آموزش برنامه نویسی اندروید

در محیط اندروید استودیو

متریال دیزاین

بخش نهم: کار با AlertDialog

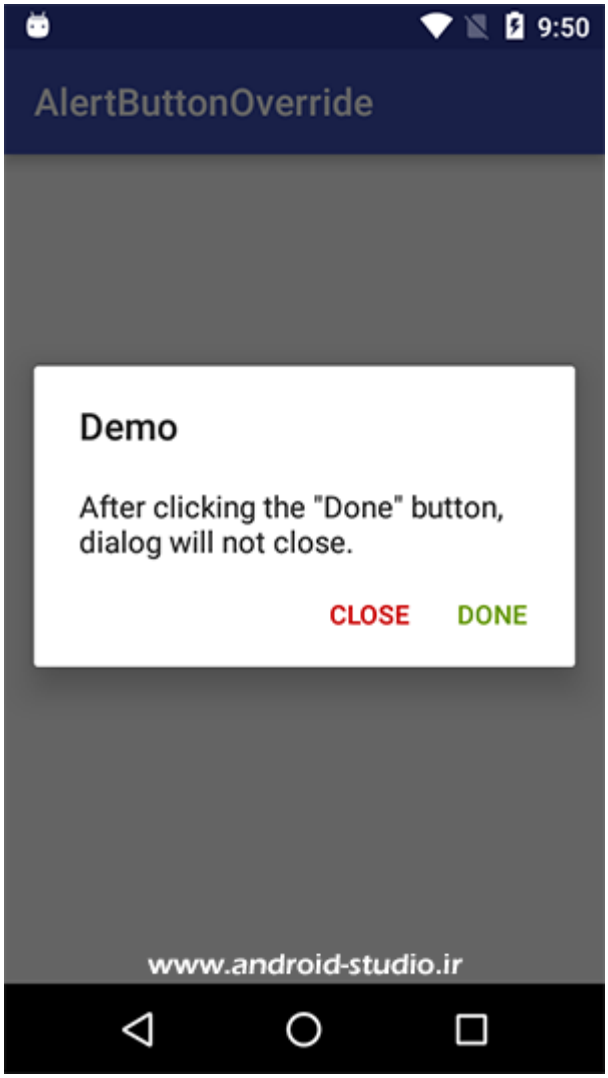
مدرس : سید مهدی مطهری

www.android-studio.ir

به نام خدا

معرفی AlertDialog:

AlertDialog پنجره کوچکی است که یک پیغام به همراه (حداکثر) سه دکمه به کاربر نمایش می دهد. این قابلیت کاربردهای متعددی دارد از جمله نمایش یک اخطار، درخواست تایید یا عدم تایید یک عملیات از کاربر، نمایش یک لیست و درخواست از کاربر جهت انتخاب یک گزینه و... .



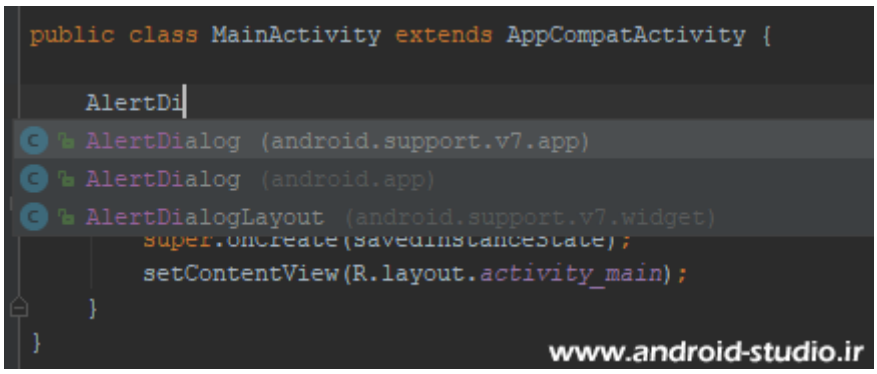
یک پروژه با نام AlertDialog و یک Empty Activity می سازم. پروژه باید حاوی کتابخانه appcompat-v7 باشد که به صورت پیش فرض هنگام ساخت پروژه اضافه شده است.

درون اکتیویتی یک نمونه از AlertDialog.Builder با نام دلخواه alertBuilder ایجاد کرده سپس درون متد onCreate بصورت زیر new می کنم.

```
package ir.android_studio.alertdialog;

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

public class MainActivity extends AppCompatActivity {

    AlertDialog.Builder alertBuilder;

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

        alertBuilder = new AlertDialog.Builder(this);

    }
}
```

AlertDialog.Builder فعلا فقط یک پارامتر ورودی گرفته که همان Context است.

