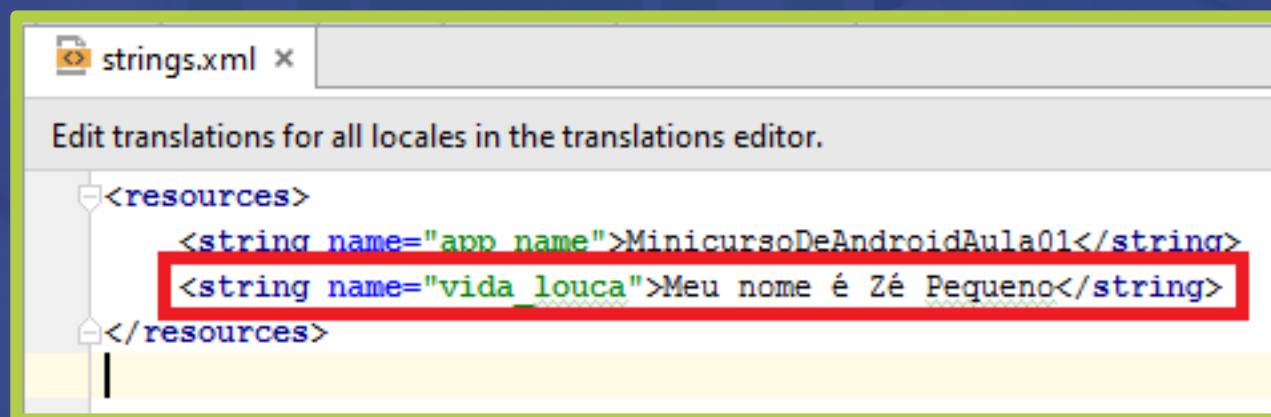


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Strings Pré-Definidas

- Para adicionar uma nova **string**, use o código:
<string name="nome_da_string"> Conteúdo </string> .



```
strings.xml x
Edit translations for all locales in the translations editor.
<resources>
  <string name="app_name">MinicursoDeAndroidAula01</string>
  <string name="vida_louca">Meu nome é Zé Pequeno</string>
</resources>
```

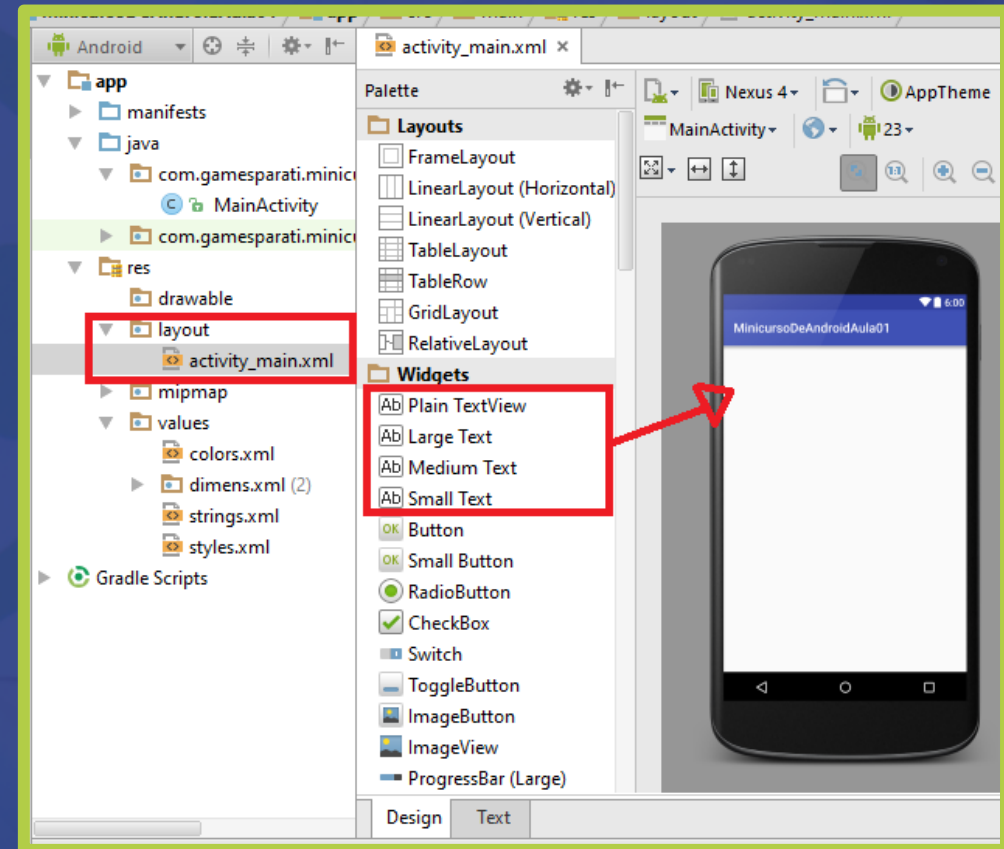
CLIQUE E CONHEÇA O
CURSO COMPLETO



TextViews

TextViews

- **TextViews** são campos de texto.
- Para adicionar um em uma tela, abra layout/activity_main.xml.
- Selecione um tipo de campo de texto e arraste-o para o celular.

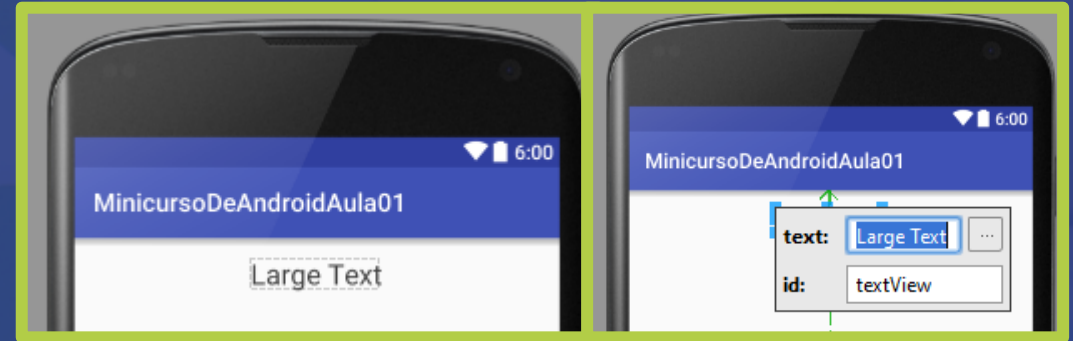


CLIQUE E CONHEÇA O
CURSO COMPLETO

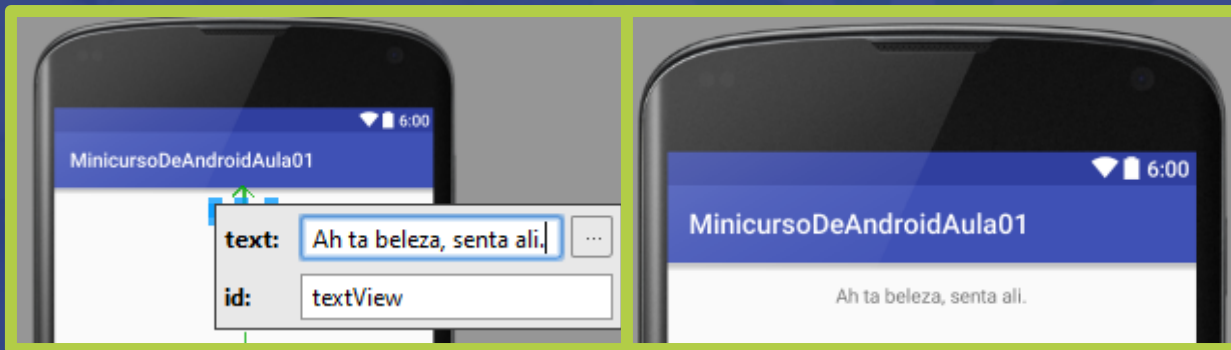
onedaycode.com

TextViews

- Ao clicar duas vezes em um **TextView**, você verá duas opções:
- **Text**: É o texto que será exibido inicialmente.
- **Id**: É o identificador do campo.



