منابع اصلی

✓ **جزوه مورد تدریس در کلاس درس: برنامه نویسی جاوا و برنامه نویسی موبایل۱ و برنامه نویسی موبایل۲**

1. Learn Android Studio: Build Android Apps Quickly and Effectively (The authors: Adam Gerber, Clifton Craig, David Selvarj)

۲. اندروید برای برنامه نویسان(Android Studio دیتیل-دیتیل)

۳. آموزش کاربردی برنامه نویسی Android در محیط Android studio(نویسنده: جی پاول کاردل و ترجمه:سیدعلیرضا قمصری انتشارات: پندارپارس)

۴. اندروید برای برنامه نویسان: با رویکرد مبتنی بر برنامه نویسی(پل دیتل، هاروی دیتل، اگزاندر والد. ترجمه: کامران سیروسیان انتشارات: نص)

۵. جاوا ۹ برای برنامه نویسان(ترجمه بهرام پاشایی انتشارات آیلار)

ارزیابی

- تکالیف ٪۲۰
- میانترم ٪۲۰
- پایان ترم ٪۶۰

برای تمریناتی که بالاتر از سطح استاندارد آماده شده باشند تا ٪۱۰ نمره اضافه در نظر گرفته خواهد شد.

تکالیف

- پیاده سازی و طراحی برنامه های مختلف با زبان جاوا در محیط اندروید استدیو برای موبایل.
- برای هر تکلیف یک گزارش کتبی شامل : پاسخ سوالات، خلاصه کارهای انجام شده، نحوه پیاده سازی برنامه ها و نتایج بدست آمده ارائه شود.

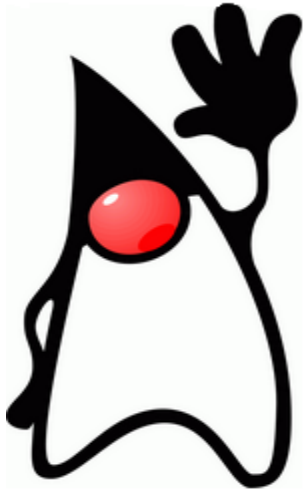
Mobile.Family-Name.rar(zip) Example :**Mobile.Bagheri.asl-Salar.rar**

- **ارسال تمرینات** afo.ac.computer@gmail.com
- **سوالات درسی** https://t.me/laptop1982 یا salar.ba@gmail.com

کانال آموزشی و ارسال منابع

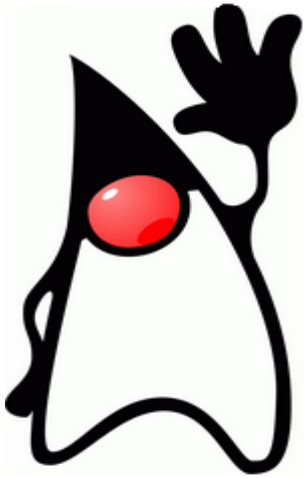
دانشجویان عزیز برای دریافت آخرین جزوات آموزشی و همچنین اطلاع رسانی های مورد نیاز و دریافت نرمافزارهای مربوطه با اسکن کد مقابل وارد کانال آموزشی در پیغام رسان تلگرام شوید

سرفصل مطالب برنامه نویسی موبایل ۱



- نصب اندروید استودیو و تنظیمات
- آشنایی با محیط اندروید استودیو
- آشنایی با چینش عناصر و TextView
- آشنایی با رنگ ها و Button, Event
- آشنایی با چرخه حیات Activity
- آشنایی با انواع منو و روش های تعریف
- آشنایی با EditText و بررسی ویژگی ها

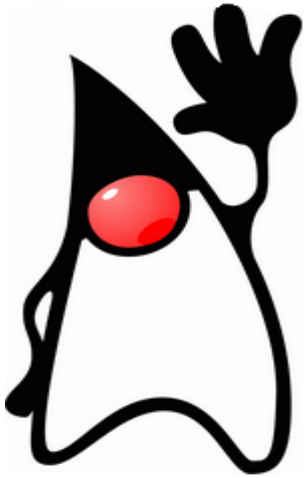
سرفصل مطالب برنامه نویسی موبایل۱



- آشنایی با Lyaout ها در اندروید
- آشنایی وکار با فرم ها
- آشنایی با Intent و حرکت بین Activity ها

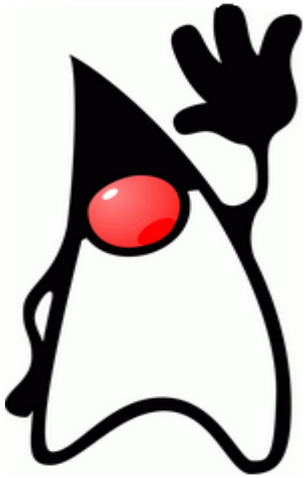
- آشنایی با ImageView
- آشنایی با فهرستها (ListView,...)
- آشنایی با Toast, snackBar و تنظیمات
- تعریف یک پروژه پایانی

سرفصل مطالب موبایل ۲



- آشنایی و ساخت OptionMenu
- طراحی فرم ورود اطلاعات
- آشنایی با Intent و حرکت بین Activity ها
- ارسال مقادیر فیلدهای فرم به اکتیویتی دیگر
- دریافت اطلاعات از اکتیویتی فرخوانی شده
- کار با مدیا(Media)
- VideoView
- پخش صوت(MediaPlayer)
- اضافه کردن Support Action Bar

سرفصل مطالب موبایل ۲



- انمیشن‌های ساده برای تصاویر
- Preference
- کار با فایل‌ها
- آشنایی با xml, json
- شی گرایی در اندروید
- کار با دیتابیس
- دریافت اطلاعات از دوربین(BarcodeScanner)
- پروژه کاربردی دیتابیس(کار با دیتابیس و دریافت اطلاعات از سرور و اسکن بارکد)
- ارسال SMS
- کار با سنسور موقعیت یاب (GPS)

درباره اندروید استودیو

درباره اندروید استودیو

- نرم افزار اندروید استودیو (Android Studio) یک محیط توسعه یکپارچه (IDE) رسمی برای توسعه پلتفرم و برنامه نویسی اندروید است.
- این نرم افزار در ۱۶ می سال ۲۰۱۳ در کنفرانس Google I/Oمعرفی شد. اندروید استودیو تحت لیسانس Apache License 2.0به صورت رایگان در دسترس قرار دارد.
- اندروید استودیو در می سال ۲۰۱۳ از ورژن ۰.۱ در مرحله پیش نمایش قرار داشت، سپس با شروع ورژن ۰.۸ وارد مرحله بتا شد که در ژوئن سال ۲۰۱۴ عرضه شد. اولین نسخه ثابت آن در دسامبر سال ۲۰۱۴ عرضه شد و با ورژن ۱.۰ شروع شد.
- براساس نرم افزار IntelliJ IDEAاز شرکت JetBrains، اندروید استدیو مخصوص توسعه اندروید طراحی شده است. لینک دانلود آن برای ویندوز، Mac OS Xو لینوکس قرار دارد و به عنوان IDEاصلی گوگل برای توسعه برنامه های محلی اندروید، جایگزین Eclipse Android Development Toolsشده است.

ویژگی های اندروید استودیو

- ویژگی های جدید با عرضه هر یک از نسخه های جدید اندروید استودیو ارائه می‌شوند. ویژگی های زیر در ورژن ثابت کنونی ارائه شده اند:

- – پشتیبانی از Build مبتنی بر Gradle
–بازنویسی کد و اصلاحات فوری مخصوص اندروید
– ابزارهای Lintبرای رفع مشکلات عملکرد، کارایی، سازگاری ورژن ها و مشکلات دیگر
– ادغام با ProGuardو قابلیت های App Signing
–پنجره های Wizardمبتنی بر Templateبرای ایجاد طرح ها و مولفه های رایج اندروید
– یک Layout Editorغنی که به کاربران اجازه می دهد مولفه های محیط کاربری را درگ و دراپ کنند و گزینه ای برای پیش نمایش Layoutها در چندین پیکربندی صفحه نمایش وجود دارد.
– پشتیبانی از ساخت برنامه های Android Wear
–پشتیبانی داخلی از پلتفرم Google Cloudکه اجازه ادغام با پیام رسانی و موتور برنامه Google Cloudرا می دهد.

ویژگی های اندروید استودیو

- اجرای فوری با Instant Run
- می توانید کدها و منابع برنامه تان را تغییر بدهید و همزمان تغییرات را روی دستگاه یا شبیه ساز مشاهده کنید.

ویژگی‌های اندروید استودیو

ویرایشگر متن هوشمند

```
            android:title="Gallery" />
        <item
            android:id="@+id/nav_slideshow"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="@string/slideshow" />

    </group>

    <item android:title="@string/communicate">
        <menu>
            <item
                android:id="@+id/nav_share"
                android:icon="@drawable/ic_menu_share"
                android:title="@string/share" />
            <item
                android:id="@+id/nav_send"
                android:icon="@drawable/ic_menu_send"
                android:title="@string/send" />
        </menu>
```

ویژگی‌های اندروید استودیو

- شبیه ساز پرسرعت و پر از ویژگی های مختلف

- می توانید برنامه هایتان را سریع تر از یک دستگاه واقعی نصب و اجرا کنید و آنها را روی هر نوع دستگاه اندرویدی که بخواهید تست کنید: گوشی های اندروید، تبلت های اندروید، Android Wearو دستگاه های Android TV.
- Android Emulator 2.0جدید از همیشه سریع تر است و اجازه می دهد به طور پویا اندازه شبیه ساز را تغییر دهید و به مجموعه ای از کنترل های سنسور دسترسی داشته باشید.

ویژگی‌های اندروید استودیو

- سیستم Buildقدرتمند و انعطاف پذیر
  - به راحتی می توانید کتابخانه های کد را به پروژه تان اضافه کنید و از یک پروژه، چندین Buildمختلف تولید کنید.
  - اندروید استودیو به همراه Gradle، یک اتوماسیون Buildبا عملکرد بالا، مدیریت وابستگی قدرتمند و پیکربندی های uild سازی ارائه می دهد.

ویژگی‌های اندروید استودیو

- **توسعه تمام دستگاه های اندروید**
  - با یک پروژه می توانید Form Factorهای بسیاری را هدف قرار دهید تا به راحتی کد را در میان ورژن های مختلف برنامه تان به اشتراک بگذارید.
  - اندروید استودیو یک محیط یکپارچه برای توسعه برنامه های گوشی ها و تبلت های اندروید، Android Wear، Android TVو Android Autoفراهم می کند.

قالب‌های کد و ادغام با GitHub

- می توانید پروژه ها را با Templateها شروع کنید تا الگوهایی مانند منوی ناوبری و ViewPaperداشته باشید یا اینکه نمونه های کد گوگل را از GitHubوارد کنید.
- با پنجره های Project Wizardاندروید استودیو، اضافه کردن کد به پروژه های جدید از همیشه راحت تر است.

سیستم مورد نیاز

	ویندوز	OS X	لینوکس
ورژن سیستم عامل	مایکروسافت ویندوز ۷/۸/۱۰ (۳۲ یا ۶۴ بیتی)	Mac OS X 10.8.5 یا بالاتر تا 10.11.4 (El Capitan)	GNOME یا KDE desktop
رم	حداقل رم ۲ GB – رم توصیه شده ۸ GB		
فضای دیسک	۵۰۰ MB فضای دیسک برای اندروید استودیو، حداقل ۱,۵ GB برای Android SDK، تصاویر سیستم Emulator و Cache		
ورژن جاوا	Java Development Kit (JDK) 8	Java Development Kit (JDK) 6	Java Development Kit (JDK) 8
رزولوشن صفحه	حداقل رزولوشن صفحه نمایش ۱۲۸۰x۸۰۰		

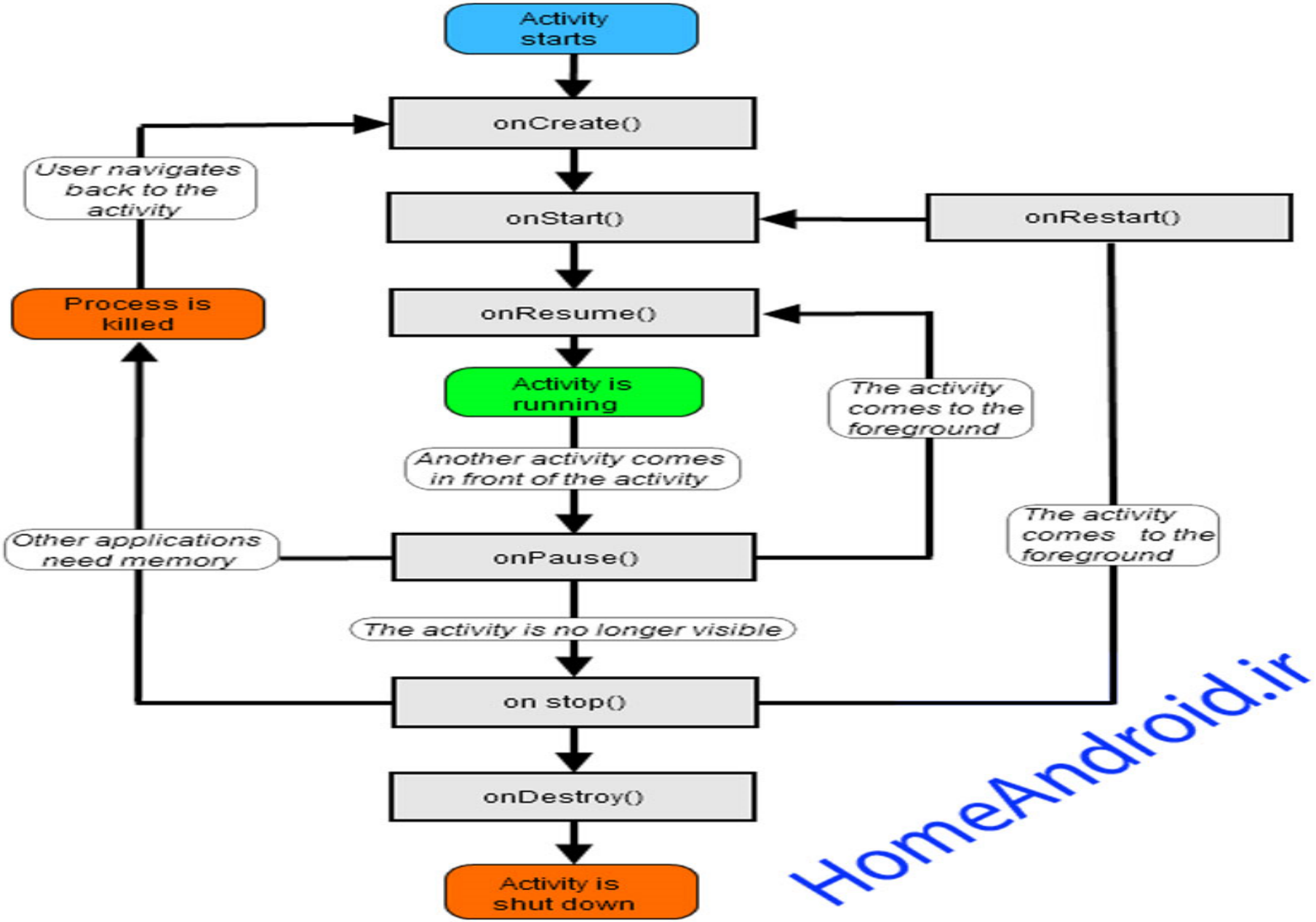
	ویندوز	OS X	لینوکس
ورژن سیستم عامل	مایکروسافت ویندوز ۲۰۰۳/XP/ویستا/ ۷/۸/۸,۱/۱۰ (۳۲ یا ۶۴ بیتی)	Mac OS X 10.8.5 یا بالاتر تا 10.10 تا 10.10.2 تا 10.10.3 روی 10.10.5 (Yosemite)	GNOME یا KDE یا Unity desktop روی Ubuntu یا Fedora یا GNU/Linux Debian
رم	حداقل رم ۲ GB – رم توصیه شده ۴ GB		
فضای دیسک	۵۰۰ MB فضای دیسک		
فضا برای Android SDK	حداقل ۱ GB برای Android SDK، تصاویر سیستم Emulator و Cache		
ورژن JDK	Java Development Kit (JDK) 7 یا بالاتر		
رزولوشن صفحه	حداقل رزولوشن صفحه نمایش ۱۲۸۰x۸۰۰		

کلیدهای میانبر در اندروید استدیو

F4 : رفتن به سورس	**مرتب سازی کد های هارد شده یا بهم ریخته شده : Ctrl+Alr+L**
رفتن به کلاس والد : CTRL + U	حذف importاستفاده نشده: Ctrl+Alt+O
تولید خودکار کانستراکتور و ... : Alt+Insert	تکمیل کد (نمایش راهنما): Ctrl+Space
تغییر وضعیت: Ctrl+~	حل سریع مشکل: Alt+Enter
کامل کردن پرانتز: Ctrl+Shift+Enter	افزودن یا حذف کامنت خطی: / + CTRL
اورراید کردن متدها: Ctrl+O	افزودن یا حذف کامنت بلوکی: / + CTRL + SHIFT
ایمپلمنت کردن متدهای اینترفیس: Ctrl+I	رفتن به متد قبل/بعد: ALT + UP/DOWN
کپی کردن کامل یک خط به خط پایینی: Ctrl + D	نمایش پارامترهای متد: CTRL + P
جست و جو: Ctrl + F	جستجوی سریع مستندات: CTRL + Q
عملیات جستجو و جایگزین کردن: Ctrl + R	حذف خط: CTRL + Y
تغییر اسم متغییر و ... در کل پروژه: Shift + F6	رفتن به کلاس: Ctrl + N
جابجایی بین تب های باز: ALT + Left-Arrow; ALT + Right-Arrow	رفتن به فایل: Ctrl + Shift + N
مراجعه به فایل های اخیر: CTRL + E	رفتن به خط: CTRL + G
رفتن به (declaration) اعلامیه: CTRL + B	جابجایی به محل آخرین تغییر: CTRL + SHIFT + BACKSPACE

چرخه حیاط اکتیویتی‌ها در اندروید استودیو

- اکتیویتی‌ها در واقع همان صفحاتی هستند که برای کاربر نشان داده می‌شوند و تمامی طراحی‌ها و چینش کامپوننت‌ها در روی آنها انجام می‌شود.
- بنابراین ممکن است که برنامه طراحی شده برای استفاده شامل بیش از ۱ اکتیویتی باشد و بنا به شرایطی بخواهیم بین اکتیویتی‌ها جابجا شویم که در این حالت اصطلاحی بنام چرخه حیاط اکتیویتی مطرح می‌گردد در اسلایدهای بعدی به این موضوع می‌پردازیم.



چرخه حیاط اکتیویتی

چرخه اکتیویتی همان طور که در تصویر قبلی مشاهده
وقتی شما یک اکتیویتی را استارت می‌کنید یا به اصطلاح پروژه خودتان را Run می‌کنید اولین متدی که فراخوانی می شود متد OnCreate می باشد توسط این متد اکتیویتی ما اجرا می شود.
خوب همان‌طور که در تصویر بالا، دید زمان های مختلفی از یک اکتیویتی نشان داده شده است. وقتی کاربر وارد یک اکتیویتی می‌شود برای اولین بار برنامه اندرویدی ما وارد متد onCreate می‌شود و کدهای موجود در این متد Run می‌گردد حالا اگه کاربر برروی یک دکمه کلیک نماید و وارد یک اکتیویتی جدید شود اکتیویتی قبلی وارد onPause خواهدشد و اگر کاربر مدت زمان زیادی را وارد اکتیویتی قبلی نشود اکتیویتی وارد متد onStop خواهد شد و بعد از مدتی هم وارد متد onDestroy خواهد شد.
در صورتی که برنامه در اکتیویتی داخل onPause باشد و کاربر به اکتیویتی قبلی برگردد کدهای موجود در onResume اجرا خواهد شد ولی اگه اکتیویتی بسته شده باشه کدهای داخل onCreate از نو شروع میشود.

چرخه حیاط اکتیویتی

فرض کنید یک صفحه تنظیمات داریم و میخواهیم وقتی کاربر دکمه برگشت را فشار داد برنامه تنظیمات را ذخیره کند، در کدام قسمت باید کد های مربوط به ذخیره سازی اطلاعات را وارد کنیم؟

onPause

حالا فکر کنید ما داخل صفحه یک دکمه تنظیمات داریم که کاربر وقتی بر روی آن کلیک می‌کند یک صفحه باز می‌شود و تنظیمات را انجام می‌دهد و وقتی دکمه برگشت را زد. داخل صفحه قبلی تنظیمات اعمال و تغییرات انجام شود!! برای اینکه تغییرات انجام بشود باید کدها را کجا قرار بدیم؟

onResume

مثلاً می‌خواهیم Font تغییرسایز شود در onResume مشخص می‌کنیم که مقدار جدید را بگیرد و FontSize را تغییر بدهد.

