

```
1  #Region Project Attributes
2      #ApplicationLabel: Project
3      #VersionCode: 1
4      #VersionName: 1.0
5      'Supports2DRenderer: true
6      #Supports2DRenderer: true
7      #CanInstallExternalStorage: false
8  #End Region
9
10 #Region Activity Attributes
11     #FullScreen: false
12     #IncludeTitle: true
13 #End Region
14
15 #Sub Global
16     'The following variables will be redeclared
17     'These variables can only be accessed from
18
19 End Sub
20
21 Sub Globals
22     'The following variables will be redeclared
23     'These variables can only be accessed from
24
25 End Sub
26
27 Sub Activity_OnStart (As Boolean)
28     'Do something here like layout file creation
29     'Activity.OnStart (Layout1")
30 End Sub
```

Android 4 و اندروید با

برنامه نویسی اندروید با

به همراه آموزش مقدماتی اچ تی ام ال



نویسنده: ع. سجادی

از سری مجموعه‌های آموزشی قهوه‌کاشی

نذرت کلیک کردن
قهوه‌کاشی

wWw.GhahveKashi.iR

با همکاری مرکز پاسخگویی به سوالات برنامه نویسی



www.AnswerCenter.ir

و انجمن برنامه‌نویس ایرانی



www.learn-net.ir



تقديم به

دوستاناران برنامه نویسی اندروید

فهرست مطالب:

۴	 مقدمه
۵	 ۱- نرم افزارهای پیش نیاز
۷	 ۲- اقدامات اولیه
۱۱	 ۳- طراحی ظاهر اپلیکیشن اندرویدی
۱۶	 ۴- بررسی View ها
۱۷	 ۵- قسمت Abstract designer
۱۹	 ۶- محیط کدنویسی Basic4Android
۲۴	 ۷- اتصال Basic4Android به شبیه ساز و گوشی
۲۸	 ۸- کدهای مهم
۴۷	 ۹- استفاده از چند اکتیویتی
۴۹	 ۱۰- کتابخانه ها (و آموزش کار با کتابخانه ی Phone)
۵۱	 ۱۱- کار با Manifest Editor
۵۳	 ۱۲- گرفتن خروجی نهایی از اپلیکیشن اندرویدی
۵۴	 ضمیمه ۱: آموزش مقدماتی HTML
۶۱	 ضمیمه ۲: معرفی و آموزش کار با کتابخانه ی AriaLib
۶۳	 ادامه ی راه؟؟؟



مقدمه:

با سلام.

از این که قصد دارید به جمع توسعه دهندگان ایرانی سیستم عامل اندروید بپیوندد خرسندیم.

در این کتاب توسعه ی اپلیکیشن با محیط B4A (Basic4Android) آموزش داده شده است که نسبت به Eclipse و Android Studio (محیط بومی توسعه ی اغلب اپلیکیشن های اندرویدی) از پیچیدگی کمتری بهره مند است و برنامه نویسی اندروید را بسیار ساده و لذت بخش می کند.

راه برنامه نویسی موبایل دراز و پیچیده است و این کتاب سعی می کند شما را در قدم های اول این راه همیاری کند و ادامه ی این راه را به شما بنمایاند.

بدون شک تدوین این اثر و تجمع اطلاعاتی جامع (و نه کامل) در این رابطه در یک کتاب کار آسانی نبوده است. در صورتی که این اثر را مناسب ارزیابی کردید با دریافت اپلیکیشن این آموزش از کافه بازار و ارتقاء آن به نسخه ی کامل، از کتاب حمایت کرده و موجب دلگرمی ما برای نگارش آثاری از این دست باشید (:

دریافت اپلیکیشن از کافه بازار: <https://cafebazaar.ir/app/com.asajadi.b4a>

تماس با نویسنده: asajadi84@yahoo.com | www.GhahveKashi.ir



۱- نرم افزارهای پیش نیاز:

اجرای B4A بر روی ویندوز نیازمند نصب بودن نرم افزارهای پیش نیاز این محیط است که می توانید از سایت های دانلود نرم افزار آن ها را دریافت کنید. نرم افزارهای پیش نیاز عبارتند از:

Java Run Time (JRE)

Android Software Development Kit (Android SDK)

Java Development Kit (JDK)

پس از دانلود و نصب موارد بالا نرم افزار Basic4Android 5.50 را نصب، کرک و اجرا کنید. سپس از منوی Tools، گزینه ی Configure Paths را انتخاب کنید.

تکمیل دو فیلد javac.exe و android.jar برای کارکرد برنامه الزامی است. در این دو فیلد مسیر دو فایل خواسته شده در کامپیوتر را وارد کنید که اولی در فولدری که JDK نصب شده و دومی در فولدری که SDK نصب شده است موجود می باشند. در صورتی که هنگام نصب این دو نرم افزار مسیر نصب آن ها را تغییر نداده باشید آدرس آن ها به صورتی است که در زیر فیلد هر کدام نوشته شده است.

نکته: شماره ی پلتفرم اندروید در این قسمت، شماره ی ورژن اندروید نیست، بلکه شماره ی ورژن api (مخفف Application Programming Interface)



به معنی رابط کاربری برنامه نویسی) می باشد. معمولاً SDK دانلود شده در فولدر platforms خود هیچ api از قبل نصب شده ای را ندارد؛ بلکه آن ها باید جداگانه توسط کاربر و بسته به نیاز دانلود شوند. api اندروید ۵ (آب نبات چوبی)، ۲۱ می باشد.



۲- اقدامات اولیه:

قبل از هر کاری باید محیط برنامه را برای ساخت اپلیکیشن موردنظر خود آماده کنید. پس از باز کردن Basic4Android طبق مراحل زیر پیش بروید.

قدم اول، ذخیره کردن پروژه: تا پروژه ذخیره نشود عملاً هیچ کاری امکان پذیر نخواهد بود. پس اولین کار این است که پروژه ای که هنوز هیچ تغییری در آن ایجاد نشده و خالی است را ذخیره کنید. برای این کار از منوی File گزینه ی Save را انتخاب کرده و یا کلیدهای Ctrl+S را فشار دهید. فایل خروجی با فرمت *.b4a می باشد. شدیداً توصیه می شود که فایل فوق را در یک New Folder ذخیره کنید. چرا که بعد از توسعه ی اپلیکیشن فایل های زیادی به سورس برنامه اضافه می شوند که اگر در یک فولدر جداگانه نباشند کاملاً گیج کننده خواهند بود.

قدم دوم، انتخاب آیکون برای اپلیکیشن: بایستی آیکون اپلیکیشن خود را در نرم افزاری مثل فتوشاپ طراحی کرده (توصیه می شود با فرمت *.png باشد) و در دو اندازه ی 128x128 و 512x512 (واحد پیکسل است) ذخیره کنید. آیکون 128x128 برای وارد کردن به Basic4Android است و آیکون 512x512 برای وقتی که اپلیکیشن برای انتشار به مارکت ارسال می شود (یعنی در کل آیکون 512x512 در برنامه نویسی نقشی ندارد).

برای وارد کردن آیکون اپلیکیشن از منوی Project گزینه ی Choose Icon را انتخاب کنید.

قدم سوم، انتخاب کردن نام پکیج اپلیکیشن: پکیج نامی است که برنامه از روی آن شناخته می شود (با نام برنامه اشتباه نشود). برای مثال نام پکیج اپلیکیشن اینستاگرام com.instagram.android است.

نام پکیج، نامی است که اپلیکیشن شما را از سایر اپلیکیشن ها متمایز می کند و در واقع هویت برنامه است. ممکن است نام اپلیکیشن بعد از مدتی تغییر کند (مثل اپلیکیشن Omni Swipe که پیش تر به Lazy Swipe معروف بود). اما نام پکیج یک برنامه ی منحصر به فرد هیچ وقت تغییر نمی کند.

قرارداد خاصی برای انتخاب نام پکیج ها وجود ندارد. تنها بایستی از حروف انگلیسی (حساس به کوچک و بزرگ بودن حروف)، اعداد و آندر اسکور (آندر لاین _) تشکیل و با یک یا چندین نقطه از هم جدا شده باشند. استفاده از حروف بزرگ، عدد و آندر اسکور توصیه نمی شود چرا که جستجوی برنامه را از روی نام پکیج دشوار می کند. در هر صورت هر چه این نام کوتاه تر باشد بهتر است. مرسوم است که توسعه دهندگان نام آدرس سایت خود را به صورت وارونه به عنوان نام پکیج انتخاب می کنند (مثل



اینستاگرام). اکثر نام پکیج ها با com شروع می شوند و دارای سه قسمت است که با دو کاراکتر نقطه از هم جدا شده اند.

صفحه ی یک اپلیکیشن در مارکت کافه بازار از روی نام پکیج:

نام پکیج/cafebazaar.ir/app/https://

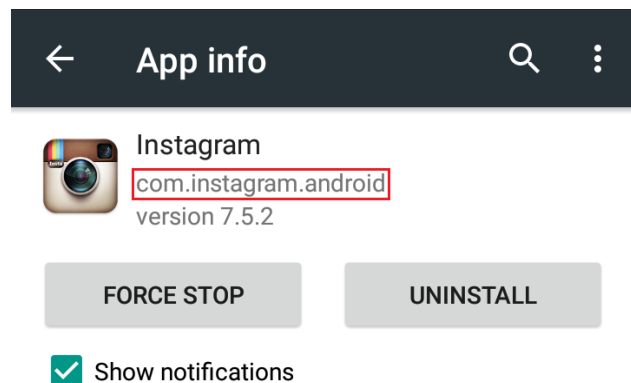
com.instagram.android/cafebazaar.ir/app/https://

صفحه ی یک اپلیکیشن در مارکت گوگل پلی از روی نام پکیج:

نام پکیج?id=com.instagram.android/play.google.com/store/apps/details?https://

com.instagram.android/play.google.com/store/apps/details?https://

برای مشاهده ی نام پکیج یک اپلیکیشن در تنظیمات گوشی به قسمت Apps رفته و اپلیکیشن موردنظر خود را از لیست انتخاب کنید. نام پکیج در زیر نام برنامه دیده می شود.



دقت کنید که نام پکیج انتخابی شما قبلاً انتخاب نشده باشد. برای تعیین نام پکیج از منوی Projects گزینه ی Build Configurations را انتخاب کرده و یا کلیدهای Ctrl+B را فشار دهید.



A Build Configurations

Configuration: Create New

Configuration Name: Delete

Package:

Conditional Symbols:

Example: Full, NoAds

Cancel OK

سپس در پنجره ی باز شده، در فیلد Package نام پکیج موردنظر خود را وارد کرده و بر روی OK کلیک کنید.

قدم چهارم، امضا کردن اپلیکیشن: برای این که هیچ کس به جز توسعه دهنده ی اصلی نتواند نسخه ی جدیدی برای اپلیکیشن خود ارائه کند اپلیکیشن توسط رمزی که توسعه دهنده بر روی آن می گذارد و به ساین (Sign به معنی امضا) معروف است محافظت می شود. به این صورت که امکان نصب دو اپلیکیشن حتی اگر نام پکیج یکسانی داشته باشند ولی با دو امضای مختلف ساین شده باشند در کنار یکدیگر وجود ندارد و پیغام خطای App not installed نمایش داده می شود (علت این که بازی مود شده را نمی توان در کنار بازی اصلی نصب کرد نیز همین است که شخصی که بازی را مود کرده است به امضای اصلی برنامه دسترسی نداشته است).

برای امضا کردن برنامه از منوی Tools گزینه ی Private Sign Key را انتخاب کنید.

A Private Sign Key

First and last name: ☒ Create new key

Organization: ☐ Load existing key

Country (two letters):

Password (at least 6 characters):

File: Save As

Signature (read-only):

Create or load a key and reopen this dialog to see the signatures.

Use debug key Cancel Ok



در صورتی که قسمت Create new key فعال باشد امضای جدیدی ساخته می شود. در قسمت First and last name نام و نام خانوادگی و در قسمت Organization نام کمپانی خود را وارد کنید. در قسمت Country باید نام کشور را به صورت دو حرفی وارد کرد که برای ایران می شود IR.

در قسمت Password رمز محافظتی موردنظر خود را وارد کرده و با کلیک بر روی کلید Save As فایل امضای خود را ذخیره کنید. فرمت این فایل keystore* است. سپس بر روی Ok کلیک کنید. بعد از این کار پیغام زیر مشاهده می شود:



Keystore was created successfully.
It is highly recommended to backup this file.
If this file is lost you will not be able to update your application in Google Play.

ترجمه: فایل امضا با موفقیت ساخته شد. شدیداً توصیه می شود که از این فایل بک آپ بگیرید. اگر این فایل از بین برود شما دیگر قادر به آپدیت اپلیکیشن خود در گوگل پلی نخواهید بود.

حال هنگام توسعه ی نسخه ی جدید یک اپلیکیشن در صورتی که بخواهید از امضای قبلی استفاده کنید در پنجره ی Private Sign Key گزینه ی Load existing key را فعال کرده و با کلیک بر روی کلید Browse فایل امضای خود را وارد برنامه کنید. سپس در قسمت Password رمز خود را وارد کرده و بر روی Ok کلیک کنید.

نکته: در صورتی که در حال تمرین هستید و محافظت از اپلیکیشن چندان اهمیتی ندارد می توانید با کلیک بر روی Use debug key از فایل امضای دیباگ برنامه که برای تست اپلیکیشن ها کاربرد دارد و در واقع عملاً محافظتی نمی کند استفاده کنید.

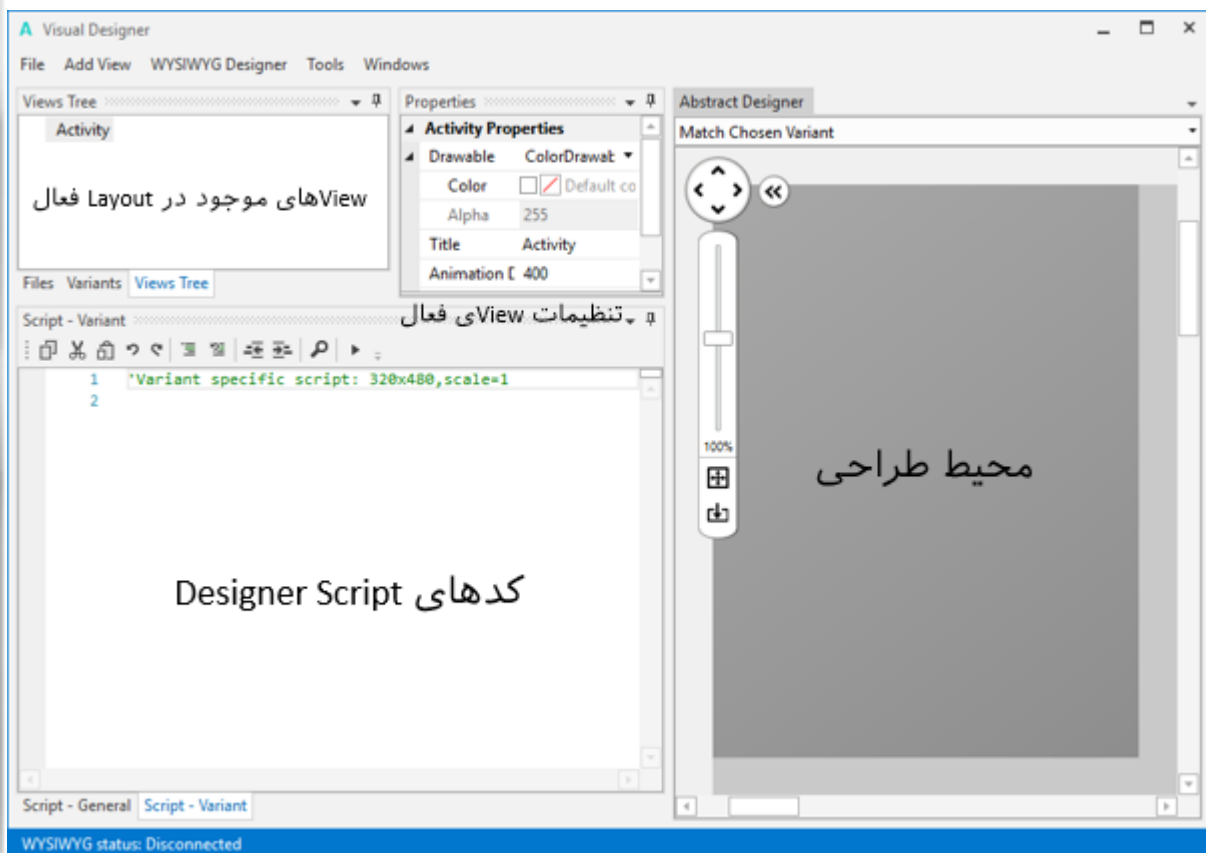
این قدم را در آخر برنامه نویسی و هنگام خروجی گرفتن از اپلیکیشن نیز می توانید انجام دهید. چرا که هنگام تست برنامه نیازی به آن نیست و برنامه می تواند از فایل امضای دیباگ استفاده کند.