- Exemplo:



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

TextViews com String Pré-Definida

- Para passar um valor pré-definido em Strings.xml para uma **TextView**, utilize o código: `@string/nome_da_string`.
- Exemplo:



CLIQUE E CONHEÇA O
CURSO COMPLETO

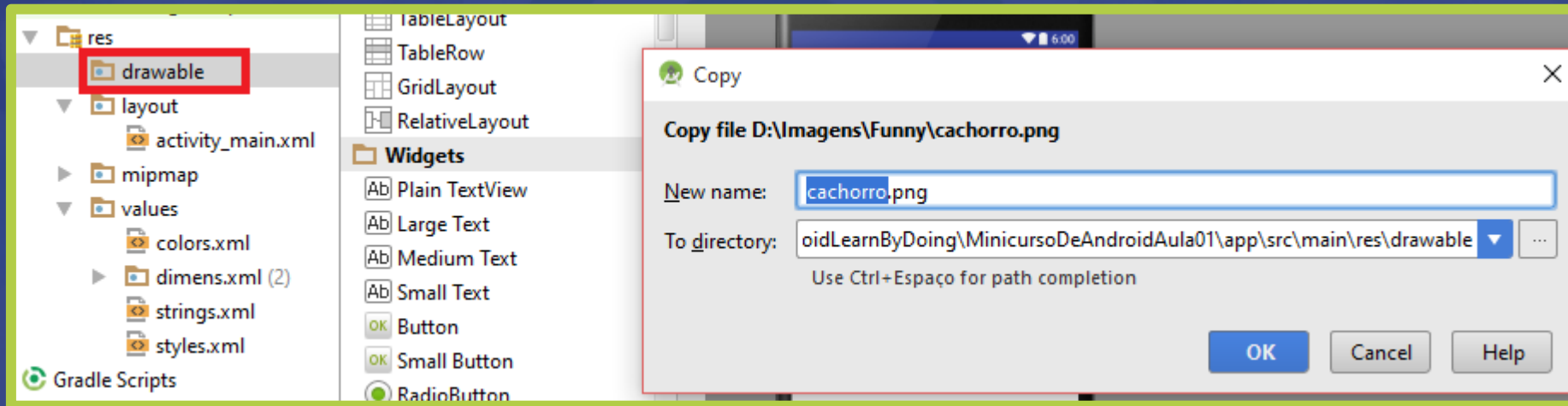
onedaycode.com



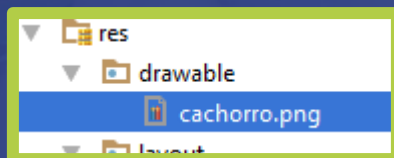
ImageViews

ImageViews

- Para adicionar imagens ao projeto, precisamos arrastá-los e soltá-los na pasta drawable .



- Resultado:

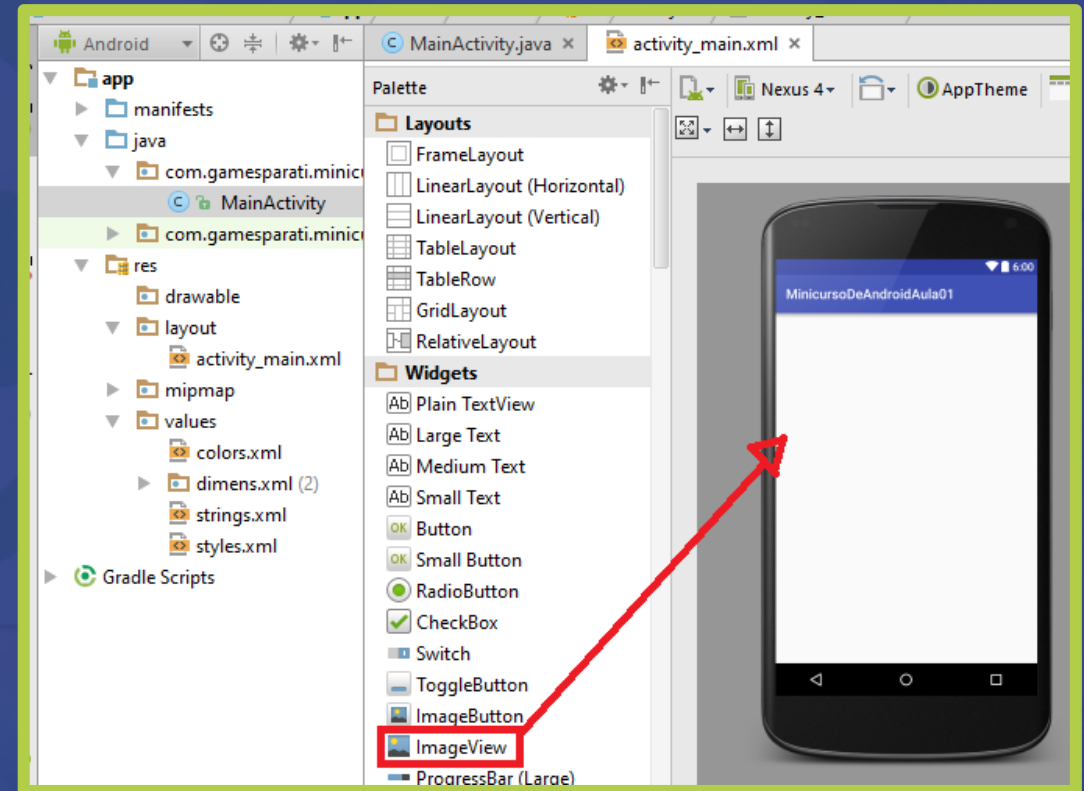


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

ImageViews

- **ImageViews** são campos de imagem.
- Para adicionar um em uma tela, abra o layout/activity_main.xml.
- Selecione o tipo **ImageView** e arraste-o para o celular.

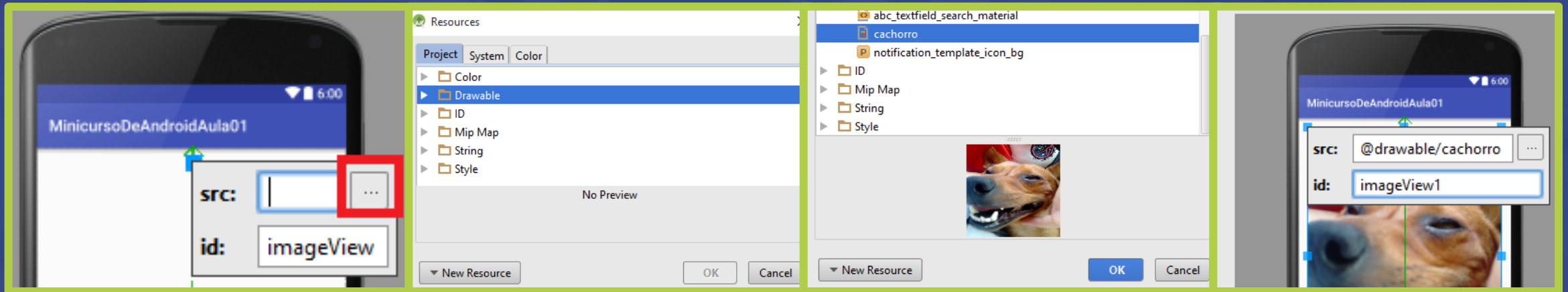


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

ImageViews

- Ao clicar duas vezes em uma **ImageView**, você verá duas opções:
- **Src**: É o local onde está a imagem e **Id**: É o identificador do campo.



