



# آموزش برنامه نویسی اندروید در محیط اندروید استودیو

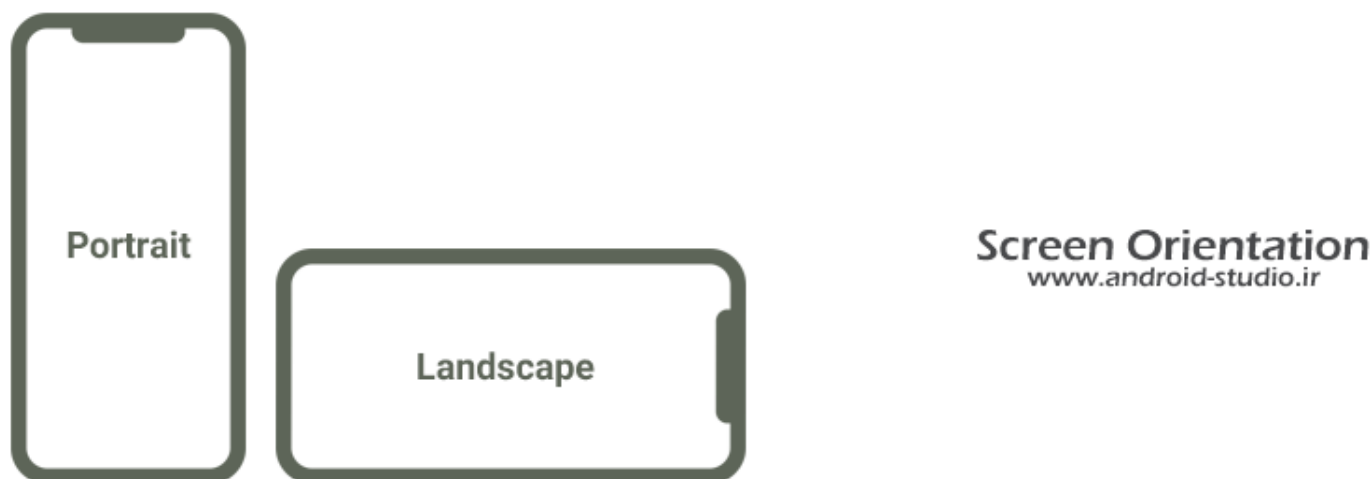
تشخیص جهت (Orientation) صفحه نمایش دستگاه اندرویدی

مدرس : سیدمهدی مطهری

[www.android-studio.ir](http://www.android-studio.ir)



## به نام خدا



در این جلسه از سری مباحث **آموزش برنامه نویسی اندروید** قصد داریم دو روش بدست آوردن جهت یا Orientation صفحه نمایش دستگاه‌های اندرویدی را بررسی کنیم. به این ترتیب می‌توانیم تشخیص دهیم صفحه نمایش در حالت عمودی (Portrait) قرار گرفته یا افقی (Landscape).

### تشخیص جهت صفحه نمایش در اندروید

در روند توسعه یک اپلیکیشن اندرویدی ممکن است در برخی مواقع لازم باشد عمل یا اعمالی بر اساس نحوه قرارگیری دستگاه انجام شوند. این عمل و تغییرات می‌تواند در رابط کاربری بوده و یا اینکه مربوط به عملکرد اصلی برنامه ما باشد.

برای تشخیص جهت قرارگیری صفحه نمایش به دو روش می‌توان عمل کرد که در ادامه و در قالب یک پروژه، هر دو روش را بررسی می‌کنیم. مطابق مبحث **آموزش ساخت پروژه در اندروید استودیو** یک پروژه اندرویدی با نام Orientation می‌سازم. اکتیویتی را از نوع Empty Activity و زبان را Java انتخاب کردم.