۳- طراحی ظاهر اپلیکیشن اندرویدی:

بعد از انجام اقدامات اولیه، نوبت به طراحی ظاهر نرم افزار می رسد. یعنی چیزی که کاربر با بازکردن برنامه مشاهده می کند. این ظاهر می تواند شامل کلید، عکس، متن، منو و... باشد.

طراحی ظاهر نرم افزار توسط قسمتی به نام Designer صورت می گیرد. برای باز کردن این قسمت از منوی Designer گزینه ی Open Designer را انتخاب کنید. هر صفحه ای که توسط این پنجره طراحی شود Layout نام می گیرد.

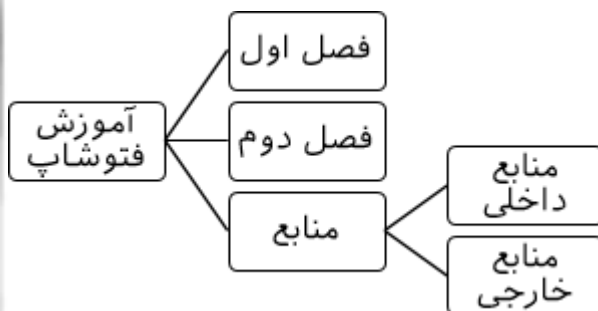


نکته: با اتصال شبیه ساز یا گوشی به WYSIWYG Designer می توانید به طور لحظه ای تغییرات ایجاد شده را در یک محیط اندرویدی مشاهده کنید. برای این کار کافی است که از منوی WYSIWYG Designer گزینه ی Connect را انتخاب کرده و یا کلید F2 را فشار دهید. البته باید گوشی یا شبیه ساز از قبل به Basic4Android متصل باشد که نحوه ی این کار در بخش «اتصال Basic4Android به شبیه ساز و گوشی» توضیح داده شده است.

Layout حکم خط کشی کردن یک دفتر مشق را دارد که قبل از نوشتن مشق باید انجام داد. حتی ممکن است چندین صفحه مشق بر روی ورق هایی با خط کشی مشابه نوشته شود. در Basic4Android هم ممکن است چند صفحه ی کاملاً متفاوت (که در ادامه گفته می شود به این صفحه ها اکتیویتی می



گویند) از یک Layout استفاده کنند. مثال فوق به روشنی این مطلب کمک می کند:



این نمودار قسمت های مختلف اپلیکیشن کتابی فرضی با موضوع آموزش فتوشاپ را نشان می دهد که با باز شدن آن لیستی شامل سه گزینه ی «فصل اول»، «فصل دوم» و «منابع» دیده می شود.

با انتخاب گزینه ی «فصل اول» محتوای فصل اول و با انتخاب گزینه ی «فصل دوم» محتوای فصل دوم نمایش داده می شود. با انتخاب گزینه ی «منابع» نیز صفحه ای با یک لیست شامل دو گزینه ی «منابع داخلی» و «منابع خارجی» پدید می آید که با انتخاب هر کدام محتوایشان نشان داده می شود.

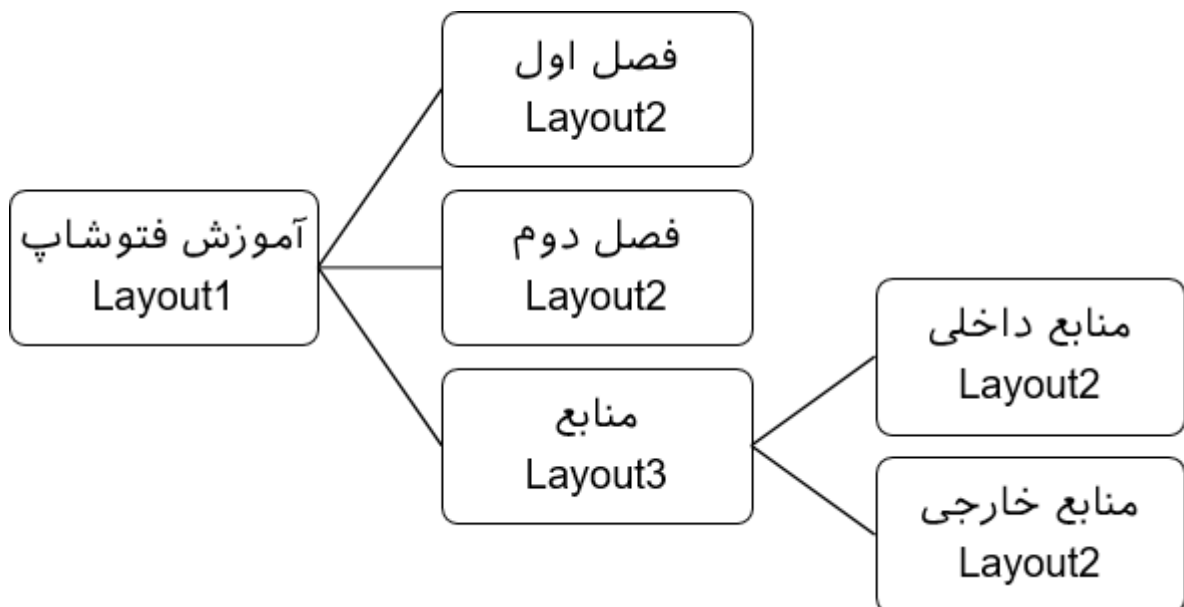
حال پرسش این جاست که برای طراحی این اپلیکیشن باید از چند Layout استفاده شود؟

پاسخ سه Layout است.

Layout1: شامل تصویر صفحه ی اصلی اپلیکیشن به همراه یک لیست

Layout2: شامل یک مستطیل بزرگ برای نمایش مطالب

Layout3: شامل تصویر صفحه ی منابع اپلیکیشن به همراه یک لیست



در واقع بسیاری از افراد تفاوت Layout و صفحه (اکتیویتی) را نمی دانند و یا آن ها را با هم اشتباه می کنند.



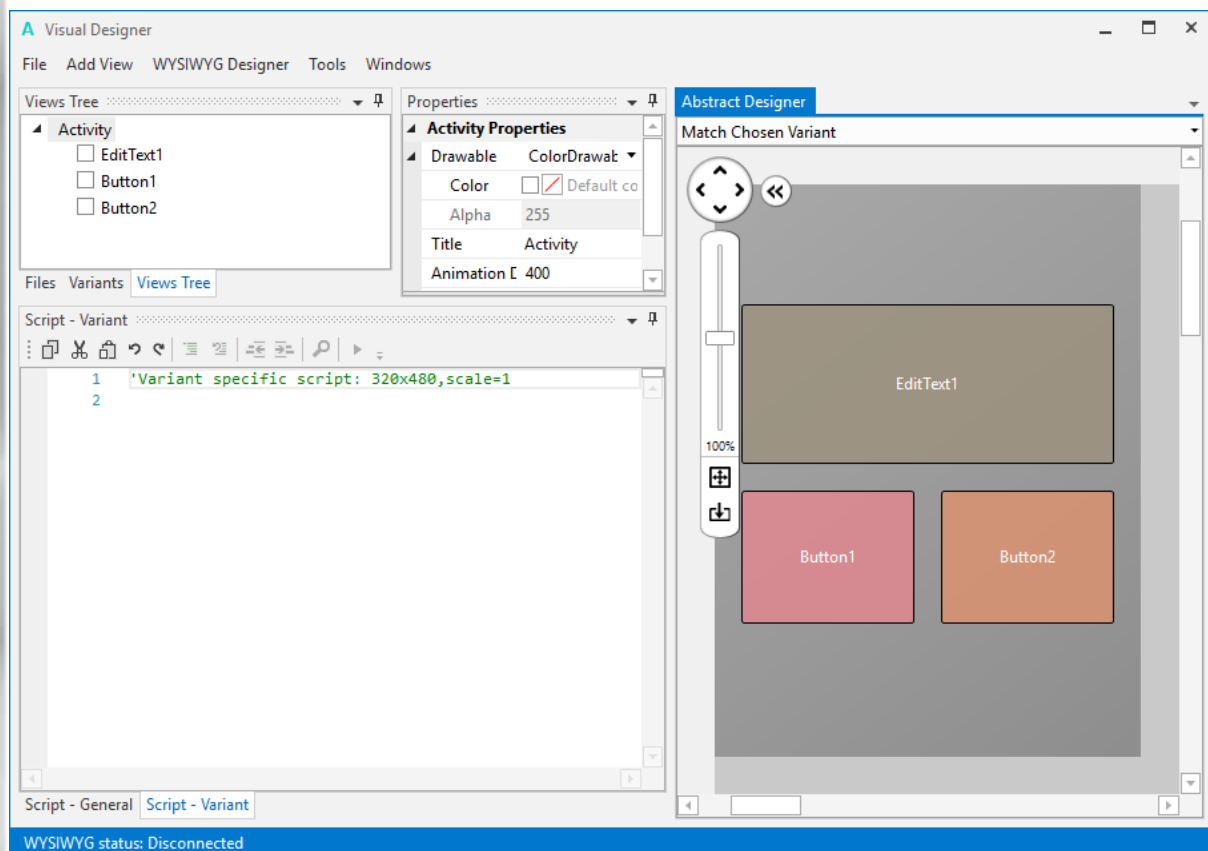
در اپلیکیشن فوق از ۵ صفحه استفاده شده است (لیست اصلی، فصل اول، فصل دوم، لیست منابع، منابع داخلی و منابع خارجی)؛ در صورتی که تنها ۳ Layout دارد.

به اجزای موجود در Layout اصطلاحاً View گفته می شود. View های مهم و کاربریشان در بخش «بررسی View ها» آموزش داده شده اند. اما فعلاً دو View پیش از موعد معرفی می شوند:

Button: کلید

EditText: فیلدی که در آن توسط کاربر متنی وارد می شود

برای اضافه کردن View به Layout در پنجره ی Designer از منوی Add View اجرای موردنظر خود را انتخاب کنید. فعلاً برای تمرین دو Button و یک EditText را اضافه کرده و آن ها را در محیط طراحی (Abstract Designer) به صورت زیر در آورید (برای جابجایی و یا تغییر اندازه ی View ی موردنظر ابتدا در محیط طراحی بر روی آن کلیک کرده و سپس اقدام به کشیدن آن با ماوس (برای جابجایی) و یا کشیدن گوشه های آن (برای تغییر اندازه) کنید).



همان طور که مشاهده می کنید در قسمت Views Tree نام تمامی View های موجود در این Layout نشان داده می شوند که با کلیک روی هر کدام (و یا

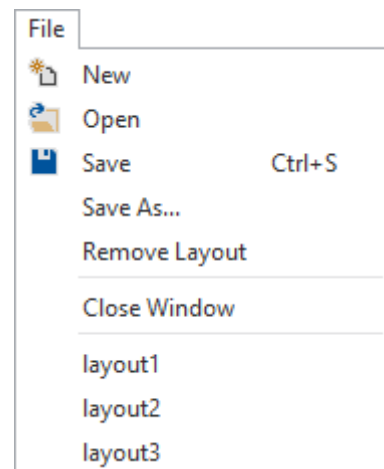


کلیک بر روی آن ها در محیط طراحی) امکان تغییر مشخصاتشان در قسمت Properties وجود خواهد داشت. نامی که هر View به طور پیشفرض دارد شامل نوع آن View و شماره ی آن (چندمین View ی کل پروژه از نوع خودش است؟) می باشد و در برنامه نویسی هم همین نام است که معرف View خواهد بود. تغییر این نام چندان توصیه نمی شود. اما در صورت تمایل می توانید برای این کار پس از کلیک بر روی نام View، از قسمت Properties و بخش Name نام آن را تغییر دهید.

نکته: این نام هیچ ربطی به چیزی که کاربر بعدها خواهد دید ندارد و تنها برای برنامه نویسی این View استفاده می شود. یعنی ممکن است کلیدی که Button1 نام دارد توسط کاربر با عبارت دیگری دیده شود.

برای تغییر نوشته ی روی کلید بر روی آن کلیک کرده و سپس در قسمت Properties و در بخش Text نام دلخواه خود را وارد کنید.

نکته: برای تغییر عکس پس زمینه ی کل Layout می توانید در قسمت Views Tree بر روی Activity کلیک کنید و سپس در قسمت Properties (که در این جا Properties کل Layout است) و بخش Drawable گزینه ی BitmapDrawable را انتخاب کنید (دو گزینه ی دیگر به ترتیب برای تغییر رنگ و قرار دادن گرادیانت (تداریج رنگ که دو رنگ در هم محو می شوند) هستند). سپس در بخش Image File تصویر موردنظر را انتخاب کنید. برای این که تصویر در این لیست دیده شود ابتدا باید آن را در بخش Files پنجره ی Designer وارد کرده باشید. برای مشاهده ی این بخش از سربزرگ های موجود در زیر Views Tree گزینه ی Files را انتخاب کرده و یا از منوی Windows آن را فراخوانی کنید سپس بر روی کلید Add Files کلیک کنید. متن تیترا بالای Layout نیز از طریق بخش Title قسمت Properties قابل تعیین است. بعد از اتمام طراحی Layout باید آن را ذخیره



کنید. برای این کار از منوی File پنجره ی Designer گزینه ی Save را انتخاب کرده و یا کلیدهایی Ctrl+S را فشار دهید. سپس در پنجره ای که باز می شود باید برای Layout خود نامی انتخاب کنید. بعد از این کار در صورتی که نیازی به Layout های بیشتر بود می توانید از منوی File گزینه ی New را انتخاب کرده و به طراحی Layout جدید بپردازید.

در نهایت لیست تمام Layout های پروژه در پایین منوی File پنجره ی Designer دیده می شوند که با کلیک روی هر کدام Layout موردنظر باز می شود.



طراحی ظاهر برنامه به پایان رسید و در پایان ذکر دو نکته الزامی است. اول آن که سعی کنید برای هر نوع View تمام موارد موجود در قسمت Properties پنجره ی Designer آن را بررسی کنید. چرا که امکان تغییر فونت، رنگ، اندازه و خیلی چیزهای دیگر در این قسمت ممکن است که توضیح خاصی برای آموزش ندارند و هر کسی می تواند کار آن ها را درک کند و دوم آن که به دلیل نیازمند بودن قسمت Designer Script به کدنویسی، این قسمت در بخش جداگانه ای بررسی شده است.