CLIQUE E CONHEÇA O
CURSO COMPLETO

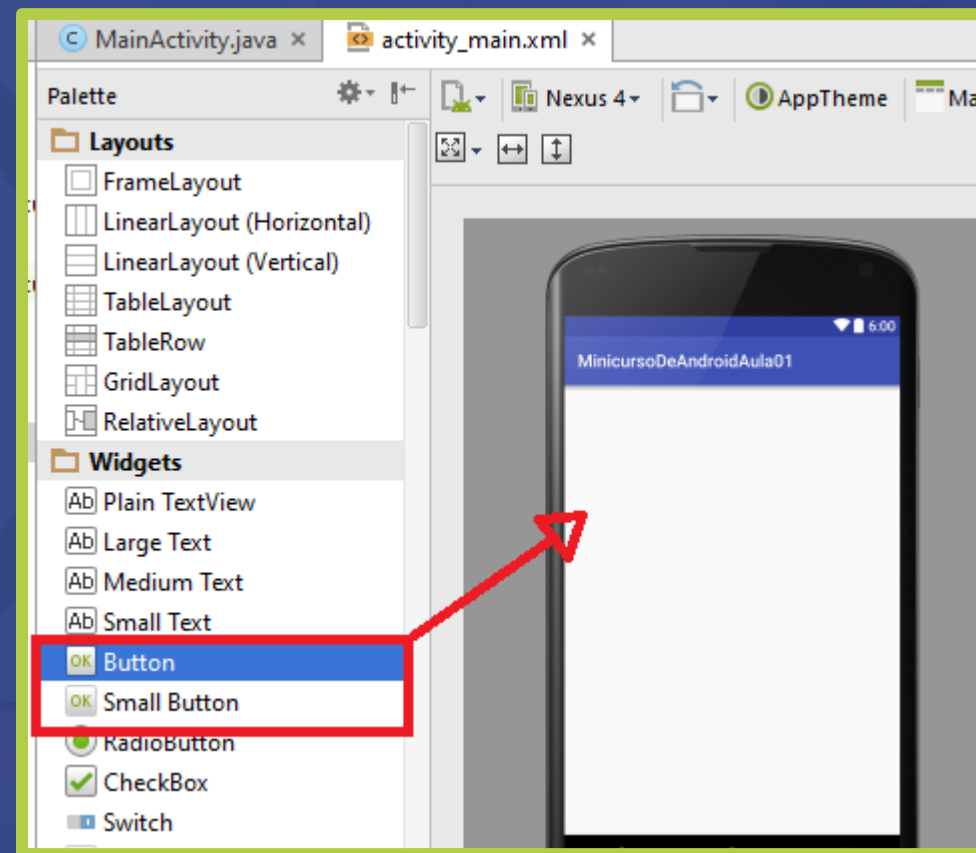
onedaycode.com



Buttons

Buttons

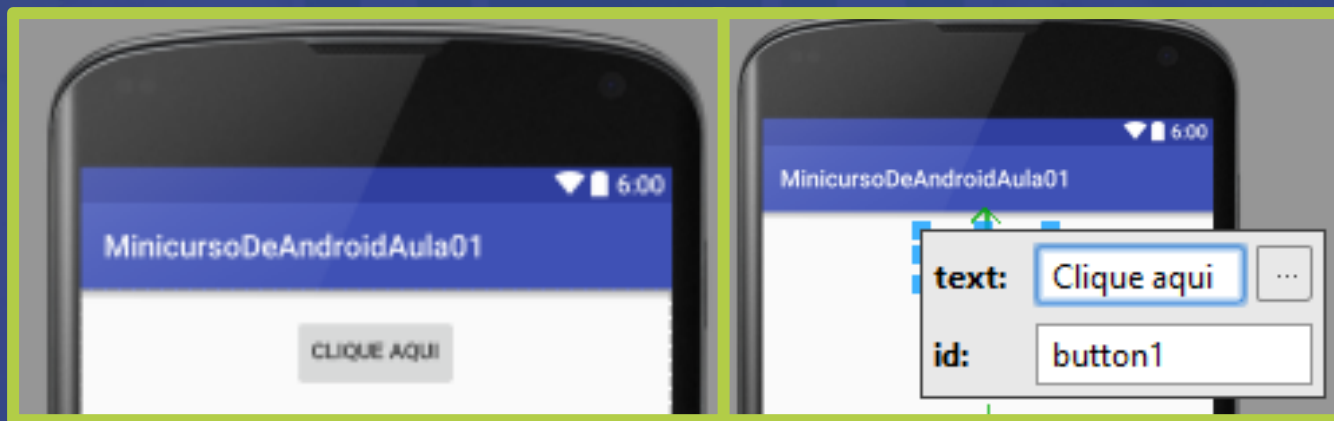
- **Buttons** são campos clicáveis.
- Para adicionar um em uma tela, abra o layout/activity_main.xml.
- Selecione um tipo de botão e arraste-o para o celular.



CLIQUE E CONHEÇA O
CURSO COMPLETO

Buttons

- Ao clicar duas vezes em um **Button**, você verá duas opções:
- **Text**: É o texto que será exibido inicialmente e **Id**: É o identificador do campo.

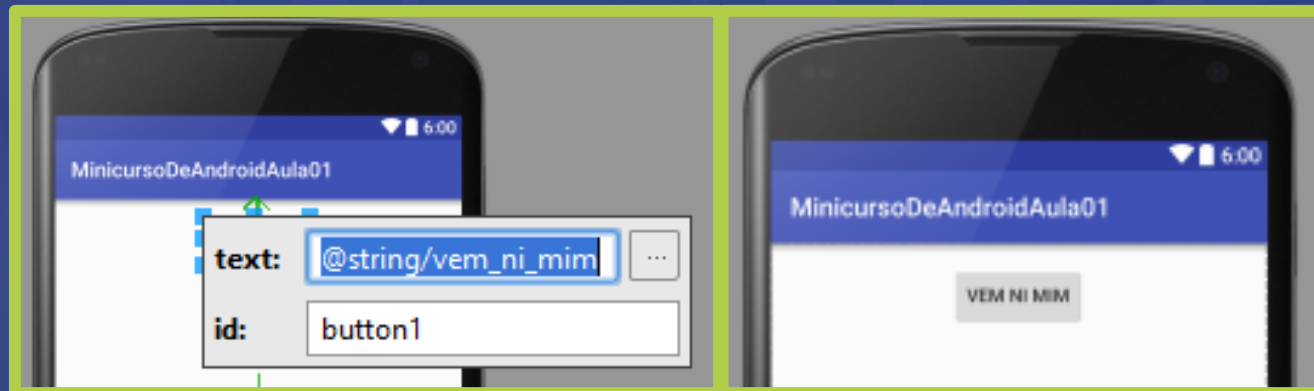


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Buttons com String Pré-Definida

- Para passar um valor pré-definido em strings.xml para uma **TextView**,
- Utilize o código: **@string/nome_da_string** .



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com



Funções

Funções

- As funções em Java são chamadas de **Métodos**, servem para agilizar e organizar os códigos.
- Para criar uma função, precisamos de 4 ingredientes:



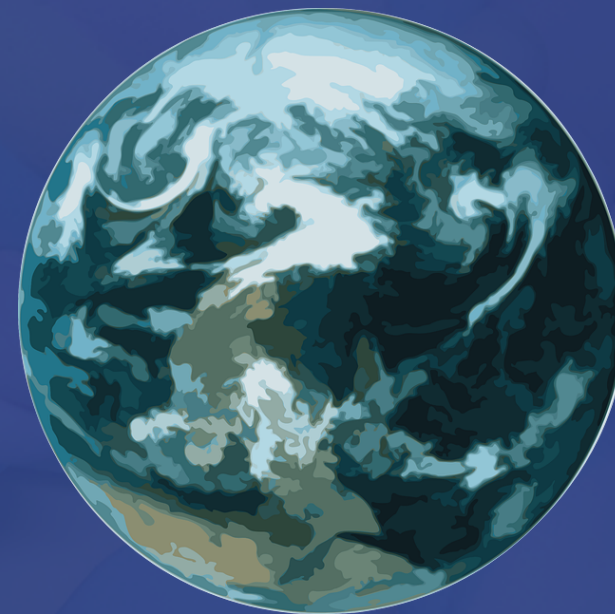
Disponibilidade + Tipo de Valor a ser Retornado + Nome + (Opcional: Parâmetros)

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Disponibilidade de Funções

- **public** : pode ser chamada em qualquer parte do projeto.
- **protected** : pode ser chamada apenas em sua própria classe ou classes que tenham extend nessa classe que está sendo criada.
- **private**: função que só pode ser chamada dentro da própria classe.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Valor Retornado de Funções

Tipo	Retorno
<u>void</u>	Não retorna valor algum.
<u>boolean</u>	Retorna True ou False.
<u>int ou Integer</u>	Retorna um inteiro (Até 11 casas).
<u>double</u>	Retorna um número quebrado.
byte	Retorna um inteiro (Até 3 casas).
short	Retorna um inteiro (Até 5 casas).
float	Retorna um inteiro (Até 10 casas).