ابتدا دو TextView در layout **اکتیویتی** تعریف می‌کنم تا خروجی هر دو روش را نمایش دهد:



## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txt_one"
        android:textSize="30dp"
        android:text="" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txt_two"
        android:textSize="30dp"
        android:text=""
        android:layout_below="@id/txt_one" />

</RelativeLayout>
```

## روش اول تشخیص جهت صفحه نمایش

در این روش با استفاده از کلاس Configuration وضعیت فعلی صفحه نمایش را بررسی کرده و با تعریف یک شرط، خروجی مناسب را دریافت می‌کنیم.

## MainActivity.java

```
package ir.android_studio.orientation;

import androidx.appcompat.app.AppCompatActivity;

import android.content.res.Configuration;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView txtOne, txtTwo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```



```
setContentView(R.layout.activity_main);

txtOne = findViewById(R.id.txt_one);
txtTwo = findViewById(R.id.txt_two);

int screenOrientation = this.getResources().getConfiguration().orientation;

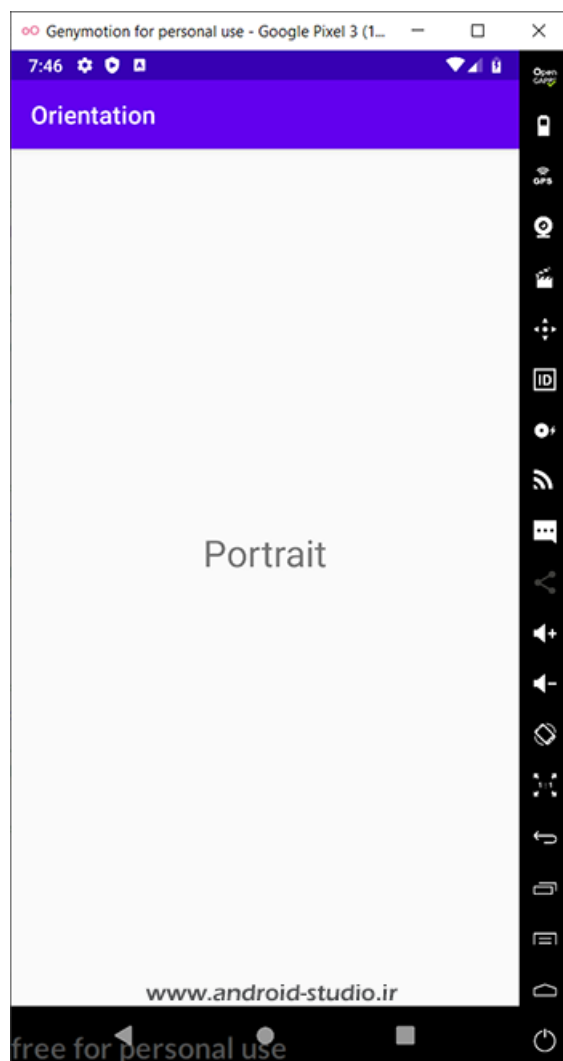
if (screenOrientation == Configuration.ORIENTATION_LANDSCAPE) {
    txtOne.setText("Landscape");
} else {
    txtOne.setText("Portrait");
}
}
```

یک متغیر با نام دلخواه screenOrientation و از جنس int تعریف کردم که orientation فعلی صفحه را در خود ذخیره می‌کند:

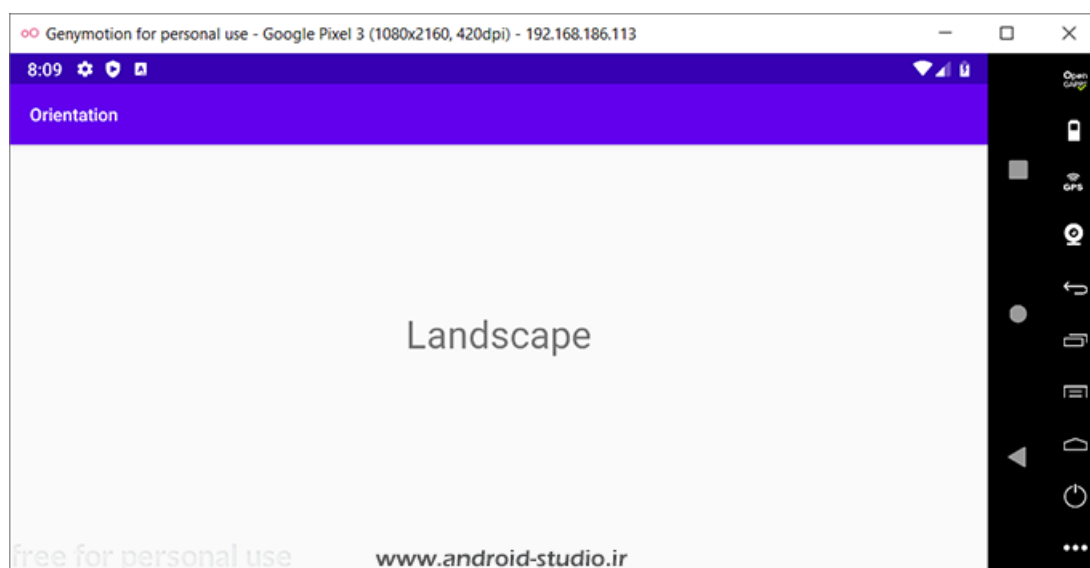
```
int screenOrientation = this.getResources().getConfiguration().orientation;
```

عدد ذخیره شده در متغیر فوق، عدد ۱ یا ۲ خواهد بود که عدد ۱ نشانه وضعیت Portrait و عدد ۲ نشانه وضعیت Landscape است. سپس یک شرط تعریف شده که بررسی می‌کند چنانچه مقدار ذخیره شده در این متغیر با Configuration.ORIENTATION\_LANDSCAPE برابر باشد (به عبارتی عدد ۲ ذخیره شده باشد) قسمت اول شرط و در غیر اینصورت قسمت دوم شرط اجرا شود. بدیهی است چنانچه از Configuration.ORIENTATION\_PORTRAIT استفاده می‌کردم خروجی شرط بلعکس می‌شد. کانتکست را this قرار داده‌ام.

پروژه را اجرا می‌کنم:



در ابتدای اجرای شبیه ساز Genymotion صفحه نمایش در حالت عمودی قرار دارد و خروجی شرط که در TextView نمایش داده شده با وضعیت دستگاه همخوانی داشته و عبارت Portrait را نمایش می‌دهد. حالا دستگاه را در حالت افقی قرار می‌دهم:

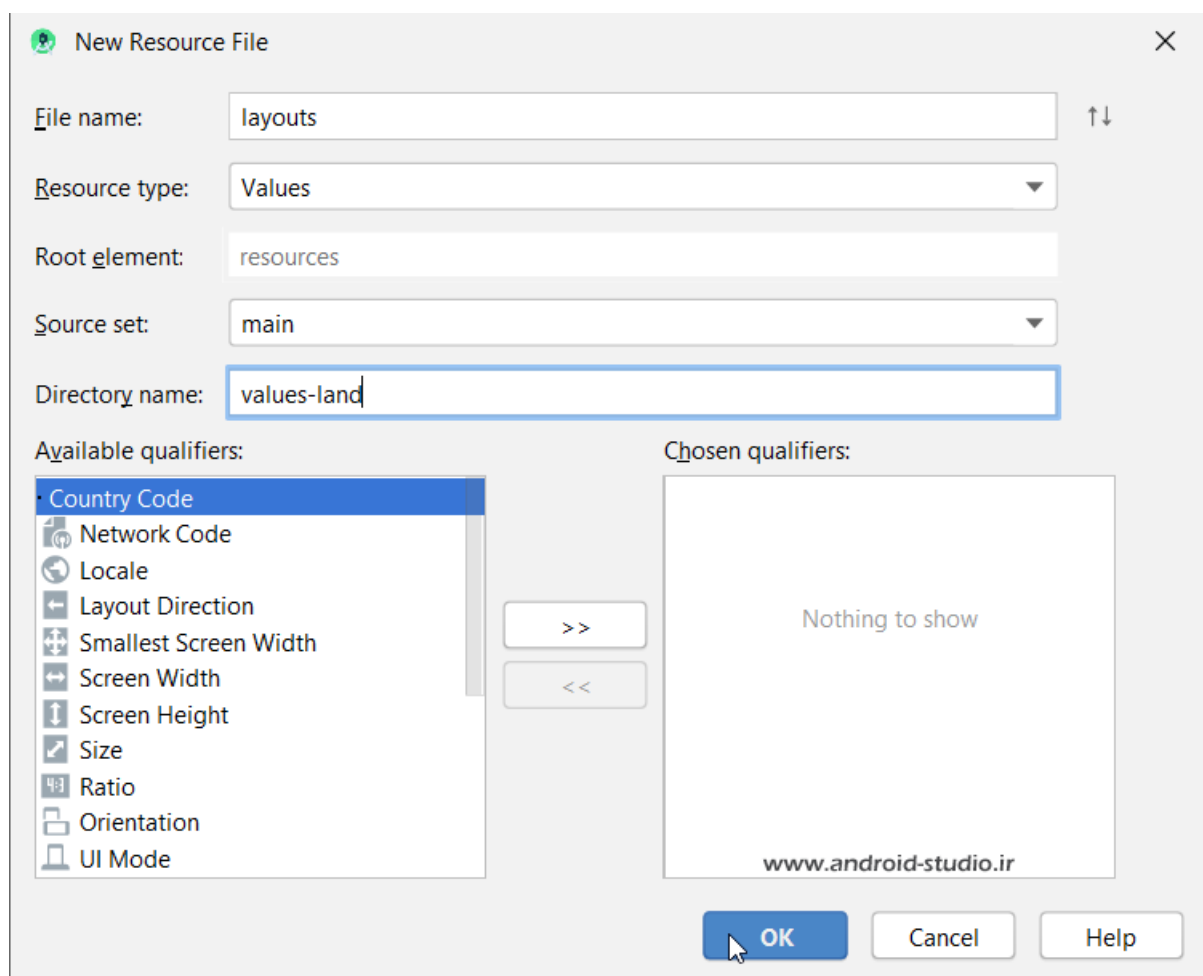




مشاهده می‌کنید با تغییر وضعیت صفحه نمایش و قرار گرفتن در حالت افقی یا Landscape و restart شدن اکتیویتی، قسمت دوم شرط اجرا شده است.

## روش دوم تشخیص جهت صفحه نمایش

در این قسمت از resource پروژه برای تشخیص وضعیت صفحه نمایش استفاده می‌کنیم. روی فولدر res پروژه راست کلیک کرده و Android Resource File را انتخاب می‌کنم:



نام فایل را layouts وارد کرده و در قسمت Directory name مطابق تصویر فوق، values-land می‌نویسم. از آنجایی که نام این دایرکتوری به land ختم شده، محتویات داخل آن در صورتی در دسترس خواهد بود که وضعیت صفحه نمایش در حالت Landscape قرار گرفته باشد. پس از ایجاد فایل layouts.xml آنرا به صورت زیر تکمیل می‌کنم:



## values-land/layouts.xml

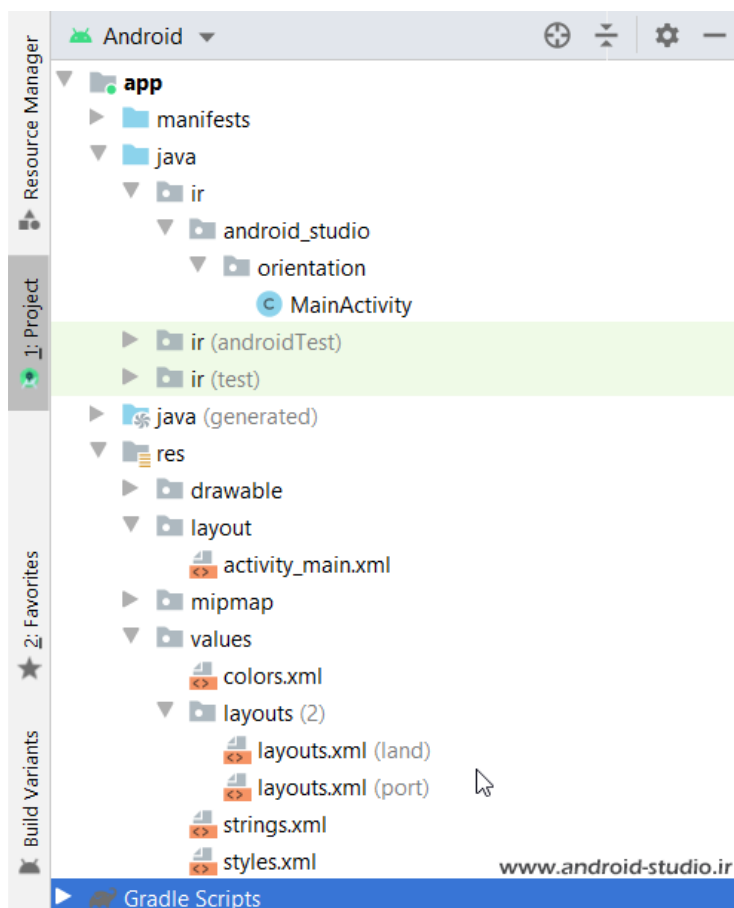
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <bool name="is_landscape">true</bool>
</resources>
```

در این فایل یک ویژگی از جنس bool (مخفف boolean) تعریف کردم با نام دلخواه is\_landscape و مقدار true. مجدد یک فایل دیگر به همین شیوه و با همان نام layouts.xml اما اینبار در دایرکتوری values-port ایجاد می‌کنم:

## values-port/layouts.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <bool name="is_landscape">false</bool>
</resources>
```

در این فایل هم یک ویژگی با نام is\_landscape تعریف کردم با این تفاوت که مقدار آن false است. این دو فایل در محیط پروژه به اینصورت نمایش داده می‌شود:





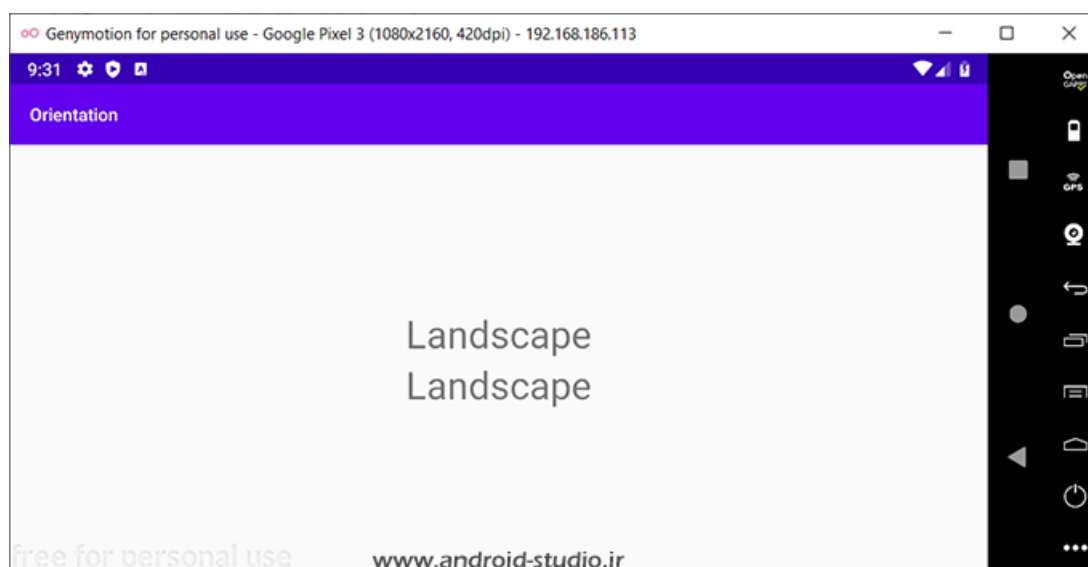
حالا در اکتیویتی یک متغیر از جنس boolean تعریف می‌کنم:

```
boolean isLandscape = this.getResources().getBoolean(R.bool.is_Landscape);
```