۴- بررسی View ها:

در این بخش View های مهم و کاربریشان بررسی می شوند. View ها همان طور که قبلاً هم اشاره شد از طریق منوی Add View ی پنجره ی Designer قابل اضافه شدن به Layout هستند.

Button: کلید یا دکمه

CheckBox: گزینه ای که امکان تیک خوردن کنار آن وجود دارد

EditText: فیلدی که در آن امکان تایپ متن وجود دارد

ImageView: قاب عکس (عکس باید در قسمت Files وارد شده باشد)

Label: نوشته

ListView: لیستی که می تواند شامل انواع مختلفی از گزینه ها باشد

Panel: جزئی که در حکم نگه دارنده و والد سایر اجزا می شود (برای مثال در ساخت ویجت).

ProgressBar: نواری که میزان پیشرفت یک عملیات را نشان می دهد (مثل دانلود یا نصب).

RadioButton: چند گزینه که تنها انتخاب یکی از آن ها برای کاربر میسر خواهد بود.

SeekBar: نوار لغزنده (مثل نوار کم و زیاد کردن روشنایی صفحه)

Spinner: منویی که با لمس گشوده می شود.

TabHost: فضایی که می تواند با استفاده از سربرگ ها چندین Layout را در خود جای دهد.

ToggleButton: کلیدی که امکان خاموش و روشن شدن دارد.

WebView: مستطیلی که امکان باز کردن یک سایت، یک فایل *.html و یا اجرای کدهای مستقل HTML را دارد.

Add View

AutoCompleteEditText
Button
CheckBox
CustomView
EditText
HorizontalScrollView
ImageView
Label
ListView
Panel
ProgressBar
RadioButton
ScrollView
SeekBar
Spinner
TabHost
ToggleButton
WebView



۵- قسمت Abstract Designer:

این قسمت، بخشی از پنجره ی Designer است و از اینجاست که برنامه نویسی برای اپلیکیشن آغاز می شود.

همه چیز در طراحی Layoutها خوب پیش می رود تا آن جا که هر گوشی اندروید اندازه ی صفحه ی متفاوتی پیدا می کند. این در صورتی است که اندازه ی Viewها ثابت است. بنابراین ممکن است در یک گوشی با صفحه ی کوچک Viewها از صفحه بیرون بزنند و در یک صفحه ی بزرگ مثل تبلت ها فضای خالی زیادی ایجاد شود. برای حل این مشکل بایستی Viewها را طوری تعریف نمود که اندازه شان به اندازه ی صفحه وابسته باشد و برای این کار به جای عدد از درصد استفاده می شود. برای مشاهده ی نحوه ی انجام این کار به مثال زیر دقت کنید:

این Layout را در نظر بگیرید. مشاهده می کنید که دارای یک WebView (مستطیل بزرگی که محتوا را نشان داده است)، یک Button (دریافت از کافه بازار) و یک Label (نشان دهنده ی حجم برنامه) است.

طول WebView از اول طول صفحه شروع می شود و تا آخر آن ادامه دارد (صفر درصد تا صد درصد طول صفحه).

عرض WebView هم از اول عرض صفحه شروع می شود ولی تا ۹۰ درصد عرض صفحه ادامه دارد (صفر درصد تا ۹۰ درصد عرض صفحه).

برای تعیین میزان اشغال بودن طول صفحه توسط طول View از دستور SetLeftAndRight و برای تعیین میزان اشغال بودن عرض صفحه توسط عرض View از دستور SetTopAndBottom

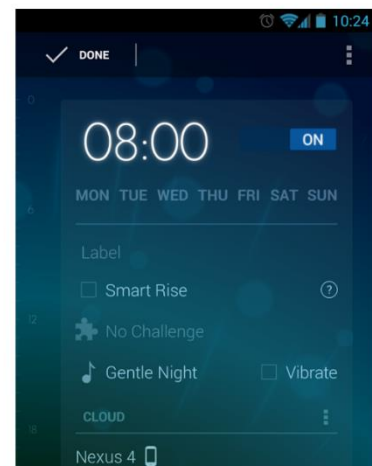
استفاده می شود. برای استفاده از این دستورها ابتدا نام View را در Designer (دقت کنید که در زیر آن سربرگ Script – General فعال باشد) نوشته سپس یک نقطه گذاشته و نام دستور را بنویسید و بعد از آن درصد و نوع محور (x برای طول و y برای عرض) را به صورت زیر مشخص کنید. کدهای زیر تمامی کدهای Script Designer این Layout هستند.

گلچین برترین نرم افزارهای اندروید



Timely

ساعت زنگدار و زمان سنج رنگارنگ



دریافت از کافه بازار

Size= 9.4 MB



۱۷

کتابخانه امید ایران

```
WebView1.SetLeftAndRight(0%x,100%x)  
WebView1.SetTopAndBottom(0%y,90%y)  
Button1.SetLeftAndRight(0%x,60%x)  
Button1.SetTopAndBottom(90%y,100%y)  
Label1.SetLeftAndRight(65%x,100%x)  
Label1.SetTopAndBottom(90%y,100%y)
```

با وارد کردن این کدها، Viewها به طور خودکار دچار تغییر اندازه می شوند. نکته: ممکن است با خارج شدن از این قسمت Viewها در قسمت محیط طراحی دچار به هم ریختگی شوند. اما این به هم ریختگی اهمیتی ندارد و در اصل به همان صورت که کدنویسی برای آن ها انجام شده نشان داده خواهند شد.

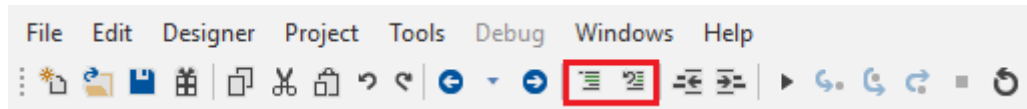


۶- محیط کدنویسی Basic4Android:

قبل از وارد شدن به کدنویسی باید به سه موضوع اشاره کرد.

کامنت کردن و آنکامنت کردن: در محیط برنامه نویسی Basic4Android می توان توضیحاتی را در لابلای کدها نوشت. این توضیحات جزء کدها حساب نمی شوند و هیچ تاثیری بر روند اجرای کدها ندارند و تنها برای توضیح کدها (مثل مواقعی که کدها گیج کننده هستند و یا چندین برنامه نویس روی یک اپلیکیشن کار می کنند و باید هر یک از کدهای دیگری سر در بیاورد) می باشند. برای نوشتن یک کامنت بایستی ابتدا در قسمت موردنظر به اول یک سطر خالی رفت سپس یک کاراکتر آپوستروف (') تایپ کرد و توضیح موردنظر خود را نوشت و یا سطر یا سطرهای موردنظر را انتخاب و از بالای برنامه بر روی کلید کامنت کلیک کرده و یا کلیدهای Ctrl+Q را فشرد.

آنکامنت کردن فرآیندی برعکس کامنت کردن است و سطر یا سطرهای انتخاب شده را از حالت کامنت در می آورد. برای انجام این کار باید بر روی کلید آنکامنت کردن کلیک کرد و یا کلیدهای Ctrl+W را فشرد.



(از چپ به راست: کلید کامنت کردن و کلید آنکامنت کردن)

نکته: هر آپوستروف تنها می تواند سطر خودش را کامنت کند.

کامنت ها در محیط برنامه نویسی Basic4Android به رنگ سبز در می آیند.

```
'This is a comment
```

مجزا بودن کدها در یک اکتیویتی (صفحه): کد هر اکتیویتی (صفحه) در واقع از چندین بخش مجزا تشکیل شده است که کد موجود در هر بخش در صورت به وقوع پیوستن شرط آن بخش اجرا می شود.

هر بخش با عبارت `Sub x` (که `x` در آن شرط موردنظر برای اجرایی شدن کد است) آغاز و با `End Sub` به پایان می رسد.

```
Sub X
    کدهایی که اگر شرط X رخ دهد اجرا می شوند
End Sub
```

همان طور که مشاهده می کنید آغاز تا پایان هر بخش با یک آکلاک خاکستری از سایر بخش ها جدا می شود که با کلیک بر روی مربع کنار `Sub` می توان کل بخش را جمع کرد تا نوشتن سایر بخش ها بدون شلوغی محیط کدنویسی همراه باشد. بعد از این کار کل بخش تنها در یک خط (به صورت `Sub X`) همراه باشد.



نشان داده می شود که با کلیک بر روی علامت مثبت بخش مجدداً گسترده می شود.

نکته: برای جمع و یا گسترده کردن تمام بخش ها کافی است از منوی Edit گزینه ی Outlining و سپس گزینه ی Toggle All را انتخاب کرده و یا کلیدهای Ctrl+Shift+O را فشار دهید (با هر بار انتخاب این گزینه به طور یک در میان عمل گسترده کردن و جمع کردن انجام می شود). دو گزینه ی دیگر قسمت Outlining یعنی Expand All و Collapse All نیز به ترتیب برای گشودن و جمع کردن کل بخش ها استفاده می شوند که با وجود گزینه ی Toggle All عملاً کاربردی ندارند.

انواع متغیر: متغیرها مانند خانه های خالی هستند که در برنامه نویسی مقادیری را می گیرند. برای مثال برنامه ای که نام شخص را در یک EditText می گیرد و سپس در جعبه ی پیامی نام او را نوشته و به وی خوش آمد می گوید ابتدا نام شخص را از EditText گرفته و در یک متغیر (خانه ی خالی) ذخیره می کند. سپس متغیر را فراخوانی کرده و در پیام به اطلاع کاربر می رساند.

مدل زیر نمونه ای فرضی از برنامه ی فوق را نشان می دهد:

۱- ایکس را در نظر بگیر

۲- ایکس را برابر متنی که شخص در EditText وارد می کند قرار بده

۳- به کاربر پیام بده: سلام جناب "ایکس"

در Basic4Android از سه نوع متغیر متناوباً استفاده می شود که String، Int و Boolean نام دارند.

متغیر String: متغیر String یا متغیر رشته ای متغیری است که می تواند هر نوع مقداری را در خود جای دهد.

متغیر Int: متغیری است که تنها می تواند شامل عدد باشد.

متغیر Boolean: متغیری است که تنها می تواند شامل دو مقدار True (آری) و False (خیر) باشد. برای مثال پاسخ سوال زیر یک متغیر Boolean است:

آیا امروز دوشنبه است؟ (جواب تنها می تواند بله و یا خیر باشد و حالتی خارج از این دو امکانپذیر نیست)

بعد از دانستن این مطالب می توان نگاهی اجمالی به قسمت کد نویسی برنامه انداخت.



Basic4Android

برنامه نویسی اندروید با

```

1 #Region Project Attributes
2     #ApplicationLabel: B4A Example
3     #VersionCode: 1
4     #VersionName:
5     'SupportedOrientations possible values: unspecified, landscape or portrait.
6     #SupportedOrientations: unspecified
7     #CanInstallToExternalStorage: False
8 #End Region
9
10 #Region Activity Attributes
11     #FullScreen: False
12     #IncludeTitle: True
13 #End Region
14
15 #Sub Process_Globals
16     'These global variables will be declared once when the application starts.
17     'These variables can be accessed from all modules.
18
19 End Sub
20
21 #Sub Globals
22     'These global variables will be redeclared each time the activity is created.
23     'These variables can only be accessed from this module.
24
25 End Sub
26
27 #Sub Activity_Create(FirstTime As Boolean)
28     'Do not forget to load the layout file created with the visual designer. For example:
29     'Activity.LoadLayout("Layout1")
30
31 End Sub
32
33 #Sub Activity_Resume
34
35 End Sub
36
37 #Sub Activity_Pause (UserClosed As Boolean)
38
39 End Sub

```

در صفحه ی کدنویسی Basic4Android کدهای بالا به صورت پیش فرض قرار دارند. در بعضی از قسمت ها خود برنامه توضیحاتی را درباره ی کاربرد آن بخش کامنت کرده است.

در ادامه تمام بخش ها توضیح داده می شوند.

بخش Project Attributes: این بخش تنها در اکتیویتی اصلی وجود دارد و مشخصات کلی اپلیکیشن را تعیین می کند.

```

#Region Project Attributes
    #ApplicationLabel: B4A Example    نام اپلیکیشن که کاربر مشاهده می کند
    #VersionCode: 1    کد ورژن
    #VersionName:    نام ورژن

```

کد ورژن نشان دهنده ی تعداد دفعاتی است که اپلیکیشن منتشر شده است. برای مثال اولین بار که اپلیکیشن منتشر می شود کد ورژن آن برابر ۱ است؛ دومین بار برابر ۲ و به همین صورت ادامه دارد.

اما نام ورژن همان ورژن معروف است که همه می گویند و قواعد خاص خود را دارد و می تواند شامل اعداد نیز باشد (و یا حتی کلمه؛ با این که اکیداً توصیه نمی شود).

```

'SupportedOrientations possible values: unspecified,
landscape or portrait.

```

#SupportedOrientations: unspecified جهت اپلیکیشن



در صورتی که این مقدار برابر portrait باشد جهت اپلیکیشن به صورت عمودی، landscape به صورت افقی و unspecified بسته به جهت گوشی خواهد بود.

امکان نصب روی حافظه ی خارجی `#CanInstallToExternalStorage: False` باشد این مقدار برابر True باشد امکان نصب اپلیکیشن روی کارت حافظه وجود خواهد داشت و در صورتی که برابر False باشد این امکان وجود نخواهد داشت.

#End Region

بخش Activity Attributes: این بخش مشخصات کلی اکتیویتی (صفحه) را تعیین می کند.

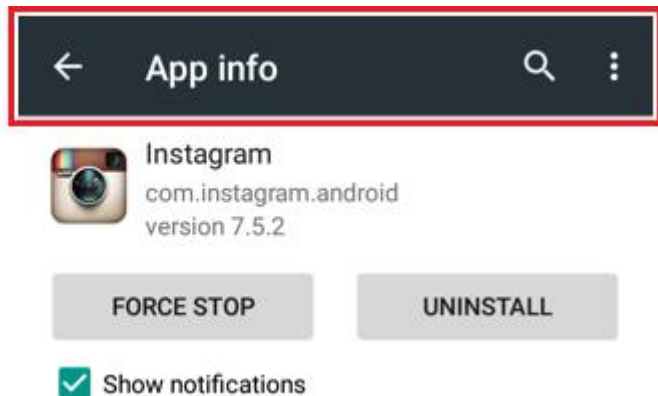
#Region Activity Attributes

تمام صفحه `#FullScreen: False` باشد این مقدار برابر False باشد نوار ابزار نشان داده خواهد شد و در صورتی که برابر True باشد نوار ابزار نشان داده خواهد شد و اکتیویتی به صورت تمام صفحه خواهد بود.

نشان داده شدن تیتیر `#IncludeTitle: True` اگر این مقدار برابر True باشد تیتیر نشان داده خواهد شد و اگر برابر False باشد تیتیر نشان داده نخواهد شد.

#End Region

تیتیر یک صفحه



Sub Process_Globals

'These global variables will be declared once when the application starts.

'These variables can be accessed from all modules.

در این بخش برخی از موارد خاص تعریف می شوند. موارد معرفی شده در این بخش در تمامی ماژول های پروژه در دسترس خواهند بود. (ماژول: اکتیویتی نوعی ماژول است. سایر ماژول ها عبارتند از کلاس، کد و سرویس)

End Sub

Sub Globals



'These global variables will be redeclared each time the activity is created.