- *Os sublinhados são os mais utilizados.

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Criando uma Função Vazia

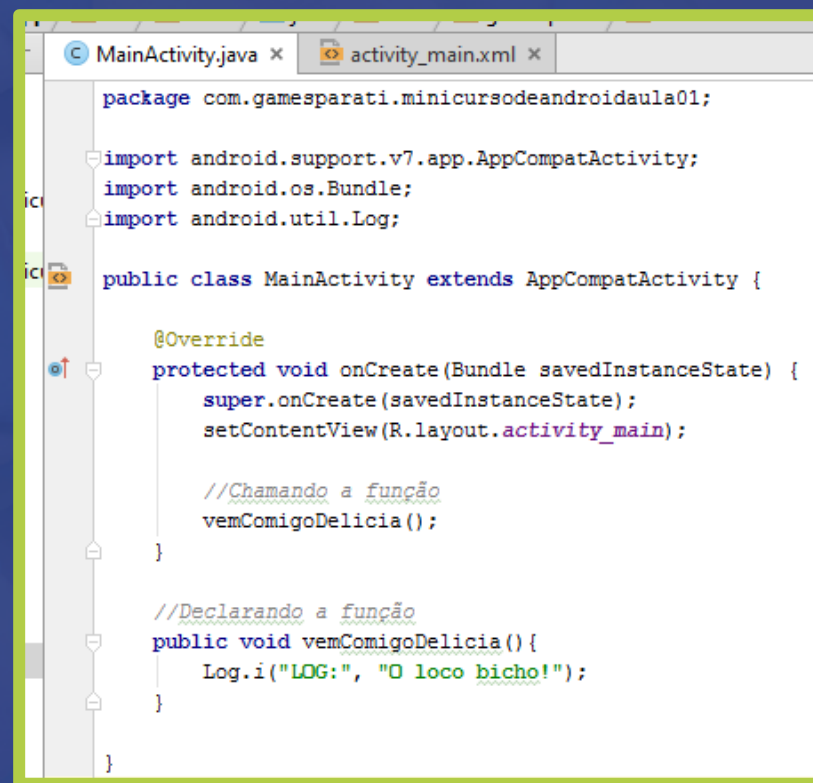
- O nome da **função** é formado por palavras juntas e sem acentos, a primeira é minúscula e as próximas tem iniciais maiúsculas.

- Exemplo de uma função pública, e que não retorna valores:

```
public void vemComigoDelicia(){  
    Log.i("LOG:", "O loco bicho! ");  
}
```

- Para chamar essa função, usaremos o código:

```
vemComigoDelicia();
```

A screenshot of an IDE window showing the MainActivity.java file. The code defines a public class MainActivity extending AppCompatActivity. It includes imports for AppCompatActivity, Bundle, and Log. The onCreate method is overridden, calling super.onCreate and setContentView. A comment indicates a function call: //Chamando a função vemComigoDelicia(). Another comment indicates a function declaration: //Declarando a função. The vemComigoDelicia() method is defined as public void, logging a message with Log.i.

```
package com.gamesparati.minicursodeandroidaula01;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Chamando a função  
        vemComigoDelicia();  
    }  
  
    //Declarando a função  
    public void vemComigoDelicia(){  
        Log.i("LOG:", "O loco bicho!");  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

Desafio

Crie uma função que exiba uma soma entre dois valores **Integer** e exiba o resultado no **Log.i** .
Dica: **String.valueOf(Variável)** converte para **String**

Slides usados:
Função , Definir Integer ,
Concatenado Variáveis ,
Operação Matemática ,
Exibir Log.i .



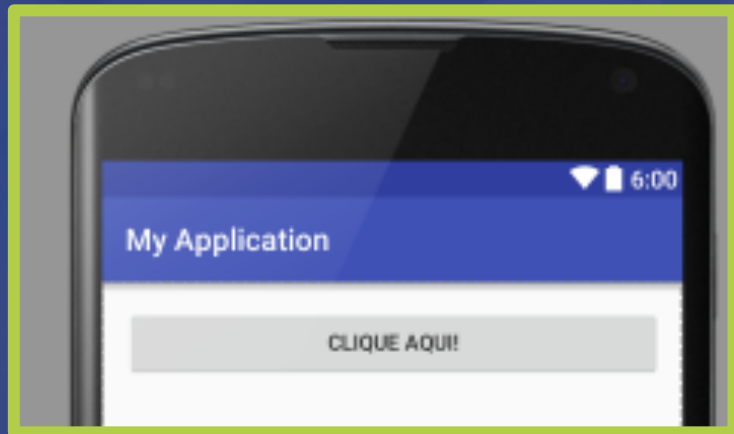
Resposta:
Exercício02



Chamar Função Com Botão

Chamando Função com Button

- Primeiramente, adicione um botão na tela (**Id** = button, **Text** = Clique aqui!)
- Agora vamos definir e referenciar o botão no código, na classe MainActivity.java :



```
package br.devfestsul.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    Button botao;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Referenciando o Botão
        botao = (Button) findViewById(R.id.button);
    }
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Chamando Função com Button

- Agora digite o “**nomeDoBotãoCriado.setOnClickListener(new On)**”.
- O Andoid Studio irá oferecer o autocomplemento do **OnClickListener(...)**.
- Selecione essa opção e ele irá gerar um código.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Referenciando o botão
    Button cliqueAquiButton = (Button) findViewById(R.id.button);

    //Colocando um Listener no botão
    cliqueAquiButton.setOnClickListener(new On);
}
```

OnClickListener {...} (android.view.View)
QuickContactBadge (android.widget)

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Chamando Função com Button

- Dentro dessas novas linhas de código, onde está indicado na imagem, ficarão **todas as ações** que ocorrerão quando o usuário clicar nesse **botão** que referenciamos.

```
public class MainActivity extends AppCompatActivity {  
  
    TextView campoDeTexto;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando o botão  
        Button cliqueAquiButton = (Button) findViewById(R.id.button);  
  
        //Colocando um Listener no botão  
        cliqueAquiButton.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                //Aqui ficarão os códigos chamados após clicar no botão  
            }  
        });  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Chamando Função com Button