دقت کنید AlertDialog مربوط به کتابخانه support را انتخاب نمائید:



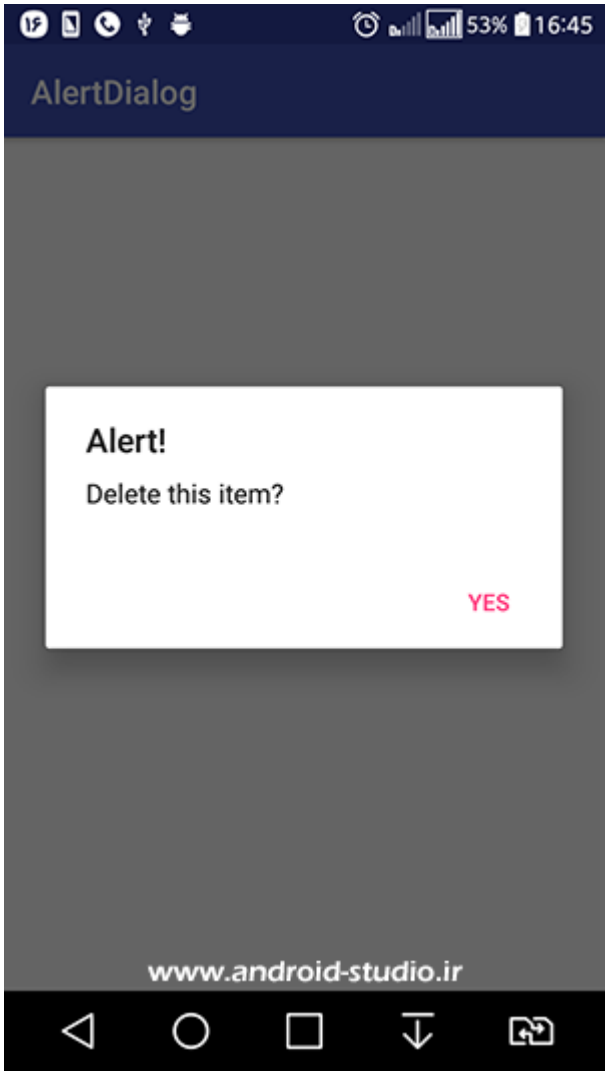
ابتدا یک AlertDialog ساده با حداقل متدها ایجاد کرده و در ادامه به ارائه جزئیات می‌پردازم:

```
package ir.android_studio.alertdialog;

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

public class MainActivity extends AppCompatActivity {

    AlertDialog.Builder alertBuilder;

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

        alertBuilder = new AlertDialog.Builder(this);

        alertBuilder.setTitle("Alert!");
        alertBuilder.setMessage("Delete this item?");

        alertBuilder.setPositiveButton("Yes", null);

        AlertDialog alert = alertBuilder.create();
        alert.show();

    }
}
```

همانطور که از نام متدهای setTitle و setMessage پیداست، این دو مورد برای تعیین عنوان و متن پیغام استفاده می شوند. در ابتدای مبحث گفتیم در AlertDialog تا **۳** دکمه می توان به دیالوگ اضافه کرد که به واسطه متدهای setPositiveButton، setNegativeButton و setNeutralButton قابل پیاده سازی هستند. در کد بالا فعلا فقط یک متد را بکار بردم. یعنی دیالوگ من باید یک دکمه با عنوان Yes داشته باشد. پارامتر دوم مربوط به Listener دکمه است که فعلا null قرار داده ام.

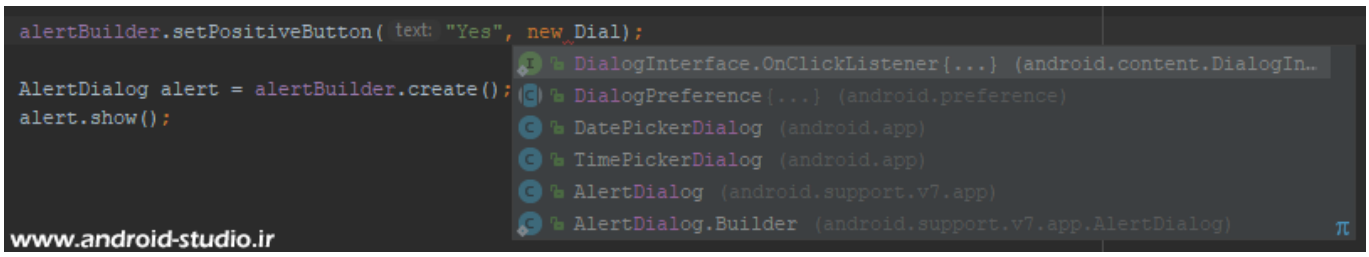
در انتها جهت ساخت و نمایش AlertDialog دو متد ()create و ()show را استفاده کردم.

پروژه را اجرا می کنم:



به همین سادگی توانستم یک پنجره پیغام را نمایش بدهم. از آنجایی که در پارامتر دوم دکمه، مقدار null تعریف شده، با انتخاب آن عملا هیچ اتفاقی رخ نداده و صرفا پنجره بسته می شود.

می خواهم برای دکمه فعلی یک Listener تعریف کنم:



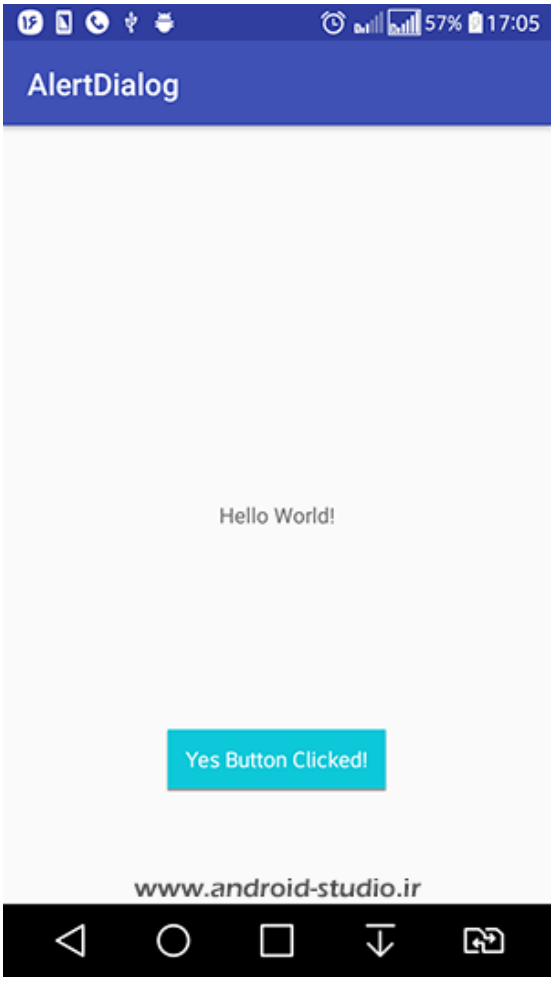
یک Listener از نوع DialogInterface.OnClickListener() جایگزین null شد:

```
alertBuilder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

    }
});
```

حالا مانند آنچه در گذشته آموختیم، هر دستوری درون onClick تعریف کنیم، با لمس دکمه اجرا خواهد شد. یک پیغام از جنس Toast به Listener اضافه کرده، پروژه را مجدد اجرا می کنم:

```
alertBuilder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "Yes Button Clicked!", Toast.LENGTH_SHORT).show();

    }
});
```

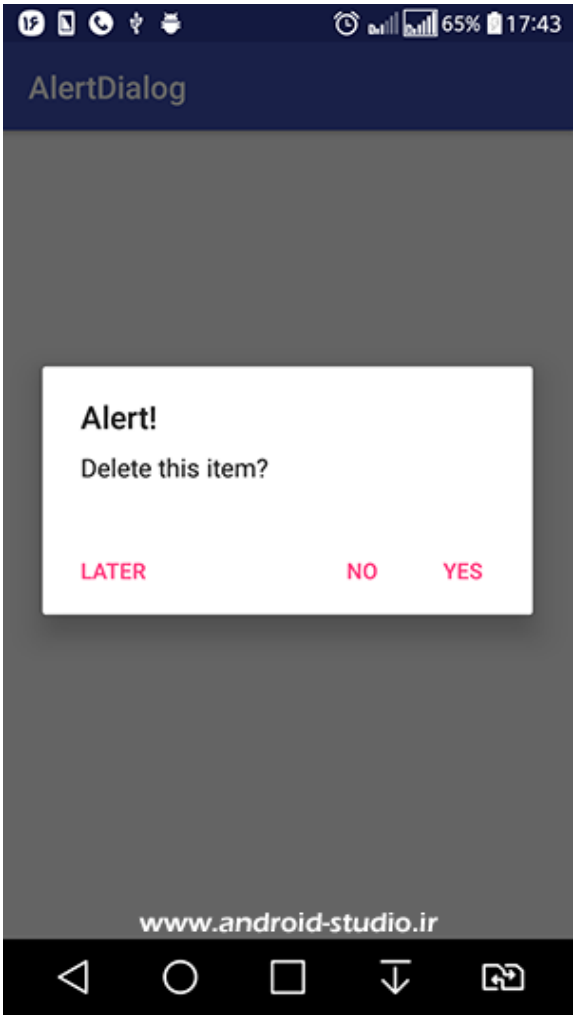
با لمس دکمه Yes، پیغام مربوطه اجرا می گردد:



دو متد مربوط به دو دکمه دیگر را نیز به اکتیویتی اضافه می کنم:

```
alertBuilder.setNegativeButton("No", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "No Button Clicked!", Toast.LENGTH_SHORT).show();

    }
});

alertBuilder.setNeutralButton("Later", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "Neutral Button Clicked!",
Toast.LENGTH_SHORT).show();

    }
});
```

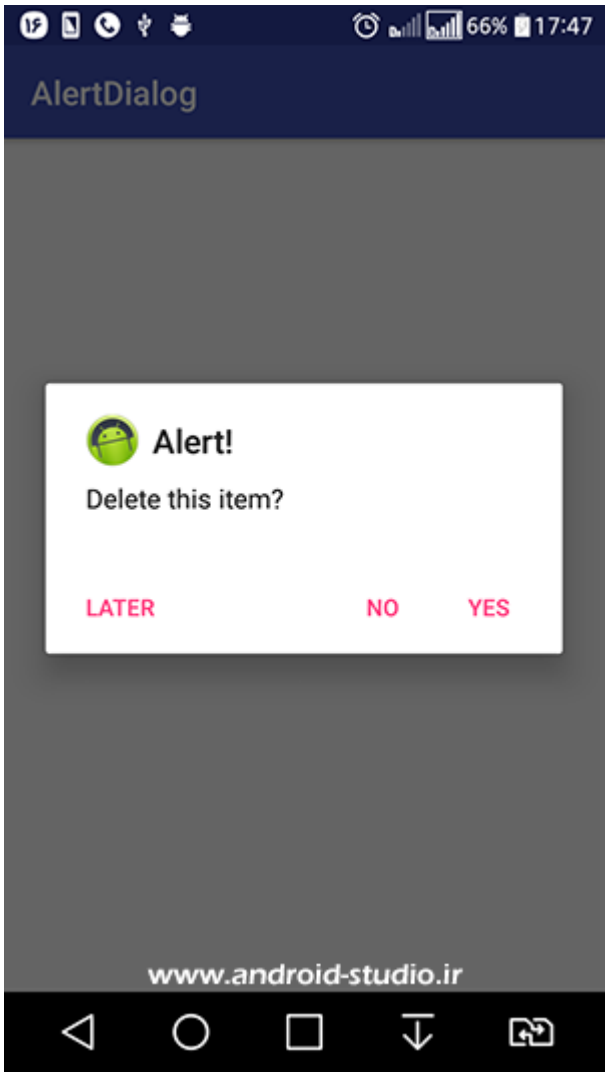
با اجرای پروژه، شاهد نمایش سه دکمه در انتهای پنجره هستیم. دکمه مربوط به NeutralButton در سمت چپ قرار می گیرد.