'These variables can only be accessed from this module.

در این بخش مواردی از قبیل متغیرها تعریف می شوند. موارد معرفی شده در این بخش تنها در همین ماژول در دسترس خواهند بود.

End Sub

Sub Activity_Create (FirstTime As Boolean)

'Do not forget to load the layout file created with the visual designer. For example:

'Activity.LoadLayout("Layout1")

کدهای موجود در این قسمت به محض باز شدن ماژول اجرا خواهند شد.

End Sub

Sub Activity_Resume

کدهای موجود در این قسمت به محض باز شدن ماژول و یا باز کردن مجدد اپلیکیشن (اگر در پیش زمینه ی دستگاه باقی مانده باشد) اجرا خواهند شد.

End Sub

Sub Activity_Pause (UserClosed As Boolean)

کدهای موجود در این قسمت هنگام خروج کاربر از محیط اپلیکیشن و یا بستن آن اجرا خواهند شد.

End Sub



۷- اتصال Basic4Android به شبیه ساز و گوشی:

هنگام برنامه نویسی نیاز به وجود یک محیط اندرویدی جهت تست لحظه ای کارکرد اپلیکیشن احساس خواهد شد. برای فراهم کردن این محیط اندرویدی دو راه ممکن است. اتصال Basic4Android به یک شبیه ساز (Emulator) و یا اتصال آن به یک گوشی موبایل واقعی.

شبیه ساز: شبیه ساز اندروید نرم افزاری است که محیط اندروید را در یک پنجره در ویندوز شبیه سازی می کند و امکان کار با یک دستگاه مجازی اندرویدی را فراهم می سازد.

چندین شبیه ساز اندرویدی وجود دارد که در این مجموعه آموزشی از YouWave استفاده شده است. این نرم افزار را از یکی از سایت های دانلود دریافت و نصب کنید. پس از اجرای برنامه با محیط زیر رو به رو می شوید:



این شبیه ساز کاملاً مستقل از Basic4Android است و حتی می توان از آن در نبود Basic4Android و برای شبیه سازی اندروید در ویندوز (بدون هیچ مقاصد برنامه نویسی) هم استفاده کرد.

برای وارد کردن یک فایل apk در این شبیه ساز ابتدا وارد مسیری که در نوار بالای پنجره ی شبیه ساز نوشته شده (که معمولاً به صورت

C:\Users\youwave\کاربر در ویندوز\نام کاربر

می باشد) شوید و فایل های موردنظر خود را این فولدر قرار دهید. سپس از منوی View شبیه ساز گزینه ی Redraw Icons را انتخاب کرده تا لیست اپلیکیشن ها ریفرش شده و اپلیکیشن های جدید در آن قرار گیرند. حال با دوبار کلیک روی هر کدام اپلیکیشن در شبیه ساز اجرا می شود.

اتصال Basic4Android به شبیه ساز: با اتصال Basic4Android به شبیه ساز پس از هر بار کامپایل، اپلیکیشن به صورت اتوماتیک در شبیه ساز اجرا می شود. البته برای این کار بایستی شبیه ساز از قبل باز شده باشد.

برای اتصال شبیه ساز به Basic4Android یک راه این است که وارد مسیری که SDK معرفی شده به Basic4Android در آن نصب شده است بشوید (قبل از آن باید Basic4Android و شبیه ساز هر دو باز باشند). این مسیر احتمالاً

C:\Program Files\Android\android-sdk-windows

می باشد (مسیر دقیق از منوی Tools و گزینه ی Configure Paths برنامه ی Basic4Android (فیلد دوم) قابل مشاهده است. البته این مسیر فایل درون یکی از فولدرهای داخل SDK را نشان می دهد نه فولدر اصلی را).

سپس وارد فولدر platform-tools شوید. حال با کلیک بر روی نوار آدرس آن را به حالت ویرایش در آورده و در ادامه ی آن

\adb connect 127.0.0.1:5558

را تایپ کنید. یعنی متن موجود در نوار آدرس باید چیزی شبیه به آدرس زیر باشد:

C:\Program Files\Android\android-sdk-windows\platform-tools\adb connect 127.0.0.1:5558

اکنون با فشردن کلید اینتر اتصال بین شبیه ساز و Basic4Android برقرار می شود.

به دلیل آن که برای اتصال باید هر دفعه این کار انجام شود چندان مناسب نیست و بهتر است از راه دوم استفاده کنید.



راه دیگر کمی به برنامه نویسی DOS وابسته است. این راه شامل نوشتن کدهایی در Command Prompt است که در نهایت منجر به اتصال می شوند. اما برای این که کار آسان تر شود می توان کدهای فوق را در یک فایل ذخیره کرد و هر بار با دوبار کلیک روی آن فایل دستورات را اجرا نمود. بدین منظور ابتدا کدهای زیر را در یک پنجره ی جدید Notepad بنویسید:

C:

```
cd C:\Program Files (x86)\Android\android-sdk\platform-tools
```

```
adb connect 127.0.0.1:5558
```

```
pause
```

سپس فایل را ذخیره کرده و فرمت آن را از *.txt به *.bat تغییر دهید. از این پس با هر بار باز کردن این فایل، اتصال برقرار می شود.

نکته: ممکن است مسیر نصب SDK با مسیر بالا تفاوت داشته باشد. در این صورت آن مسیر را وارد کنید. همچنین در صورتی که SDK در درایو دیگری به جز درایو C قرار دارد هم بایستی در خط اول این کد نام آن درایو را وارد کنید.

نکته: به همین طریق می توان فایل قطع ارتباط نیز به وجود آورد. برای این کار کافی است به جای کلمه ی connect در کدهای بالا، کلمه ی disconnect را بنویسید.

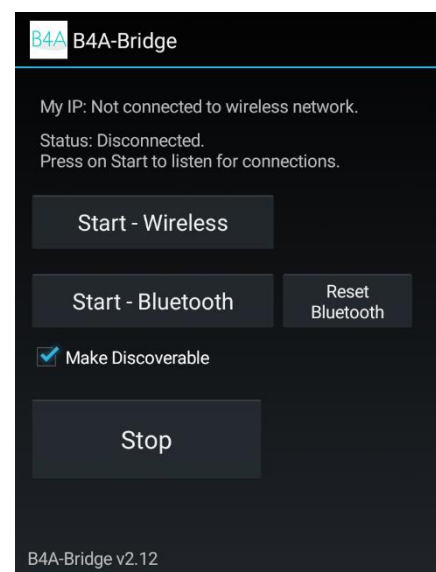
اتصال Basic4Android به گوشی: علاوه بر شبیه ساز، امکان اجرا و تست اپلیکیشن ها بر روی گوشی واقعی نیز وجود دارد. اما برای این کار به اپلیکیشن اختصاصی Basic4Android برای موبایل با نام B4A-Bridge نیاز است.

لینک دریافت این اپلیکیشن از گوگل پلی:

<https://play.google.com/store/apps/details?id=anywheresoftware.b4a.b4abridge>

(در مارکت کافه بازار نیز نمونه های وطنی مشابه وجود دارد)

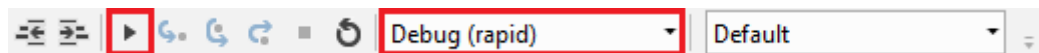
بعد از نصب و اجرای این برنامه می توان با استفاده از وایرلس و یا بلوتوث (امکان اتصال با بلوتوث در نسخه های جدید Basic4Android وجود ندارد) گوشی را به Basic4Android متصل کرد. برای این کار (در اینجا اتصال با



وایرلس توضیح داده می شود) باید گوشی و کامپیوتر به یک شبکه (و یا هر دو به اینترنت) متصل باشند. سپس در اپلیکیشن B4A-Bridge موجود بر روی گوشی کلید Start – Wireless را انتخاب کرد. بعد از این کار در کنار My IP، IP دستگاه (چهار عدد که با نقطه از هم جدا شده اند) دیده خواهد شد.

حال بایستی از منوی Tools برنامه ی Basic4Android روی کامپیوتر گزینه ی B4A Bridge و سپس Connect و New IP را انتخاب کرد. سپس در پنجره ای که باز می شود IP نوشته شده در My IP اپلیکیشن B4A-Bridge را وارد کرد. همواره در گوشه ی پایین سمت چپ برنامه وضعیت اتصال B4A-Bridge قابل مشاهده است.

کامپایل کردن اپلیکیشن: در هر مرحله از برنامه نویسی می توان پیش نمایشی از اپلیکیشن را به صورت آنی بر روی شبیه ساز و یا گوشی مشاهده کرد (نرم افزار شبیه ساز باید باز باشد و به Basic4Android هم متصل باشد). برای این کار ابتدا مطمئن شوید که در منوی بالای نرم افزار گزینه ی Debug (Rapid) فعال باشد سپس بر روی کلید ▶ کلیک کنید و یا کلید F5 را فشار دهید.



در هنگام تست برنامه در صورت رخ دادن خطایی، علت و توضیحات آن در سربرگ Logs ستون سمت راست برنامه نوشته خواهد شد. این قسمت از طریق کلیک بر روی سربرگ Logs این ستون و یا انتخاب گزینه ی Logs از منوی Windows و یا فشردن کلید F6 صورت می گیرد.

نکته: هنگامی که گزینه ی Debug (rapid) فعال است چون خروجی صرفاً برای تست کردن است برنامه با فایل دیباگ امضا می شود.



۸- کدهای مهم:

از این بخش کدنویسی در Basic4Android پررنگ می شود. این بخش خود شامل چندین قسمت است.

قبل از بررسی کدها باید کدهای شرطی توضیح داده شوند. کدهای شرطی به این صورت هستند که اگر شرط موردنظرشان صدق کند اجرا می شود و در غیر این صورت اجرا نمی شوند. ساختار کدهای فوق به این صورت است:

اگر «شرط اول صدق کند»

آنگاه «دستورات a رخ دهند»

در غیر این صورت؛

اگر «شرط دوم صدق کند»

آنگاه «دستورات b رخ دهند»

در غیر این صورت؛

.

.

.

اگر هیچ کدام از دستورات صدق نکردند

آنگاه «دستوراتی که اگر هیچ شرطی صدق نکرد رخ دهند»

پایان شرط ها

برای مثال:

اگر «نمره ی دانش آموز بالاتر از ۱۸ بود»

آنگاه «پیام بده: عالی»

در غیر این صورت؛

اگر «نمره ی دانش آموز بین ۱۵ تا ۱۸ بود»

آنگاه «پیام بده: خوب»



در غیر این صورت؛

اگر «نمره ی دانش آموز بین ۱۲ تا ۱۵ بود»

آنگاه «پیام بده: متوسط»

در غیر این صورت؛

اگر «نمره ی دانش آموز بین ۱۰ تا ۱۲ بود»

آنگاه «پیام بده: ضعیف»

در غیر این صورت؛

«پیام بده: دانش آموز مردود شده است»

پایان شرط ها

کدهای شرطی در Basic4Android به این صورت نوشته می شوند:

```
If Then شرط اول صدق کند
    دستورات a رخ دهند
Else If Then شرط دوم صدق کند
    دستورات b رخ دهند
Else If Then شرط سوم صدق کند
    دستورات c رخ دهند
Else
    دستوراتی که اگر هیچ شرطی صدق نکرد رخ دهند
End If
```

به عبارت دیگر:

```
If Then شرط اول صدق کند
    دستورات a رخ دهند
Else If Then شرط دوم صدق کند
    دستورات b رخ دهند
Else If Then شرط سوم صدق کند
    دستورات c رخ دهند
Else
    دستوراتی که اگر هیچ شرطی صدق نکرد رخ دهند
End If
```



علاوه بر این حالت کلی دو حالت دیگر هم وجود دارد.

۱- حالتی که اگر یک شرط اتفاق افتاد دستوری اجرا شود و اگر اتفاق نیفتاد دستور دیگر اجرا شود (به عبارت دیگر تنها یک شرط دارد):

```
If Then شرط صدق کند
    دستورات a رخ دهند
```

```
Else
    دستورات b رخ دهند
```

```
End If
```

در این حالت دیگر از Else If خبری نیست.

۲- حالتی که اگر یک یا چندین شرط صدق کرد کدهای مربوط به آن شرط ها اجرا شود ولی اگر هیچ شرطی صدق نکرد اتفاق خاصی نیفتد: در این حالت دیگر نوشتن Else و دستورات رخ ندادن هیچ کدام از شرط ها لزومی ندارد.

قسمت اول (کدهای کلی): این کدها در بسیاری از قسمت های کدنویسی قابل استفاده هستند. برای مثال هم در ابتدای باز شدن اپلیکیشن قابلیت اجرا شدن دارند و یا هنگام لمس یک کلید یا انتخاب یک گزینه از منوی اپلیکیشن. اما برای تمرین و تا قبل از آموختن نحوه ی کدنویسی برای View ها و... این کدها را در قسمت Activity_Create بنویسید تا به محض باز شدن برنامه اجرا شوند. توصیه می شود که تک تک کدهای این قسمت را در برنامه نوشته و سپس کامپایل کرده و نتیجه را مشاهده کنید.

```
Sub Activity_Create (FirstTime As Boolean)
    'Do not forget to load the layout file created with the
    visual designer. For example:
    'Activity.LoadLayout("Layout1")
```

کدهای موجود در این قسمت به محض باز شدن ماژول اجرا خواهند شد. بنابراین تمامی کدهای قسمت اول توضیحات را باید در این Sub بنویسید.

```
End Sub
```

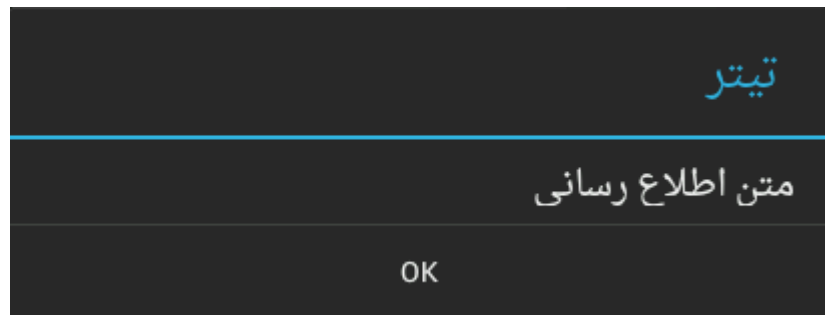


کد لود کردن Layout: معمولاً اولین کدی که در این قسمت نوشته می شود کدی است که Layout از قبل طراحی شده که ظاهر اپلیکیشن را تشکیل می دهد را لود می کند.

```
Activity.LoadLayout ("نام Layout")
```

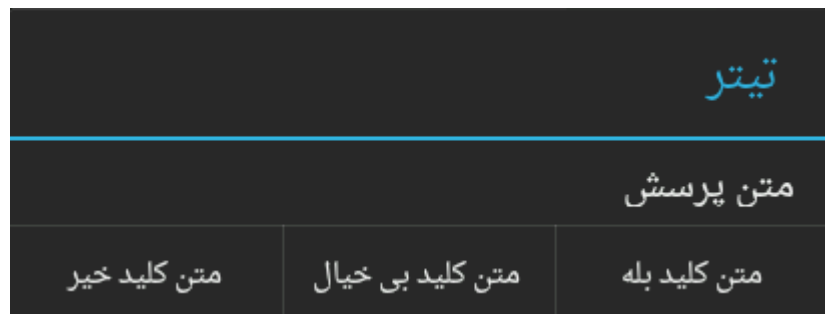
کد اطلاع یک موضوع در یک جعبه پیام (Msgbox):

```
Msgbox ("تیترا", "متن اطلاع رسانی")
```



کد پرسیدن یک سوال (بله یا خیر) از کاربر در جعبه ی پیام و جواب گرفتن از آن (Msgbox2):

```
Dim x As Int
x = MsgBox2 ("متن کلید بی", "متن کلید بله", "تیترا", "متن پرسش")
(Null, "متن کلید خیر", "خیال")
If x = DialogResult.POSITIVE Then
    'کدهایی که اگر کلید بله انتخاب شود اجرا می شوند'
else if x = DialogResult.CANCEL Then
    'کدهایی که اگر کلید بی خیال انتخاب شود اجرا می شوند'
else if x = DialogResult.NEGATIVE Then
    'کدهایی که اگر کلید خیر انتخاب شود اجرا می شوند'
End If
```



چند نکته در رابطه با MsgBox2 وجود دارد. اول آن که در کدهایش از دستورات Dim و If استفاده می کند که نحوه ی کار آن در قسمت قبلی توضیح داده شد.