چرخه حیاط اکتیویتی

onCraete()

هنگامی که اکتیویتی برای اولین بار ایجاد می شود، فراخوانی می شود.

onStart()

هنگامی که اکتیویتی به کاربر نمایش داده می شود فراخوانی می گردد.

onResume()

هنگامی که اکتیویتی شروع به تعامل با کاربر می کند، فراخوانی می شود.

onPause()

هنگامی که اکتیویتی فعلی موقتا نگه داشته می شود و اکتیویتی قبلی در حال شروع به کار شدن است، فراخوانی می شود.

onStop()

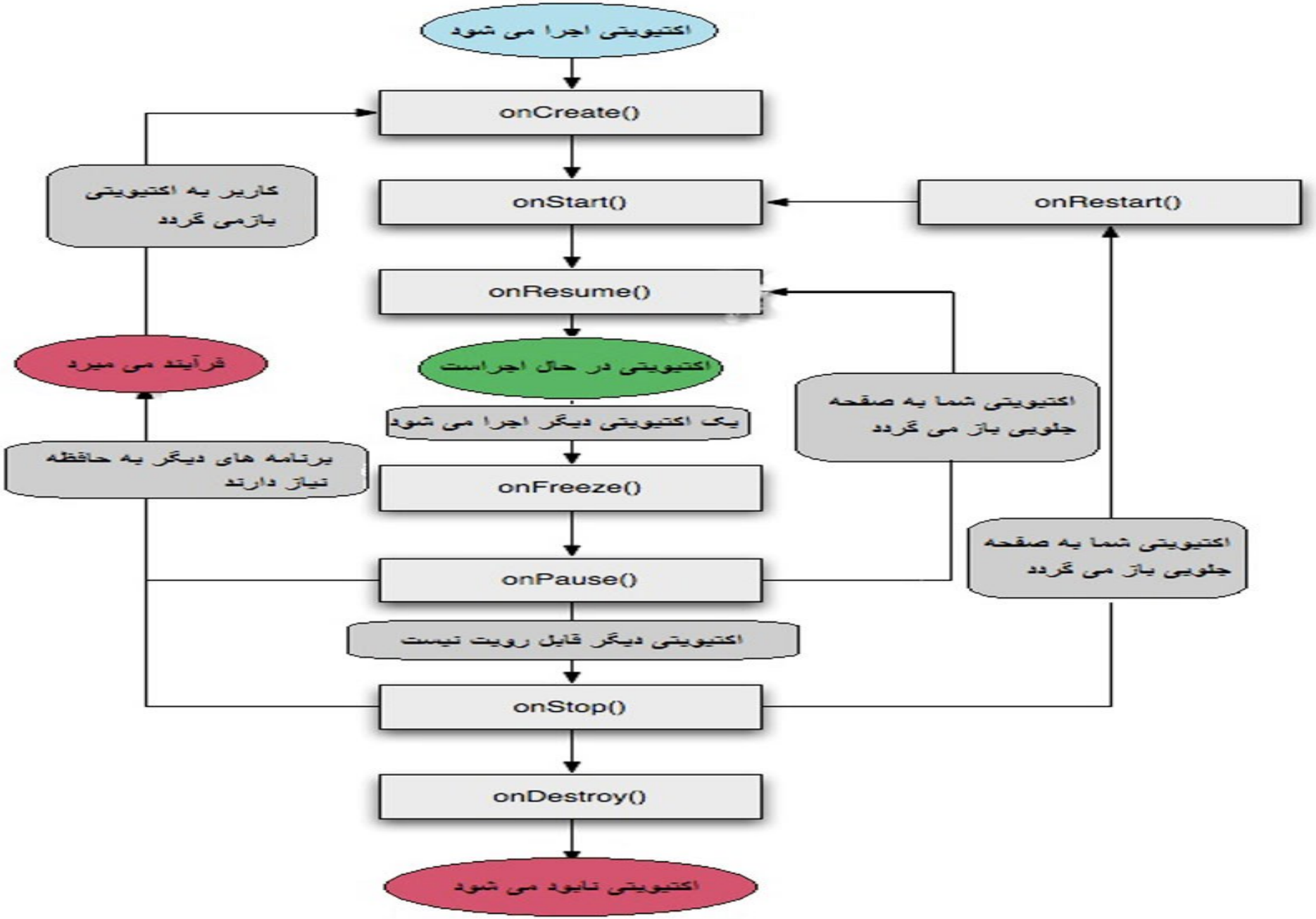
هنگامی که دیگر اکتیویتی به کاربر نمایش داده نمی شود، فراخوانی می شود.

onDestroy()

قبل از اینکه اکتیویتی کاملا از بین برود، توسط سیستم فراخوانی می شود.

onRestart()

هنگامی که اکتیویتی پس از توقف دوباره شروع به کار می کند، فراخوانی می شود.


اکتیویتی اجرا می شود
onCreate()
onStart()
onRestart()
onResume()
اکتیویتی در حال اجراست
یک اکتیویتی دیگر اجرا می شود
onFreeze()
onPause()
اکتیویتی دیگر قابل رویت نیست
onStop()
onDestroy()
اکتیویتی نابود می شود
کاربر به اکتیویتی بازمی گردد
فرآیند می میرد
برنامه های دیگر به حافظه نیاز دارند
اکتیویتی شما به صفحه جلویی باز می گردد
اکتیویتی شما به صفحه جلویی باز می گردد

Gradle چیست؟!

- ***Gradle Scripts***چیست و چه کاربردی دارد اصلا چرا اندروید استودیو با گریدل دیباگ و ارزیابی می‌شود در پاسخ این سوال می توان جواب های زیادی را ارائه داد.
- برای آشنایی با ***Gradle Scripts***اول باید با build system آشنا بشید. اما بیلدسیستم چیست؟
- بیلد سیستم یک ابزار نرم‌افزار می‌باشد که کار کامپایل کردن کدها را به صورت اتوماتیک انجام می‌دهد. هدف اصلی بیلدسیستم‌ها کامپایل و اجرا کردن کدها می‌باشد.

Gradle چیست؟! ادامه...

- برای هر زبان برنامه‌نویسی بیلدسیستم‌های مختلفی به وجود آمدند.
- جاوا سه نوع BuiledSystem اصلی دارد:

۱) Ant-محصول Apache معرفی شده در سال ۲۰۰۰ – براساس زبانXML

۲) Maven-محصول Apache معرفی شده در سال ۲۰۰۴ – براساس زبانXML

۳) Gradle-محصول Hans Docker معرفی شده در سال ۲۰۱۲ – براساس زبانGroovy

Gradle چیست؟! ادامه...

- Gradle یک بیلدسیستم اتوماتیک متن‌باز می‌باشد و برخلاف Ant و Maven که براساس زبان XMLهستند، براساس زبان Groovy شکل گرفت. Gradle مزایای هر دو بیلدسیستم Ant و Mavenرا در خود جمع کرد و خیلی زود بخاطر قدرت و انعطاف‌پذیریش موردتوجه قرار گرفت.

- برای اولین بار گوگل در کنفرانس Google I/O در سال ۲۰۱۳ اعلام کرد که از بیلدسیستم Gradle در IDEجدیدش یعنی آندروید استودیو استفاده خواهد کرد. گوگل، Gradleرو به وسیله یک افزونه به نام Android Plug-in for Gradleبه آندروید استودیو آورد.

مزایای بیلدسیستم گریدل:

- **Project Dependency:**

- Dependencyمی‌تواند یک Modules یا JAR File یا Library باشد که یا روی کامپیوتر خود شماست یا روی یک سرور. گریدل می‌تواند این Dependency ها رو مدیریت و به پروژه اضافه کند.

- **Testing:**

- **gradle-testing**

- Gradleبه صورت اتوماتیک یک پوشه تست از پروژه شما درست می‌کند و محتویات پروژه و یک فایل تست از APK پروژه شما را داخلش نگه می‌دارد و هنگام Build شدن پروژه، عملیات تست را روی اپ شما انجام می‌دهد.

مزایای بیلدسیستم گریدل:

- **Publishing:**
- **gradle-publishing**

Gradle می‌تواند مراحل Sign کردن اپ شما را مدیریت کند.

- Multiple APKs:

Gradle می‌تواند از پروژه شما چند خروجی APK بگیرد. برای مثال وقتی که می‌خواهید برای دیوایس‌های مختلف با سخت‌افزارهای مختلف اپ‌های جداگانه داشته باشید.

تشریح فایل‌های گریدل:

- هروقت پروژه‌ای در اندروید استودیو ساخته بشه، به طور اتوماتیک فایل‌هایی همراه پروژه ساخته می‌شوند که در اندروید استودیو زیرمجموعه Gradle Scripts هستند.

- **gradle-scripts**

- Gradle Buidیا همان فایلbuild.gradle

هر پروژه آندروید استودیو حداقل دارای دو فایل build.gradle هست:

- **۱- Top Level Build File**

- تنظیمات اصلی که مربوط به کل پروژه است در این فایل قرار می‌گیرد. ما نیازی به تغییر و دستکاری این فایل نداریم.

- **۲- Module Level Build File**

- هر ماژول، دارای یک build.gradle جداگانه است که تنظیمات مربوط به ماژول مخصوص خودش داخل آن قرار می‌گیرد. می‌توان گفت در اینجا ماژول به معنای پروژه‌های زیرمجموعه پروژه اصلی‌ست. ما در طول مدت برنامه‌نویسی در اندروید استودیو همیشه با این فایل سر و کار داریم.

تشریح فایل‌های گریدل:

- gradle-multiple-modules

- همانطور که در تصویر بعدی می‌بینید build.grade اول همان Top Level Build File ما هست که مربوط به کل پروژه است و فایل‌های build.gradle بعدی همان Module Level Buildما هستند که داخل پرانتز روبروی هرکدام کارایی آن مشخص شده است. این یعنی یک پروژه اندروید در تصویر وجود دارد و نسخه‌های مختلف این اپ برای دستگاه‌های مختلف مثل موبایل، تلویزیون و پوشیدنی در حال توسعه است.

تشریح محتویات فایل‌ها:

- **Top Level Gradle Build File:**
  - محتویات این فایل همیشه ثابت هست و نیازی به تغییر دادن داخل این فایل نیست.

- **Module Level Gradle Build File:**
  - هر پروژه‌ای که داخل پروژه اصلی تعریف شود، یک Module Level Gradle Build جداگانه خواهد داشت. داخل این فایل اطلاعات کلی اپ مثل min sdk ، target sdk، نسخه اپ و dependency های پروژه تعریف می‌شوند. در اندروید استودیو نیازی نیست اطلاعاتی که بالا گفتم داخل فایل manifest تعریف بشه، و تمام این اطلاعات داخل همین فایل قرار می‌گیره.

تشریح محتویات فایل‌ها:

- **gradle-wrapper.properties:**

این فایل به دیگران اجازه می‌دهد که کدهای شما را Build کنند حتی اگر Gradle روی کامپیوتر خود نصب نداشته باشند.

- این فایل بررسی می‌کند که چه نسخه‌ای از Gradle برای Build کردن کدها نیاز است و اگر نسخه مورد نظر را پیدا نکند، اقدام به دانلود کردن نسخه‌ی مورد نیازش برای اجرای کدها می‌کند. وقتی شما پروژه‌ای در اندروید استودیو می‌سازید امکان دارد این فایل همراه پروژه ساخته نشود، ولی اگر پروژه‌ای را از اینترنت مثل سایت github بگیرید، بعضی مواقع این فایل را داخلش پیدا می‌کنید.

- **Settings.gradle:**

این فایل تمام زیرپروژه‌هایی همان (Module) که پروژه ما داخلش دارد را معرفی می‌کند.

- **Gradle.properties:**

- اطلاعات کلی پروژه داخل این فایل ذخیره می‌شوند. به‌طور پیش‌فرض این فایل خالی است.

- **Local.properties:**

- این فایل محل ذخیره android sdk را به پلاگ‌این Android Gradle معرفی می‌کنه.

پنجره‌های گریدل در اندروید استودیو

- چند پنجره در اندروید استودیو هستند که مربوط به گریدل می‌شوند.
- پنجره:Gradle Task

- **gradle-task**

- با استفاده از این پنجره در آندروید استودیو می‌توان دستورات Gradle را مشاهده و یا با کلیک روی هرکدام از آن‌ها، آن را اجرا کرد.
- پنجره:Gradle Console

- **gradle-console**

- در پنجره Gradle Console خروجی دستورات Gradle به همراه نتیجه و پیغام‌های خطای آن‌ها نمایش داده می‌شود.

اضافه کردن Dependency به پروژه:

- وقتی در حال توسعه اپ اندرویدی هستید، همیشه لازم هست تا dependency هایی به پروژه اضافه کنید.
- چند راه برای اضافه کردن Dependency به پروژه وجود دارد:
- ۱ -قرار دادن فایل موردنظر در داخل پوشه libs در داخل پوشه پروژهlibs
- ۲ -استفاده از تب dependencies در قسمت project structure در اندروید استودیو در ویندوز با زدن کلیدهای ترکیبی alt+ctrl+shift+sمی‌توان پنجره project structure را مشاهده کنید
- gradle-project-structure
- ۳ -اضافه کردن dependency به صورت مستقیم در قسمت dependencies فایل build.gradle پروژه
- gradle-dependencies
- اضافه کردن dependency به هر یک از روش‌های بالا در نهایت باعث اضافه شدن dependencyبه فایل build.gradle در قسمت dependenciesخواهد شد و بلافاصله بعد از اضافه کردن یک dependency فایل build.gradle به‌روزرسانی و dependency موردنظر به پروژه اضافه می‌شود.

نمایش رشته ها اندروید استودیوString

android:text="Start"

- در یک اپلیکیشن ساده تعریف عبارت‌ها و متن‌ها به حالت پیش فرض که در بالا مشخص می‌باشد در داخل فایل xml یا از بخش Attribute برای هر View تعریف می‌کنیم، شاید ساده و راحت باشد. اما برای یک اپلیکیشن که از چندین اکتیویتی تشکیل شده و یا محتوای زیادی را شامل می شود، قطعا دردسرساز خواهد شد. فرض می کنیم اپلیکیشن ما شامل ۵ اکتیویتی بوده که در انتهای هرکدام دکمه ای با متن "مراجعه به وب سایت ما" قرار داده شده. در حالت عادی شما باید در هر اکتیویتی که دکمه را تعریف کرده اید، متن را هم تعیین کنید. حالا تصمیم گرفته ایم "مراجعه به وب سایت ما" را به "بازدید از وب سایت" تغییر دهیم. تنها چاره این است که تغییرات را در هر ۵ اکتیویتی روی متن اعمال کنیم که زمان بیشتری از توسعه دهنده می گیرد و علاوه بر آن احتمال وقوع اشتباه را نیز افزایش می دهد. یا بخواهیم زبان اپلیکیشن را عوض کنیم. باید تمامی دکمه ها و متن ها و تکست فیلدها را یک به یک پیدا کرده و متون و کلمات فارسی را با زبان انگلیسی جایگزین کنیم. ضمن اینکه این تکرار متن بخصوص در متن های با تعداد کاراکتر بالا باعث شلوغی کد و در نتیجه سنگینی و افزایش حجم نیز خواهد شد. روش بهینه استفاده از رشته ها (String) می باشد. در ادامه به نحوه کار با این روش می‌پردازیم.

نمایش رشته ها اندروید استودیوString

- اگر به ساختار پروژه اندروید استودیو خود نگاهی بیندازید مشاهده می‌کنید در بخش درختی و در مسیر زیر یک فایل بنام strings.xml وجود دارد.

- App\res\values\strings.xml

- اگر روی این فایل strings.xml دوبار کلیک کنید محتوی آن قابل مشاهده خواهد بود. که بطور پیشفرض فقط شامل تگ(Tag) زیر می‌باشد.

```
<resources>
    <string name="app_name">My Application</string>
</resources>
```

- برای تعریف یک یا چند متن بصورت رشته‌ای باید در داخل تگ resources بصورت زیر اقدام کنیم

```
<string name = "btn_login"> ورود به سایت</string>
```

- یعنی در هربار تعریف از تگ string بشکل بالا استفاده خواهیم کرد و یک متغییر رشته‌ای که در این مثال آن را btn_login نامگذاری کرده‌ایم بکار خواهیم برد و عبارات و متون دلخواه را خواهیم نوشت و در ادامه با روش فراخوانی این متغییرها آشنا خواهیم شد.

فراخوانی رشته های‌تعریف شده درstrings.xml

- برای فراخوانی رشته‌های تعریف شده بصورت **Hardcoded string** در فایل **activity*.xml** در هر تگ که اقدام به تعریف **View** می‌کنیم در داخل تگ مربوطه و در خاصیت **android.text:** بصورت زیر فراخوانی می‌کنیم.

```
android:text="@string/btn_login"
```

- یعنی در خاصیت **text** مربوط به تگ **View** مورد نظر عبارت **@** را نوشته و از لیست باز شده از زیرگروه **string** متغییر رشته‌ای تعریف شده را انتخاب می‌کنیم.

کاربرد واحد های اندازه گیری dp و فونت sp اندروید استودیو

- ***اندازه گیری dp و فونت sp***

- سوالی که در ذهن تمامی افراد وقتی برای اولین بار با اندروید استودیو کار می کنند پیش می آید اینکه وقتی می خواهید عرض و ارتفاع و یا مارجین به دکمه ها و اشیاهای داخل اکتیویتی اعمال کنید باید از واحد dpاستفاده کنید.این واحد مخفف کلمه Density-Pixels هست که یک واحد اندازه گیری دقیق برای گوشی اندروید است .

- **فونت سایز : Scale_Pixels**

- Scale_Pixelsمخفف spکه برای فونت سایز استفاده می شود در برنامه نویسی اندروید بسیار هوشمند عمل می کند زیرا با تغییر فونت گوشی شما از بخش Settings واکنش نشان می دهد و سایز فونتی که شما تنظیم کردید در بخش تنظیمات گوشی با توجه به آن تغییرات واکنش نشان می دهد و سایز فونت بزرگ و یا کوچک می شود.

متدfindViewById

این متد برای بدست آوردن شناسه(نام یا ID) ابزارهای قرار گرفته روی اکتویتی‌ها استفاده می‌شود. توجه داشته باشید که هر ابزاری روی اکتویتی قرار می‌دهیم باید داری یک نام منحصر بفرد باشد تا بتوانیم توسط آن نام به این ابزارها دسترسی داشته و متدها و خصوصیات مربوط به آن را کنترل نماییم. حالت کلی استفاده از این متد بصورت زیر می‌باشد:

۱) ابتدا از کلاس مربوط به View یک آبجکت تعریف می‌نماییم.

۲) سپس آبجکت تعریف شده را با متد findViewById مقداردهی می‌نماییم.

✓ در ادامه یک نمونه با فرض وجود در اکتویتی را مشاهده می‌نمایید.

```
Button button;
button = findViewById(R.id.btnSendAgain);
```

ابزار Button

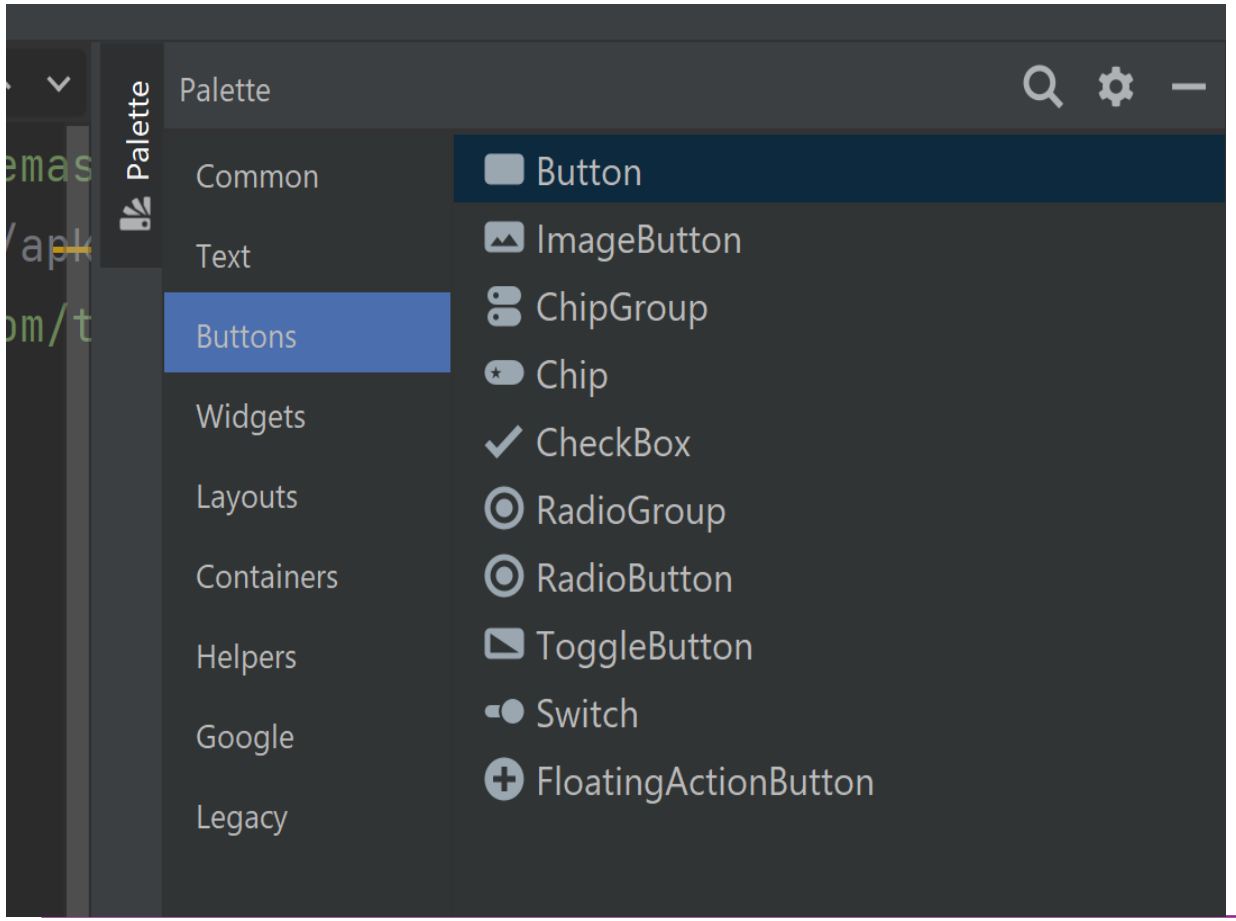
- همانگونه که در تمام برنامه‌های ویندوز و تحت وب و یا هر نرم‌افزار و App که وجود دارد ابزار Button معمولاً یک جزء جدا ناپذیر برنامه‌های نوشته شده می‌باشد این ابزار نیز با تمام ویژگی‌هایی که از آن شناخت دارید در اندروید و برنامه‌های جاوا نیز وجود دارد از طریق آن می‌توانیم یکسری رویدادها یا متدها را برحسب نیاز و لمس شدن دکمه(کلیک‌شدن) فراخوانی نماییم. برای اضافه کردن این ابزار همانند سایر ابزارهای موجود از ۲ روش معمول در این مبحث استفاده خواهیم کرد.