- Agora na classe MainActivity.java vamos criar a **função** com o nome de funcao.
- Exemplo de função vazia que imprime texto no LogCat:

```
public void funcao(View view){  
    Log.i("LOG", "Sassenhora, você é um mito!");  
}
```

```
public class MainActivity extends AppCompatActivity {  
  
    TextView campoDeTexto;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando o botão  
        Button cliqueAquiButton = (Button) findViewById(R.id.button);  
  
        //Colocando um Listener no botão  
        cliqueAquiButton.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                //Aqui ficarão os códigos chamados após clicar no botão  
                funcao();  
            }  
        });  
    }  
  
    public void funcao(){  
        Log.i("LOG:", "Sassenhora, você é um mito");  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com



Funções Integer e String

Função que Retorna String

- Exemplo de uma **função pública**, e que retorna uma string:
- **public string vemComigoString(){
 return "Estou emocionado, aluno.";**
}
- Como essa **função** retornando uma frase, podemos criar uma Variável para receber o resultado da **função**:
String texto = vemComigoString();

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Passando o valor da função para a variável texto  
        String texto = vemComigoString();  
    }  
  
    //Declarando a função  
    public String vemComigoString() {  
        return "Estou emocionado, aluno";  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

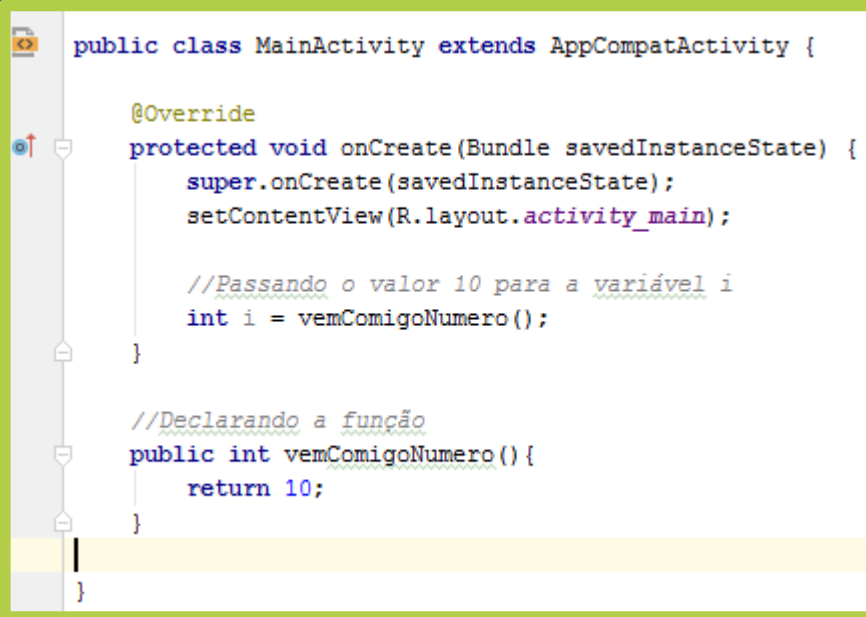
Função que retorna um Inteiro

- Exemplo de uma **função pública**, e que retorna um **inteiro**:

```
public int vemComigoNumero(){  
    return 10;  
}
```

- Como essa **função** retorna o valor 10, podemos criar uma **Variável** para receber o resultado da **função**:

```
int i = vemComigoNumero();
```



```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Passando o valor 10 para a variável i  
        int i = vemComigoNumero();  
    }  
  
    //Declarando a função  
    public int vemComigoNumero(){  
        return 10;  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie uma função que retorne uma **String** e outra um **Integer**.
Crie uma variável que recebe o valor da função **String** e outra da função **Integer**.
Exiba a **String** com o **Integer** em um **Log.i**.

Slides usados:
Definir Integer,
String e Concatenar ,
Função String , Função integer
e Exibir Log.i .



Resposta:
Exercicio03



TextViews, ImageViews, Buttons, EditTexts com Códigos

Definindo Componentes - Importante

- Para podermos acessar os Componentes como:
- **TextView**, **Button**, **EditText** dentro de funções separadas no código, sempre seguiremos a ordem
- Primeiro: Define.
- Segundo: Referencia.
- Terceiro: Utiliza.

```
public class MainActivity extends AppCompatActivity {  
  
    Button botao1, botao2;  
    TextView texto1, texto2;  
    EditText campo1, campo2;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando TextViews com Código

- Crie um **TextView** com o Id textView no layout activity_main.xml, vamos defini-lo com o código:
TextView campoDeTexto; logo após o início da classe.
- Depois vamos referenciá-lo com o código:
campoDeTexto = (TextView) findViewById(R.id.textView);
- Exemplo:

```
public class MainActivity extends AppCompatActivity {  
  
    TextView campoDeTexto;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        campoDeTexto = (TextView) findViewById(R.id.textView);  
  
    }  
}
```

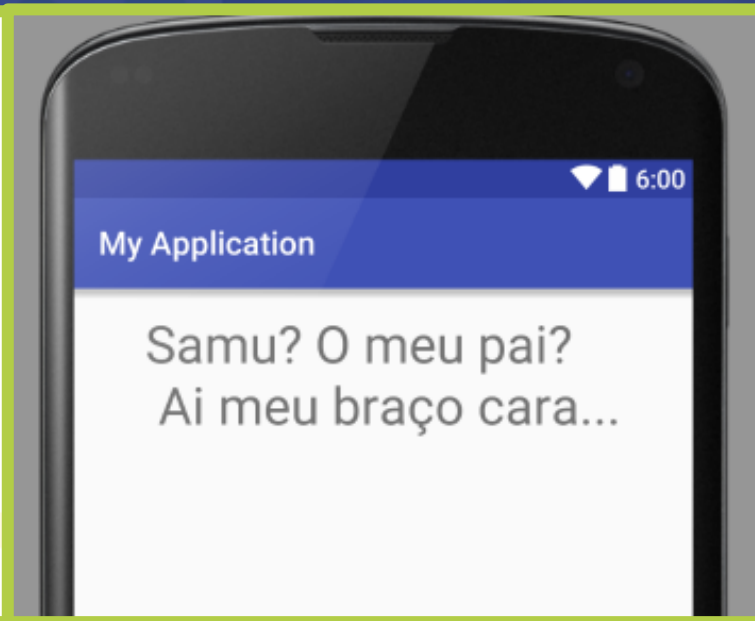
CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando TextViews com Código

- Agora utilizamos o **TextView** referenciado, com a propriedade **setText("Mensagem")** para alterar seu conteúdo. Para pular linhas, usa-se `\n` . *O resultado só será visto ao dar Play.*
- Exemplo: `campoDeTexto.setText("Samu? O meu pai? \n Ai meu braço cara...");`

```
public class MainActivity extends AppCompatActivity {  
  
    TextView campoDeTexto;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        campoDeTexto = (TextView) findViewById(R.id.textview);  
  
        //Modificando o texto:  
        campoDeTexto.setText("Samu? O meu pai? \n Ai meu braço cara...");  
    }  
}
```



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie um projeto que troca o valor de uma **TextView**, quando você clicar em um **Button**.

Slides usados:

Definindo Componentes ,
Modificar valor de TextView,
Função Chamada por Button.



Resposta:
Exercicio04

Alterando ImageViews com Código

- Crie uma **ImageView** com o Id imageView no layout activity_main.xml,
- Vamos defini-la com o código: **ImageView imagem;**
- Referenciamos com o código: **imagem = (ImageView) findViewById(R.id.imageView);**
- No exemplo abaixo, perceba que será preciso importar o pacote, use ALT+ENTER .

```
public class MainActivity extends AppCompatActivity {  
  
    ImageView imagem;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        ? android.widget.ImageView? Alt+Enter  
        //Referenciando os componentes  
        imagem = (ImageView) findViewById(R.id.imageView);  
    }  
}
```

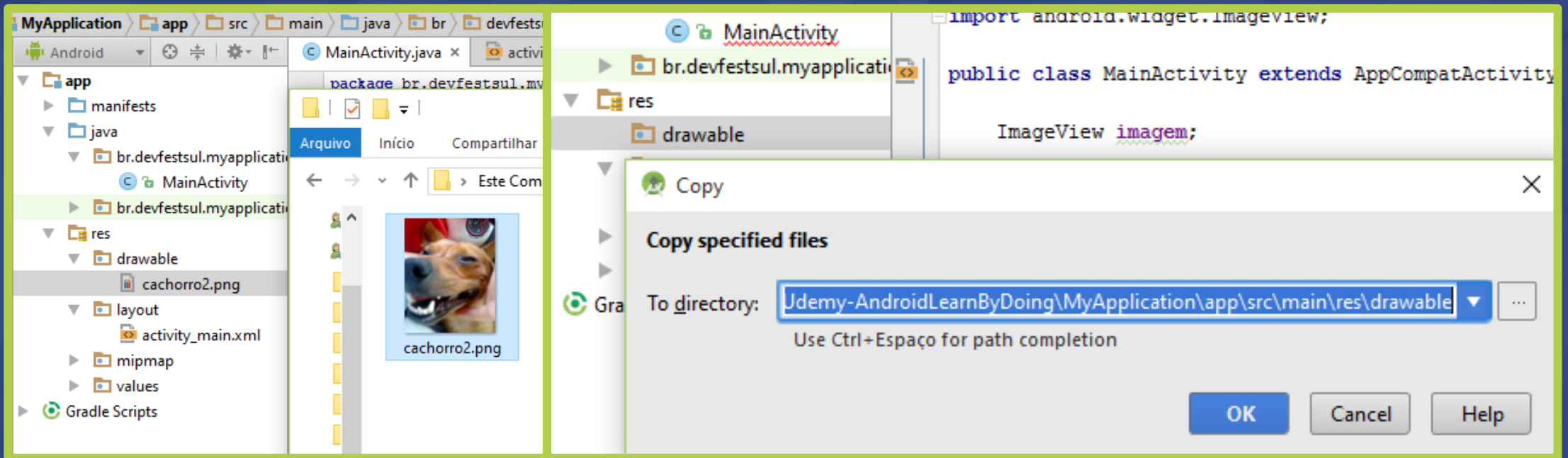
```
import android.widget.ImageView;  
  
public class MainActivity extends AppCompatActivity {  
  
    ImageView imagem;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        imagem = (ImageView) findViewById(R.id.imageView);  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando ImageViews com Código