نکته ی دیگر آن است که اگر به هر یک از سه کلید بالا نیاز نداشتید برای حذف آن کافی است در قسمت متن کلید تنها دو گیومه را باقی بگذارید و نوشته ی داخل آن را حذف کنید. همچنین می توانید کل کدهای مرتبط با آن کلید که در قسمت if وجود دارد را هم حذف کنید.

دیگر آن که امکان گذاشتن آیکون نیز برای MsgBox2 وجود دارد (این آیکون در گوشه ی بالایی جعبه مشاهده خواهد شد). برای قرار دادن آیکون ابتدا باید فایل آن را در برنامه وارد کنید (برای وارد کردن یک فایل به پروژه از ستون سمت راست برنامه سربرگ Files Manager را انتخاب یا از منوی Windows آن را فراخوانی کنید. سپس بر روی کلید Add Files کلیک کرده و فایل موردنظر خود را وارد کنید) سپس به جای عبارت Null در قسمت بالا عبارت `LoadBitmap(File.DirAssets, "icon.png")` را وارد نمایید (به جای icon.png بایستی نام فایل را وارد نمود). دقت کنید که در این حالت در آخر دستور دو پرانتز وجود دارد.

دستوراتی که در این بخش ها وارد می شوند حتماً لازم نیست که با عنوان دستور مطابقت کنند. برای مثال در قسمت NEGATIVE می توان دستوری را وارد کرد که کاربر به جای خروج وارد صفحه ی امتیازدهی اپلیکیشن در کافه بازار شود.

در نهایت باید گفت که در این مثال x نام متغیری است که جواب کاربر در آن قرار می گیرد و نام این متغیر می تواند هر چیز دیگری (کاملاً دلخواه) باشد.

کد اطلاع رسانی در قالب ToastMessage: ToastMessage عبارتی است که در یک مستطیل در پایین اپلیکیشن درج شده و پس از مدتی محو می شود. از ToastMessage بیشتر هنگامی استفاده می شود که کاربر کلید بازگشت را فشار دهد. در این حالت در یک ToastMessage پیام «برای خروج یک بار دیگر دکمه ی بازگشت را فشار دهید» به کاربر داده می شود.

`ToastMessageShow("متن ToastMessage", False)`

در صورتی که در دستور بالا عبارت False نوشته شود ToastMessage مدت کوتاهی نمایش داده می شود و در صورتی که به جای آن عبارت True نوشته شود مدت زمان نمایش آن طولانی تر خواهد شد.

کد اضافه کردن منو به اکتیویتی: منو لیستی است که با فشردن کلید منوی گوشه ی باز می شود و با انتخاب هر گزینه اتفاق مشخصی می افتد.



سه نوع آیتم برای اضافه کردن به منو وجود دارد. ساده ترین نوع آن (MenuItem) تنها شامل یک متن است.

```
Activity.AddItem("متن منوی اول", "m1")
Activity.AddItem("متن منوی دوم", "m2")
Activity.AddItem("متن منوی سوم", "m3")
Activity.AddItem("متن منوی چهارم", "m4")
```



در کدهای فوق گیومه ی اول مربوط به نام نمایشی منو و گیومه ی دوم مربوط به نام رویداد منو است. نام رویداد در برنامه نویسی این که هنگام انتخاب آن گزینه چه کدی اجرا شود استفاده می شود.

کدهایی که با انتخاب آیتم های منو اجرا می شوند در بخش Activity_Create نوشته نمی شوند؛ بلکه هر کدام نیاز به یک بخش مستقل دارند. پس دقت کنید که کدهای فوق نباید در داخل Subها نوشته شوند (چون خودشان Sub حساب می شوند). بهتر است این کدها را در آخر صفحه ی کدنویسی بنویسید.

```
Sub m1_click
    کدهایی که اگر منوی اول انتخاب شود اجرا می شوند
End Sub

Sub m2_click
    کدهایی که اگر منوی دوم انتخاب شود اجرا می شوند
End Sub

Sub m3_click
    کدهایی که اگر منوی سوم انتخاب شود اجرا می شوند
End Sub

Sub m4_click
    کدهایی که اگر منوی چهارم انتخاب شود اجرا می شوند
End Sub
```

(به جای m1، m2، m3 و m4 نام رویداد منوها نوشته می شود)



نکته: سایر انواع آیتم ها هم با کدهای بالا برنامه نویسی می شوند و تنها تفاوتشان در ظاهر آن ها است.

نوع دوم آیتم ها (MenuItem2) علاوه بر متن امکان داشتن آیکون را هم دارد.

```
Activity.AddItem2 ("متن", "m5", LoadBitmap(File.DirAssets,
("نام فایل آیکون"))
```

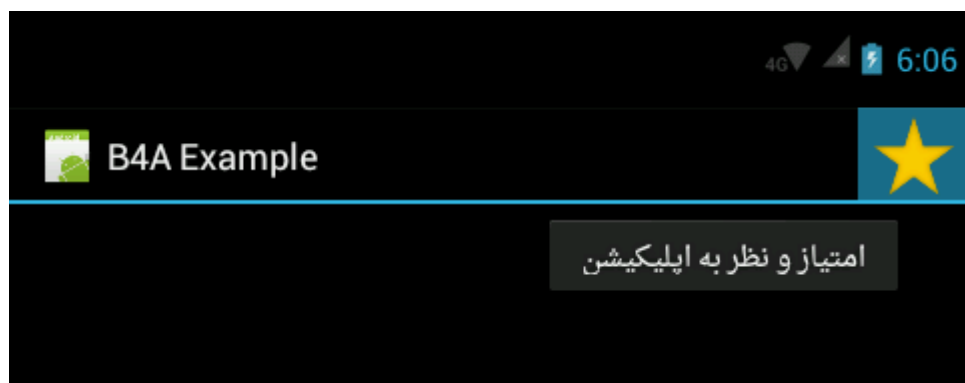
دقت کنید که فایل آیکون باید در برنامه وارد شده باشد.

نکته: تم فعلی اندروید یعنی Holo از آیکون در منوها پشتیبانی نمی کند؛ مگر آن که در اکشن بار قرار داشته باشند. بنابراین در صورت استفاده از آیکون یا باید تم اپلیکیشن را تغییر دهید و یا آیتم را در اکشن بار قرار دهید که در ادامه روش قرار دادن آیتم در Action Bar و در بخش «کار با Manifest Editor» نحوه ی تغییر تم اپلیکیشن توضیح داده شده است.

نوع سوم آیتم ها (MenuItem3) علاوه بر متن و آیکون، دارای امکان قرار داده شدن در اکشن بار نیز هست. اکشن بار نواری در بالای اکتیویتی است که در سمت چپ آن Activity (یا نام دلخواه) نوشته شده است. در صورتی که آیتمی در این نوار قرار بگیرد تنها آیکون آن دیده خواهد شد که با انتخاب آن کدهای مربوطه اجرا می شوند و با لمس کردن و نگه داشتن آن متن منو در یک ToastMessage نشان داده خواهد شد. البته اگر در اکشن بار فضای خالی موجود نباشد آیتم به لیست منوها منتقل خواهد شد.

```
Activity.AddItem3 ("متن", "m6", LoadBitmap(File.DirAssets,
("نام آیکون فایل"), True)
```

آخرین متغیر این دستور یک متغیر Boolean (بله یا خیر) است که اگر برابر True باشد آیتم در اکشن بار نشان داده شده و اگر برابر False باشد در اکشن بار نشان داده نمی شود.



کد مثال موجود در تصویر:

```
Activity.AddItem3 ("و نظر به اپلیکیشن", "امتیاز", "m6", LoadBitmap(File.DirAssets, "star.png"), True)
```



در این تصویر بر روی آیکون لمس و نگه داشته شده است تا ToastMessage آن نیز نمایش داده شود.
کد خارج شدن از اپلیکیشن:

```
ExitApplication
```

قسمت دوم (اجرا کردن یک کد پس از فشردن یکی از کلیدهای فیزیکی گوشی): دقت کنید که این قسمت هم نباید داخل یک Sub دیگر قرار بگیرد (به طور کلی هیچ Sub ای نباید داخل Sub دیگر قرار بگیرد).

```
Sub Activity_KeyPress (KeyCode As Int) As Boolean
    If KeyCode = KeyCodes.نام کلید Then
        کدهایی که با فشردن این کلید اجرا می شوند
    End If
End Sub
```

نام کلیدهای فیزیکی (گوشی های معمول):

KEYCODE_BACK : کلید بازگشت

KEYCODE_HOME : کلید خانه

KEYCODE_MENU : کلید منو

KEYCODE_POWER : کلید خاموش/روشن

KEYCODE_VOLUME_DOWN : کلید کم کردن صدا

KEYCODE_VOLUME_UP : کلید زیاد کردن صدا

نکته: با نوشتن عبارت `Return True` در قسمت کدهای فشردن یک کلید فیزیکی می توان فشردن آن را بی تاثیر کرد.

قسمت سوم (کدهای معرفی View های یک Layout به برنامه و عملکرد آن ها): تمام View های مورد استفاده در یک Layout بایستی در صفحه ای که قصد استفاده از آن ها را دارید معرفی شوند. راه دستی معرفی یک View آن است که داخل Sub Globals کد زیر نوشته شود:

```
Private نام View As View نوع
```

برای مثال:

```
Sub Globals
    Private ListView1 As ListView
```




```
Private WebView1 As WebView
Private Button1 As Button
End Sub
```

یک راه آسان تر برای این کار این است که از پنجره ی Designer نسبت به معرفی View ها اقدام شود. در این حالت کدهای فوق به طور اتوماتیک به این Sub اضافه می شوند. برای این کار در پنجره ی Designer بر روی View ی موردنظر و یا نام آن در Views Tree راست کلیک کرده و از منوی پدیدار شده گزینه ی Generate و Dim ... As ... را انتخاب کنید.

حال پس از تعریف View ها باید مشخص کرد که با انجام هر عمل (برای مثال لمس کردن یا لمس کردن و نگه داشتن) بر روی یک View ی مشخص چه کدهایی اجرا شوند. کدهایی که باید بعد از رخ دادن هر رویداد اجرا شوند، هر کدام نیازمند Sub جداگانه ای هستند. ساختار این Sub به صورت زیر است:

```
Sub نوع عمل نام View
    کدهایی که با عمل فوق بر روی View اجرا می شوند
End Sub
```

برای مثال:

```
Sub Button1_Click
    کدهایی که پس از کلیک (لمس) بر روی Button1 اجرا می شوند
End Sub
```

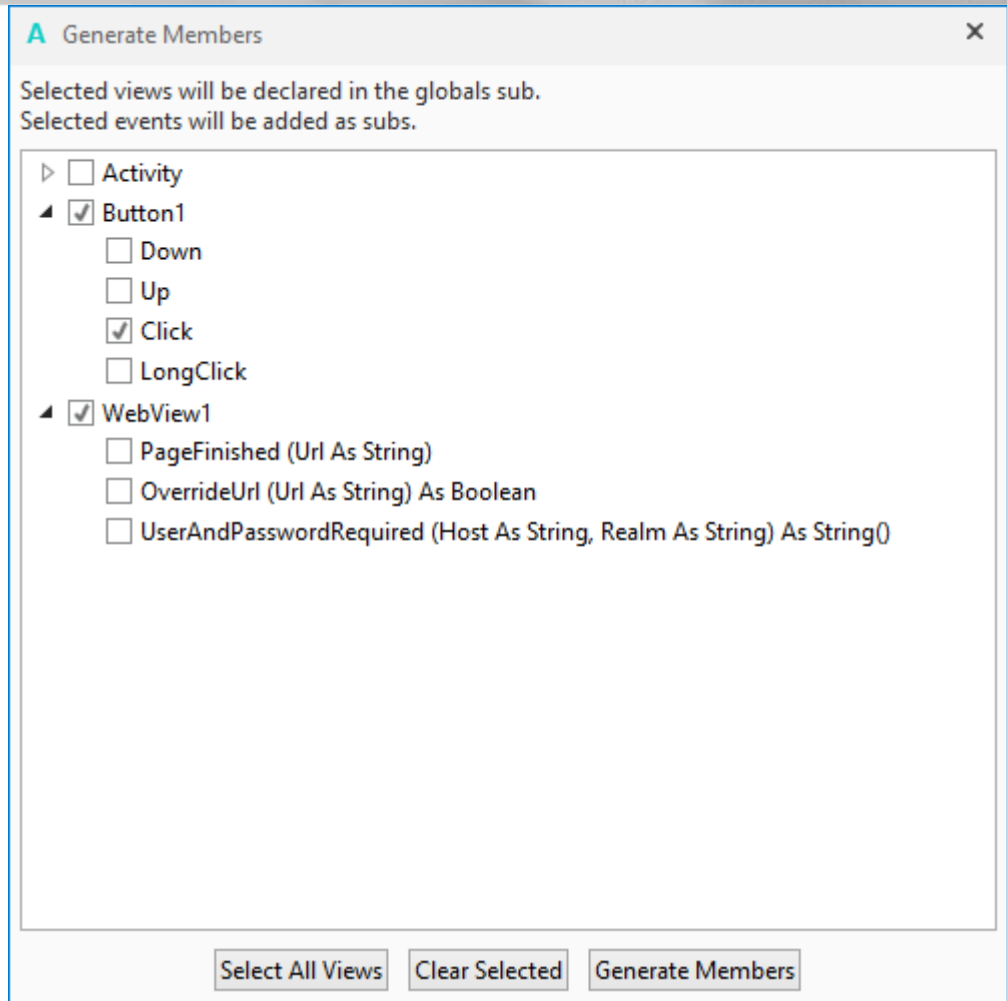
یا:

```
Sub Button1_LongClick
    کدهایی که پس از لمس طولانی بر روی Button1 اجرا می شوند
End Sub
```

البته نوشتن این کدها به صورت دستی چندان مطلوب نیست و امکان درج این کدها هم مثل کد معرفی View ها به صورت اتوماتیک و از طریق پنجره ی Designer وجود دارد. برای این کار در پنجره ی Designer بر روی View ی موردنظر و یا نام آن در Views Tree راست کلیک کرده و از منوی پدیدار شده گزینه ی Generate را انتخاب کنید. در زیر گزینه ی Dim ... As ... رویدادهایی که برای View ی موردنظر امکان پذیر هستند مشاهده می شوند که با انتخاب هر کدام Sub مربوطه در قسمت کدها درج خواهد شد.