۱. روش اول استفاده از عمل Drag & Drop که می‌توانیم از بخش Palette و زیر گروه Buttons یکی از دکمه‌های موجود را روی اکتیویتی درگ نماییم.

۲. روش دوم استفاده از بخش Xml مربوط به اکتیویتی هست که می‌توان با استفاده از TAG مربوط به ساخت دکمه از آن استفاده نمود. در ادامه هر دو روش را مشاهده خواهید کرد.

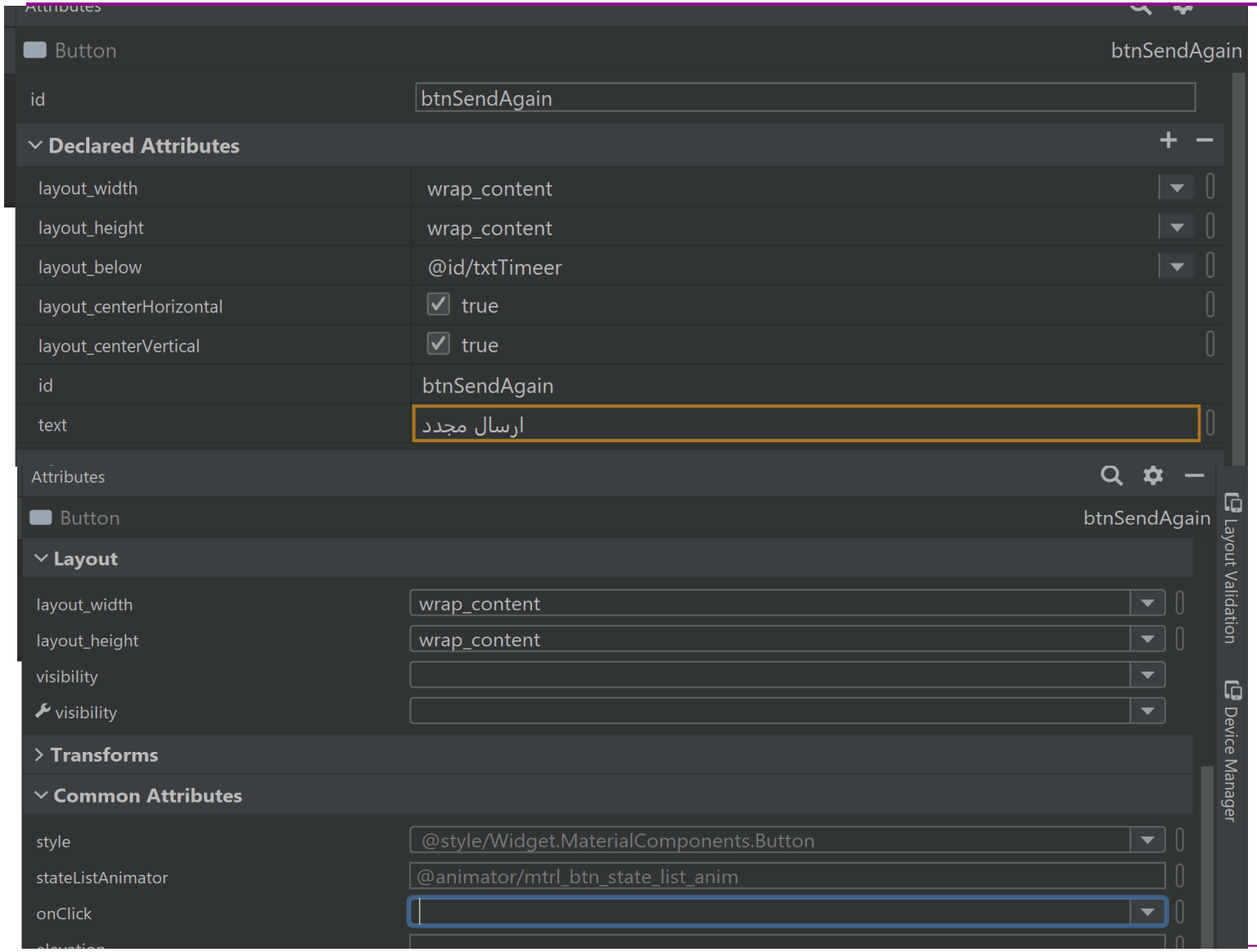
اضافه کردن Button به اکتیویتی

توجه داشته باشید روند برنامه نویسی و طراحی برنامه‌ها و چینش اشیا بر روی اکتیویتی‌های موجود یک روش سلیقه‌ای و شخصی می‌باشد.و نسبت به سلیقه هر فرد برنامه نویسی و طراحی متفاوت خواهد بود.



- با توضیحات فوق یعنی شما هیچ اجباری برای قرار دادن و چینش اشیاء روی اکتویتی ندارید و روش کاملاً شخصی و راحتی که با آن تعامل بهتری دارید را استفاده نمایید. درنظر بگیرید بمحض استفاده از روش چینش دستی کدهای XML معادل آن، اتوماتیک وار تولید و در محل مورد نظر درج خواهد گردید.

تنظیمات و مشاهده Attributeهای ابزارها



در تصاویر روبرو یکسری از خصوصیات مربوط به ابزار Button را مشاهده می‌نمایید توجه داشته باشید این خصوصیات در بخش طراحی اندروید استودیو(حالت Design) و بخش Attributes قرار دارد و نسبت به هر ابزار انتخاب شده متفاوت خواهد بود یعنی هر ابزار یا اصطلاحاً View برای خود علاوه بر تنظیمات عمومی یکسری تنظیمات منحصر بفردی خواهد داشت. در ضمن دقت داشته باشید تمام این تنظیمات نیز با استفاده از کدنویسی و از طریق XML قابل کدنویسی و تغییر می‌باشد.

افزودن Button با استفاده از کدهای XML

```
<Button
    android:id="@+id/btnSendAgain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/txtTimeer"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:text="ارسال مجدد"/>
```

- دستورات مقابل ۱ عدد button روی اکتویتی با مشخصات مشخص شده ایجاد خواهد کرد. همانطور که قبلاً هم اشاره شد می‌توانیم تمام خصوصیات و تنظیمات مربوط به ابزار را از طریق کدهای XML انجام دهیم. توجه داشته باشید بعضی ابزارها خاصیتی بنام textAllCaps دارند و در صورتی در حالت True قرار بگیرد تمام عبارت نوشته شده روی ابزار با حروف بزرگ دیده می‌شود(پیشفرض در دکمه) و اگر بصورت False قرار بگیرد متن دقیقاً در حالتی که نوشتیم نمایش داده خواهد شد(برای عبارات لاتین).

خصوصیات و متدهای Button

- متد ***setOnClickListener*** :
  - این متد گوش به زنگ میماند که Button روی اکتیویتی کلیک شده است یا خیر و شامل متد داخلی دیگری بنام onClick() می‌باشد در داخل آن متد می‌توانیم دستورات موردنیاز خود را برای انجام کارهای موردنظر بنویسیم از قبل می‌دانید این دستورات خود می‌توانند یک متد دیگر یا مجموعه‌ای از دستورات مختلف باشند. همچنین متد داخلی onClick() یک پارامتر ورودی از نوع کلاس View دریافت می‌کند و می‌توانیم توسط این پارامتر ورودی در صورت نیاز یک مجموعه دستورات شرطی قرار دهیم تا اگر چندین Button روی اکتیویتی باشد تشخیص دهیم کدام Button کلیک(لمس)شده است. در اسلاید بعدی حالت کلی متد ***setOnClickListener*** را مشاهده خواهید کرد.
- متد ***setOnLongClickListener***:
  - این متد زمانی روی خواهد داد که روی دکمه مشخص عمل کلیک طولانی(LongClick) انجام بگیرد بقیه توضیحات این متد نیز مانند متد setOnclickListener می‌باشد.

متد setOnClickListener

در تصویر زیر حالت کلی مربوط به متد setOnClickListener را مشاهده می‌نمایید همانطور که مشخص هست خود متد اصلی شامل متد داخلی دیگری بنام OnClick() می‌باشد تا بر اساس Button کلیک شده دستورات مشخص را انجام دهد.(در این نمونه مثال این متد را مستقیم روی یک Button خاص فعال کرده‌ایم)

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        /*
        کدهای خود را در این بخش وارد نمایید
        */

    }
});
```

نکته مهم:

توجه داشته باشید هنگام تعریف Button می‌توان از طریق کدهای XML و خاصیت OnClick به دکمه موجود تاکید داشته باشیم تا متد خاصی را اجرا نماید. همچنین این خاصیت از طریق Attributes نیز قابل اعمال هست.

متد setOnLongClickListener

در تصویر زیر حالت کلی مربوط به متد setOnLongClickListener را مشاهده می‌نمایید همانطور که مشخص هست خود متد اصلی شامل متد داخلی دیگری بنام OnLongClick() می‌باشد تا بر اساس Button کلیک شده دستورات مشخص را انجام دهد.

```
button1.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {
        /*
        کدهای خود را در این بخش وارد نمایید
         */

        return false;
    }
});
```

در این نمونه مثال و نمونه های قبلی این متد روی یک button خاص فعال شده است بنابراین نیازی به بررسی مقدار ورودی شی view نمی‌باشد.

استفاده از رنگ

- رنگ‌ها در برنامه‌نویسی و طراحی برنامه‌ها جز جدایی ناپذیری از ساختار برنامه‌ها می‌باشند و با استفاده از رنگ‌های موجود در جعبه رنگ می‌توان جلوه و زیبایی خاصی به طراحی‌های انجام شده اعمال نمود. در ادامه مباحث به شیوه‌های مختلف استفاده از رنگ در اندروید استدیو خواهیم پرداخت
- روش اول استفاده از TAGهای موجود XML
- روش دوم استفاده از رنگ‌های موجود در کتابخانه Color

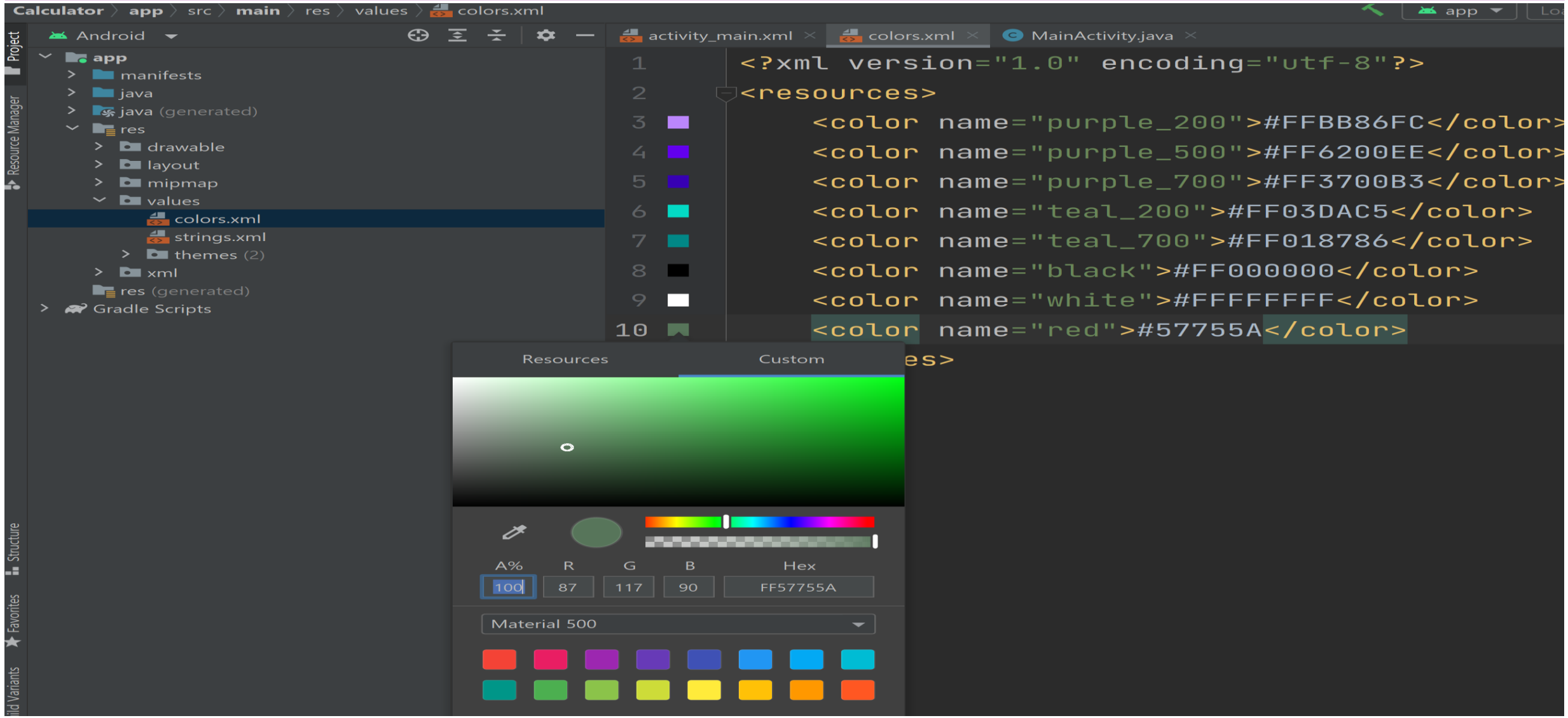
تعریف متغییرهای رنگ

- برای دسترسی به فایل **colors.xml** که محتوی آن مربوط به متغیرهای رنگ می‌باشد وارد مسیر زیر می‌شویم.

- **App\res\values\colors.xml**

محتوی فایل مربوطه تگ **Resource** می‌باشد که شامل زیر تگ‌های **color** بوده و داخل آن اقدام به تعریف متغیر و مقدار رنگ می‌نماییم. این روش تعریف متغیر و استفاده از آن قبلاً در خصوص رشته‌ها مورد بررسی قرار گرفته است. همانطور که در کدهای مشخص شده مشاهده می‌شود یک متغیر با نام **red** و با یک مقدار رنگ بصورت کد هگزا تعریف شده است.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <color name="red">#FF0000</color>
</resources>
```


Calculator app src main res values colors.xml
app
Android
activity_main.xml
colors.xml
MainActivity.java
app
manifests
java
java (generated)
res
drawable
layout
mipmap
values
colors.xml
strings.xml
themes (2)
xml
res (generated)
Gradle Scripts
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3 <color name="purple_200">#FFBB86FC</color>
4 <color name="purple_500">#FF6200EE</color>
5 <color name="purple_700">#FF3700B3</color>
6 <color name="teal_200">#FF03DAC5</color>
7 <color name="teal_700">#FF018786</color>
8 <color name="black">#FF000000</color>
9 <color name="white">#FFFFFFFF</color>
10 <color name="red">#57755A</color>
es>
Resources
Custom
A%
R
G
B
Hex
100
87
117
90
FF57755A
Material 500
Project
Resource Manager
Structure
Favorites
ild Variants

نحوه استفاده از رنگ تعریف شده

- برای استفاده از مقادیر رنگ تعریف شده بصورت متغیر XML می‌توانیم نام متغیر رنگ را در هنگام طراحی با استفاده از خاصیت مربوطه به View مورد نظر مقدار دهی نماییم در ادامه یک نمونه از مقدار دهی مربوط به این روش را مشاهده می‌نمایید.

```
63
64      <Button
65          android:layout_width="wrap_content"
66          android:layout_height="wrap_content"
67          android:text="HaHa"
68          android:onClick="clickme"
69          android:textAllCaps="false"
70          android:textColor="@color/red"/>
```

در این نمونه مثال یک دکمه ساده طراحی شده است و با استفاده از خاصیت textColor مقدار رنگ تعریف شده در فایل colors.xml را بعنوان مقدار رنگ متن دکمه طراحی شده تنظیم نموده‌ایم. توجه داشته باشید برای استفاده از این روش حتماً در فایلXML مربوط به Viewمورد نظر مقدار دهی را انجام دهیم.

تغییر رنگ با کلاس Color

- با استفاده از دستورات جاوا در محیط برنامه نویسی اندروید می‌توانیم اقدام به تغییر رنگ نماییم برای این منظور از کلاس Color استفاده خواهیم کرد توجه داشته باشید این کلاس در داخل کلاس اصلی graphics قرار دارد که باید به برنامه import شود. نکته مهم که مدنظر باید قرار داد هنگام استفاده از کلاس Color کتابخانه بصورت خودکار به داخل برنامه import خواهد شد و نیاز به درگیر شدن جهت import نمی‌باشد. در هر صورت اگر مشکلی پیش آید می‌توان روی نام کلاس Color رفته و با زدن Alt + Enter کتابخانه را بصورت دستی به برنامه import نمود.

```
import android.graphics.Color;
```

```
editText.setBackgroundColor(Color.rgb(200,100,190));
editText.setTextColor(Color.parseColor("yellow"));
editText.setHighlightColor(Color.GREEN);
```

ابزار EditText

✓ این ابزار برای ایجاد یک جعبه متن ورود اطلاعات از طریق صفحه کلید مورد استفاده قرار می‌گیرد. ابزار EditText نیز دقیقاً مانند سایر ابزارهای موجود می‌تواند از طریق TAGهای XML یا از طریق عمل Drag & Drop روی اکتویتی قرار گیرد.

➢ یکسری از خصوصیات و متدهای ابزار EditText

- متد setText
- متد getText
- متد setBackgroundColor
- متد setTextColor
- متد requestFocus
- متد addTextChangedListener

- خاصیت setHint
- خاصیت inputType
- خاصیت textAllCaps

متدهای مربوط به addTextChangedListener

- این متد زمانی فعال خواهد شد که در جعبه متن عملیات تغییر کردن متن(نوشتن عبارت یا حذف متن داخل جعبه) انجام شود که شامل سه تابع به ترتیب زیر می باشد.

```
edtName.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {

    }
});
```

beforetextchanged

این متد برای شمارش کاراکترها، حرف اول عبارتها، حرف آخر عبارت و مقدار کاراکتری که جایگزین مقدار قبلی می شود را نمایش می دهد.

ontextchanged

این متد زمانی صدا زده می شود که کاراکترها را می شمارد، وکاراکترهای جدید را به جای قدیمی جایگزین می نماید.

aftertextchanged

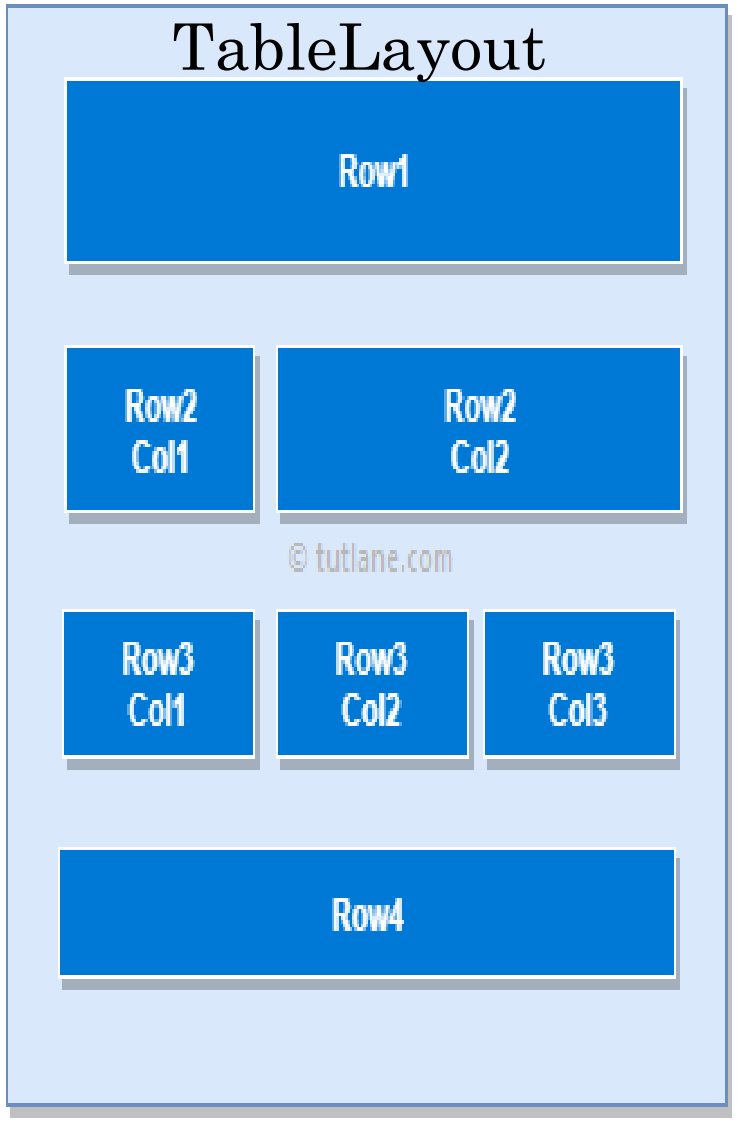
این متد زمانی فراخوانی می شود که کاربر اطلاعاتی را وارد کرده است، و قرار است این اطلاعات نمایش داده شود، و بعد از وارد شدن متن مورد نظر متد فراخوانی میگردد.