- Agora adicionaremos a imagem de um cachorro engraçado no projeto, utilizando CTRL+C, CTRL+V. Então selecione o “OK”.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando ImageViews com Código

- Agora utilizamos o **ImageView** referenciado com a propriedade: **setBackgroundResource(R.drawable.imagem_nova);** para alterar seu conteúdo.
- Exemplo: **imagem.setBackgroundResource(R.drawable.cachorro2);**

```
public class MainActivity extends AppCompatActivity {  
  
    ImageView imagem;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        imagem = (ImageView) findViewById(R.id.imageView);  
  
        //Altera a imagem  
        imagem.setBackgroundResource(R.drawable.cachorro2);  
    }  
}
```

- Obs.: O resultado só será visto ao dar Play.

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie um projeto que troca o valor de uma **ImageView**, quando você clicar em um **Button**.

Slides usados:
Definir Componente ,
Modificar valor de ImageView,
Função Chamada por Button.



Resposta:
Exercicio05

Animando ImageViews com Código

- Agora utilizamos o **ImageView** referenciado, com as propriedades :
- Para diminuir a escala da imagem para 50%:
 - **`imagem.animate().scaleX(0.5f).scaleY(0.5f).setDuration(2000);`**
- Para rotacionar a imagem em 1800 graus em 2 segundos:
 - **`imagem.animate().rotation(1800f).setDuration(2000);`**
- *Obs.: O resultado só será visto ao dar Play.*

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    public void animate(View view) {  
        ImageView imagem1 = (ImageView) findViewById(R.id.imageView1);  
  
        //Animação de escala  
        imagem1.animate().scaleX(0.5f).scaleY(0.5f).setDuration(2000);  
  
        //Animação de rotação  
        imagem1.animate().rotation(1800f).setDuration(2000);  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando Buttons com Código

- Crie um **Button** com o **id** button no layout activity_main.xml .
- Vamos defini-lo com o código: **Button botao;**
- Depois vamos referenciá-lo com o código: **Botao = (Button) findViewById(R.id.button);**

```
public class MainActivity extends AppCompatActivity {  
  
    Button botao;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        ? android.widget.Button? Alt+Enter  
        //Referenciando os componentes  
        botao = (Button) findViewById(R.id.button);  
    }  
}
```

```
public class MainActivity extends AppCompatActivity {  
  
    Button botao;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        botao = (Button) findViewById(R.id.button);  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Alterando Buttons com Código

- Agora utilizamos o **Button** referenciado, com a propriedade **setText("Mensagem")** para alterar seu conteúdo. Para pular linhas, usa-se **\n**.
- Exemplo: **botao.setText("Bom dia meu Bebê \n Te amo meu Bebê");**

```
public class MainActivity extends AppCompatActivity {  
  
    Button botao;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        botao = (Button) findViewById(R.id.button);  
  
        //Altera o texto  
        botao.setText("Bom dia meu Bebê \n Te amo meu Bebê");  
    }  
}
```



- O resultado só será visto ao dar Play.

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

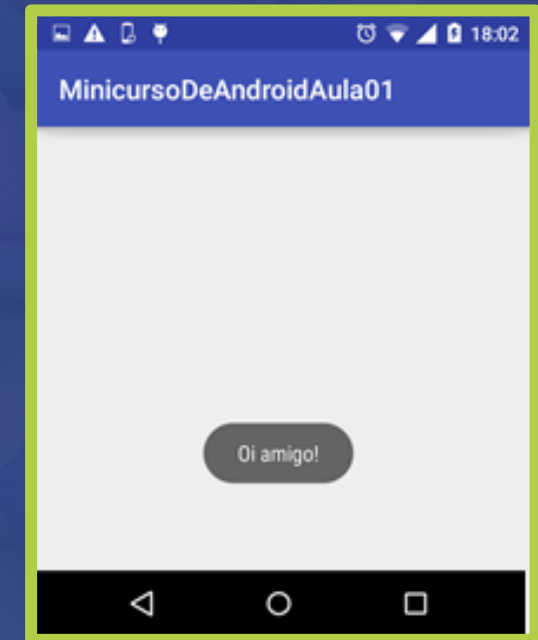


Alertas na Tela Toast

Toasts (Alertas na Tela)

- Exemplo: `Toast.makeText(Contexto, "Mensagem", Toast.Duração).show();`
- Contexto: para facilitar, vamos utilizar `getApplicationContext()` que automaticamente encontra-o.
- Mensagem: conteúdo a ser mostrado.
- Duração: `LENGTH_LONG` (Longo) e `LENGTH_SHORT` (Curto).

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Exibindo uma mensagem na tela  
        Toast.makeText(getApplicationContext(), "Oi amigo!", Toast.LENGTH_LONG).show();  
    }  
}
```



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie uma função que alerta um **Toast**.
Essa função tem que ser chamada por um **Button**.

Slides usados:
Exibir Toast,
Função Chamada por Button.



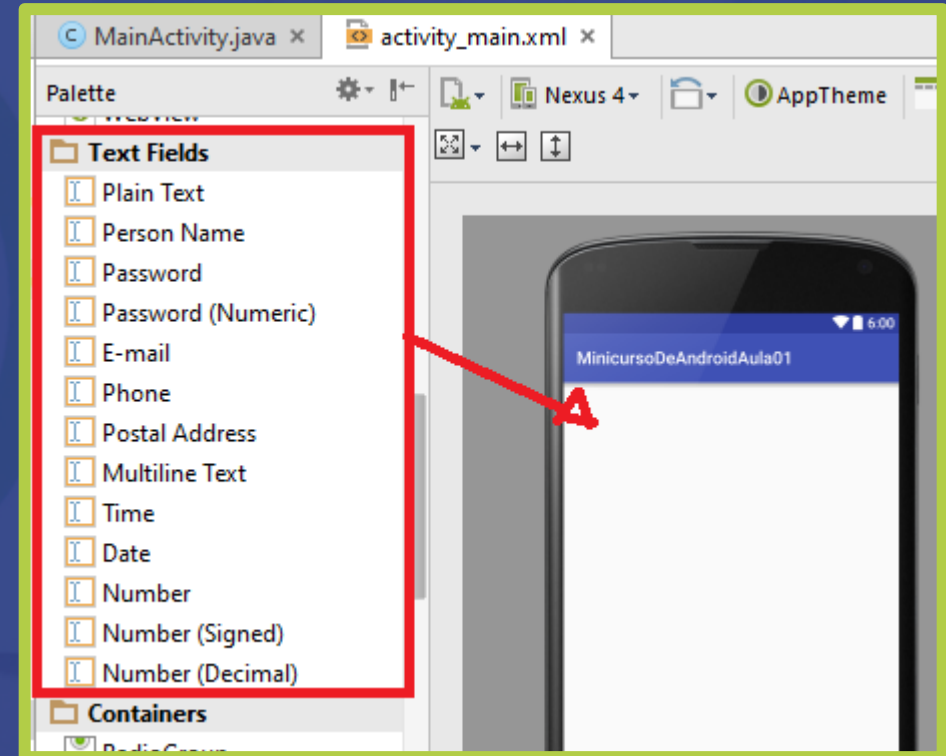
Resposta:
Exercicio06



EditTexts

EditTexts

- **EditTexts** são campos de inserção de texto.
- Para adicionar um em uma tela, abra o `layout/activity_main.xml`.
- Selecione um tipo de campo de texto e arraste-o para o celular.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