نکته: راه معقول تر برای معرفی و کدنویسی برای View ها استفاده از گزینه ی Generate Members است. برای این کار ابتدا در پنجره ی Designer از منوی Tools گزینه ی Generate Members را انتخاب کنید.





در پنجره ی باز شده لیستی از View های Layout انتخابی نمایش داده می شود که با تیک دار کردن هر کدام و کلیک بر روی کلید Generate Members به برنامه معرفی می شوند. همچنین با کلیک بر روی مثلث کنار آن ها منویی درختی باز شده و لیستی از تمام رویدادهای ممکن برای آن View نمایش داده می شود که با انتخاب هر کدام و کلیک بر روی Generate Members، Sub آن رویداد به صفحه ی کدنویسی اضافه می گردد.

قسمت چهارم (کدهای وابسته به View ها): کدهای این قسمت مانند کدهای قسمت اول جنبه ی عمومی دارند و در اکثر مکان ها امکان نوشتن آن ها وجود دارد. اما به دلیل آن که هر کدام از آن ها بر روی یک View خاص اثر دارند قسمتی جداگانه برای آن ها در نظر گرفته شده است.

اول از همه باید گفت که ساختاری کلی برای کدهایی که با اجرا شدن، ظاهر View ها (یا حتی خود اکتیویتی که برای تغییر صفات آن باید در کدهای زیر به جای نام View کلمه ی Activity را تایپ نمود) را تغییر می دهند وجود دارد که به صورت زیر است:



نام ویو . بخش مدنظر برای تغییر = تغییرات (از چپ به راست)

برای مثال برای تغییر (یا تعیین) متن یک Label که در این مثال Label1 نام دارد از این کد استفاده می شود:

```
Label1.Text="متن موردنظر"
```

توجه داشته باشید که پس از تایپ کاراکتر نقطه در یک باکس تمامی حالات ممکن برای ادامه ی این کد آشکار می شود که در میانشان آن هایی که در کنار خود آیکون آپارفرانسه دارند صفات View را مشخص می کنند. این صفات از طریق قسمت Properties پنجره ی Designer هم قابل تعیین هستند.

در ادامه برخی از کدهای خاص مربوط به Viewها توضیح داده می شود.

کدهای مربوط به ListView (نام ListView در این کدها ListView1 در نظر گرفته شده است):

کد اضافه کردن یک آیتم یک خطی به ListView:

```
ListView1.AddSingleLine2("متن موردنظر", x)
```

در این کد به جای x بایستی یک عدد گذاشته شود (عبارت هم می توان نوشت اما باید درون گیومه باشد). از این عدد در کدنویسی مربوط به این آیتم استفاده می شود.

کد اضافه کردن یک آیتم دو خطی (تیترو توضیح) به ListView:

```
ListView1.AddTwoLines2("توضیح", "تیترو", X)
```

کد اضافه کردن یک آیتم دو خطی (تیترو توضیح) به همراه آیکون به ListView:

```
ListView1.AddTwoLinesAndBitmap2("توضیح", "تیترو", y, x)
```

در این کد به جای y بایستی Bitmap آیکون نوشته شود.

برای استفاده از یک Bitmap ابتدا باید در قسمت Sub Globals آن را به صورت زیر تعریف نمود (به جای y نام دلخواه خود را وارد کنید):

```
Dim y as Bitmap
```

سپس در قسمت Sub Activity_Create با کد زیر آن را فعال کنید:

```
y.Initialize(File.DirAssets, "نام فایل آیکون")
```



حال برای کدنویسی آیتم های یک منو باید از کد زیر استفاده کرد:

```
Sub ListView1_ItemClick (position As Int, value As Object)
    If value=1 Then
        ' کدهایی که اگر بر روی آیتم ۱ کلیک شد اجرا می شوند'
    Else If value=2 Then
        ' کدهایی که اگر بر روی آیتم ۲ کلیک شد اجرا می شوند'
    .
    .
    .
End If
End Sub
```

در این کد value همان x است.

نکته: استفاده از If در این دستور در صورتی که تعداد آیتم ها خیلی زیاد باشد به دلیل طولانی شدن چندان معقول نیست و به جای آن توصیه می شود از دستور Select Case استفاده شود. این دستور در قسمت سایر کدها بررسی شده است.

کدهای تغییر ظاهر ListView (این کدها با مثال توضیح داده شده اند):

تغییر ظاهر SingleLine:

```
ListView1.SingleLineLayout.Label.TextSize=22
ListView1.SingleLineLayout.Label.Gravity=Gravity.CENTER
ListView1.SingleLineLayout.Label.TextColor=Colors.Black
```

در این کدها اندازه ی متن بر روی ۲۲، تراز آن به صورت وسط چین و رنگ آن مشکی تنظیم شده است (همان طور که ملاحظه می کنید رنگ ها در Basic4Android به صورت نام رنگ. Colors هستند)

تغییر ظاهر TwoLines:

```
ListView1.TwoLinesLayout.Label.TextSize=22
ListView1.TwoLinesLayout.SecondLabel.Typeface=Typeface.FromAssets("bnazanin.ttf")
ListView1.TwoLinesLayout.Label.Gravity=Gravity.CENTER
ListView1.TwoLinesLayout.SecondLabel.Gravity=Gravity.CENTER
ListView1.TwoLinesLayout.Label.TextColor=Colors.Black
ListView1.TwoLinesLayout.SecondLabel.TextColor=Colors.DarkGray
```

در این کدها اندازه ی تیتر بر روی ۲۲ و فونت آن bnazanin.ttf تنظیم شده است (فایل فونت باید وارد برنامه شود). همچنین تراز تیتر و توضیح بر



روی وسط چین قرار گرفته است. رنگ تیترا مشکی و رنگ توضیح نیز خاکستری تیره است.

تغییر ظاهر TwoLinesAndBitmap:

```
ListView1.TwoLinesAndBitmap.Label.TextSize=22
ListView1.TwoLinesAndBitmap.SecondLabel.TextSize=14
ListView1.TwoLinesAndBitmap.SecondLabel.Typeface=Typeface.Load
FromAssets("bnazanin.ttf")
ListView1.TwoLinesAndBitmap.Label.Gravity=Gravity.LEFT
ListView1.TwoLinesAndBitmap.SecondLabel.Gravity=Gravity.LEFT
ListView1.TwoLinesAndBitmap.Label.TextColor=Colors.Black
ListView1.TwoLinesAndBitmap.SecondLabel.TextColor=Colors.DarkGray
```

این قسمت توضیح خاصی ندارد و مثل TwoLines است. هرچند که دیگر در قسمت آن از کلمه ی Layout استفاده نشده است.

نکته: کدنویسی برای Spinner نیز ساختاری شبیه به ListView دارد. اما با آن دارای تفاوت های کوچکی است. برای مثال برای اضافه کردن آیتم به Spinner بایستی از کد زیر استفاده کرد (نام Spinner در این مثال Spinner1 فرض شده است):

```
Spinner1.Add("آیتم اول")
Spinner1.Add("آیتم دوم")
Spinner1.Add("آیتم سوم")
.
```

سپس بایستی کدهای زیر را برای کدنویسی آیتم ها در برنامه وارد کرد:

```
Sub Spinner1_ItemClick (position As Int, value As Object)
    If position=0 Then
        ' کدهایی که اگر آیتم اول انتخاب شود اجرا می شوند'
    Else If position=1 Then
        ' کدهایی که اگر آیتم دوم انتخاب شود اجرا می شوند'
    Else If position=2 Then
        ' کدهایی که اگر آیتم سوم انتخاب شود اجرا می شوند'
    .
    .
    .
End If
End Sub
```



نکته: در کدنویسی بالا، ملاک جایگاه آیتم (position) است که شماره ی جایگاه از صفر شروع می شود (برای اولین آیتم).

نکته: برای ListView هم می توان مانند کد بالا به جای مقدار از جایگاه استفاده کرد (هر چند که توصیه نمی شود). برای این کار کافی است به جای AddSingleLine2، AddTwoLines2 و AddTwoLinesAndBitmap2 به ترتیب AddSingleLine، AddTwoLines و AddTwoLinesAndBitmap را وارد کرد. همچنین باید آخرین قسمت کدها یعنی مقدار (x) را نیز حذف کرد.

کدهای مربوط به TabHost: TabHost فضایی است که در بالای آن چندین سربرگ قرار می گیرد و با انتخاب هر سربرگ، یک Layout خاص (که قبلاً در پنجره ی Designer طراحی شده است) نشان داده می شود.

قبل از هر چیز، باید گفت که هنگامی که برای مثال در Designer یک Layout با نام test ساخته می شود، آن Layout در قالب یک فایل با نام test.bal در فولدر اصلی پروژه به وجود می آید. وارد کردن layoutها در سربرگ های TabHost با نام فایل آن ها (نام layout + پسوند دات bal) صورت می پذیرد.

همچنین باید بدانید که TabHost خود نیز باید مانند هر View دیگری در layout اصلی قرار بگیرد و Layoutهایی که باید درون آن باشند هیچ ارتباطی به layout اصلی ندارند و همگی باید تک تک در پنجره ی Designer طراحی شوند.

برای اضافه کردن سربرگ به TabHost از کد زیر استفاده می شود:

```
TabHost1.AddTab("نام سربرگ", "نام فایل layout")
```

مثال:

```
TabHost1.AddTab("سربرگ یک", "layout1.bal")
TabHost1.AddTab("سربرگ دو", "layout2.bal")
TabHost1.AddTab("سربرگ سه", "layout3.bal")
TabHost1.AddTab("سربرگ چهار", "layout4.bal")
```

کدهای مربوط به WebView (نام WebView در این کدها WebView1 در نظر گرفته شده است):

کد باز کردن یک آدرس وب در WebView (در اینجا سایت گوگل):

```
WebView1.LoadUrl("http://google.com")
```



کد اجرا کردن یک کد html در WebView:

```
WebView1.LoadHtml("کد html")
```

در صورتی که قصد دارید یک فایل *.html را در WebView اجرا کنید ابتدا باید آن را وارد پروژه کرده سپس از کد زیر استفاده کنید:

```
WebView1.LoadUrl("file:///android_asset/sample.html")
```

به جای sample.html نام فایل خود را وارد کنید.

فایل *.html در ساخت کتاب به کار می رود. برای ساخت یک فایل *.html سه راه وجود دارد. راه اول آن است که مطلب موردنظر را در نرم افزار Word نوشته و سپس Save As Web Page کنید (توصیه نمی شود). راه دوم آن است که کدهای خود را به صورت دستی و به زبان HTML کدنویسی کنید و راه سوم این است که از نرم افزارهایی مثل CoffeeCup یا ویرایشگرهای وبلاگی مثل وردپرس (و یا حتی بلاگفا) استفاده شود. توضیحات بیش تر در این باره در قسمت «ضمیمه ۱: آموزش مقدماتی HTML» داده شده است.

قسمت پنجم (کد معرفی یک متغیر و نحوه ی استفاده از متغیر): برای استفاده از متغیرها (String، Int و Boolean)، آن ها نیز باید مثل Viewها تعریف شوند. تعریف متغیر نیز در قسمت Sub Globals صورت می گیرد و به صورت زیر است:

```
Dim اسم متغیر As نام متغیر
```

مثال:

```
Dim a As Int
```

می توان بعد از این خط متغیر را مقداردهی کرد و یا هر دو کار را در یک خط انجام داد:

```
Dim a As Int
```

```
a=20
```

و یا

```
Dim a As Int = 20
```

دقت کنید که باید مقدار متغیرهای String در داخل گیومه قرار بگیرد. اما برای Int و Boolean این کار لازم نیست.



پس از تعریف متغیر می توان از آن استفاده کرد. استفاده از متغیر در همه جا ممکن است.

برای مثال این دو دستور یک کار را انجام می دهند:

```
ToastMessageShow("hello world", False)
```

و

```
Dim h As String = "hello world"
ToastMessageShow(h, False)
```

نام متغیر به هیچ وجه نباید درون گیومه قرار بگیرد. چون در این صورت به عنوان یک رشته شناخته می شود و به جای مقدار درونش، خودش بازگردانی می شود.

متغیرهای Int در عملیات های محاسباتی قابل استفاده هستند.

مثال: برنامه ای که عددی را دریافت کرده (به هر طریقی)، به علاوه ی ۲ و سپس در یک جعبه ی پیام نشان می دهد:

```
Dim number As Int
Dim answer As Int
answer=number+2
Msgbox(answer, "result")
```

انواع دیگری از متغیرها نیز وجود دارند که به View ها وابسته هستند. برای مثال EditText1.Text برابر مقداری است که توسط کاربر در EditText1 تایپ کرده است و یا CheckBox1.Checked متغیری Boolean است که بررسی می کند آیا CheckBox1 تیک خورده است یا نه.

کد قرار دادن محتوای یک فایل *.txt به عنوان مقدار یک متغیر:

```
txt=File.ReadString(File.DirAssets, "sample.txt")
```

در این جا نام متغیر برابر txt و نام فایل متنی sample.txt در نظر گرفته شده است. دقت کنید که این فایل باید به پروژه اضافه شده باشد.

کد ایجاد یک عدد تصادفی (در بازه ی a تا b) و قرار دادن آن در متغیر:

```
random=Rnd(a, b)
```

(نام متغیر random در نظر گرفته شده است)



قسمت ششم (تایمر): با استفاده از تایمر می توان کدهای دلخواه را پس از یک وقفه ی زمانی اجرا کرد و یا اجرای آن ها را در فواصل زمانی دلخواه تکرار نمود.

برای استفاده از یک تایمر ابتدا باید آن را در قسمت **Sub Process_Globals** تعریف نمود:

```
Dim Timer1 As Timer
```

سپس باید آن را در قسمتی که قصد استفاده از تایمر را دارید فعال کنید:

```
Timer1.Initialize("نام رویداد", x)
```

به جای نام رویداد نام دلخواه خود و به جای x مدت زمان تایمر (به میلی ثانیه) را وارد کنید.

حال برای شروع تایمر از دستور:

```
Timer1.Enabled=True
```

استفاده کنید. حال برای نوشتن کدهایی که باید پس از گذشت زمان تایمر اجرا شوند بایستی آن ها را در Sub زیر قرار داد:

```
Sub Tick نام رویداد
```

کدهایی که پس از گذشت زمان تایمر اجرا می شوند'

```
End Sub
```

کدهای موجود در این Sub هر x میلی ثانیه (مقداری که در دستور initialize تایمر داده شده است) یک بار متناوباً اجرا می شوند. از این خاصیت می توان برای ساخت ساعت به همراه دقیقه شمار و ثانیه شمار استفاده کرد. مثال:

```
Sub Activity_Create(FirstTime As Boolean)
```

```
    Timer1.Initialize("time", 1000)
```

```
    Timer1.Enabled=True
```

```
    Activity.LoadLayout("layout1")
```

```
End Sub
```

```
Sub time_tick
```

```
    label1.Text=DateTime.Time(DateTime.Now)
```

```
End Sub
```

با کد `DateTime.Time(DateTime.Now)` می توان زمان را در قالب HH:MM:SS دریافت کرد و تایمر بالا این مقدار را هر یک ثانیه (هزار میلی ثانیه) یک بار ریفرش می کند.



در صورتی که می خواهید از تایمر برای تاخیر اجرای یک کد استفاده کنید و نه تکرار آن (برای مثال صفحه ی خوش آمد گویی به کاربر که دو ثانیه نشان داده شود و پس از آن صفحه ی اصلی باز شود) باید در Sub اتمام زمان تایمر، آن را با کد `Timer1.Enabled=False` متوقف کنید. با این کار در واقع تایمر تنها یک دور اجرا شده و پس از آن متوقف می شود.