با استفاده از متد setIcon می توان یک آیکون نیز به عنوان AlertDialog افزود:

```
alertBuilder.setIcon(R.drawable.logo);
```

در خط بالا آیکون با نام logo.png در دایرکتوری drawable پروژه به عنوان آیکون این دیالوگ تعریف شده.



در حال حاضر اگر هنگامی که AlertDialog روی صفحه نمایش قرار دارد، کاربر خارج از پنجره دیالوگ را لمس کند، پنجره بسته خواهد شد. برای رفع این ایراد کافیست از متد setCancelable استفاده کرده و مقدار آن را false قرار دهیم:

```
alertBuilder.setCancelable(false);
```

با اضافه شدن این متد، تا زمانی که کاربر یکی از دکمه های تعریف شده را انتخاب نکند، پنجره دیالوگ روی صفحه نمایش باقی می ماند و با کلیک روی اطراف آن یا حتی دکمه Back دیوایس، از بین نمی‌رود.

کد کامل AlertDialog:

```
alertBuilder.setTitle("Alert!");
alertBuilder.setMessage("Delete this item?");
alertBuilder.setIcon(R.drawable.logo);
alertBuilder.setCancelable(false);

alertBuilder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "Yes Button Clicked!", Toast.LENGTH_SHORT).show();

    }
});

alertBuilder.setNegativeButton("No", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "No Button Clicked!", Toast.LENGTH_SHORT).show();

    }
});

alertBuilder.setNeutralButton("Later", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        Toast.makeText(MainActivity.this, "Neutral Button Clicked!",
Toast.LENGTH_SHORT).show();

    }
});

AlertDialog alert = alertBuilder.create();
alert.show();
```

در نهایت می توانم کد را بصورت زیر خلاصه کنم که تا حدودی از حجم برنامه کاسته می شود:

```
new AlertDialog.Builder(this)
        .setTitle("Alert!")
        .setMessage("Delete this item?")
        .setIcon(R.drawable.logo)
        .setCancelable(false)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

                Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

                Toast.makeText(MainActivity.this, "No Button Clicked!",
Toast.LENGTH_SHORT).show();

            }
        })
        .setNeutralButton("Later", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {

                Toast.makeText(MainActivity.this, "Neutral Button Clicked!",
Toast.LENGTH_SHORT).show();

            }
        })

.create()
.show();
```

تذکر: کد قبل از اصلاح و بهینه سازی فوق، در فایل جداگانه ای با نام MainActivity_first.java در کنار سورس اصلی قرار گرفته است.

برای راحتتر کردن فرایند تست دیالوگ، یک Button به اکتیویتی اضافه می کنم. سپس کد مربوط به AlertDialog را داخل یک تابع جداگانه تعریف کرده و آنرا درون متد این دکمه فراخوانی می کنم تا با هربار لمس دکمه، پنجره مجدد نمایش داده شود و نیاز به اجرای مجدد پروژه نباشد:

:activity_main.xml

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Run AlertDialog" />

</RelativeLayout>
```

:MainActivity.java

```
package ir.android_studio.alertdialog;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    //AlertDialog.Builder alertBuilder;
    Button alertButton;

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

        alertButton = findViewById(R.id.button);
        alertButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                runAlert();
            }
        });

    }

    private void runAlert() {

        new AlertDialog.Builder(this)
                .setTitle("Alert!")
                .setMessage("Delete this item?")
                .setIcon(R.drawable.logo)
                .setCancelable(false)
                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                        Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                        Toast.makeText(MainActivity.this, "No Button Clicked!",
Toast.LENGTH_SHORT).show();

                    }
                })
                .setNeutralButton("Later", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                        Toast.makeText(MainActivity.this, "Neutral Button Clicked!",
Toast.LENGTH_SHORT).show();

                    }
                })
```

```
        .create()
        .show();

    }

}
```

تذکر: تمرین فوق، با نام AlertDialog One در پوشه Exercises قرار دارد.

در ادامه قصد دارم سایر قابلیت های AlertDialog را معرفی کنم. پروژه جدیدی نمی‌سازم و همان پروژه قبل را ویرایش خواهم کرد.

یکی از کاربردهای AlertDialog نمایش یک لیست از آیتم‌هاست که کاربر باید یک گزینه را انتخاب کند. مانند یک بازی که ابتدا از کاربر می‌خواهد سطح بازی را تعیین کند.

ابتدا داخل بدنه اکتیویتی یک نمونه از []String می سازم و چند مقدار برای آن تعریف می کنم:

```
String[] items = {"Easy", "Medium", "Hard", "Very Hard"};
```

سپس این مقادیر را توسط متد setSingleChoiceItems نمایش می دهم:

```
private void runAlert() {

    new AlertDialog.Builder(this)
            .setTitle("Select Game level")
            .setCancelable(false)
            .setSingleChoiceItems(items, 2, null)
            .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                    Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

                }
            })

            .create()
            .show();

}
```

این متد سه پارامتر ورودی می‌گیرد. پارامتر اول همان items است، یعنی مقادیر آیتم‌های لیست. در پارامتر دوم می‌توانیم تعیین کنیم در حالت پیش فرض کدام آیتم انتخاب شده باشد. من مقدار 2 را

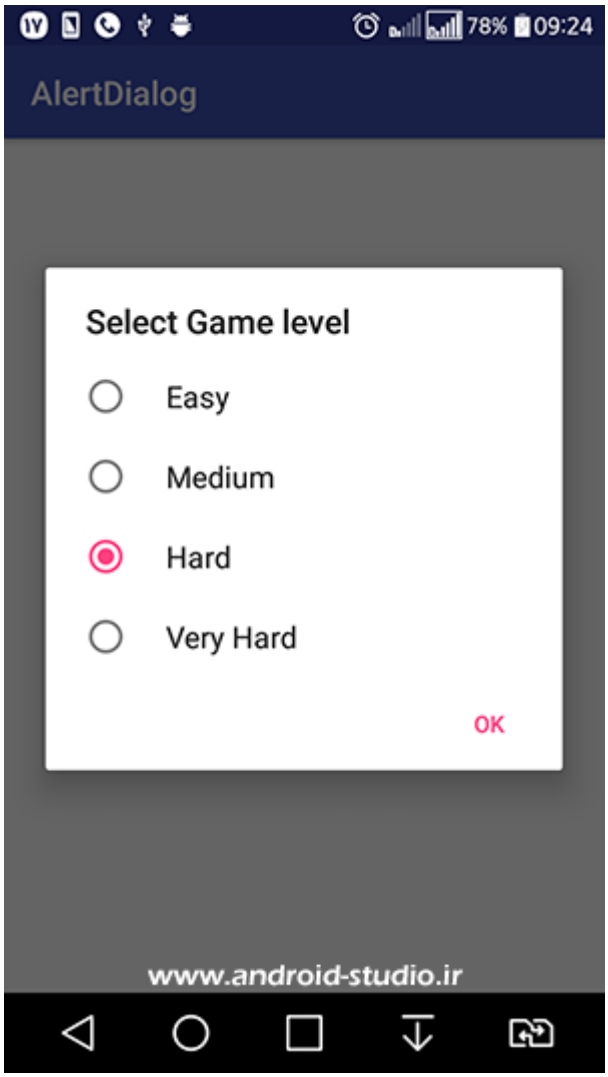
تعیین کردم یعنی گزینه Hard (شمارش از 0 آغاز می شود). اگر بخواهیم هیچ گزینه ای در ابتدا انتخاب نشده باشد مقدار 1- وارد می‌کنیم. پارامتر سوم هم مربوط به Listener است که فعلا null تعریف کردم.

توجه داشته باشید از متدهای setMessage و setSingleChoiceItems نمی‌توان همزمان استفاده کرد.

کد کامل MainActivity.java:

```
package ir.android_studio.alertdialog;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button alertButton;
    String[] items = {"Easy", "Medium", "Hard", "Very Hard"};

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

        alertButton = findViewById(R.id.button);
        alertButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                runAlert();
            }
        });

    }

    private void runAlert() {

        new AlertDialog.Builder(this)
                .setTitle("Select Game level")
                .setCancelable(false)
                .setSingleChoiceItems(items, 2, null)
                .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialogInterface, int i) {

                        Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

                    }
                })

                .create()
                .show();

    }

}
```

پروژه را اجرا می کنم:



مانند دکمه های AlertDialog، برای متد setSingleChoiceItems هم امکان تعریف Listener وجود دارد.

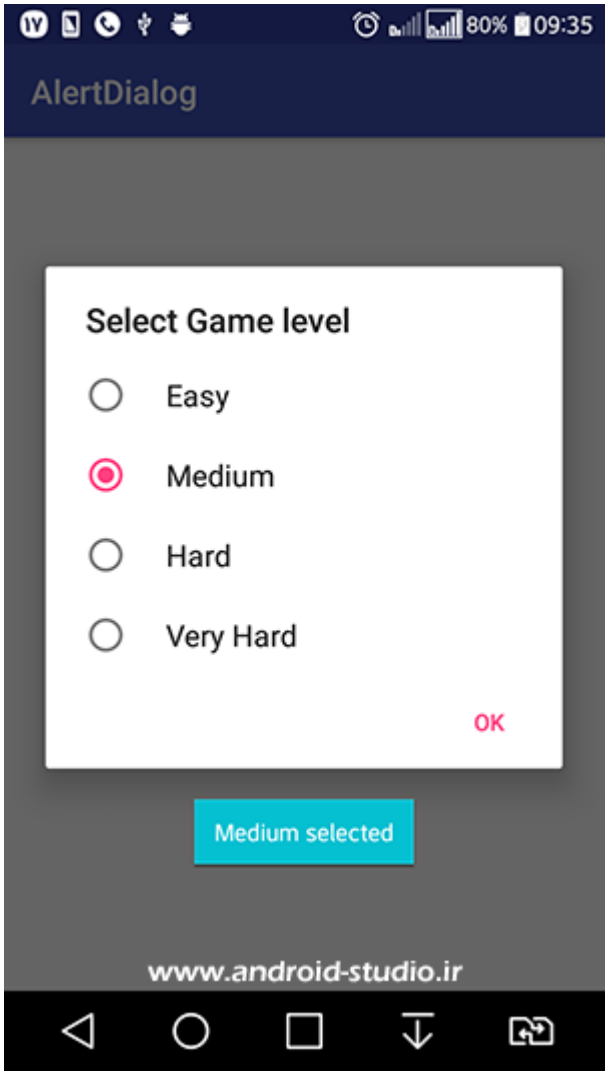
یک نمونه از String با نام دلخواه درون بدنه اکتیویتی تعریف می کنم:

```
String result;
```

حالا Listener را به اینصورت تکمیل می کنم:

```
.setSingleChoiceItems(items, 2, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

        result = items[i];
        Toast.makeText(MainActivity.this, result + " selected", Toast.LENGTH_SHORT).show();

    }
})
```

در کد بالا با انتخاب هر آیتم توسط کاربر، نام آن آیتم در یک Toast چاپ می شود:



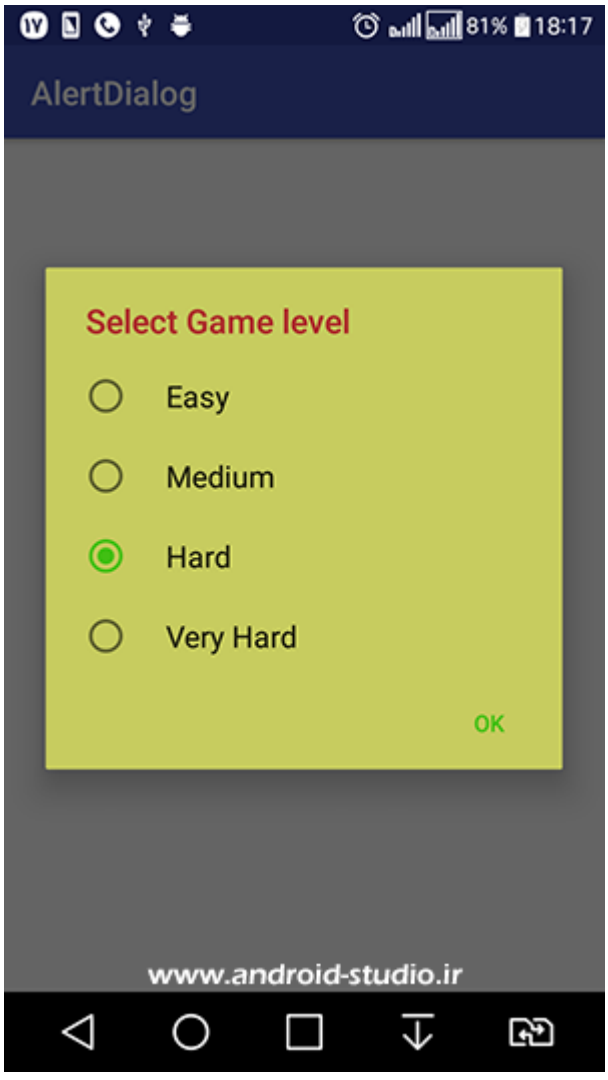
استایل دهی به اجزای AlertDialog نیز امکان‌پذیر است.

یک استایل جدید با نام دلخواه DialogTheme به styles.xml اضافه کردم که از Theme.AppCompat.Light.Dialog.Alert ارث بری شده:

```
<style name="DialogTheme" parent="Theme.AppCompat.Light.Dialog.Alert">
    <item name="colorAccent">#41bf13</item>
    <item name="android:textColorPrimary">#ac2a2a</item>
    <item name="android:background">#c7cc5f</item>
</style>
```

برای اتصال استایل به دیالوگ می‌بایست آنرا به عنوان پارامتر ورودی دوم AlertDialog.Builder تعریف کنیم:

```
new AlertDialog.Builder(this, R.style.DialogTheme)
```



ممکن است بخواهید برای پنجره دیالوگ یک نوار حاشیه (border) یا انحنا در چهار گوشه (radius) تعریف کنید. برای اینکار ابتدا یک Resource file درون دایرکتوری drawable ایجاد می کنم:

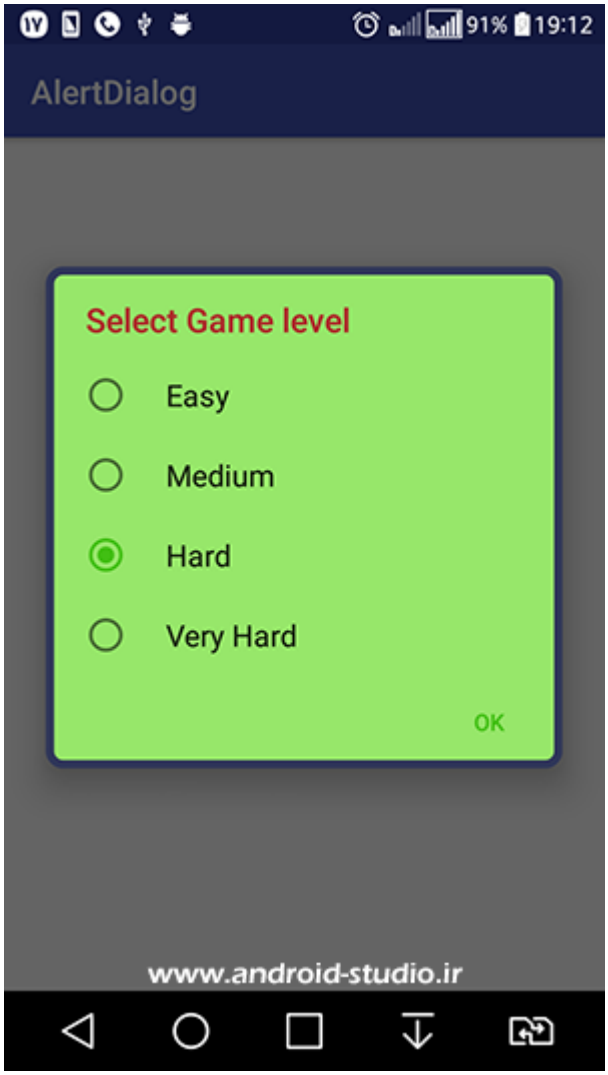
فایل dialog_background.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<inset xmlns:android="http://schemas.android.com/apk/res/android"
    android:insetLeft="16dp"
    android:insetTop="16dp"
    android:insetRight="16dp"
    android:insetBottom="16dp">

    <shape android:shape="rectangle">
        <corners android:radius="8dp" />
        <solid android:color="#97e76a" />
        <stroke android:width="5dp" android:color="#303759" />
    </shape>
</inset>
```

لینک توضیحات مربوط به InsetDrawable در انتهای مبحث قید شده. با اینحال نام آیتم ها و خواص بکار رفته در کد بالا گویای کاربرد آنهاست. insetLeft و سه مورد دیگر جهت تعیین فاصله پنجره دیالوگ از حاشیه صفحه نمایش هستند. سپس یک تگ shape داریم که درون آن به ترتیب مقدار انحنای گوشه ها، رنگ پس زمینه و ضخامت نوار حاشیه و رنگ آن تعریف شده اند.

حالا در استایل DialogTheme خط زیر را جایگزین خاصیت background می‌کنم:

```
<item name="android:windowBackground">@drawable/dialog_background</item>
```


91% 19:12
AlertDialog
Select Game level
Easy
Medium
Hard
Very Hard
OK
www.android-studio.ir

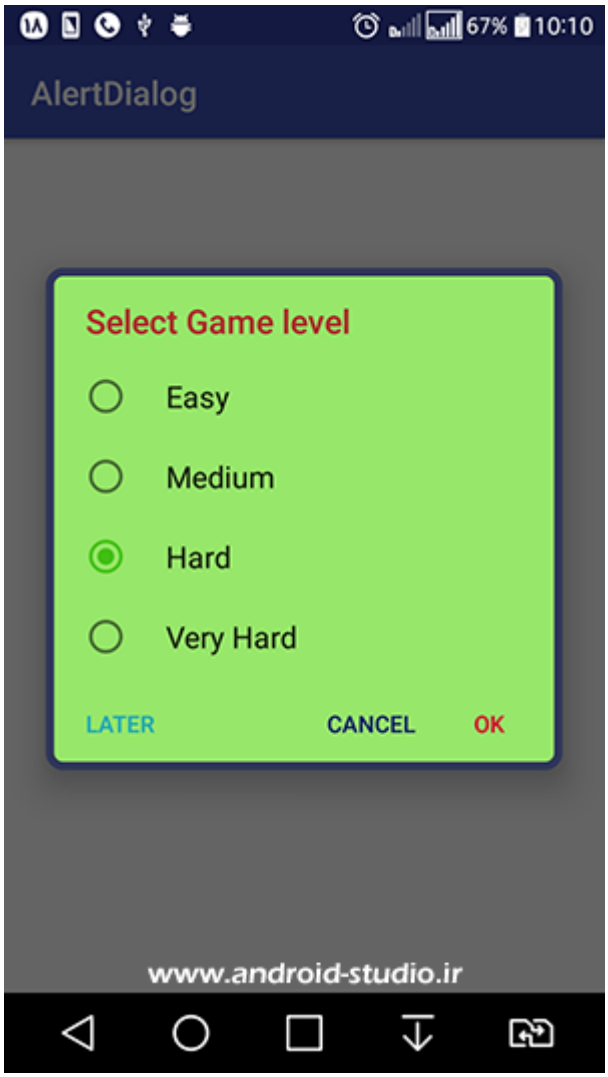
برای هرکدام از دکمه ها هم می‌توانیم یک رنگ جداگانه تعیین کنیم. مجدد دو دکمه دیگر به دیالوگ اضافه می کنم تا در مجموع سه دکمه در اختیار داشته باشم:

```
private void runAlert() {

    new AlertDialog.Builder(this, R.style.DialogTheme)
            .setTitle("Select Game level")
            .setCancelable(false)
            .setSingleChoiceItems(items, 2, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                    result = items[i];
                    Toast.makeText(MainActivity.this, result + " selected",
Toast.LENGTH_SHORT).show();

                }
            })
            .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                    Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

                }
            })
            .setNegativeButton("Cancel", null)
            .setNeutralButton("Later", null)

            .create()
            .show();

}
```