EditTexts

- Ao clicar duas vezes em um **EditView**, você verá duas opções:
- **Text**: É o texto que será exibido inicialmente e **Id**: É o identificador do campo.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

EditTexts Hints

- **Hints** são dicas para o usuário saber o que inserir no campo.
- Para acessar essas propriedades, selecione o campo e use CONTROL+B.
- Nessa nova tela, adicione a propriedade **android:hint = "Dica"**.



```
MainActivity.java x activity_main.xml x
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText1"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:text="10" />

</RelativeLayout>
```

```
MainActivity.java x activity_main.xml x
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <EditText
        android:hint="Digite um valor aqui."
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText1"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:text="10" />

</RelativeLayout>
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

EditTexts Recebendo Valor Inserido

- Crie um **EditText** com o Id `editText` no layout `activity_main.xml`,
- Vamos defini-lo com o código: **`EditText campo;`**
- Vamos referenciá-lo com o código: **`campo = (EditText) findViewById(R.id.editText);`**
- Exemplo:

```
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    EditText campo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Referenciando os componentes
        campo = (EditText) findViewById(R.id.editText);
    }
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

String Recebendo Valor de EditTexts

- Agora utilizamos o **EditText** referenciado, com a propriedade **getText()** para receber o conteúdo.
- Exemplo: **campo.getText();**
- Agora que sabemos como receber o valor, podemos passar ele para uma variável se convertermos o campo antes com **.toString()** com o código:
String textoDoCampo= campo.getText().toString();

```
public class MainActivity extends AppCompatActivity {  
  
    EditText campo;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando os componentes  
        campo = (EditText) findViewById(R.id.editText);  
  
        //Variável que recebe o valor do campo  
        String textoDoCampo = campo.getText().toString();  
  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie uma função que alerta um **Toast** com um valor de um **EditText**. Essa função tem que ser chamada por um **Button**.

Slides usados:
Captando valor do EditText,
Exibir Toast,
Função Chamada por Button.



Resposta:
Exercicio07

EditTexts Recebendo Valor Numérico

- Agora utilizamos o **EditText** referenciado, com a propriedade **getText()** para receber o conteúdo.
- Exemplo: **campo1.getText();**
- Agora que recebemos o valor, podemos passar ele para uma variável se convertermos o campo antes com **Integer.parseInt()** com o código:

Integer valorCampo1Integer = Integer.parseInt(campo1.getText().toString());

```
public class MainActivity extends AppCompatActivity {  
  
    EditText campo1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando o Campo  
        campo1 = (EditText) findViewById(R.id.editText);  
  
        //Recebendo valor inserido e convertendo-o  
        Integer valorCampo1Integer = Integer.parseInt(campo1.getText().toString());  
    }  
}
```

**CLIQUE E CONHEÇA O
CURSO COMPLETO**

onedaycode.com

EditTexts Recebendo Valor Numérico

- **Cuidado!** É importante conferir se o valor que está vindo do **EditText** é **vazio**, senão o app irá dar erro e fechar.
- Então só use esse código se tiver certeza que no momento em que ele é chamado, o campo não está nulo.

```
public class MainActivity extends AppCompatActivity {  
  
    EditText camp01;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando o Campo  
        camp01 = (EditText) findViewById(R.id.editText);  
  
        //Recebendo valor inserido e convertendo-o  
        Integer valorCamp01Integer = Integer.parseInt(camp01.getText().toString());  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

Desafio

Crie uma função que alerta um **Toast** com o valor da soma de dois **EditTexts**.
Essa função tem que ser chamada por um **Button**.

Slides usados:
Valor Numérico do EditText,
Operação de Soma,
Exibir Toast,
Função Chamada por Button.



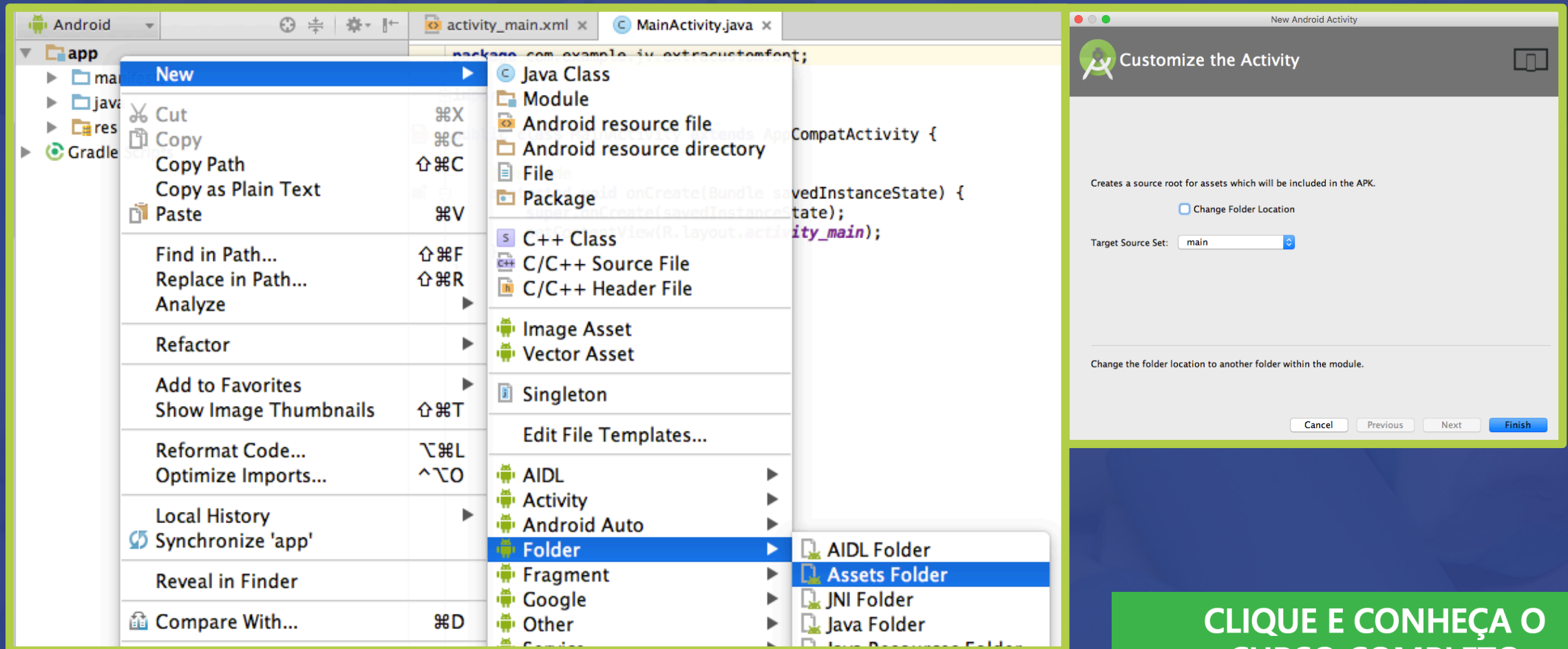
Resposta:
Exercicio08



Extra: TextViews com Fontes Próprias

TextViews com Fontes Próprias

- Primeiramente precisamos criar uma nova pasta de Assets em nosso projeto.