قسمت هفتم (سایر کدها):

کد `Select Case`: این کد ساختاری شبیه به کد `If/Then` دارد. اما از آن خلاصه تر است و کار با آن آسان تر می باشد.

برای توضیح این کد در یک مثال ابتدا یک دستور با کد `If/Then` نوشته شده، سپس همان دستور با کد `Select Case` شرح داده می شود.

این دستور اعداد یک تا پنج را تبدیل به حروف می کند و در یک `ToastMessage` نشان می دهد.

دستور فوق با کد `If/Then`:

```
If n=1 Then
    ToastMessageShow("one", False)
Else If n=2 Then
    ToastMessageShow("two", False)
Else If n=3 Then
    ToastMessageShow("three", False)
Else If n=4 Then
    ToastMessageShow("four", False)
Else If n=5 Then
    ToastMessageShow("five", False)
End If
```

همین دستور با کد `Select Case`:

```
Select n
Case 1
    ToastMessageShow("one", False)
Case 2
    ToastMessageShow("two", False)
Case 3
    ToastMessageShow("three", False)
Case 4
    ToastMessageShow("four", False)
Case 5
    ToastMessageShow("five", False)
```



End Select

همان طور که مشاهده می کنید این دستور نیاز به تایپ کم تری دارد و نسبت به کد If/Then از سادگی بیشتری نیز برخوردار است.

کد Try/Catch: گاهی اوقات اجرای یک کد در یک گوشی با موفقیت انجام نمی شود. برای مثال اگر یک کد که وظیفه ی ارسال اس ام اس را بر عهده دارد در حالی اجرا شود که کاربر به شبکه متصل نیست اپلیکیشن دچار خطا می شود و ارور می دهد و یا هنگامی که یک کد کاربر را به صفحه ی نظرخواهی اپلیکیشن در کافه بازار هدایت می کند در حالی که کاربر این مارکت را بر روی گوشی خود نصب نکرده است باز هم ممکن است با خطا مواجه شود. برای این که از نمایش صفحه ی ارور به کاربر خودداری شود و هنگام ناموفق بودن اجرای یک دستور، دستور خاصی اجرا شود از کد Try/Catch استفاده می شود. ساختار این کد به صورت زیر است:

Try

کدهای اصلی'

Catch

کدهایی که اگر امکان اجرای کدهای اصلی نبود اجرا می شوند'

End Try

برای مثال:

Try

sms.Send("شماره ی موبایل", "متن پیامک")
ToastMessageShow("پیامک ارسال شد", False)

Catch

ToastMessageShow("پیامک ارسال نشد", False)

End Try

(ارسال اس ام اس در بخش کتابخانه ها آموزش داده شده است)

نکته ی مهم (نحوه ی concat کردن): برای این که در یک مکان (مثل متنی که قرار است یک ToastMessage و یا یک MsgBox نشان دهد) بتوان در کنار هم از متغیرها و مقادیر رشته ای استفاده کرد باید آن ها را concat نمود. برای این کار از کاراکتر & استفاده می شود.

مثال:

name=EditText1.Text

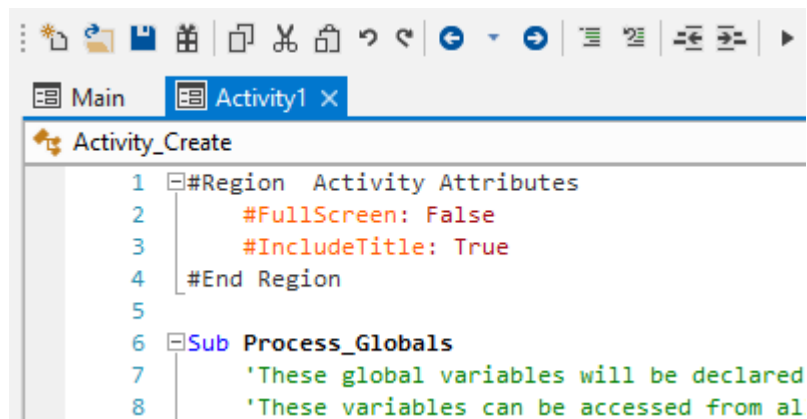
Msgbox("Welcome dear "&name,"Hello")



۹- استفاده از چند اکتیویتی:

تا اینجا آموزش ساخت یک صفحه (اکتیویتی) داده شد. اما اپلیکیشن ها بیش از یک صفحه دارند. برای اضافه کردن یک صفحه ی دیگر به اپلیکیشن از منوی Project گزینه ی Add New Module و سپس Activity Module را انتخاب کنید.

در پنجره ای که باز می شود نام دلخواه خود را برای صفحه وارد نموده و بر روی کلید Ok کلیک کنید.



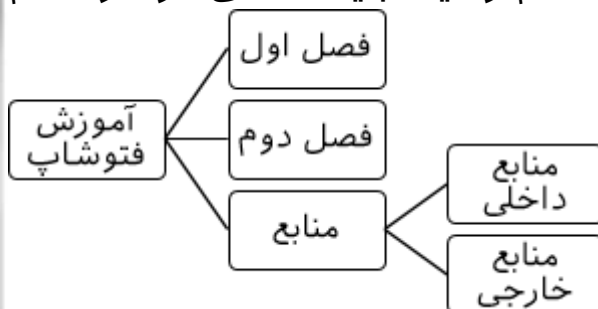
همان طور که مشاهده می کنید در لیست سربرگ های بالای قسمت کدنویسی نام اکتیویتی اضافه می شود. پس از این با کلیک روی هر سربرگ پنجره ی کدنویسی جداگانه ای که مخصوص همان اکتیویتی است گشوده می شود که در هر کدام باید کدنویسی یک صفحه صورت بگیرد.

نکته: با بستن سربرگ ها اکتیویتی ها حذف نمی شوند و برای مشاهده ی مجدد آن ها باید از ستون سمت راست برنامه بر روی سربرگ Modules کلیک و یا از منوی Windows گزینه ی Modules را انتخاب کرد. حال با دوبار کلیک روی نام هر کدام مجدداً به سربرگ اضافه می شوند.

برای حذف کردن یک اکتیویتی باید در حالی که آن اکتیویتی فعال است از منوی Project گزینه ی Remove Module را انتخاب کرد.

پس از این که کدنویسی هر صفحه به اتمام رسید، باید امکانی در هر کدام فراهم کرد که بتوان از طریق آن به سایر صفحات نیز دسترسی داشت.

برای مثال در این اپلیکیشن فرضی در صفحه ی منابع بایستی با انتخاب هر یک از آیتم های «منابع داخلی» و «منابع خارجی» اکتیویتی مربوطه باز



شود. همچنین با فشردن شدن کلید فیزیکی بازگشت هم باید اکتیویتی صفحه ی اصلی برنامه باز شود.

برای باز کردن یک اکتیویتی دیگر از کد زیر استفاده می شود:

```
StartActivity("نام اکتیویتی")  
Activity.Finish
```

کد `Activity.Finish` به تنهایی منجر به بسته شدن اکتیویتی فعال می شود و استفاده از آن در این جا تنها برای این است که پیش از باز شدن یک اکتیویتی جدید، اکتیویتی قبلی بسته شود.



۱۰- کتابخانه ها (و آموزش کار با کتابخانه ی Phone):

کتابخانه ها امکانی هستند که هر کدام می توانند قابلیت های جدیدی را به Basic4Android اضافه کنند. هر کتابخانه از دو فایل *.jar و *.xml تشکیل شده است. برای اضافه کردن یک کتابخانه یک راه اضافه کردن فایل های کتابخانه به فولدر اصلی کتابخانه ها است (که در مسیر نصب Basic4Android و با نام Libraries قرار دارد) و راه دیگر (که توصیه می شود) آن است که فایل کتابخانه های موردنظر در یک فولدر قرار گرفته و مسیر آن فولدر در بخش Additional Libraries پنجره ی Configure Paths واقع در منوی Tools وارد شود.

پس از انجام مراحل فوق کتابخانه های جدید در لیست کتابخانه های Basic4Android (ستون سمت راست برنامه سربرگ Libraries Manager که از طریق منوی Windows هم قابل فراخوانی است) نمایش داده می شود. با تیک دار کردن نام کتابخانه در این لیست امکان استفاده از آن در پروژه ی فعلی وجود خواهد داشت.

در ادامه چند مورد از کارهایی که می توان با یکی از کاربردی ترین و مهم ترین کتابخانه های Basic4Android یعنی Phone (به طور پیشفرض در لیست کتابخانه ها قرار دارد) انجام داد بررسی می شود. اما قبل از آن تیک کنار Phone در لیست کتابخانه ها را بزنید.

ارسال اس ام اس:

ابتدا یک متغیر PhoneSms در بخش Sub Globals تعریف کنید (در این جا با نام sms تعریف شده است).

```
Dim sms as PhoneSms
```

سپس از کد زیر در مکان دلخواه استفاده کنید.

```
sms.Send("متن پیامک", "شماره ی موبایل")
```

ارسال ایمیل:

ابتدا یک متغیر Email در بخش Sub Globals تعریف کنید (در این جا با نام mail تعریف شده است).

```
Dim mail As Email
```



سپس از کد زیر در مکان دلخواه استفاده کنید.

```
mail.To.Add("آدرس ایمیل")
mail.Subject = "موضوع ایمیل"
mail.Body = "متن ایمیل"
StartActivity(mail.GetIntent)
```

برقراری تماس تلفنی:

ابتدا یک متغیر PhoneCalls در بخش Sub Globals تعریف کنید (در این جا با نام phonecall تعریف شده است).

```
Dim phonecall As PhoneCalls
```

سپس از کد زیر در مکان دلخواه استفاده کنید.

```
StartActivity(phonecall.Call("شماره تلفن"))
```

جلوگیری از خاموش شدن صفحه:

ابتدا یک متغیر PhoneWakeState در بخش Sub Globals تعریف کنید (در این جا با نام phonewake تعریف شده است).

```
Dim phonewake As PhoneWakeState
```

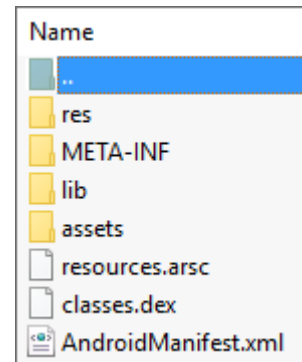
سپس از کد زیر در مکانی که قصد دارید صفحه خاموش نشود استفاده کنید.

```
phonewake.KeepAlive(True)
```



۱۱- کار با Manifest Editor:

اگر هر اپلیکیشن اندرویدی را با برنامه ای مثل WinRAR باز کنید مشاهده می کنید که درون آن یک فایل با نام AndroidManifest.xml وجود دارد.



وظیفه ی این فایل این است که اپلیکیشن را به سیستم عامل اندروید بشناساند. همچنین دسترسی های اپلیکیشن نیز در این فایل قرار دارد (برای مثال برای این که برنامه بتواند اس ام اس ارسال کند باید دسترسی این کار به آن داده شود).

Manifest Editor پنجره ای است که در منوی Project قرار گرفته است و وظیفه ی ویرایش این فایل را بر عهده دارد.

در ادامه سه کار که می توان با Manifest Editor انجام داد شرح داده شده است.

اضافه یا حذف کردن یک دسترسی:

برای اضافه کردن یک دسترسی باید کد فوق را به Manifest Editor اضافه کرد:

`AddPermission("نام دسترسی")`

نام دسترسی ها را در این تاپیک مشاهده کنید: [لینک](#)

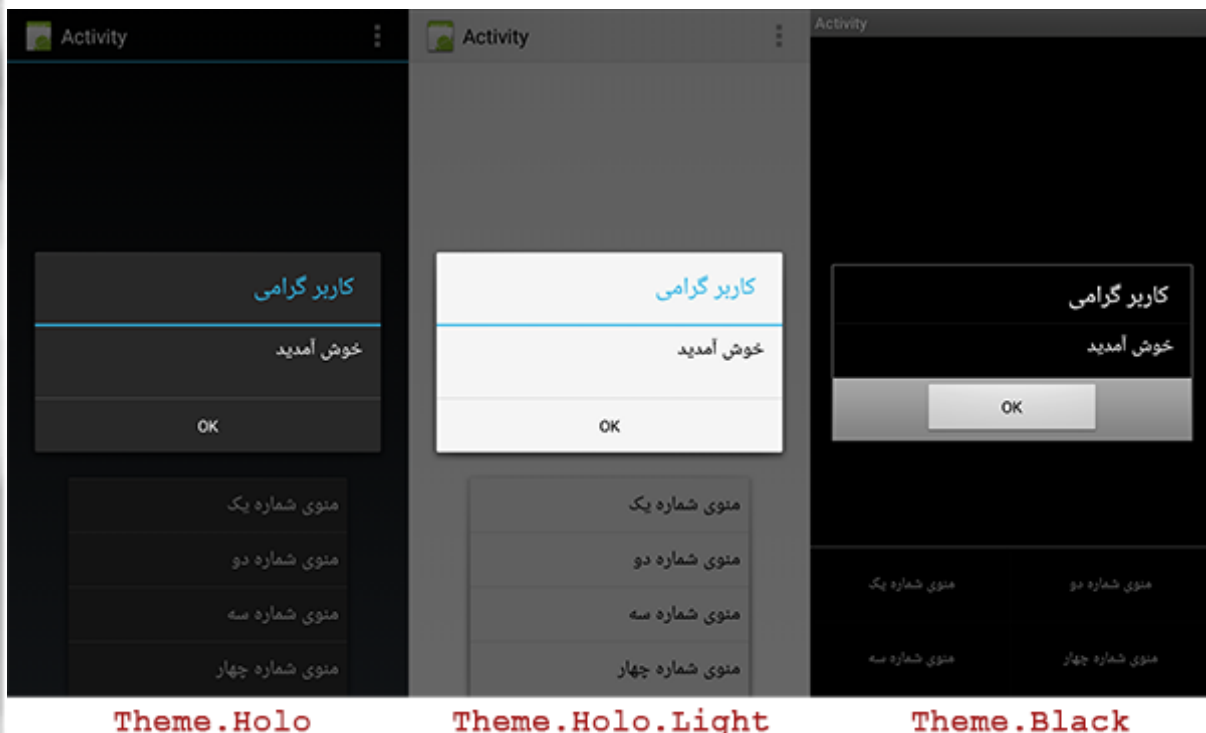
گاهی اوقات برخی کتابخانه ها به طور اتوماتیک دسترسی هایی را به اپلیکیشن می دهند که مورد تایید مارکت ها نیست.

برای حذف یک دسترسی باید کد فوق را به Manifest Editor اضافه کرد:

`RemovePermission("نام دسترسی")`



تغییر تم برنامه:



تعیین تم اپلیکیشن (که به صورت پیشفرض بر روی Theme.Holo است) نیز در قسمت Manifest Editor صورت می گیرد. برای تغییر تم در Manifest Editor عبارت Theme.Holo را پیدا کرده و نام تم موردنظر خود را جایگزین کنید (نام تم به بزرگی و کوچکی حروف حساس است).