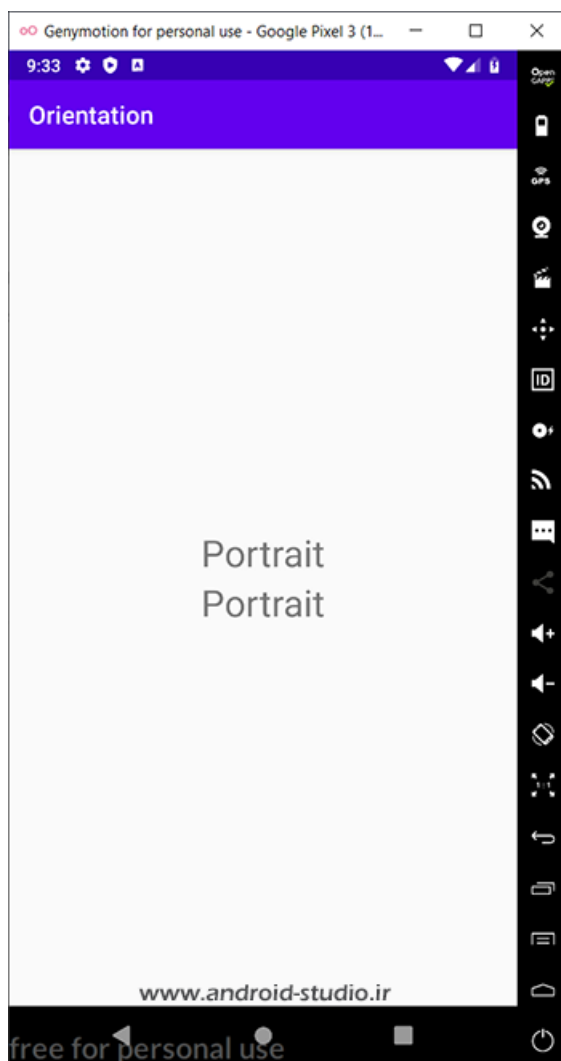
توسط کد فوق، مقدار ویژگی bool با نام is\_landscape از layouts.xml ای که با توجه به وضعیت صفحه نمایش در دسترس است، در متغیر isLandscape ذخیره می‌شود. یعنی چنانچه صفحه نمایش در حالت افقی قرار داشته باشد، layouts.xml دایرکتوری values-land در دسترس خواهد بود در نتیجه مقدار true را برمی‌گرداند و همینطور برای حالت عمودی مقدار false. در نهایت شرطی را برای بررسی خروجی این متغیر مانند قسمت قبل تعریف می‌کنم:

```
if (isLandscape) {  
    txtTwo.setText("Landscape");  
} else {  
    txtTwo.setText("Portrait");  
}
```

مجدد پروژه را اجرا می‌کنم:







ملاحظه می‌کنید خروجی هردو روش بر اساس افقی یا عمودی بودن دستگاه بروز می‌شود.  
کد کامل اکتیویتی:

### MainActivity.java

```
package ir.android_studio.orientation;

import androidx.appcompat.app.AppCompatActivity;

import android.content.res.Configuration;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView txtOne, txtTwo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```



```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

txtOne = findViewById(R.id.txt_one);
txtTwo = findViewById(R.id.txt_two);

int screenOrientation = this.getResources().getConfiguration().orientation;

if (screenOrientation == Configuration.ORIENTATION_LANDSCAPE) {
    txtOne.setText("Landscape");
} else {
    txtOne.setText("Portrait");
}

boolean isLandscape = this.getResources().getBoolean(R.bool.is_Landscape);

if (isLandscape) {
    txtTwo.setText("Landscape");
} else {
    txtTwo.setText("Portrait");
}
}
```

موفق و پیروز باشید.

مطالعه بیشتر:

<https://developer.android.com/reference/android/content/res/Configuration>

**توجه:** سورس پروژه درون پوشه Exercises قرار دارد

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش‌های بهتر یاری فرمائید.  
این فایل رایگان بوده و انتشار آن (بدون دخل و تصرف) مانعی ندارد.

[www.android-studio.ir](http://www.android-studio.ir)