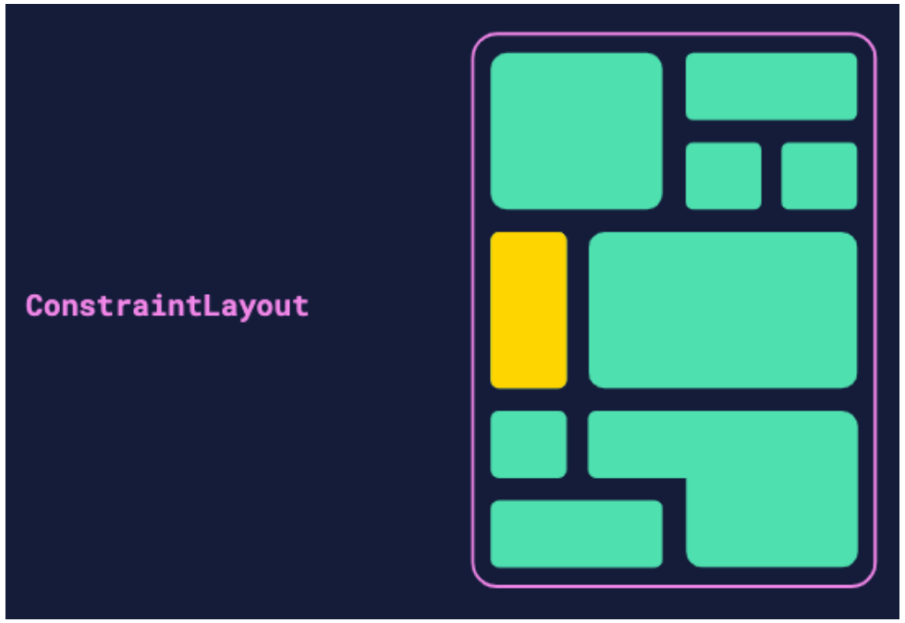
مثال برای متد addTextChangedListener

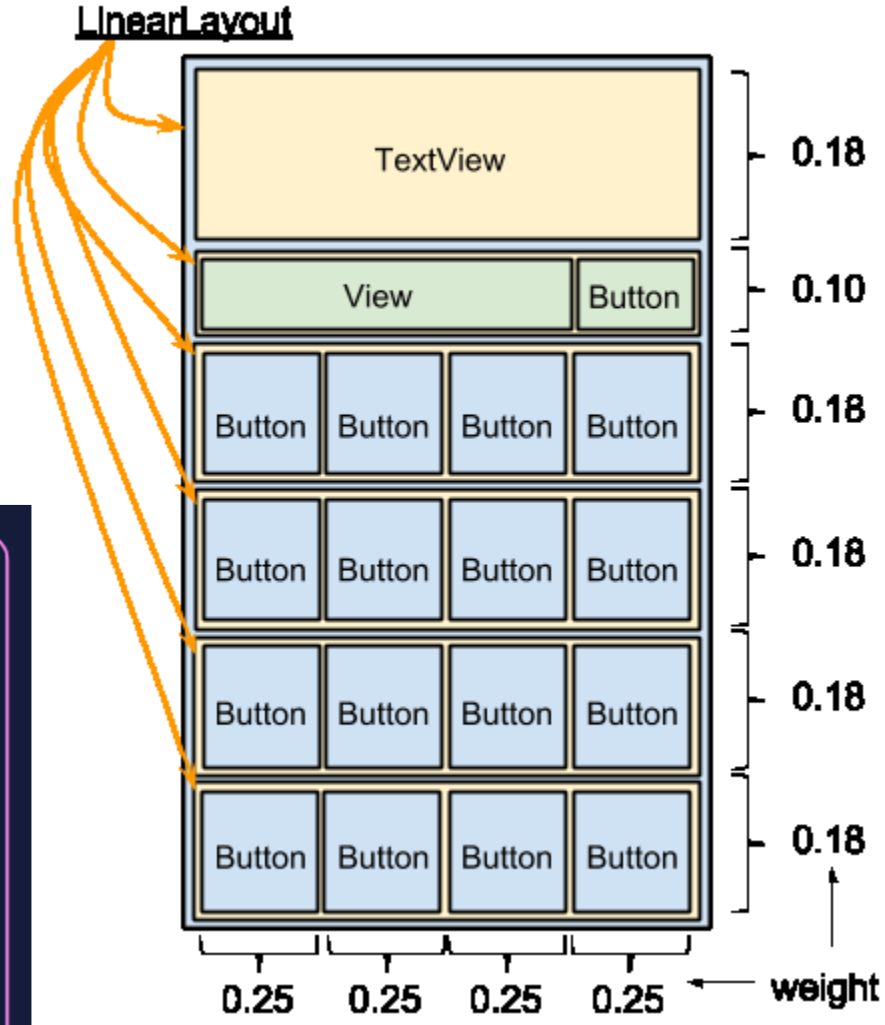
```
edtName.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {


    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        Toast.makeText( context: MainActivity.this,  text: edtName.getText().toString() + " "+ count
                + " " + before + " " + start + " " + s, Toast.LENGTH_SHORT).show();
    }

    @Override
    public void afterTextChanged(Editable s) {

    }
});
```

در این مثال از متد *Toast* برای نمایش مقادیر وارد شده و همچنین شمارش کارکترهای وارد شده و همچنین اولین حرف آخرین عبارت(آخرین کلمه) استفاده شده است.

لایه‌ها در اندروید







آشنایی با Layout ها

- مهمترین ابزارهایی که در طراحی ظاهر برنامه‌ها مورد استفاده قرار می‌گیرد تا از آن ابزارها استفاده کرده و برای قرار گیری سایر Viewها روی اکتویتی استفاده کنیم ابزارهای Layout می‌باشند این Layoutها هرکدام نحوه چینش خاصی را مطرح می‌کنند و هرکدام شامل یکسری ویژگی‌های عمومی و همچنین یکسری ویژگی‌های اختصاصی می‌باشند. ابزارهای چینش به ما امکان می‌دهند تا بتوانیم در سریعترین حالت و در محل های دلخواه Viewهای موردنیاز را قرار دهیم. در حالت کلی می‌توانیم این Layoutها را شبیه به تگ div یا table در Html مقایسه نماییم. توجه داشته باشید اکتویتی باید شامل یکی از Layout ها باشد. که بعنوان Layout والد(Parent)شناخته می‌شود.
- نکته مهمی که نیاز به توجه هست می‌توان از ابزارهای چینش بصورت تودرتو نیز استفاده نمود. یعنی بعنوان مثال در داخل ابزار چینشTable Layout می‌توانیم از ابزار چینش Linear Layout استفاده نماییم

☜انواع Layoutها:

1) Relative Layout
2) Linear Layout
3) Frame Layout
4) Table Layout
5) Constraint Layout

Relative Layout

- این امکان را فراهم می کند تا مشخص کنید هر ویجت با کدام یک از ویجت‌های داخل صفحه در ارتباط باشد. در واقع این روش چینش یک روش طرح نسبی(فامیلی) می‌باشد. موقعیت هر Viewمی تواند از رابطه با والد خودش (که Relative Layoutباشد) یا Viewهای دیگری که حکم فامیلی را دارند شکل بگیرد.
- با استفاده از Relative Layout میتوان دو عنصر را از لبه سمت راست تراز کنیم، یا یک عنصر را زیر عنصر دیگری قرار دهیم و یا در صورت نیاز آن را به وسط صفحه منتقل کنیم به صورت پیش فرض تمامی عناصرها(ویجت ها) زمان منتقل شدن به لایه چینش Relative Layout در سمت بالا-چپ صفحه طراحی قرار می‌گیرند. بنابراین باید با استفاده از تنظیمات متنوع و اختصاصی که برای RelativevLayout تعریف شده موقعیت عناصر قرار گرفته روی صفحه (ویجت ها) را مدیریت و تنظیم نماییم.

Linear Layout

- توسط ابزار چینش خطی می‌توانیم عناصر (ویجت‌ها) را بصورت های افقی و عمودی روی صفحه طراحی قرار دهیم توجه داشته باشید که این ابزار چینش خاصیتی بنام orientation دارد که وضعیت این نوع چینش را در حالت افقی(horizantal) یا حالت عمودی (vertical) مشخص میکند. می‌توانیم از این ابزار چینش و استفاده آن بصورت تودرتو و در وضعیت‌های افقی یا عمودی روش چینش Table Layout را نیز شبیه سازی نماییم. اما توجه داشته باشید همانطور قبلاً مطرح شد هر ابزار چینش یکسری خصوصیات خاص خود را دارد که در سایر ابزارهای چینش ممکن هست غیر قابل دسترسی باشد.

Frame Layout

- این ابزار چینش شبیه یک حفره یا در واقع شبیه iframe موجود در طراحی وب (فریم شناور) می‌باشد که روی صفحه طراحی(اکتویتی)قرار میگیرد و می توانیم از این روش برای نمایش یک View/wedgetدیگر استفاده نماییم میتوانیم از این ابزار استفاده کرده و چند لایه یا view را روی هم قرار دهیم و معمولاً ۲ کاربرد ویژه دارد که یکی از این کاربردها در پاراگراف قبلی مطرح شد(قرار دادن چند View روی یکدیگر) و کاربرد عمده دیگر استفاده از ابزارهای Live مانند بازکردن دوربین گوشی در داخل برنامه طراحی شده یا استفاده از ویجت‌های طراحی در داخل برنامه می‌باشد. موردی که بیان شد توسط همه ابزارهای چینش قابل پیاده سازی هست اما Frame layout سبکتر و سریعتر از بقیه می‌تواند عمل نماید.

Table Layout

- توسط ابزار چینش Table Layout می‌توانیم یکسری سطر و ستون ایجاد کرده و هر View/Widget را که نیاز داشتیم در سطر و ستون مشخص قرار دهیم اگر قبلاً با تَگ Table در زبان نشانه گذاری HTML کار کرده باشید یا با آن آشنایی داشته باشید متوجه خواهید شد این ابزار چینش دقیقاً شبیه به آن عمل کرده و می‌تواند یک صفحه جدول مانند را در اختیار برنامه نویس برای چینش اشیا قرار دهد در ادامه با این روش چینش و یک مثال نمونه کار خواهیم تا بیشتر با این روش چینش آشنا شویم. توجه داشته باشید این ابزار چینش یکی از راحت ترین ابزارهایی هست که میتوان توسط آن برنامه‌هایی شبیه به ماشین حساب را طراحی نمود.

Table Layout

- این ابزار چینش برای ساخت هر سطر از تگ <TableRow> </TableRow> استفاده می‌کند که ابزارهای چینش یا همان اشیا را داخل تگ فوق قرار خواهیم داد. توجه داشته باشید بر خلاف HTML نیاز به ایجاد ستون(سلول) نیست و هر ابزاری که قرار می‌دهیم یک ستون ایجاد خواهد کرد. این ابزار چینش اشیا را بصورت افقی و پشت سرهم قرار می‌دهد.
- یکسری از خصوصیات مهم و پرکاربرد:
  - ✓ Layout_column
  - ✓ Layout_span
  - ✓ streachColumns
  - ✓ collapseColumns

Layout_column

با تنظیم نمودن این خاصیت مشخص می‌کنیم تا View/widget قرار گرفته در سطر جدول از خانه ذکرشده شروع شود توجه داشته باشید هر عنصری که روی اکتویتی و داخل ابزار چینش قرار می‌گیرد موقعیت آن از عدد 0 شروع می‌شود یعنی اگر ۳ عنصر روی صفحه قرار دهیم اولین عنصر با موقعیت 0 و عنصر بعدی با موقعیت ۱ و عنصر سوم با موقعیت ۲ مشخص خواهد شد. نمونه زیر را در نظر بگیرید. همانطور که مشخص هست این شماره گذاری ها از عدد 0 شروع شده است. در اسلاید بعدی نمونه یک کد و نتیجه آن در اندروید استودیو را مشاهده خواهید کرد.

0	1	2
Button 1	Button 2	Button 3

Layout_column

```xml
<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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"
    tools:context=".MainActivity">
    <TableRow android:gravity="center_horizontal">
        <Button
            android:text="1"
            android:layout_margin="5dp"
            android:layout_column="2"/>
        <Button
            android:text="2"
            android:layout_margin="5dp"/>
        <Button
            android:text="3"
            android:layout_margin="5dp"/>
    </TableRow>
```

همانطور که در شبیه ساز مشاهده می‌نمایید در صورتی که مقدار خاصیت layout_colmn یک ابزار را مقدار دهی نماییم آن ابزار از موقعیت مشخص شده روی صفحه قرار خواهد گرفت

Layout_span

با تنظیم نمودن این خاصیت مشخص می‌کنیم تا View/widget قرار گرفته در سطر جدول در خانه قرار گرفته به میزان مشخص شده خانه(سلول) جا بگیرد یعنی اگر بعنوان مثال مقدار این حاصیت برای عنصری برابر ۲ باشد به اندازه ۲برابر خود جا اشغال خواهد کرد میتوان این موضوع را چیزی شبیه به ادغام(Merge) خانه‌های جدول در نظر گرفت بعنوان مثال نمونه مشخص شده زیر را در نظر بگیرید که عنصر Button 2 به میزان ۲ خانه از سطر را اشغال نموده است. در اسلاید بعدی نمونه کد XML و خروجی آن را مشاهده خواهید کرد.

0	1	2
Button 1	Button 2	

Layout_span

```
<TableRow android:gravity="center_horizontal">
    <Button
        android:text="7"
        android:layout_margin="5dp"
        />
    <Button
        android:text="8"
        android:layout_span="2"
        android:layout_margin="5dp"/>

</TableRow>
```

در این نمونه Button8 معادل ۲ عنصر Button از خانه های جدول را اشغال کرده است که این امر از حالت کشیدگی آن مشخص می‌باشد.

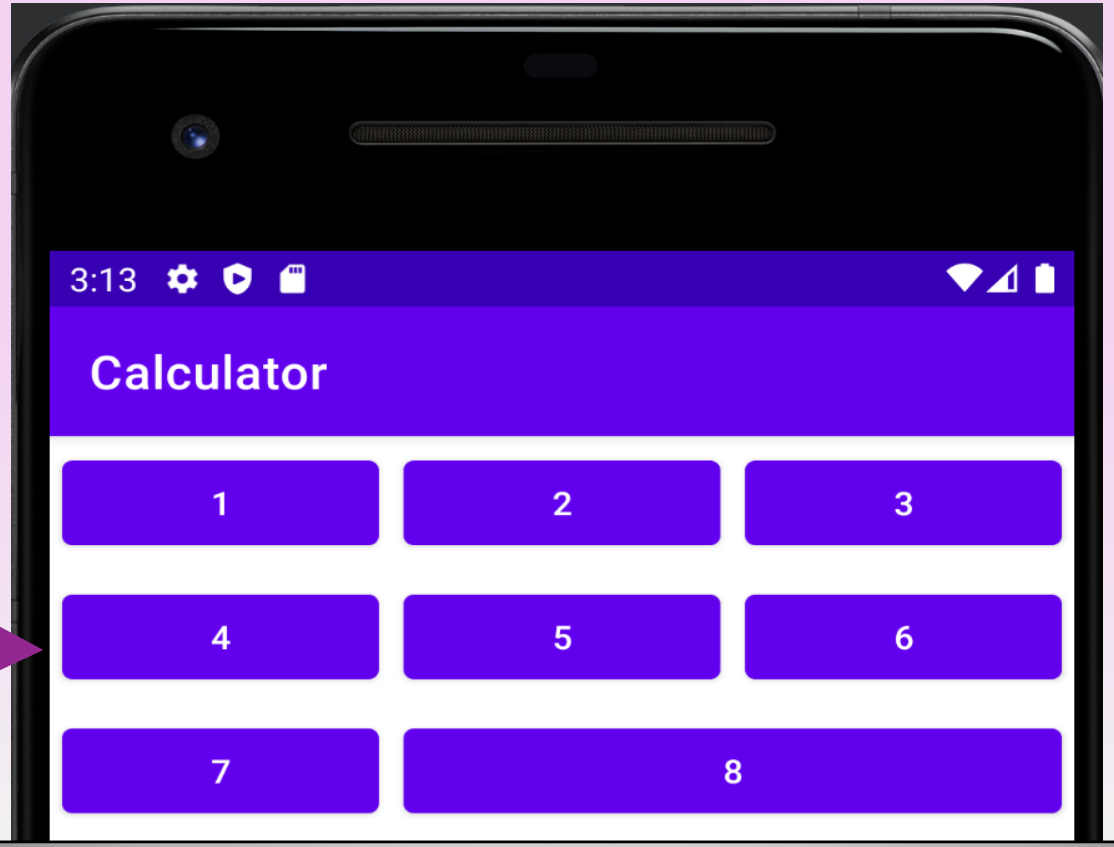
streachColumns

با تنظیم نمودن این خاصیت مشخص می‌کنیم تا View/widgetهای قرار گرفته در سطرهای جدول در خانه قرار گرفته به میزان مشخص حالت کشیدگی داشته و بنوعی یکدست از لحاظ اندازه باشند نکته مهم اول توجه داشته باشید اگر بخواهید این خاصیت اعمال شود باید تعیین کنید تمام عناصر موجود روی صفحه بصورت مرتب بدون دستکاری در موقعیت خانه ها روی اکتویتی قرار بگیرند(از خاصیتLayout_columnاستفاده نشود)بعنوان مثال نمونه مشخص شده زیر را در نظر بگیرید که تمام عناصرButton به میزان یکسان در خانه های جدول کشیدگی دارند. *نکته مهم بعدی اینست که این خاصیت باید در داخل تگ اصلی چینش یا همان Table Layout نوشته شود چون از خصوصیات اصلی آن می‌باشد.* در اسلاید بعدی نمونه کد XML و خروجی آن را مشاهده خواهید کرد.

0	1	2
Button 1	Button 2	Button 3

streachColumns با مقدار *

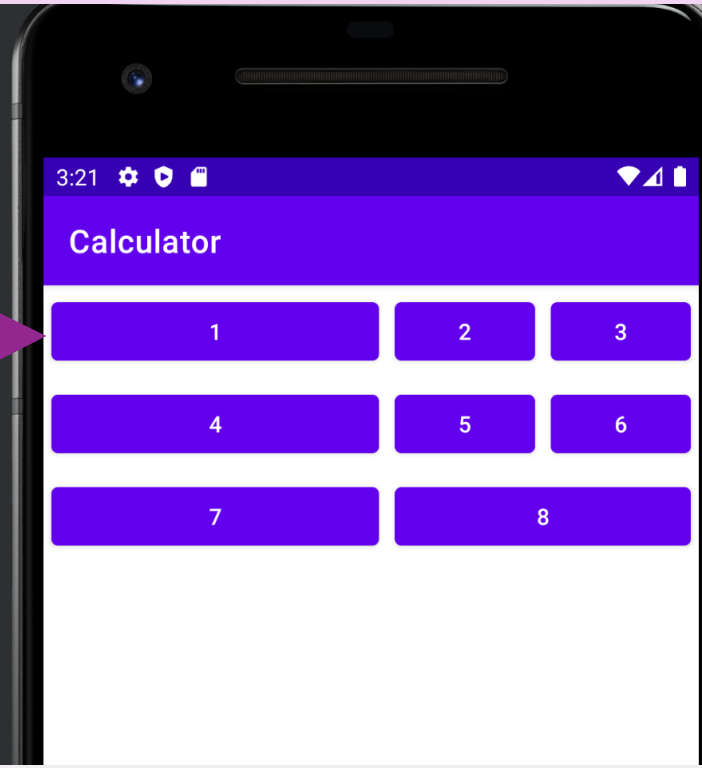
```xml
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:stretchColumns="*">
    <TableRow android:gravity="center_horizontal">
        <Button
            android:text="1"
            android:layout_margin="5dp"/>
        <Button
            android:text="2"
            android:layout_margin="5dp"/>
        <Button
            android:text="3"
            android:layout_margin="5dp"/>
    </TableRow>
```



با قرار دادن مقدار * بجای عدد مشخص می‌کنیم تمام عناصر یکدست باشند. و همانطور که مشاهده می‌کنید عناصر موجود به ترتیب موقعیت روی صفحه قرار گرفته اند

streachColumns با مقدار 0 یا همان شماره موقعیت ستون‌ها

```xml
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/androi
    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"
    tools:context=".MainActivity"
    android:stretchColumns="0">
    <TableRow android:gravity="center_horizontal">
        <Button
            android:text="1"
            android:layout_margin="5dp"/>
```



*با قرار دادن مقدار * بجای عدد مشخص می‌کنیم تمام عناصر یکدست باشند.* بجای مقدار * می‌توانیم شماره موقعیت هر ستون را مشخص نماییم تا عناصر موجود در آن ستون‌ها یکدست شوند مشاهده میکنید با قرار دادن مقدار 0 تعیین کردیم تا تمام عناصر موجود در ستون اول(ستون شماره 0) از لحاظ اندازه یکدست باشند. که می‌توان با قرار دادن و بین شماره ستون ها بقیه را نیز تنظیم نماییم.

collapseColumns

با تنظیم نمودن این خاصیت مشخص می‌کنیم تا ستون یا ستون‌هایی از دید کاربر و یا موقع طراحی مخفی شوند(در واقع حذف ستون انجام میگیرد). این خاصیت نیز مانند خاصیت streachColumns یک عدد یا مجموعه اعداد را بعنوان شماره موقعیت ستون دریافت کرده و باعث حذف ستون یا ستون‌های مشخص شده می‌نمایند. برخلاف خاصیت قبلی نمیتواند مقدار * را بعنوان همه شماره های ستون‌ها دریافت نماید و باید جداگانه شماره ستون داده یا مجموعه شماره ها را با جدا کننده , وارد نماییم.(0,1,2)

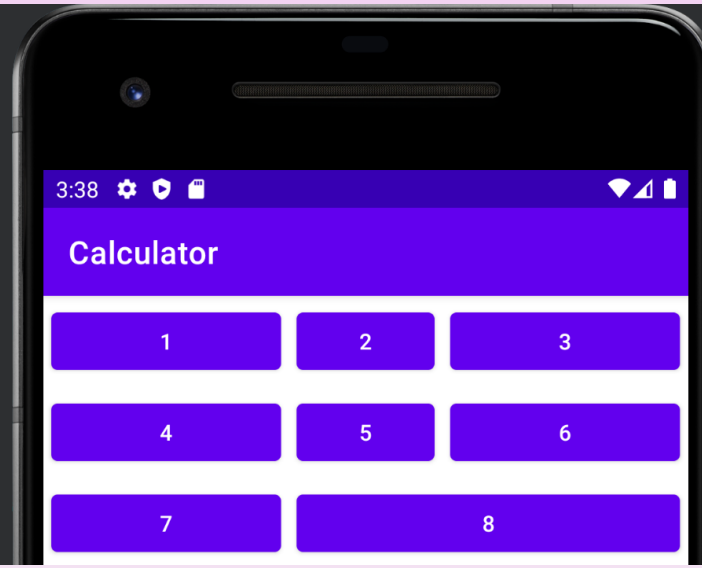
نکته مهم و قابل توجه اینست که خاصیت ذکر شده باید در داخل تگ اصلی چینش یا همان Table Layout نوشته شود چون از خصوصیات اصلی آن می‌باشد.