در قدم بعد یک استایل برای هر دکمه تعریف کرده و سپس در استایل DialogTheme به اینصورت اضافه می کنم:

```
<style name="DialogTheme" parent="Theme.AppCompat.Light.Dialog.Alert">
    <item name="colorAccent">#41bf13</item>
    <item name="android:textColorPrimary">#ac2a2a</item>
    <item name="android:windowBackground">@drawable/dialog_background</item>
    <item name="buttonBarNegativeButtonStyle">@style/NegativeButton</item>
    <item name="buttonBarPositiveButtonStyle">@style/PositiveButton</item>
    <item name="buttonBarNeutralButtonStyle">@style/NeutralButton</item>
</style>

<style name="PositiveButton" parent="Widget.AppCompat.Button.ButtonBar.AlertDialog">
    <item name="android:textColor">#d6062d</item>
</style>

<style name="NegativeButton" parent="Widget.AppCompat.Button.ButtonBar.AlertDialog">
    <item name="android:textColor">#131e68</item>
</style>

<style name="NeutralButton" parent="Widget.AppCompat.Button.ButtonBar.AlertDialog">
    <item name="android:textColor">#1da6ba</item>
</style>
```

با اجرای پروژه مشاهده می کنید هرکدام از دکمه ها رنگ مختص خود را نشان می‌دهد:



تذکر: کد اکتیویتی فوق، در فایل جداگانه ای با نام MainActivity_first.java در کنار سورس اصلی قرار گرفته است.

یک قابلیت دیگر که در این مبحث آشنا می‌شویم، امکان استفاده از Custom Layout در AlertDialog است. فرض کنید می‌خواهید در یک پنجره دیالوگ، دو فیلد Username و Password نشان دهید تا کاربر از این طریق وارد حساب کاربری خود در اپ شود.

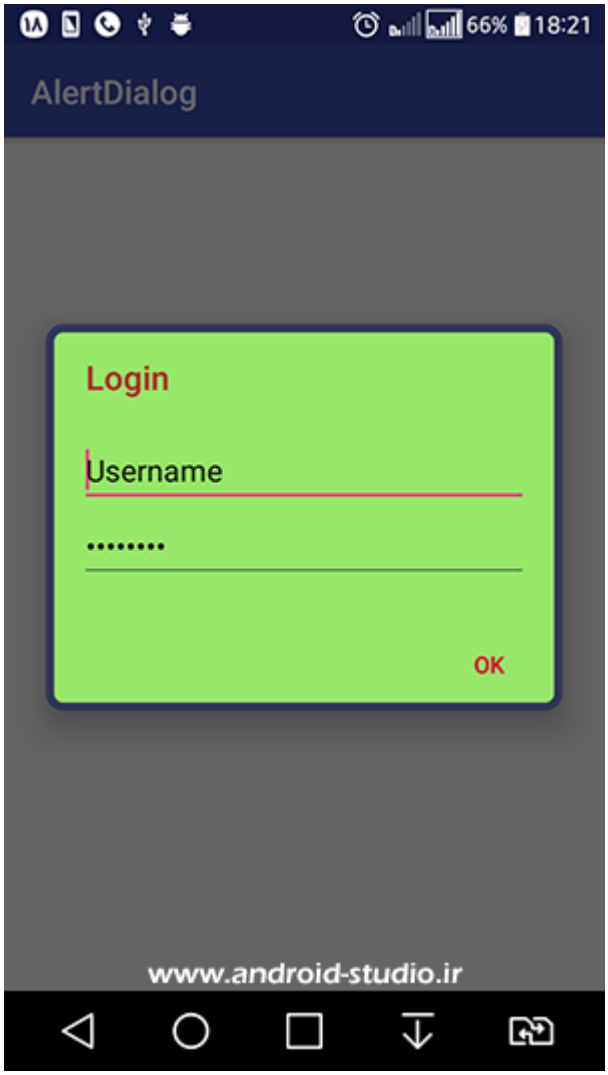
ابتدا یک layout به پروژه اضافه می کنم:

:custom_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Username" />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPassword"
            android:text="Password" />

    </LinearLayout>

</RelativeLayout>
```

سپس با استفاده از LayoutInflater، لایه فوق را inflate کرده و در نهایت توسط متد ()setView به AlertDialog اضافه می کنم:

```
private void runAlert() {

    LayoutInflater aInflater = getLayoutInflater();
    View alertLayout = aInflater.inflate(R.layout.custom_dialog, null);

    new AlertDialog.Builder(this, R.style.DialogTheme)
            .setTitle("Login")
            .setCancelable(false)
            .setView(alertLayout)
            .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {

                    Toast.makeText(MainActivity.this, "Yes Button Clicked!",
Toast.LENGTH_SHORT).show();

                }
            })

            .create()
            .show();

}
```



تذکر: تمرین فوق، با نام AlertDialog Two در پوشه Exercises قرار دارد.

منابع تکمیلی:

https://developer.android.com/reference/android/app/AlertDialog

https://material.io/guidelines/components/dialogs.html

https://developer.android.com/reference/android/graphics/drawable/InsetDrawable

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.
www.android-studio.ir