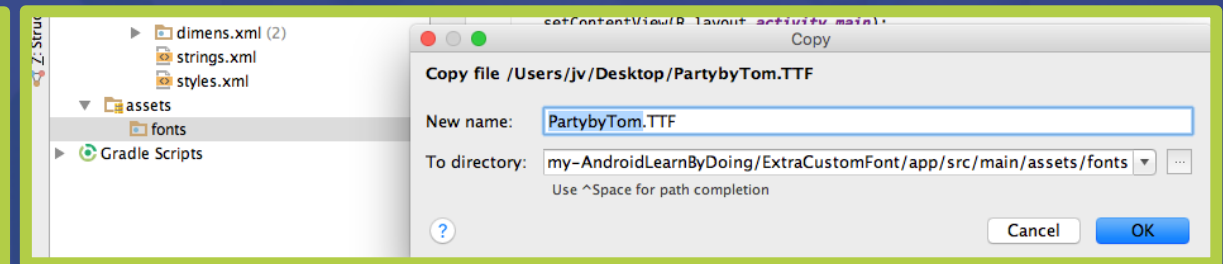
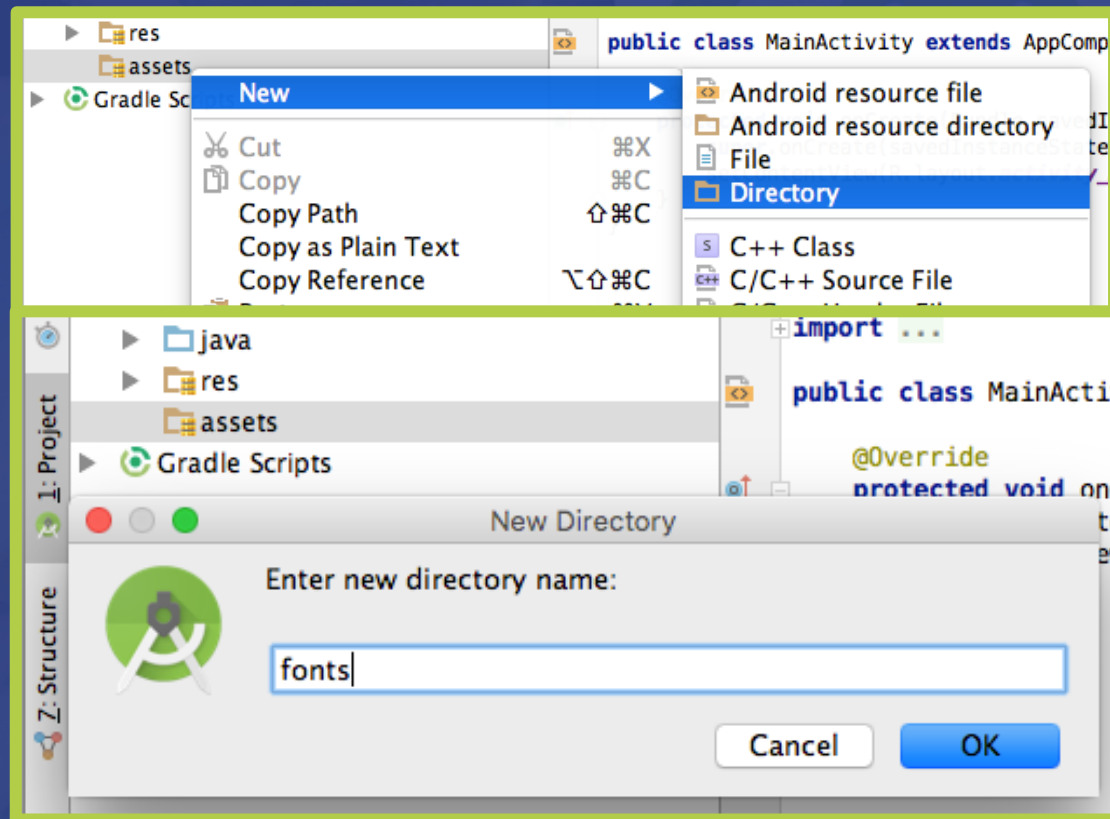


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

TextViews com Fontes Próprias

- Depois , criamos um **Directory** dentro de Assets com o nome de fonts e colamos uma fonte com a extensão .TTF.

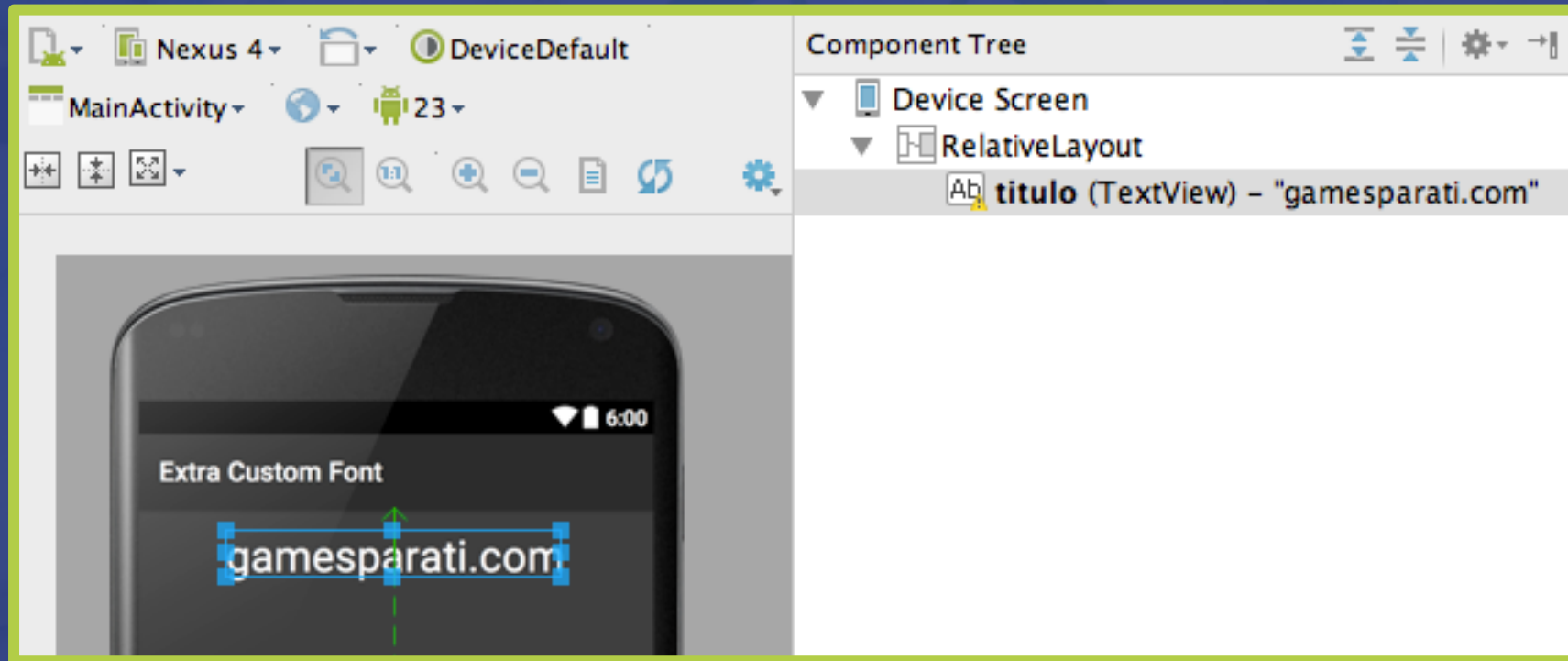


CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

TextViews com Fontes Próprias

- Então criamos uma **TextView** com o id de “titulo”.



CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

TextViews com Fontes Próprias

- Precisamos definir e referenciar nossa **TextView**.

```
public class MainActivity extends AppCompatActivity {  
  
    //Definindo Componentes  
    TextView titulo;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Referenciando Componentes  
        titulo = (TextView) findViewById(R.id.titulo);  
    }  
}
```

CLIQUE E CONHEÇA O
CURSO COMPLETO

onedaycode.com

TextViews com Fontes Próprias

- Para criar a fonte no código usamos:
- `Typeface nomeDaFonte = Typeface.createFromAsset(getAssets(), "fonts/nomedafonte.TTF");`



- Para passar a fonte para a TextView usamos:
- `nomeDaTextView.setTypeface(nomeDaFonte);`

CLIQUE E CONHEÇA O
CURSO COMPLETO

Desafio

Crie um projeto que tenha uma **TextView** com uma fonte própria.

Slides usados:

TextView com Fontes Próprias.



Resposta:
ExtraCustomFont



**Parabéns por chegar até aqui!
Muito obrigado!**



**Parabéns por chegar até aqui!
Muito obrigado!**