در اسلاید بعدی نمونه کد XML و خروجی آن را مشاهده خواهید کرد.

نمیتوانیم مقدار * را بعنوان شماره موقعیت هر ستون مشخص نماییم.ستون‌ها با مشخص شدن شماره آنها که از 0 شروع می‌شود حذف خواهند شد می‌توان با قرار دادن , بین شماره ستون ها بقیه را نیز حذف نماییم.

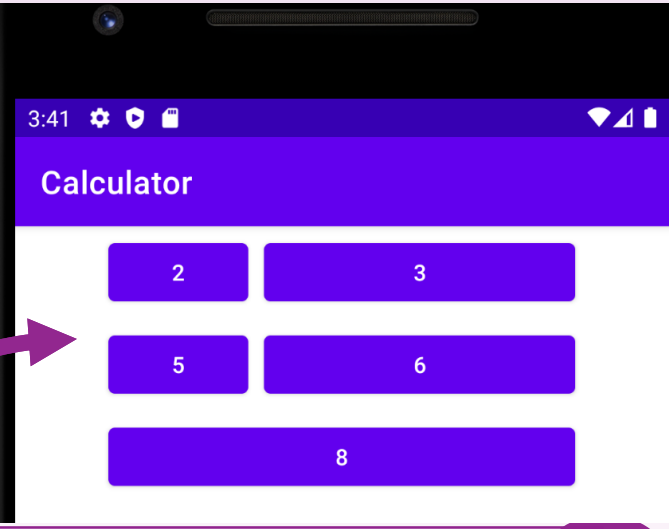
```xml
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:stretchColumns="0,2">
    <TableRow android:gravity="center_horizontal">
```

قبل از اعمال حذف ستون




```xml
<TableLayout xmlns:android="http://schemas.android.com/apk/res/androi
    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"
    tools:context=".MainActivity"
    android:stretchColumns="0,2"
    android:collapseColumns="0">
```

پس از اعمال حذف ستون



Constraint Layout

برای اولین بار در کنفرانس Google I/O در سال ۲۰۱۶ معرفی شد. اصلی ترین هدف گوگل از معرفی این Layout جدید، امکان طراحی لایه‌های پیچیده و در عین حالflat (تخت) بود بر خلاف Layoutهای قبلی که برای طراحی یک صفحه ناگزیر به استفاده از لایه‌های تو در تو و سلسله مراتبی بودیم. این یعنی دیگر لازم نیست چندین لایه RelativeLayoutو LinearLayoutو... درون یکدیگر و به صورت تو در تو تعریف شود تا بتوان یک صفحه چند قسمتی را ساخت. بنابراین علاوه بر سادگی پیاده سازی این لایه و کاهش حجم نهایی کدهای آن، در طراحی صفحات با ConstraintLayoutکارایی و Performance اکتیویتی بهبود یافته و می‌تواند در افزایش سرعت نمایش UIروی دستگاه کاربر تاثیر مثبت بگذارد.

ConstraintLayout از اندروید استودیو ۲,۲ اضافه شد و نسخه ابتدایی آن Beta 1نام داشت. در طی این چند سال مرتباً اصلاحاتی روی آن انجام و همچنین قابلیت‌های جدیدی نیز اضافه شده است. این لایه از API 9به بعد پشتیبانی می‌شود بنابراین مطمئن خواهیم بود تمامی دستگاه‌های اندرویدی که در حال حاضر در اختیار افراد هستند به خوبی می‌توانند صفحاتی که با این layoutطراحی شده‌اند را نمایش دهند.

Constraint Layout

توسط Constraint Layout می‌توانیم وضعیت قرارگیری تمامی View/Widgetها را نسبت به یکدیگر و یا از طریق Parrent مربوطه تعیین کنیم. برای مثال می‌توانیم یکTextViewرا به نحوی در زیر یک Image Viewمتصل کنیم که در جهت افقی، متن همواره در مرکز تصویر قرار گیرد. یا اینکه یک یا چند view بر اساس سایز صفحه نمایش دستگاه، در جهت افقی و در جهت عمودی همواره در مرکز قرار گرفته و علاوه بر آن، فاصله آنها از یکدیگر نیز درصدی ثابت باشد. این یعنی یک انعطاف پذیری بسیار بالا که تا حدود زیادی مشابه طراحی صفحات Responsiveدر صفحاتاینترنتی است.

توجه داشته باشید در نسخه‌های ابتدایی اندروید استودیو که Constraint Layout اضافه شده بود باید ابزار آن به صورت جداگانه در SDK و در زیر مجموعه Support Repositoryدریافت و نصب می‌شد اما در نسخه‌های اخیر، این گزینه از لیست SDK Toolsحذف شده و با سایر ابزار اصلی توسعه اندروید ادغام شده است. بطوری که لایه پیشفرض موقع ساخت پروژه جدید نیز می‌باشد.

توجه داشته باشید ابزارهایی قرار گرفته در این لایه اگر روی آنها کلیک نماییم یکسری خطوط در ۴ جهت آن نمایش داده می‌شوند که این خطوط یا خود ابزار شامل یکسری Nodeهای تنظیم نیز خواهد بود میتوانیم بصورت دستی و از طریق ماوس موقعیت ابزار مورد نظر در صفحه را تعیین و تنظیم نماییم همچنین توسط دستورات XML از طریق اکتویتی مورد نظر نیز قابلیت تنظیم و تعیین موقعیت دارد یا میتوانیم از بخش attribute در سمت راست صفحه و رفتن به بخش Layouts نیز تنظیمات موردنیاز را انجام دهیم. در ضمن میتوان در صورت نیاز خطوط اتصال مربوط به ابزار را حذف کرد و آن سمت از تعیین موقعیت را از موقعیت دهی خارج نمود.

آموزش استفاده از وب ویو : *WebView*

- در طراحی برنامه‌ها برای اندروید می‌توانیم یک **Browser** برای خودمان طراحی کنیم برای اینکار باید از **Object** مربوط به مشاهده وب بنام **WebView** بهره ببریم برای اینکار به صورت زیر عمل می‌کنیم.
- ۱)ابتدا یک **View** از نوع **WebView** و از بخش **Palette** و زیر گروه **widget** بر روی اکتیویتی اضافه می‌کنیم. یا از طریق فایل **activity*.xml** با استفاده از تگ مربوط به **View** مشاهده صفحات وب بصورت زیر تعریف می‌کنیم.

```
<WebView
    android:layout_width="match_parent"
    android:layout_height="match_parent"></WebView>
```

- سپس در داخل کدهای اکتیویتی که **WebView** روی آن قرار دارد ابتدا یک **Object** از نوع **WebView** را **Cast** می‌کنیم و با استفاده از متد **findViewById** شناسه **View** را پیدا کرده و با متد loadUrl صفحه وب را بارگذاری می‌کنیم.

WebView

- با استفاده از تگ‌های زیر در داخل فایل activity*.xml اقدام به تعریف یک WebView می‌نماییم.

```
<WebView
    android:layout_width="match_parent"
android:layout_height="match_parent"
    android:id="@+id/web"></WebView>
```

- سپس در داخل فایل activity*.java اقدام به نوشتن دستورات مربوط به شناسایی و Cast نمودن View می‌نماییم.

```
WebView webView;
webView = findViewById(R.id.web);
webView.loadUrl("http://afo.ac.ir/");
```

نکته: در بعضی از مثال‌ها برای مشخص کردن اسم فایل * قرار داده‌ام به این دلیل که اسم فایل ممکن است هرچیزی باشد. بعنوان مثال Activity_main.xml

دادن مجوز دسترسی

- پس از نوشتن کدهای مشخص شده قبلی و اجرا کردن برنامه در شبیه‌ساز یا دستگاه متصل به سیستم کامپیوتری هنگام لود شدن صفحه اینترنت با پیغام خطای زیر مواجه خواهیم شد.

- برای رفع این مشکل باید درداخل فایل AndroidMainfest.xml که در داخل فولدر mainfestes قرار دارد مجوز دسترسی به منابع را بدهیم.برای این منظور در فایل مورد نظر قبل از تگ application با استفاده از تگ uses-permission طبق روال زیر مجوز را اعمال می‌کنیم.

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

تنظیمات متفاوت برای WebView

- یکسری تنظیمات متفاوت و بیشتری می‌توانیم بر روی ابزار **WebView** اعمال نماییم در ادامه به این موارد اشاره می‌نماییم:
- تنظیم اندازه قلم:

```
webView.getSettings().setDefaultFontSize(45);
```

- تنظیمات کلی و عمومی:
  - تنظیم مربوط به فعال شدن لینک توکار
  - تنظیم فعال شدن جاوا اسکریپت
  - فعال شدن حالت بزرگنمایی داخلی
  - فعال شدن حالت اووراید مد برای مشاهده صفحات

```
button.setOnClickListener(new View.OnClickListener() {
    @SuppressLint("SetJavaScriptEnabled")
    @Override
    public void onClick(View view) {
        webView.setWebViewClient(new WebViewClient());
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setLoadWithOverviewMode(true);
        webView.getSettings().setBuiltInZoomControls(true);
        webView.loadUrl("https://afo.ac.ir/");
    }
});
```

متد onKeyDown برای برگشت به لینک قبلی

- توسط متد **onKeyDown** می‌توانیم لمس شدن دکمه **Back** روی دیوایس را کنترل کنیم. در حالت معمولی اگر این کلید لمس شود باعث برگشت به اکتیویتی قبلی شده و یا از اکتیوتی فعلی خارج خواهد شد. می‌خواهیم کدهایی داشته باشیم که در صورت لمس نمودن کلید **Back** اگر در داخل ابزار **WebView** وارد لینکی شده باشیم به صفحه قبلی برگردیم و از برنامه خارج نشویم درواقع شبیه سازی دکمه **Back** مربوط به مرورگرهای اینترنتی را شبیه سازی نماییم.
- نکته مهم این است که حتماً باید دستور `webView.setWebViewClient(new WebViewClient());` اجرا نماییم تا لینک‌های کلیک شده در داخل خود ابزار WebView باز شوند و امکان برگشت داشته باشیم.

کدهای متد onKeyDown

```java
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if(event.getAction() == KeyEvent.ACTION_DOWN){
        if (keyCode == KeyEvent.KEYCODE_BACK) {
            if (webView.canGoBack()) {
                webView.goBack();
            } else {
                finish();
            }
            return true;
        }
    }
    return super.onKeyDown(keyCode, event);
}
```

✓ نکته مهم در خصوص محل نوشتن کدها بعد از متد onCreate می باشد.

onBackPressed()

- این متد زمانی روی می‌دهد که کاربر در یک اکتیوتی از دکمه Back روی صفحه موبایل یا دستگاه اندرویدی استفاده کند در این حالت این متد فراخوانی شده و دستوارتی که داخل آن وجود دارد طبق روال اجرا خواهد شد.در ادامه با نحوه فراخوانی و همچنین درخواست تایید جهت خروج از یک اکتیویتی آشنا خواهیم شد.
- شکل کلی متد onBackPressed برای یک اکتیویتی بصورت زیر می‌باشد.

```
public void onBackPressed() {
    mesg_box_exit();
    return;
}
```

- در این متد زمانی که فراخوانی می‌شود یک تابع بنام msg_box_exit() صدا زده می‌شود که در داخل آن از یکسری دستورات برای تایید خروج یا عدم خروج سوالی پرسیده می‌شود.

تابع msg_box_exit()

- محتوی دستورات موجود در تابع **msg_box_exit**

```
public void mesg_box_exit()
{
    AlertDialog.Builder Alert_close = new AlertDialog.Builder(MainActivity.this);
    Alert_close.setTitle("هشدار")
            .setMessage("آیا قصد خروج دارید؟")
            .setPositiveButton("بله", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    finish();
                }
            })
            .setNegativeButton("خیر", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.cancel();
                }
            })
            .show();
}
```

پیغام مربوط به زدن دکمه back

متد Toast

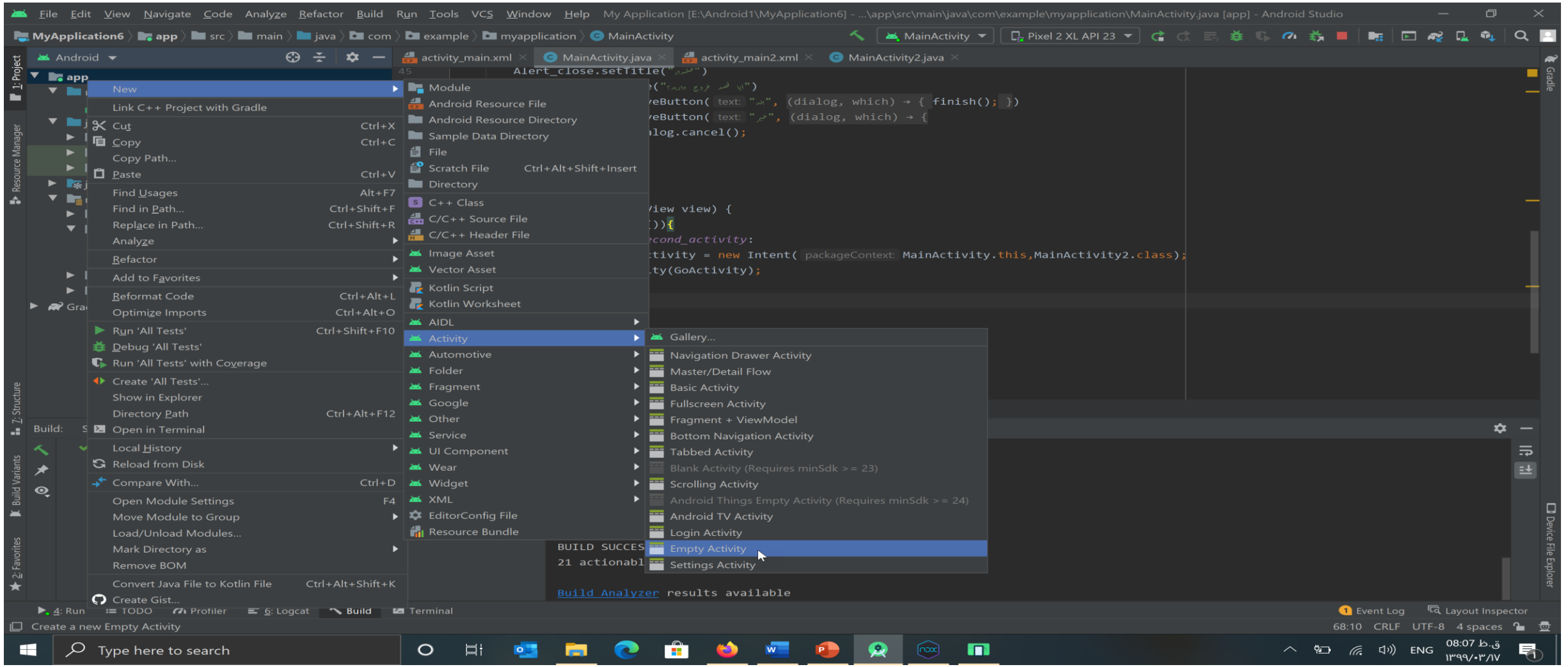
- در اندروید تابعی بنام Toast وجود دارد که توسط آن می‌توانیم در مواقع خاصی اقدام به نمایش پیغام روی صفحه نماییم شکل کلی این متد(تابع) بصورت زیر می‌باشد.

```
Toast.makeText(this, "", Toast.LENGTH_SHORT).show();
```

- this درواقع اکتیویتی(فعلی) می‌باشد که میخواهیم در آن اکتیویتی اقدام به نمایش پیغام نماییم.
- متن مورد نظر را داخل " متن دلخواه" خواهیم نوشت.
- این متد پیغام را در ۲ مدت زمان کوتاه و مدت زمان بلند نمایش می‌دهد. حالت پیش‌فرض نمایش پیغام مدت زمان کوتاه می‌باشد. برای تغییر حالت مدت زمان می‌توانیم از خاصیت LENGTH_SHORT و همچنین LENGTH_LONG استفاده نماییم.

حرکت بین Activityهای موجود

- گاهی ممکن هست که در برنامه‌های طراحی شده بیش از یک اکتیویتی داشته باشیم و لازم باشد در بین این اکتویتی‌ها بنا به شرایط جابجا شویم که در ادامه با نحوه حرکت بین اکتیویتی‌ها آشنا شده و هم اینکه با مثالی چرخه حیاط اکتیویتی‌ها را بصورت عملی مشاهده خواهیم کرد.
- برای حرکت بین اکتویتی‌ها ابتدا نیاز داریم تا حداقل ۲ اکتیویتی جداگانه داشته باشیم.برای اضافه کردن اکتویتی دوم روی app در بخش درختی Project کلیک راست می‌کنیم سپس New و بعد از آن Activity و در نهایت از لیست باز شده یکی از اکتویتی‌ها را انتخاب می‌کنیم.(پیشنهاد توسعه دهنده‌گان آپ‌های اندروید همواره انتخاب Empty Activity می‌باشد که بر اساس نیاز خود اکتیویتی را طراحی نماییم.)
- در روی هر اکتویتی یک Button با عناوینی مشخص جهت حرکت بین اکتیویتی‌ها قرار خواهیم داد که با کلیک روی هر دکمه به اکتیویتی دیگر منتقل شویم.
- در تصویر بعدی نحوه ساخت اکتیویتی را مشاهده خواهید کرد.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
My Application [E:\Android1\MyApplication6] - ...\app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio
MyApplication6 app src main java com example myapplication MainActivity
MainActivity Pixel 2 XL API 23
Android
activity_main.xml MainActivity.java activity_main2.xml MainActivity2.java
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path...
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Run 'All Tests' Ctrl+Shift+F10
Debug 'All Tests'
Run 'All Tests' with Coverage
Create 'All Tests'...
Show in Explorer
Directory Path Ctrl+Alt+F12
Open in Terminal
Local History
Reload from Disk
Compare With... Ctrl+D
Open Module Settings F4
Move Module to Group
Load/Unload Modules...
Mark Directory as
Remove BOM
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
Create Gist...
Module
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
AIDL
Activity
Automotive
Folder
Fragment
Google
Other
Service
UI Component
Wear
Widget
XML
EditorConfig File
Resource Bundle
Gallery...
Navigation Drawer Activity
Master/Detail Flow
Basic Activity
Fullscreen Activity
Fragment + ViewModel
Bottom Navigation Activity
Tabbed Activity
Blank Activity (Requires minSdk >= 23)
Scrolling Activity
Android Things Empty Activity (Requires minSdk >= 24)
Android TV Activity
Login Activity
Empty Activity
Settings Activity
BUILD SUCCES
21 actionabl
Build Analyzer results available
4: Run TODO Profiler 6: Logcat Build Terminal
Event Log Layout Inspector
Create a new Empty Activity
68:10 CRLF UTF-8 4 spaces
Type here to search
ENG
08:07 ق.ظ
۱۳۹۹/۰۳/۱۷

حرکت بین اکتیویتی‌ها

- در اکتیویتی اول یک دکمه فشاری بصورت زیر تعریف می‌کنیم.

```
<Button
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:text="Seetings"
    android:id="@+id/go_second_activity"
    android:onClick="btns_click"/>
```

- سپس در اکتیویتی مربوطه فایل کدها .java دستورات مربوط به تابع btns_click زیر را وارد می‌کنیم.

```
public void btns_click(View view) {
    switch (view.getId()){
        case R.id.go_second_activity:
            Intent GoActivity = new Intent(MainActivity.this,MainActivity2.class);
            startActivity(GoActivity);
            break;
    }
}
```

توضیحات مربوط به کدهای حرکت بین اکتیوتی‌ها

- کلاسIntent :

Intent در برنامه نویسی اندروید، یک توصیف انتزاعی از یک عملیاتی است که انجام می شود. Intent می‌تواند به عنوان شروع کننده‌ی یک اکتیویتی برای راه اندازی یک (اکتیویتی) ، فرستادن هر مولفه ای از broadcastIntentبه BroadcastReceiver، و برقراری ارتباط بین متد stratService(Intent)و سرویس‌های پس زمینه استفاده شود.ساختار کلی استفاده از کلاس Intent برای تعریف یک شی جدید بصورت زیر می‌باشد.

```
Intent GoActivity = new Intent(MainActivity.this,MainActivity2.class);
```

بعد از تعریف شی مورد نظر از کلاس Intent طبق روال فوق مشخص می‌کنیم که از کدام اکتویتی به کدام اکتیویتی حرکت خواهیم کرد فقط در اکتیویتی دوم باید کلاس آن را فراخوانی نماییم. و سپس با استفاده از متدstartActivity شی مورد نظر را که شامل اکتیویتی مدنظر می‌باشد فراخوانی می‌کنیم.

نکته مهم اینکه حتماً باید کتابخانه موردنظر Import شود که توسط دستور زیر اینکار انجام خواهد شد.

```
startActivity(GoActivity);
```

**این عمل با انتقال مکان‌نما به زیر کلاس Intent و زدن کلیدهای Alt+Enter و دوباره Enter نیز انجام می‌شود.

```
import android.content.Intent;
```

ترتیب فراخوانی متدهای اکتیویتی

هنگام اجرا شدن اکتیویتی برای بار اول متدهای زیر به ترتیب اجرا می‌شوند.

onCreate();

onStart();

onResume();

زمانی که اکتیویتی بعدی توسط اکتیوتی اول فراخوانی شد متدهای زیر به ترتیب فراخوانی میشوند.

اکتیویتی جاری

onPause();

اکتیویتی دوم

onStart();

onResume();

اکتیویتی جاری

onStop();

ادامه ترتیب فراخوانی متدهای اکتیویتی

- اگر از دکمه بک روی گوشی استفاده کنیم
- اکتیویتی جاری
- onPause();
- اکتیویتی مورد نظر
- onRestart();
- onStart();
- onResume();
- اکتیویتی جاری
- onStop();
- onDestroy();

ادامه ترتیب فراخوانی متدهای اکتیویتی

- اگر از دکمه برگشت که خودمان طراحی کردیم استفاده کنیم

- اکتیویتی جاری

- onPause();

- اکتیویتی موردنظر

- onCreate();
- onStart();
- onResume();

- اکتیویتی جاری

- onStop();

پیاده سازی چرخه حیات اکتیویتی‌ها

- همانطور که قبلاً در خصوص چرخه حیات اکتیوتی(Activity LifeCycle) مطرح شد برای حرکت بین اکتویتی‌ها متدهایی اجرا می‌شوند. در ادامه با این متدها و نحوه نوشت متدها آشنا می‌شویم.

```
@Override
protected void onResume() {
    super.onResume();
    Toast.makeText(this, "MainActivity Resume .....", Toast.LENGTH_SHORT).show();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Toast.makeText(this, "MainActivity Destroy .....", Toast.LENGTH_SHORT).show();

}
```

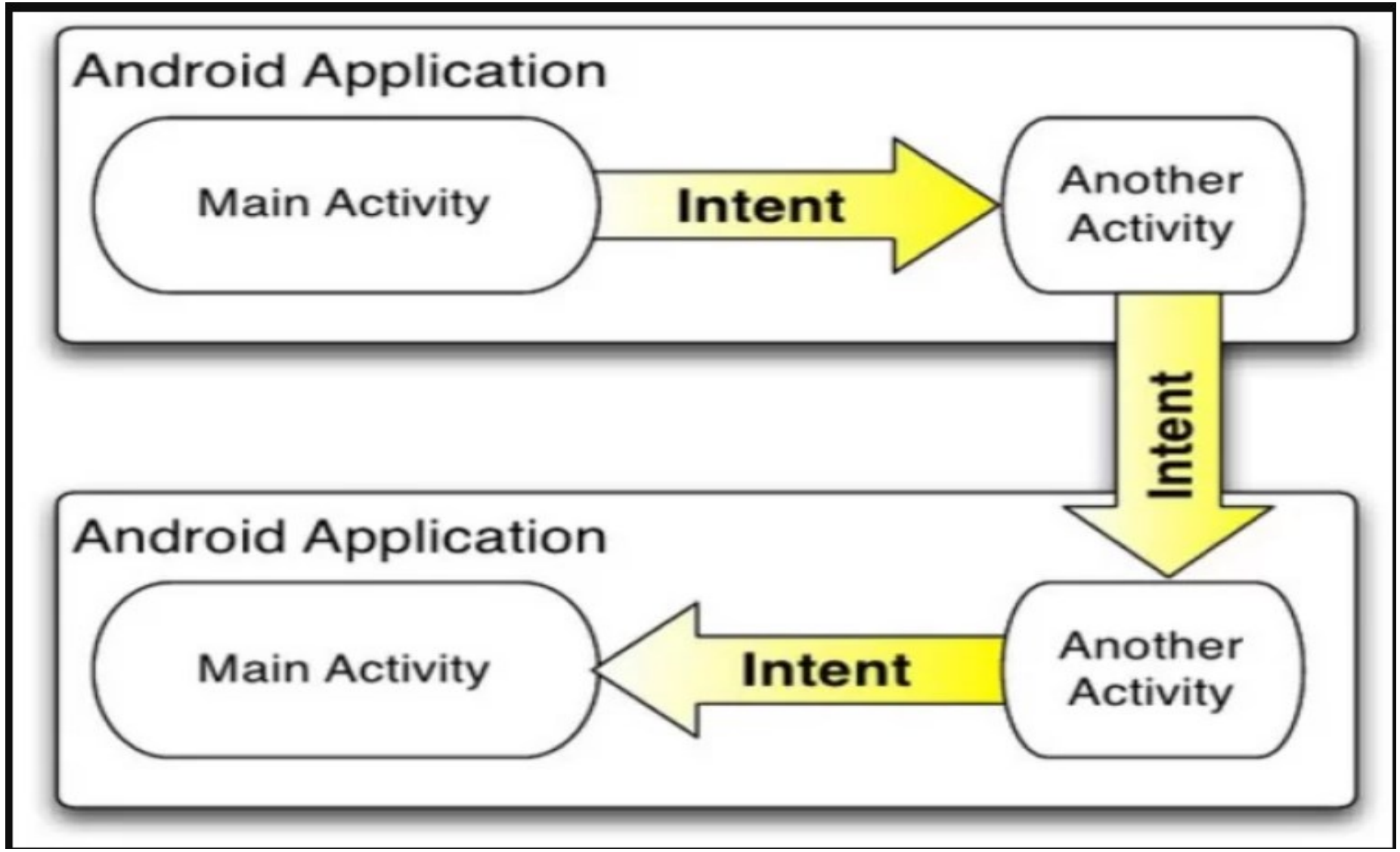
ادامه متدهای چرخه حیات

```
@Override
protected void onStart() {
    super.onStart();
    Toast.makeText(this, "MainActivity Start .....", Toast.LENGTH_SHORT).show();

}

@Override
protected void onStop() {
    super.onStop();
    Toast.makeText(this, "MainActivity Stop .....", Toast.LENGTH_SHORT).show();

}
```

ادامه متدهای چرخه حیات

```
@Override
protected void onPause() {
    super.onPause();
    Toast.makeText(this, "MainActivity Pause .....", Toast.LENGTH_SHORT).show();

}

@Override
protected void onRestart() {
    super.onRestart();
    Toast.makeText(this, "MainActivity Restart .....", Toast.LENGTH_SHORT).show();

}
```

Intent و فراخوانی Activity



ارسال مقادیر فیلدهای فرم به اکتیویتی دیگر

اکنون که با موفقیت فیلدها را اعتبار سنجی نمودیم نوبت به ارسال مقادیر فیلدهای دریافت شده به یک اکتیویتی دیگر می‌رسد. برای انجام اینکار باید یکسری متغییرهای همنوع با فیلدهای فرم تعریف نماییم و پس از مقدار دهی متغییرها با استفاده از *Intenet* به اکتیویتی دیگر بصورت پارامتر ارسال می‌کنیم. برای ارسال از متد زیر استفاده خواهیم کرد.

```
intent.putExtra("نام فیلد یا کلید", متغییرمقداردهی شده در فرم);
intent.putExtra(“firstName” , name);
```

همانطور که در دستور بالا مشخص هست مقادیر فیلدها را با استفاده از یک کلید”*key*” ارسال می‌کنیم. در مثال مشخص هست یک کلید بنام ”*firstName*“ تعریف شده و با محتوی متغییر”name” مقدار دهی خواهد شد.

دریافت مقادیر ارسال شده به Activity

- جهت دریافت مقادیر در اکتیویتی مورد نظر از کلاس *bundle* یک شی تعریف می‌کنیم. سپس با استفاده از متدهای *getExtras* اقدام به دریافت مقادیر ارسال شده خواهیم کرد.

```
Bundle extra = getIntent().getExtras();
```

```
متغییر = extra.getString("نام کلید | فیلد ارسال شده");
name = extra.getString("firstName");
```

- در مثال فوق با توجه با اینکه یک رشته دریافت میکنیم از متد *getString* استفاده نموده‌ایم.

دریافت مقادیر ارسال شده به Activity

نکته قابل توجه برای خواندن محتوی کلیدها و مقادیر ارسال شده وجود دارد بررسی وجود فیلدها می‌باشد که از فرم اصلی ارسال شده است یا خیر، برای کنترل از متد `containsKey` استفاده خواهیم کرد.

```
if(extra != null){
    String name = "", email = "", phone = "";
    if(extra.containsKey("name")){
        name = extra.getString("name");
    }
```

در مثال فوق با استفاده از دستور `if` و درصورت خالی نبودن شی `extra` از دستور `if` برای بررسی وجود کلید `name` استفاده شده است.

کدهای دریافت اطلاعات فرم در Activity

```
Bundle extra = getIntent().getExtras();
EditText tv;
tv = findViewById(R.id.tv);
if(extra != null){
    String name = "", email = "", phone = "";
    if(extra.containsKey("name")){
        name = extra.getString("name");
    }
    if(extra.containsKey("phone")){
        phone = extra.getString("phone");
    }
    if(extra.containsKey("email")){
        email = extra.getString("email");
    }
    tv.setText("Name: " + name + "\n");
    tv.append("Phone: " + phone + "\n");
    tv.append("Email: " + email);
}
```

کدهای فوق برای راحتی و صرف آموزشی بودن در متد *onCreate* نوشته شده است.

ادامه Intenet (دریافت نتیجه از اکتیویتی با استفاده از Intent)

- در اسلایدهای قبل با استفاده از *startActivity(activity)* اقدام به فراخوانی اکتیوتی می کردیم و یک سری مقادیر در صورت نیاز به داخل اکتیویتی پاس می‌دادیم که این مقادیر در داخل همان اکتویتی فراخوانی شده قابل استفاده بود. در مقابل این عمل ما می‌خواهیم از اکتویتی که فراخوانی کردیم مقادیری دریافت کنیم(در واقع return شدن مقادیر)این عمل با فراخوانی متد *startActivityForResult()* اتفاق می‌افتد.این متد برای فراخوانی در حالت معمولی یک intent که در واقع همان اکتیویتی موردنظر می‌باشد به همراه یک *request_code* که در واقع یک کد شناسایی یا درخواست هست، بعنوان ورودی دریافت می‌کند و نتیجه فراخوانی این متد به داخل متد دیگری به نام onActivityResult منتقل می‌شود.

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
```

متد onAvtivityResult

- زمانی که یک اکتویتی با متد startActivityForResult فراخوانی شد و مقداری برگشت داد متد onActivityResult فراخوانی می‌شود.

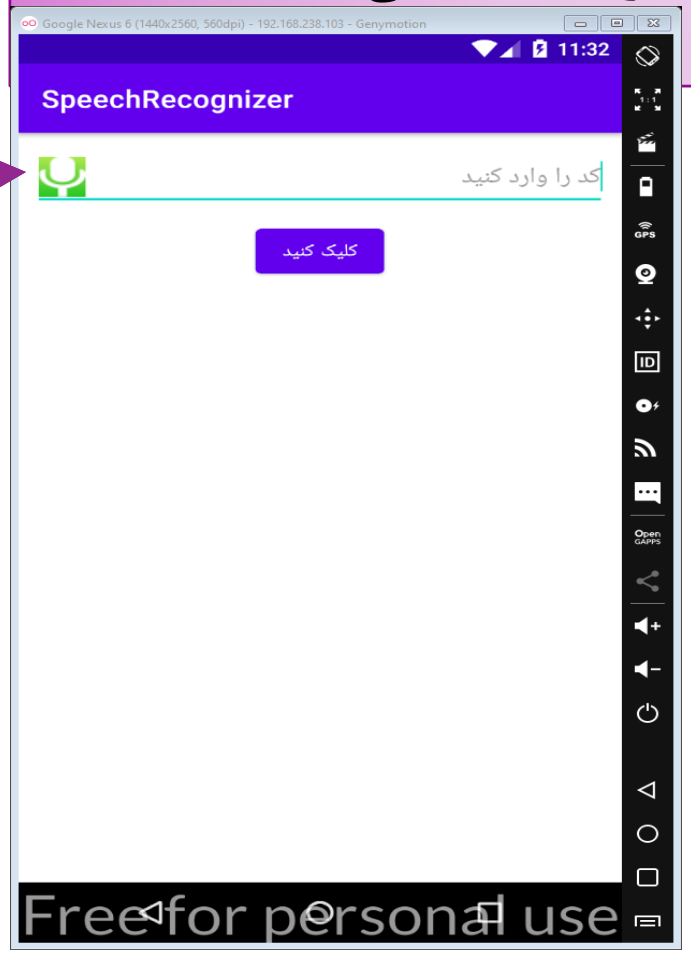
```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
```

- همانطور که مشاهده می‌کنید این متد شامل ۳ ورودی بوده که بر اساس مقادیر هر ورودی می‌توانیم کد فراخوانی شده و نتایج و یکسری داده که بصورت intent ارسال شده دسترسی داشته باشیم. در این متد requestCode در واقع یک کد شناسایی هست که به intent داده ایم تا در زمانی که اگر تاخیر در پاسخگویی بود با این کد متوجه شویم کدام اکتویتی مقداری برگشت داده است(ممکن هست چند اکتیویتی همزمان فراخوانی شوند).
- resultCode مشخص می‌کند که آیا نتیجه برگشت داده شده بصورت RESULT_OK هست یا RESULT_CANCEL تا بر اساس آن تصمیم گیری نماییم.
- data نیز مقدار برگشتی از اکتویتی فراخوانی شده می باشد
- در ادامه یک مثال کاربردی تبدیل گفتار به متن را به همراه این روش فراخوانی متدها بررسی خواهیم نمود.

تبدیل گفتار به متن و استفاده از متد onActivityResult

```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"
   tools:context=".MainActivity">
   <EditText
      android:id="@+id/edtSearchCode"
      android:layout_width="fill_parent"
      android:layout_height="51dip"
      android:drawableLeft="@drawable/mic"
      android:hint="کد را وارد کنید"
      android:inputType="text"
      android:layout_margin="10dp"/>
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/edtSearchCode"
      android:layout_centerVertical="true"
      android:layout_centerHorizontal="true"
      android:text="کلیک کنید"
      android:id="@+id/btnSearch"
      />
</RelativeLayout>
```

در این مثال روی اکتیویتی ۲ ابزار از نوع EditText و Button قرار می‌دهیم که در ادامه کدهای xml را مشاهده می‌کنید.



ظاهر برنامه مانند شکل مقابل خواهد بود چون فقط از ۲ ابزار استفاده شده است. در ضمن با خاصیت drawableleft در سمت چپ جعبه متن یک تصویرکوچک قرار داده‌ایم.

تبدیل گفتار به متن و استفاده از متد onActivityResult

```
public class MainActivity extends AppCompatActivity {
    Button btnSearch;
    EditText txtname;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnSearch = findViewById(R.id.btnSearch);
        txtname = findViewById(R.id.edtSearchCode);

     btnSearch.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if(!SpeechRecognizer.isRecognitionAvailable(MainActivity.this)){
                Toast.makeText(MainActivity.this, "امکانات صوتی پیدا نشد", Toast.LENGTH_SHORT).show();
                return;
            }
```

ادامه کدهای تبدیل گفتار به متن

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
      intent.putExtra(RecognizerIntent.LANGUAGE_MODEL_FREE_FORM,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
      intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,"en");
      intent.putExtra(RecognizerIntent.EXTRA_PROMPT,"چه چیزی نیاز دارید؟!");
      startActivityForResult(intent,1209);
    }
 });
}

 @Override
   protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
      super.onActivityResult(requestCode, resultCode, data);
      if(requestCode==1209 && resultCode== Activity.RESULT_OK){
         ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
         for(String text:result){
            txtname.setText(text);
         }
      }
   }
}
```

هنگام فراخوانی برای اکتویتی یک کد بشماره ۱۲۰۹ مشخص شده است این کد زمانی که چند اکتویتی فراخوانی شده و قرار بر ارسال مقادیر داشته باشند باید مشخص کنیم مقدار کدام اکتویتی در حال دریافت هست(در واقع کدشناسایی)

با توجه به اینکه در این مثال عمل تشخیص گفتار اتفاق می افتد مقادیر دریافت شده بصورت جدا جدا برگشت داده میشود ما از یک ArrayList رشته‌ای برای دریافت مقدار موجود در آبجکت data استفاده کرده‌ایم.

همانطور که در کدها مشخص می باشد در این مثال یک intent از نوع شناسایی گفتار تعریف شده است و با استفاده از متدهای putExtra که قبلا برای پاس دادن مقادیر استفاده کرده ایم این بار نیز یکسری پارامتر از قبیل مدل زبان، نوع زبان گفتاری، و یک پیغام برای نمایش در پنجره تشخیص صوت ارسال کرده‌ایم

ادامه کدهای تبدیل گفتار به متن

- توجه داشته باشید برای اینکه این مثال برای ما کار کند و بتوانیم از امکانات آن بهره ببریم باید ۲نوع دسترسی(permission) را در فایل AndroidManifest.xml مربوط به پروژه اعمال کنیم تا استفاده از میکروفن و همچنین اینترنت برای جستجو مقدور باشد.

activity_main.xml | MainActivity.java | AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.speechrecognizer">
4
5      <uses-permission android:name="android.permission.INTERNET"/>
6      <uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

۱

۲

- همانطور که مشخص هست ۲ دسترسی برای اینترنت و ابزار ضبط صدا به برنامه داده شده است. که در غیر اینصورت تبدیل گفتار عمل نخواهد کرد.

startActivityForResult و اکتویتی مورد نظر

- در مثال قبل چون از ابزار SPEECH استفاده کرده بودیم حالت parse شدن(تجزیه شدن) توسط خود ابزار و بصورت جداجدا برگشت داده می‌شد. زمانی که بخواهیم از یک اکتویتی دلخواه مقداری دریافت کنیم از روش زیر برای دریافت مقادیر استفاده خواهیم کرد.
- ابتدا در اکتویتی فرخوانی شده یک intent تعریف می‌کنیم که توسط متد setData، اینتنتِ تعریف شده را مقدار دهی خواهیم کرد. در ادامه با کدهای مربوط به این روش فراخوانی آشنا خواهیم شد. فرض بر این است که در همان پروژه قبلی یا هر پروژه دیگر ۲ اکتویتی داریم. اکتویتی دوم را بنام secondActivity تعریف کرده و روی آن ۲ ابزار با نوع EditText و Button قرار داده‌ایم، می‌خواهیم هر چیزی داخل جعبه متن نوشتیم با زدن روی Button یا روی دادن هر اتفاق دیگر به اکتویتی اول(فراخوانی کننده) برگشت داده شود.

ادامه کدهای مربوط به برگشت دادن مقادیر (secondActivity)

```
public void onClick(View v)
{
   Intent myData = new Intent();
   EditText text = findViewById(R.id.edtText);
   myData.setData(Uri.parse(text.getText().toString()));
   setResult(RESULT_OK, myData);
   finish();
}
```

- با توجه به اینکه فقط ۲ عنصر روی صفحه قرار دارد تمام حالتهای bind کردن و مقدار دهی نمودن و برگشت دادن مقدار را در رویداد onClick() مربوط به Button انجام داده‌ایم.

- در دستورات فوق همانطور که مشخص هست ابتدا یک آبجکت(شی) از نوع Intenet تعریف کردیم که در واقع این آبجکت مقادیر را برگشت خواهد داد. سپس عمل Bind کردن جعبه متن را انجام داده‌ایم، و در دستور شماره ۳ آبجکت که از نوع اینتنت می باشد با استفاده از متد setData توسط تجزیه کننده(Parser) کتابخانه Uri مقداردهی کرده‌ایم. و توسط متد setResult نتیجه RESULT_OK و مقدار آبجکت myData را به اکتویتی فراخوانی کننده و متد onActivityResult ارسال نمودیم.

ادامه کدهای مربوط به برگشت دادن مقادیر(mainActivity)

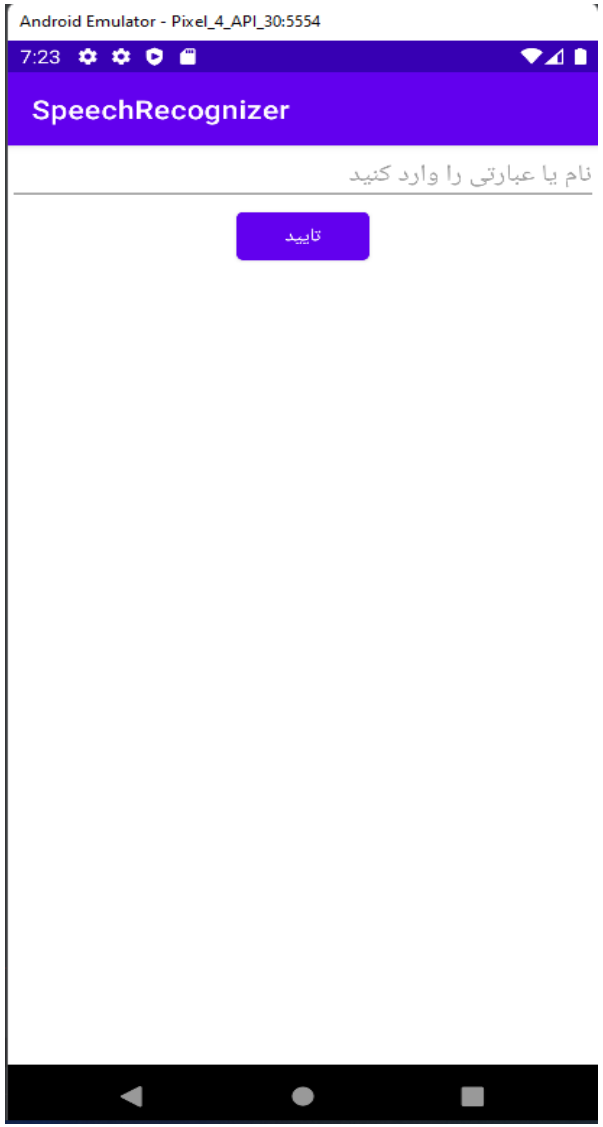
```
btnSecondActivity = findViewById(R.id.btnSecondActivity);
btnSecondActivity.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new  Intent(MainActivity.this,SecondActivity.class);
        startActivityForResult(new Intent(intent),1360);
    }
});
```

در کدهای مقابل پس از Bind کردن Button، یک آبجکت intent برای دسترسی به اکتویتی دوم تعریف شده است. سپس با استفاده از متد startActivityForResult اینتنت مورد نظر با requestCode =1360 فراخوانی شده است. که در هنگام برگشت تشخیص دهیم کدام Activity مقداری برگشت داده است.

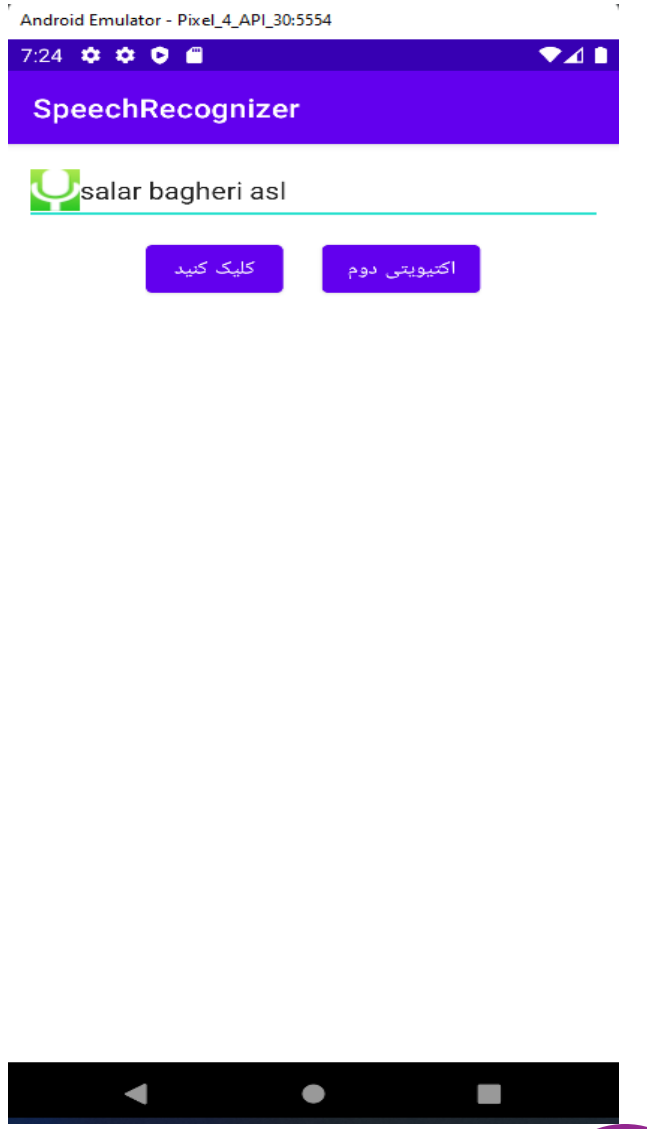
```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==1360 && resultCode == RESULT_OK){
        txtname.setText(data.getData().toString());
    }
    if(requestCode==1209 && resultCode== Activity.RESULT_OK){
        ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
  for(String text:result){
            txtname.setText(text);
        }
    }
}
```

در کدهای مشخص شده روبرو با استفاده از if ابتدا کدشناسایی(requestCode) و همچنین نتیجه موفق بودن(RESULT_OK) مقدار برگشتی بررسی شده و مقدار دریافت شده در جعبه متن با استفاده از متد setText نمایش داده می‌شود. همچنین عمل دریافت مقدار intent پاس شده توسط متد getData() و تبدیل به رشته انجام شده است. بخش دوم if هم قبلا بررسی شده است.

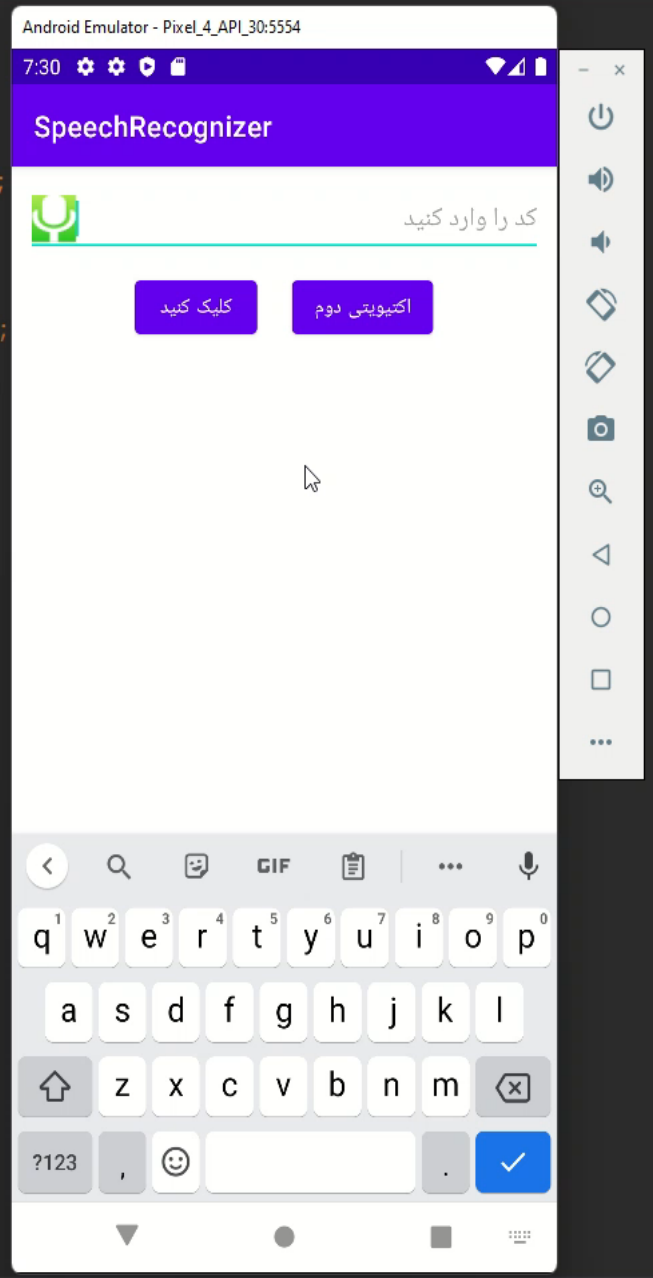

Android Emulator - Pixel_4_API_30:5554
7:23
SpeechRecognizer
کد را وارد کنید
کلیک کنید
اکتیویتی دوم
1


Android Emulator - Pixel_4_API_30:5554
7:23
SpeechRecognizer
نام یا عبارتی را وارد کنید
تایید


Android Emulator - Pixel_4_API_30:5554
7:24
SpeechRecognizer
salar bagheri asl
تایید
2


Android Emulator - Pixel_4_API_30:5554
7:24
SpeechRecognizer
salar bagheri asl
کلیک کنید
اکتیویتی دوم

```
    btnSearch.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(!SpeechRecognizer.isRecognitionAvailable( context: MainActivity.this)){
                Toast.makeText( context: MainActivity.this,  text: "امکانات صوتی پیدا نشد", Toast.LENGTH_SHORT).show();
                return;
            }
            Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
            intent.putExtra(RecognizerIntent.LANGUAGE_MODEL_FREE_FORM,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
            intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, value: "en");
            intent.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "چه چیزی نیاز دارید؟!");
            startActivityForResult(intent, requestCode: 1209);
        }
    });
    }



    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==1360 && resultCode == RESULT_OK){
            txtname.setText(data.getData().toString());
        }
        if(requestCode==1209 && resultCode== Activity.RESULT_OK){
            ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            for(String text:result){
                txtname.setText(text);
            }
        }

    }

}
```

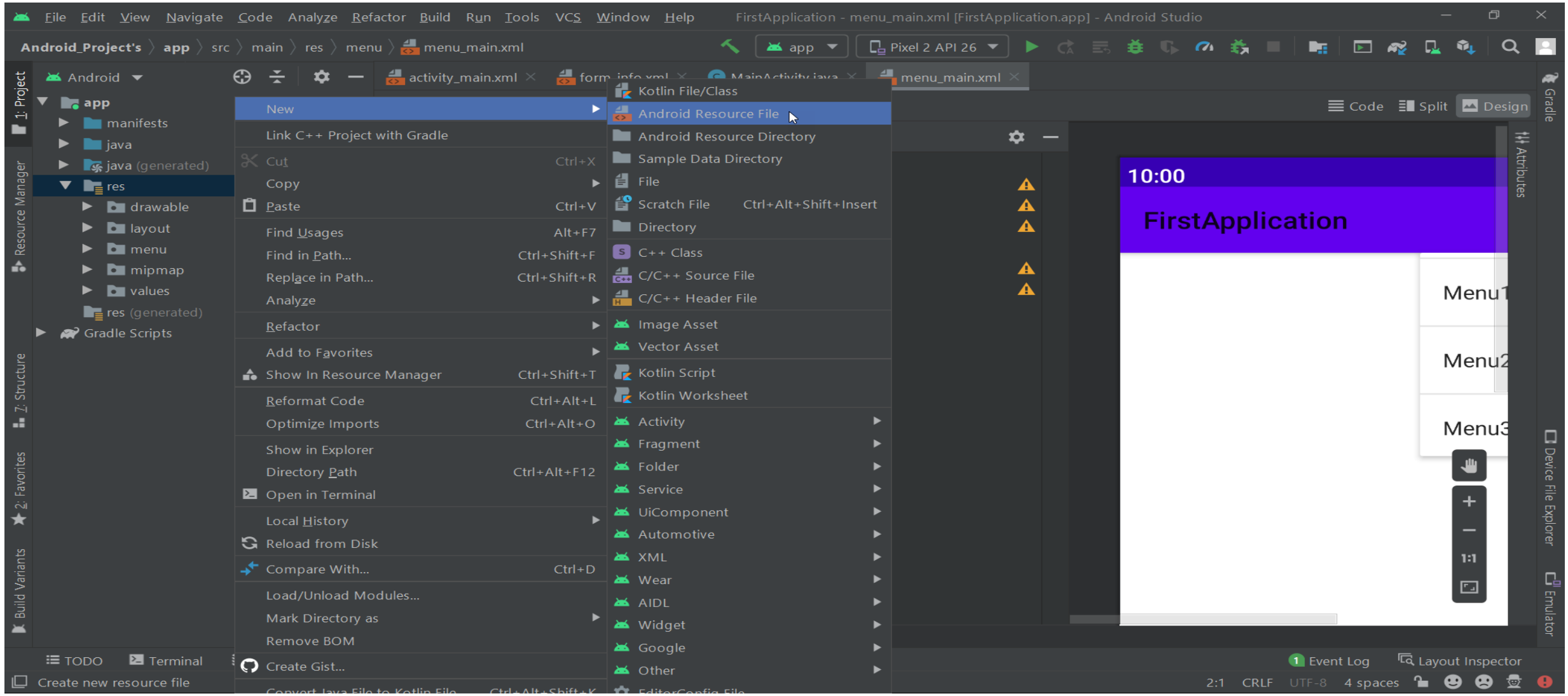


طراحی منو(OptionMenu) و ساخت فایل xml

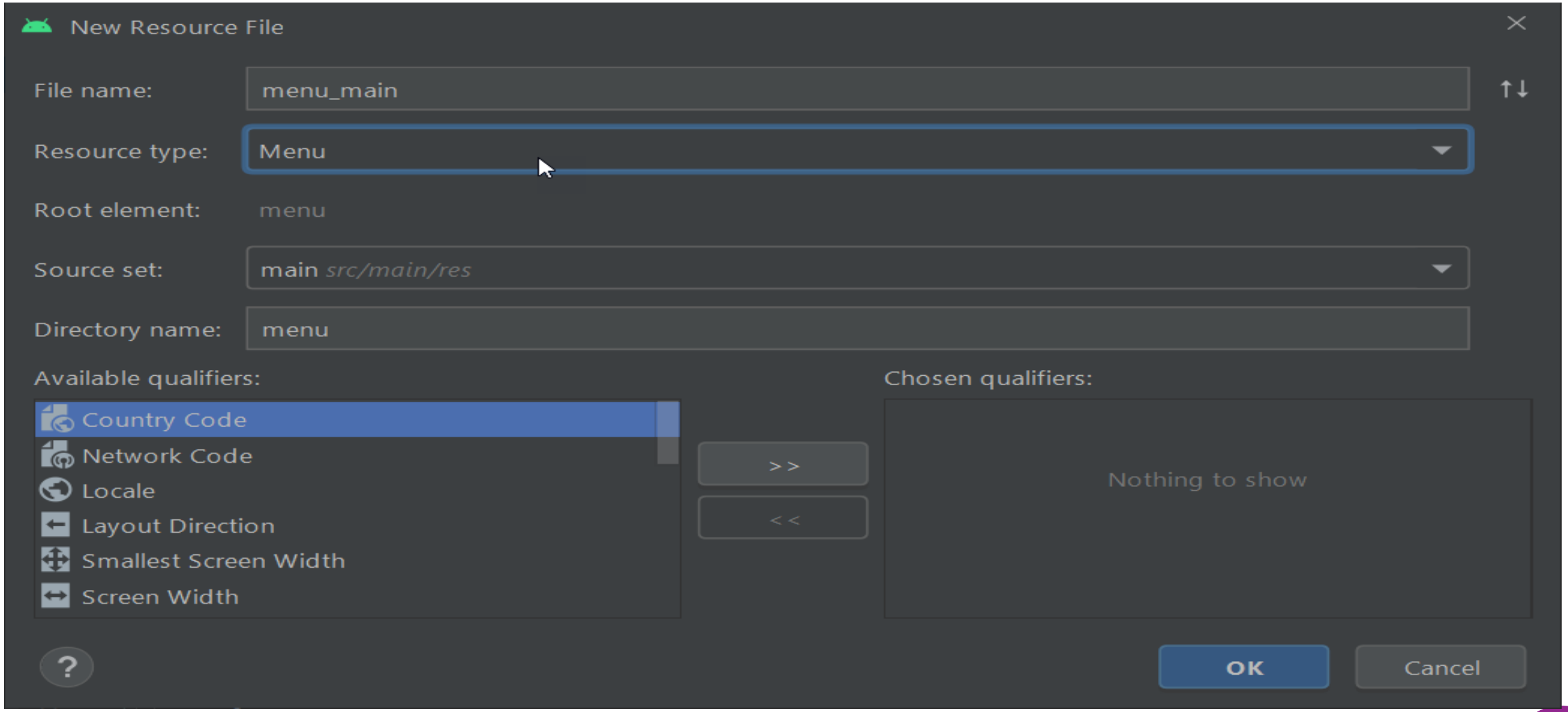
- طراحی منویی به شکل زیر در Activity

به این نوع منو **Option Menu** گفته می‌شود. همانطور که مشاهده می‌کنید شامل ۳ منوی اصلی به همراه یک زیر منو(**SubMenu**) وجود دارد که در داخل خود می‌تواند شامل چند منوی دیگر باشد.

ساخت فایل Resource برای طراحی منو



مشخصات فایل xml برای طراحی منو



نمونه محتوی فایل xml

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

    <item android:title="Menu1" />
    <item android:title="Menu2" />
    <item android:title="Menu3">
        <menu >
            <item android:title="SubMenu1" />
            <item android:title="SubMenu2"/>
        </menu>
    </item>

</menu>
```

Root Element که بصورت تگ menu می باشد

گزینه های منو

ساخت زیر منو SubMenu

فراخوانی و فعال سازی OptionMenu

- تا اینجا منو طراحی شده و در هنگام طراحی دیده می‌شود اما اگر بخواهیم در زمان اجرا نیز منوی طراحی شده را مشاهده کنیم باید با استفاده از دستورات زیر در Activity که میخواهیم ظاهر شود منوها را فراخوانی نماییم.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
}
```

تشخیص کلیک شدن (لمس‌شدن)منوها

- ساده ترین روش برای تشخیص لمس شدن منو استفاده از متد
- onOptionsItemSelected می‌باشد.در ادامه شکل کلی این متد و روش استفاده از متد را مشاهده می‌کنید.

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    Toast.makeText(this, item.getTitle(), Toast.LENGTH_SHORT).show();
    return super.onOptionsItemSelected(item);
}
```

- همانطور که درمتدonOptionsItemSekcted مشخص هست یک پارامتر item از نوع MenuItem وجود دارد که آیتم کلیک شده را برمی‌گرداند.در این مثال روی هر آیتم کلیک کنیم با getTitle عنوان را گرفته و در Toast نمایش می‌دهد.

تشخیص کلیک شدن (لمس‌شدن)منوها روش دوم

- در روش قبل اگر دقت کرده باشید برای منوها ID تعریف نکردیم و بطور مستقیم از متد `onOptionsItemSelected` برای تشخیص لمس شدن استفاده نمودیم. حال میخواهیم برای هر منو یک ID اختصاص دهیم و از خاصیت `OnClick` برای تشخیص لمس شدن استفاده می‌کنیم.
- متد `onOptionsItemSelected` بطورخودکار تشخیص می‌دهد کدام منو لمس شده است.

محتوی فایل menu_main.xml

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

    <item android:title="Menu1"
        android:id="@+id/action_menu1"
        android:onClick="menu_click"/>
    <item android:title="Menu2"
        android:id="@+id/action_menu2"
        android:onClick="menu_click"/>
    <item android:title="Menu3">
        <menu >
            <item android:title="SubMenu1"
                android:id="@+id/action_submen1"
                android:onClick="menu_click"/>
            <item android:title="SubMenu2"
                android:id="@+id/action_submenu2"
                android:onClick="menu_click"/>
        </menu>
    </item>
```

تشخیص لمس شدن آیتم‌های منو

- با استفاده از تابع زیر که در داخل فایل MainActivity.java می‌نویسیم مشخص می‌کنیم که کدام آیتم منو لمس شده است.

```
public boolean menu_click(MenuItem item) {
    Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();
    return true;
}
```

- همچنین با توجه به اینکه هر منو شامل ID می‌باشد می‌توانیم از حالت شرط گذاری هم برای تشخیص لمس شدن منو استفاده کنیم.

```
public boolean menu_click(MenuItem item) {
    if(item.getItemId()==R.id.action_menu1) {
        Toast.makeText(this, "Tap On Menu1", Toast.LENGTH_LONG).show();
    }
    return true;
}
```

تعریف منو با استفاده از کدهای جاوا

- تا کنون با استفاده از component از بخش pallete و همچنین با استفاده از تگ‌های xml اقدام به تعریف OptionMenu نمودیم که این کار را با استفاده از دستورات بصورت زیر نیز می‌توانیم انجام دهیم.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //getMenuInflater().inflate(R.menu.menu_main, menu);
    menu.add("Item1");
    menu.add("Item2");
    MenuItem item = menu.add("Item3");
    SubMenu submenu = menu.addSubMenu("Submenu");
    submenu.add("زیرمنو1");
    submenu.add("زیرمنو2");
    return super.onCreateOptionsMenu(menu);
}
```

اضافه کردن Support Action Bar به اکتویتی تصاویر

- اکشن بار معمولا در اکتویتی‌ها در گوشه بالا سمت چپ به شکل یک ← دیده می‌شوند.
- برای اضافه کردن این اکشن بار از دستور زیر


```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

- استفاده می‌کنیم.
- و برای کنترل بهتر که اکتویتی حتما شامل اکشن بار باشد آن را داخل یک شرط قرار می‌دهیم. این دستور را در متد onCreate اکتویتی قرار داده‌ایم و با استفاده از متد *onOptionsItemSelected* مشخص می‌کنیم با تپ شدن روی آن چه اتفاقی رخ دهد. دستورات کامل در اسلاید بعدی

```
if(getSupportActionBar() !=null){
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}
```

دستوارت کامل ایجاد اکشن منو بار و کنترل عملیات

- همانطور که در دستورات کاملا مشخص هست با تپ کردن روی OptionMenu ابتدا ID آن خوانده می‌شود و بطور پیشفرض ID برابر با مقدار *home* می‌باشد به دلیل اینکه سیستم خود این ID را در نظر گرفته و ما دخالتی نداشته‌ایم.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_image_view);
    if(getSupportActionBar() !=null){
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
    }

    public boolean onOptionsItemSelected(MenuItem item){
        int id = item.getItemId();
        if (id == android.R.id.home ){
            finish();
        }
        return super.onOptionsItemSelected(item);
    }
```

اضافه کردن یک Activity برای کار با ImageView

- برای کار با ImageView بهتر هست که یک اکتیویتی جداگانه داشته باشیم، با نحوه ایجاد اکتیویتی آشنا هستید. برای ادامه کار فرض می کنیم اکتیویتی جدید برای مشاهده تصاویر ایجاد کرده ایم و نام اکتیـویتی را *ImageViewActivity* قرار داده‌ایم برای فراخوانی اکتیویتی در *MainActivity* قبلی پخش فایل صوتی، یک *Button* دیگر با مشخصه زیر اضافه می‌کنیم و متد *setOnclickListener* آن را هم فعال می‌نماییم.

Componnet	Text	ID	Action
Button	Image View	btnImageActivity	باز کردن اکتیویتی تصاویر

- در متد OnClick هم به شرط Switch که برای کنترل دکمه های زده شده وجـود دارد شـرط و دسـتور مقابل را اضافه می‌کنیم.

```
case R.id.btnImageActivity:
    startActivity(new Intent(MainActivity.this, ImageViewActivity.class));
    break;
```

ImageView

ابزار ImageView برای نمایش تصاویر استفاده می‌شود. این View را همانند سایر Viewها می توانیم هم در فایل XML بصورت تگ اضافه کنیم و هم از لیست Componentها در حالتDesign معرفی ImageView با تگهای xml همانطور که مشخص هست شامل

یکسری از خصوصیات سایر اشیا می‌باشد(عمومی) و همچنین شامل یکسری *خصوصیات اختصاصی* که صرفاً برای ImageView طراحی شده است.

که با رنگ زرد مشخص شده اند.

مقدار alpha بین ۰ تا ۱ می‌باشد که هرچقدر به ۱ نزدیک باشد تصویر کامل دیده می‌شود. و به ۰ نزدیک باشد تصویر محو می‌شود.

خصوصیات rotation برای تغییر و چرخش تصویر استفاده می‌شود.

خاصیت srcCompat مسیر فایل تصویر را نمایش می‌دهد.

```
android:id="@+id/imageView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"
android:alpha=""
android:rotationX=""
android:rotationY=""
android:translationX=""
android:translationY=""
android:translationZ=""
app:srcCompat="@drawable/snow"
android:scaleType="centerCrop"
```

انیمیشن ساده fade

- در ادامه می‌خواهیم با استفاده از متد animate و خاصیت alpha یک انیمیشن fade به تصویر مورد نظر که در imageView نمایش داده می‌شود اعمال کنیم. فرض بر اینست که یک ImageView بر روی اکتیویتی وجود دارد. و میخواهیم با هر بار تپ رو آن افکت fade شبیه سازی شود.

مقدار دهی اولیه و فعال کردن متد OnClickListener برای ImageView

```
ImageView imageView;
imageView = findViewById(R.id.imageView);
imageView.setOnClickListener(this);
```

```
public void onClick(View v) {
    if(v.getId() == imageView.getId()){
        fade();
    }
}
private void fade() {
    if(imageView.getAlpha() == 0f){
        imageView.animate().alpha(1f).setDuration(2000);
    } else {
        imageView.animate().alpha(0f).setDuration(2000);
    }
}
```

در کدهای مقابل همانطور که مشخص کردیم اگر کاربر روی imageView تپ نماید متدی بنام fade() فراخوانی می‌شود.

این متد جزء متدهای کتابخانه اندروید استدیو نیست و خودمان باید ایجاد کنیم. با یک شرط مشخص نمودیم که اگر مقدار خاصیت alpha برابر ۰ باشد در مدت زمان ۲۰۰۰ میلی ثانیه مقدار alpha را به ۱ تغییر دهد. و همچنین برعکس

ورود تصویر و چرخش تصویر

- دستورات مقابل هنگام اجرای اکتیویتی از مختصات 2000- تصویر را وارد صفحه

```
// ورود تصویر از بالا به داخل
imageView.setTranslationY(-2000f);
imageView.animate().translationYBy(2000).setDuration(2000);
//
```

میکند. همچنین برای چرخش تصویر از متدهای Rotate استفاده خواهیم کرد. فرض بر این است که یک دکمه(Button) روی اکتویتی وجود دارد که خاصیت name آن را btnAnimate در نظر گرفتیم و همچنین مشخص کردیم که با کلیک(تپ)شدن متد animate فرخوانی شود(توسط خاصیت onClick در تگ تعریف شده فایل xml

```
Button btnAnimate;
btnAnimate = findViewById(R.id.btnAnimation);
```

```
private void animate() {
        imageView.animate().rotationXBy(180f).rotationBy(180f).setDuration(2000);
}
```

- متدهای دیگری نیز برای کار با تصویر وجود دارد که بعنوان فعالیت بر عهده شما گذاشته می‌شود.

متدهای بیشتر کار با افکتهای تصاویر

- imageView.setTranslationX(Xf);
- imageView.setTranslationY(Xf);
- imageView.setScaleX(Xf);
- imageView.setScaleY(Xf);
- animation().translationX();
- animation().translationY();
- animation().rotationBy();
- .setDuration();

متدهای مقابل یکسری از خصوصیات و نحوه نمایش افکتهای مختلف تصاویر را مشخص می‌کنند. علاوه بر این موارد چند متد دیگر نیز وجود دارد که می‌توانید خودتان تمرین کنید.
همچنین در بخش آخر همانطور که مشخص شده می‌توانیم ترکیب همه این خصوصیات و متدها را در یک دستور استفاده کنیم که بر اساس نیاز شما جهت پخش ونمایش افکت می‌توانید استفاده نمایید.

- ***imageView.animation().translationX(). translationY().rotationBy().setDuration();***

فهرست‌ها و ListView

- توسط ابزار ListView می‌توانیم یکسری داده را در روی اکتیویتی به کاربر نمایش دهیم این داده‌ها می‌توانند شامل اطلاعات خوانده شده از دیتابیس یا داده‌های وارد شده توسط کاربر یا همچنین اطلاعاتی از داخل خود برنامه یا اطلاعات خوانده شده از یک فایل XML یا فایل JSON باشند.
- نکته قابل توجه در خصوص Itemهای موجود در ListView این هست که هر آیتم می‌تواند بصورت کلیک خور باشد و با کلیک روی هر آیتم بتوانیم کار خاصی انجام دهیم. همچنین هر آیتم شامل موقعیت قرار گیری در لیست بوده که این موقعیت از 0 شروع می‌شود و تا n-1 ادامه دارد.

فهرست‌ها و ListView

برای استفاده از ListView به ترتیب زیر عمل خواهیم کرد.

۱) آیتم ها بصورت آرایه تعریف می‌شوند.

۲) از آداپتر(رابط) آرایه استفاده می‌شود.

۳) یک شی(ابزار) ListView روی صفحه قرار داده می‌شود.

۴) یک فایل xml با لایه TextView ساخته شده و از آن برای دسترسی به آیتم‌ها استفاده می‌شود.

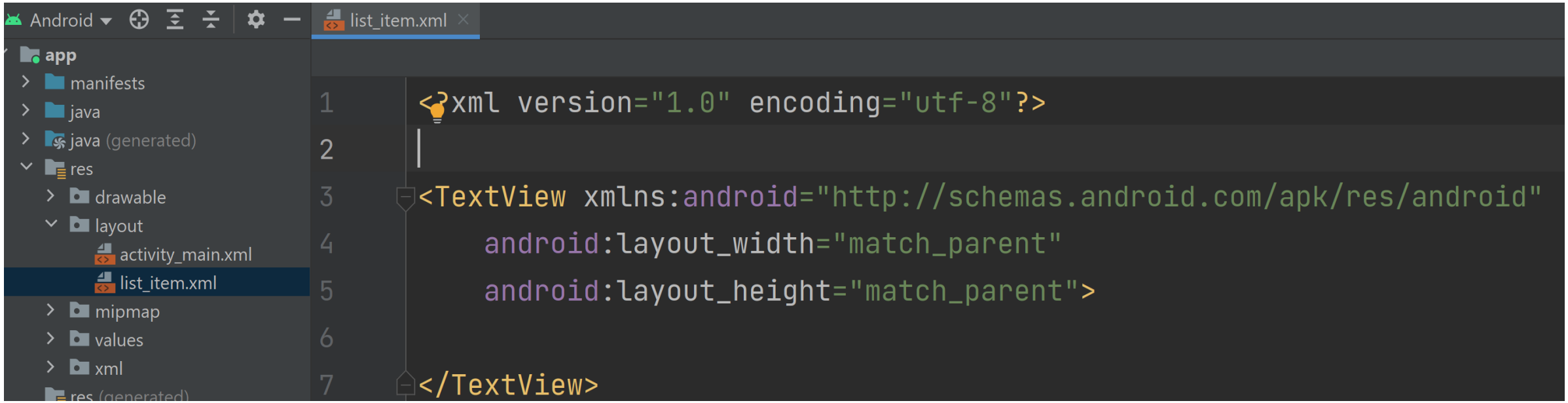
۵) لیست آرایه ساخته می‌شود.

۶) رابط آداپتر مقدار دهی می‌شود.

۷) ارتباط ListView با رابط آداپتر برقرار می‌شود.(برای نمایش)

فایل xml برای ارتباط و نمایش آیتم‌ها

- برای ساخت فایل XML از مسیر زیر اقدام می‌کنیم. در ادامه نمونه فایل ساخته شده را مشاهده می‌نمایید.
- app➔res➔layout➔Right-click➔New➔Layout XML File



- همانطور که مشاهده می‌کنید محتوی این فایل XML بسیار ساده بوده و در کل می‌توان گفت خالی هست و تنها شامل یک تگ ساده TextView می‌باشد. نام فایل را برحسب اختیار (list_item.xml) قرار داده‌ایم.

کدهای جاوا برای مقدار دهی و نمایش آیتم‌ها

- در این نمونه مثال با فرض وجود یک ListView روی اکتیویتی با شناسه(ID) listview کدها را مورد بررسی قرار می‌دهیم.

```
ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = findViewById(R.id.listView);
        listViewItem();
        listItemClick();
    }
```

```
private void listViewItem() {
    // Create List Item
    String[] listItems = {"قرمز", "سبز", "آبی","سفید"};
    // Build Adapter
    ArrayAdapter<String> adapter = new ArrayAdapter<~>(
            context: this, // Context برای اکتویتی
            R.layout.list_item, // لایه برای استفاده نمایش
            listItems); // لیست آیتمهایی که نمایش داده میشود
    // Configure the list View
    listView.setAdapter(adapter);
}
```

در کدهای بالا پس از مقدار دهی و اختصاص شناسه ListView به شی تعریف شده بلافاصله ۲ متد نوشته شده توسط کاربر فرخوانی می‌شود. متد ListViewItem برای مقدار دهی و نمایش عناصر مورد استفاده قرار می‌گیرد. و توسط متد listItemClick تشخیص می‌دهیم کدام آیتم موجود کلیک شده است.

کدهای جاوا برای تنظیم کلیک شدن آیتم‌ها

```
private void listItemClick() {
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView
                , View view
                , int position
                , long id) {
            TextView textView = (TextView) view;
            String message =  "CLick In Item "
                    + "(" + textView.getText().toString() + ")"
                    + " In Position ==>" + position;
            Toast.makeText( context: MainActivity.this
                    , message
                    , Toast.LENGTH_SHORT).show();
        }
    });
}
```

- در کدهای مقابل متد

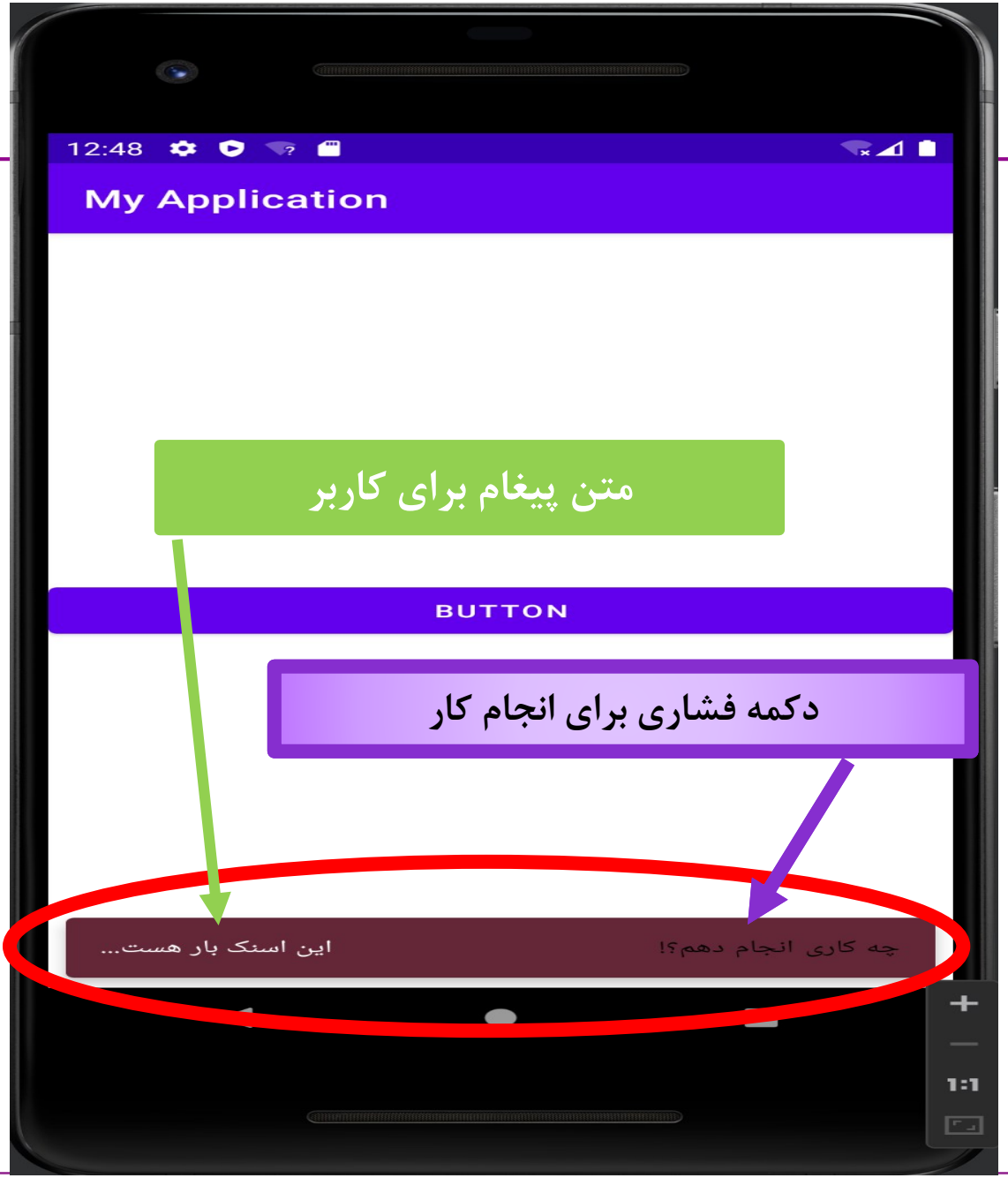
setOnItemClickListener

برای تشخیص کلیک شدن آیتم‌ها مورد استفاده قرار گرفته است. توجه داشته باشید که این متد صرفاً برای تشخیص کلیک عناصر لیست استفاده شده است. ورودی آن رابطلیست می‌باشد. در ادامه مشخص کردیم با کلیک روی هر آیتم عنوان آن و موقعیت آیتم کلیک شده توسط Toast نمایش داده می‌شود.

- اسنک‌بار بهترین تعریف برای آن جایگزینی مناسب برای Toastدر اندروید می‌باشد که همراه با مباحث **متریال‌دیزاین** مطرح شده است. همانطور که قبلاً هم بررسی کردیم در مباحث معمولی و خارج از *تکنیک‌های متریال‌دیزاین*، برای اطلاع دادن یا نمایش پیغام‌های معمول لحظه‌ای به کاربر در خصوص اتفاقات رخ داده مختلف که در اپلیکیشن روی می‌داد، از ابزار Toast استفاده میکردیم. با این حال با پیشرفت تکنیک‌های کد نویسی و زیبا سازی طراحی‌ها ابزاری کاربردی‌تر ازToast طراحی و ساخته شد که می‌توانیم از آن به جایToast استفاده نماییم. توجه داشته باشید Snack Bar امکانات بسیار بیشتر و کاربردی تر و انعطاف پذیرتر نسبت به Toast در اندروید استودیو دارد که همین دلیل باعث برتری Snack Bar در مقابل ابزار Toast شده‌است.

خروجی اسنک بار

خروجی اسنک بار



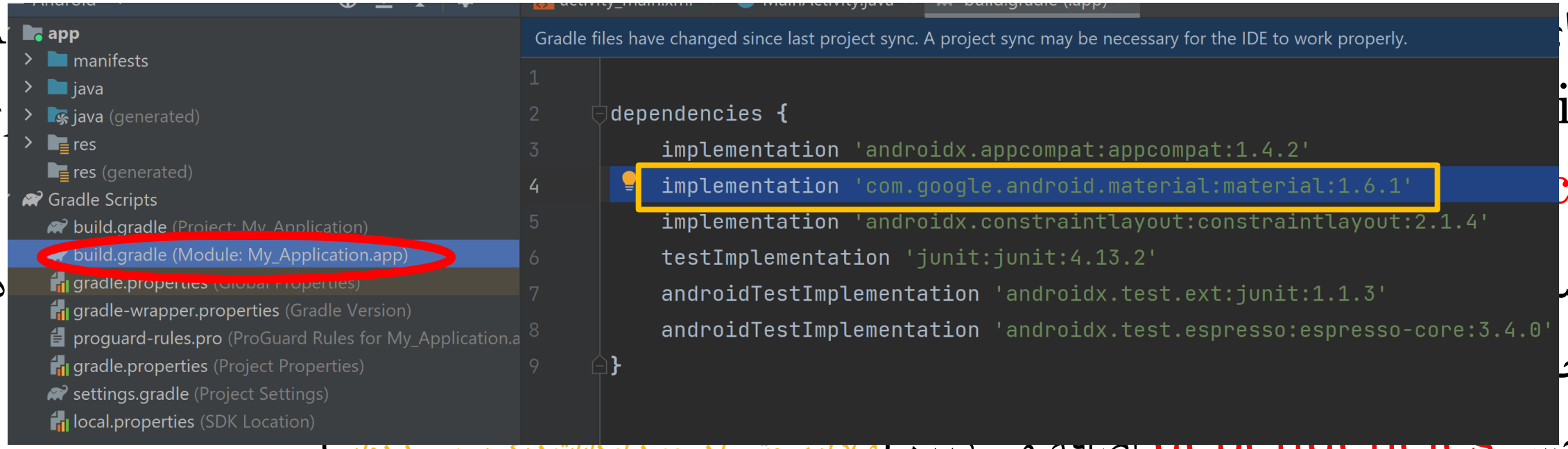
Snackbar

- برای استفاده از اسنک‌بار در برنامه‌های اندروید در نسخه‌های قبلی محیط توسعه Android Studio ابتدا باید یکسری implementation در داخل برنامه انجام می‌دادیم و یکسری dependency به پروژه خود اضافه می‌نمودیم. در ادامه این روش توضیح داده می‌شود.
- اضافه کردن dependency زیر به داخل فایل build.gradle(Module:app) که در بخش Gradle Scripts قرار دارد. این implementation را در داخل متن فایل و بخش dependencies اضافه می‌کنیم.(*قبلا این روش را در مورد امکانات دیگر بررسی کرده‌ایم*)

➢ implementation 'com.google.android.material:material:VersionNumber'

➢ VersionNumber نسخه متریال‌دیزاین موردنظر می‌باشد. در نسخه‌های جدید اندروید استودیو زمانی که می‌خواهیم از Snackbar استفاده کنیم یا آن را make نماییم همه کارهای بالا اتوماتیک انجام می‌شود.

- برا
- اض



بخش dependencies اضافه می کنیم.(قبلا این روش را در مورد امکانات دیگر بررسی کرده‌ایم)

- implementation 'com.google.android.material:material:VersionNumber'
- VersionNumber نسخه متریال‌دیزاین موردنظر می‌باشد. در نسخه‌های جدید اندروید استودیو زمانی که می‌خواهیم از Snackbar استفاده کنیم یا آن را make نماییم همه کارهای بالا اتوماتیک انجام می‌شود.

ساخت و فراخوانی Snackbar

- قالب کلی استفاده از اسنک‌بار بصورت زیر می‌باشد.

```
Snackbar.make(LayoutID, “Message”, Duration).show()
```

- همانطور که در شکل کلی متد Snackbar مشخص هست شامل ۳ پارامتر ورودی به ترتیب زیر می باشد:
  - LayoutID: محلی می‌باشد که می‌خواهیم اسنک‌بار روی آن ظاهر شود.
  - Duration: مدت زمانی که میخواهیم پیغام را نمایش دهد:
    - Snackbar.LENGTH_INDEFINITE ✓
    - Snackbar.LENGTH_LONG ✓
    - Snackbar.LENGTH_SHORT ✓

امکانات و قابلیت‌های اسنک بار

- حالت قبلی ذکر شده فقط برای ساخت اسنک‌بار و نمایش پیغام کفایت می‌نماید. اما یکسری ویژگی‌های متفاوت برای تنظیمات بیشتر و همچنین تعریف یک گزینه کلیک‌خور و تغییر رنگ نوشته یا پس‌زمینه اسنک‌بار بصورت زیر وجود دارد:
  - متد setAction: برای تعریف عبارت کلیک خور روی اسنک‌بار استفاده می‌شود.
  - متد setActionTextColor : برای تغییر رنگ پیغام ظاهر شده استفاده می‌شود.
  - متد setAnimationMode: برای تعریف حالت نمایش پیغام استفاده می‌شود.
  - متد setBackgroundTint: برای تعریف رنگ پس‌زمنیه اسنک‌بار استفاده می‌شود.

✓ و شامل یکسری متدها و قابلیت‌های دیگر که بررسی آن بر عهده خودتان واگذار می‌گردد.

```
import ...
public class MainActivity extends AppCompatActivity {
    private Button btnClick;
    LinearLayout mainLinerLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnClick = findViewById(R.id.button2);
        mainLinerLayout = findViewById(R.id.mainLinerLayout);
        btnClick.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(mainLinerLayout, text: "...این اسنک بار هست", Snackbar.LENGTH_SHORT)
                        .setAction( text: "چه کاری انجام دهم؟!", new View.OnClickListener() {
                            @Override
                            public void onClick(View view) {
                                Toast.makeText( context: MainActivity.this, text: "http://afo.ac.ir", Toast.LENGTH_SHORT).show();
                            }
                        })
                        .setActionTextColor(getResources().getColor(R.color.black))
                        .setAnimationMode(BaseTransientBottomBar.ANIMATION_MODE_SLIDE)
                        .setBackgroundTint(Color.parseColor( colorString: "#642738"))
                        .show();
            }
        });
    }
}
```