افقی کردن یک اکتیویتی خاص:

با اضافه کردن این کد به Manifest Editor می توان یک اکتیویتی خاص را افقی کرد:

```
SetActivityAttribute(activity, android:screenOrientation, "landscape")
```

(به جای activity باید نام اکتیویتی موردنظر را وارد نمود)



۱۲- گرفتن خروجی نهایی از اپلیکیشن اندرویدی:

و سرانجام نوبت به دریافت فایل apk* اپلیکیشن می رسد. برای گرفتن خروجی از اپلیکیشن ابتدا مطمئن شوید که اپلیکیشن امضای صحیحی داشته باشد (و دیباگ یا اشتباه نباشد) سپس از منوی بالای برنامه گزینه ی (Rapid) Debug را به Release تغییر دهید و یک بار برنامه را کامپایل کنید. بعد از این کار فایل اپلیکیشن در محلی که فایل b4a* پروژه ذخیره شده است در فولدری با نام Objects قرار می گیرد.

نکته ی آخر آن که سورس اپلیکیشن (پروژه ی قابل ویرایش در محیط برنامه نویسی) تنها به فایل b4a* خلاصه نمی شود و برای دریافت سورس کامل بایستی از منوی File برنامه گزینه ی Export As Zip را انتخاب کرد تا تمامی فایل های لازم برای اجرای پروژه ی اپلیکیشن در Basic4Android در یک فایل zip* قرار داده شده و ذخیره شود.



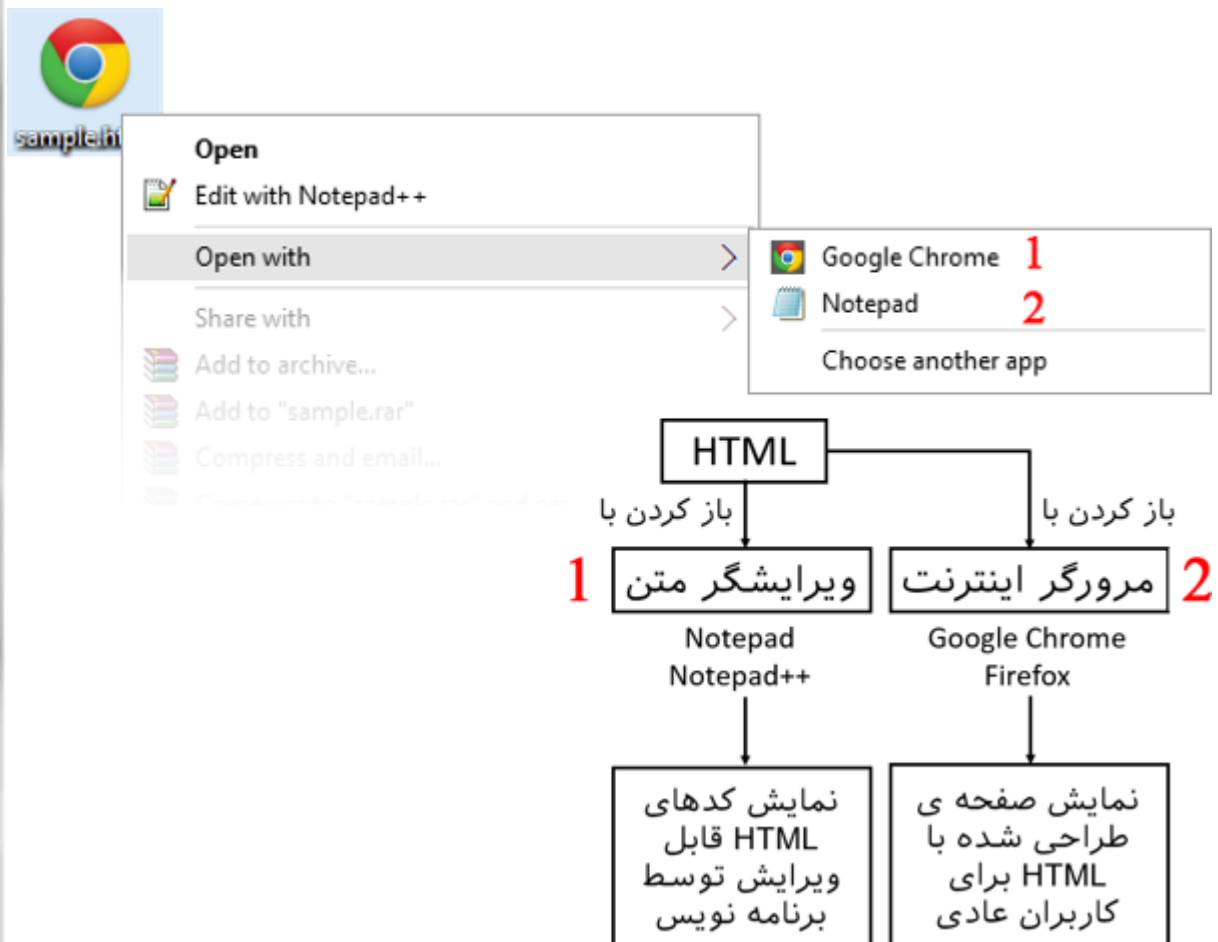
ضمیمه ۱: آموزش مقدماتی HTML

HTML که مخفف عبارت Hypertext Markup Language به معنی «زبان نشانه گذاری ابرمتنی» است زبانی است که صفحات وب را تشکیل می دهد. در یک فایل *.html شیوه ی نمایش عناصر به صورت مستتر قرار دارد و در واقع به مرورگر دستور می دهد که چگونه آن صفحه ی وب را نمایش دهد (به همین دلیل به آن زبان نشانه گذاری می گویند).

بهترین نرم افزار برای HTML نویسی در ویندوز نرم افزار Notepad++ است که می توانید آن را از سایت های دانلود دریافت نمایید. دقت شود که نباید در سیستم عامل ویندوز از برنامه هایی مثل Wordpad و یا Microsoft Word استفاده کرد. چون این برنامه ها با هدف انتشار کتاب ساخته شده اند و کدهایی به جهت پاراگراف بندی و غیره به فایل اضافه می کنند.

نکته: فایل های HTML با دو فرمت *.htm و *.html موجود هستند که در عمل تفاوتی ندارند و *.html فرمتی جدیدتر است.

ساختار یک فایل *.html:



نکته: اگر پسوند فایل ها برای شما قابل مشاهده نیست، یعنی مثلاً فایل sample.html به صورت sample تنها نمایش داده می شود باید امکان نمایش پسوند فایل را در سیستم عامل خود فعال کنید (از طریق برداشتن تیک جلوی عبارت Hide extensions for knowm file types پنجره ی Folder Options یا File Explorer Options در ویندوز ۱۰). دقت کنید که در صورت غیرفعال بودن این گزینه افزودن دستی پسوند کمکی نمی کند. یعنی اگر بخواهید فایل متنی حاوی کدهای خود را که فایلی با فرمت *.txt است (و به جای sample.txt، sample نشان داده شده است) به فایل *.html تغییر دهید نمی توانید با افزودن *.html به نام فایل آن را به فایل *.html تبدیل کنید. چون نام فایل به صورت sample.html.txt در می آید.

پیش نیاز کدنویسی با HTML: در ابتدا باید یادآور شد که HTML یک زبان برنامه نویسی نیست؛ بلکه یک زبان نشانه گذاری (Markup) است. یعنی کد HTML شامل متون، آدرس تصاویر و هر چیز دیگری است که مرورگر باید نمایش دهد. اساس کار HTML با تگ ها است.

در HTML دو نوع تگ به نام های مجرد (تکی) و مزدوج (دوتایی) وجود دارد. تگ های تکی به شکل `<x>` (به جای x کلمات متفاوتی قرار می گیرد) و تگ های مزدوج به شکل `<x>Content</x>` (به جای Content محتوای موردنظر قرار می گیرد) هستند. گاهی اوقات تگ تکی و تگ مزدوج اولی خصوصیتی را نیز دارا هستند که به صورت زیر در تگ قرار می گیرند:

```
<x attribute1="value1" attribute2="value2">
```

نکته: ترتیب خصوصیت ها مهم نیست. همچنین قبل از نوشتن هر attribute درج کاراکتر فاصله الزامی است و در جلوی آن attribute علامت مساوی و مقدار موردنظر در بین دو گیومه ی انگلیسی نوشته می شود (از آپوستروف نیز می توان استفاده کرد که چندان توصیه نمی شود).

نکته: اگر یک attribute نوشته نشود، مرورگر آن را برابر با مقدار پیش فرض attribute آن تگ در نظر می گیرد.

هر چیزی علی الخصوص متن که تمایل به اعمال شرایطی بر روی آن وجود دارد باید بین دو تگ مزدوج قرار بگیرد. تگ مزدوج شامل تگ آغازین و تگ پایانی است. عبارتی که بین دو تگ آغازین و پایانی قرار می گیرد مشمول شرایط تگ می شود.

مثال: کد HTML عبارت Iran is a country in Middle East که در آن عبارات Iran و Middle East به صورت توپر (Bold) نمایش داده می شوند به صورت زیر است (تگ مزدوج `<b>Content</b>` عبارت مابین خود را Bold می کند).

```
<b>Iran</b> is a country in <b>Middle East</b>
```



در جدول زیر تعدادی از تگ های ویراستاری را مشاهده می کنید:

<code>Content</code>	متن توپر (Bold)
<code><i>Content</i></code>	متن کج (Italic)
<code><u>Content</u></code>	متن زیر خط دار (Underline)
<code>Content</code>	متن خط خورده
<code><h1>Content</h1></code>	تیترا (عدد پس از h می تواند از ۱ تا ۶ باشد که بزرگترین تیترا h1 و کوچکترین تیترا h6 است)
<code><p>Content</p></code>	پاراگراف
<code><sub>Content</sub></code>	اندیس
<code><sup>Content</sup></code>	توان

بررسی خصوصیات تگ (attributeها) به طور دقیق تر: گاهی اوقات یک تگ نیازمند افزودن جزئیات می شود. برای مثال نوشتن کدی برای نمایش متنی آبی رنگ با فونت B Koodak. در این صورت خصوصیات موردنظر (در اینجا رنگ و فونت) به تگ آغازین اضافه می شوند.

نکته ی خیلی مهم: یکی از مهم ترین تفاوت های زبان فارسی با زبان های مطرح جهت نوشتن آن است که rtl (مخفف right to left به معنی چپ به راست) می باشد. در صورتی که attribute های پیش فرض تگ ها (که هیچ وقت هم نیازی به نوشتن آن ها نیست) جهت را به صورت ltr (راست به چپ) تعریف می کنند. این موضوع گاهی اوقات باعث به هم ریختگی متنی فارسی که در آن عبارات انگلیسی نیز هست می شود و جای متون فارسی دو طرف عبارت انگلیسی با هم عوض می شود. برای حل این مشکل باید در تگ هایی مثل پاراگراف attribute جهت را به صورت راست به چپ «بازتعریف» نمود. برای این کار از کد زیر استفاده می شود:

```
<p dir="rtl">Content</p>
```

مثال: این کد برای نمایش عبارت I love HTML با فونت tahoma و رنگ آبی و کلمه ی love منحصراً با رنگ قرمز است:

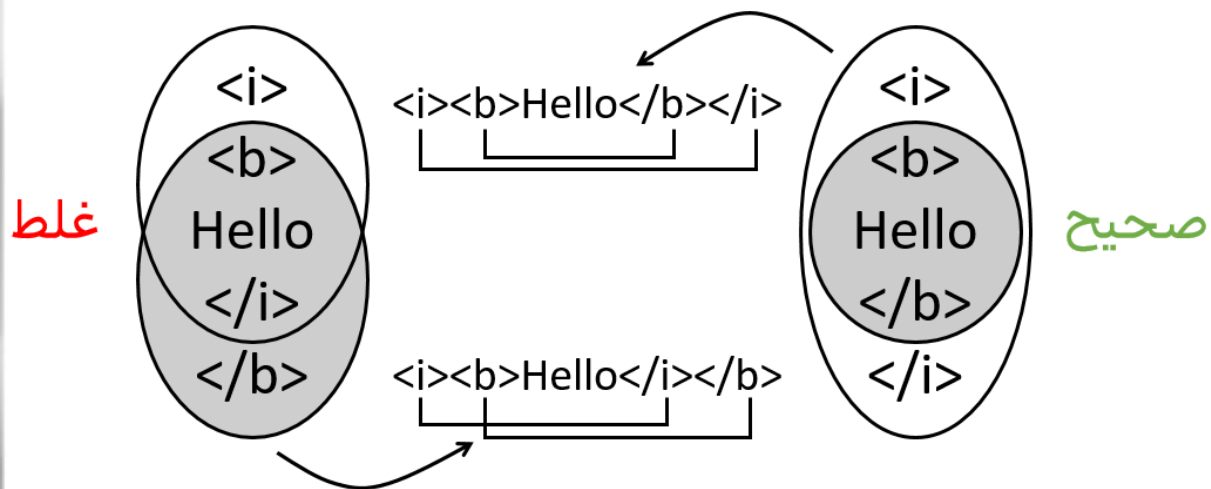
```
<font face="tahoma" color="blue">I <font color="red">love</font> HTML</font>
```



در اینجا کلمه ی love مشمول هر دو رنگ آبی و قرمز شده است ولی همواره اولویت با تگ داخلی تر است و بنابراین قرمز نمایش داده خواهد شد.

نکته: تگ به تنهایی کاربرد ندارد و از آن برای attribute هایش استفاده می کنند.

نکته ی مهم: تگ ها هیچ گاه به صورت زنجیر مانند در نمی آیند و همیشه داخل همدیگر هستند. یعنی در تگ های تو در تو تگی زودتر بسته می شود که دیرتر باز شده باشد.



اولین تجربه ی کدنویسی با HTML: برنامه ی Notepad++ یا Notepad را باز کنید (می توانید از طریق فشردن کلیدهای ویندوز+R پنجره ی Run را باز کرده و عبارت Notepad را در آن تایپ و اینتر را بزنید تا یک صفحه ی Notepad باز شود).

اولین تگی که همیشه باید نوشته شود تگ <html> است که بیان می کند کدهای HTML از این جا شروع شده اند. آخرین تگی که باید نوشته شود نیز </html> است که به پایان رسیدن کدهای HTML را نشان می دهد. سایر تگ ها و کدها را باید بین این دو تگ نوشت.

نکته: شکستن کد با کلید اینتر خلی در روند کار ایجاد نمی کند. یعنی یک کد را چه در یک خط و چه در چندین خط بنویسید هیچ فرقی نمی کند. بنابراین بهتر است برای مثال بعد از پایان یک تگ و قبل از شروع تگی دیگر برای جلوگیری از گیج شدن از کلید اینتر استفاده شود.

مثال: عملکرد کد

```
<h1>Hello world</h1>
```

با عملکرد کد



<h1>

Hello world

</h1>

یکسان است.

برای شکستن متن در زبان HTML از تگ مجرد
 استفاده می شود. تگ های مجرد برخلاف تگ های مزدوج که تغییراتی را در عبارات بین خود ایجاد می کنند باعث ایجاد یک تغییر تنها در محلی که خودشان قرار دارند می شوند و شروع و پایان ندارند. از دیگر تگ های مجرد می توان <hr> را نام برد که در محل خود یک خط افقی (جداکننده) ایجاد می کند. تگ <hr> چندین attribute از جمله color (برای تعیین رنگ آن) و width (برای تعیین عرض آن به پیکسل) دارد.

باید بدانید هر فایل *.html از دو بخش اصلی head (سر) و body (بدن) تشکیل شده است که هر کدام هم تگ مربوط به خود را دارند.

به طور کلی ساختار یک فایل *.html به صورت زیر است:

<html>

<head>

در بین تگ head کدهایی قرار می گیرند که در نمایش سایت به بازدیدکننده هیچ نقشی ندارند. مثل عنوان صفحه که به صورت استاتوس در مرورگر کاربر نمایش داده می شود. این بخش در صفحه ی مرورگر نمایش داده نخواهد شد.

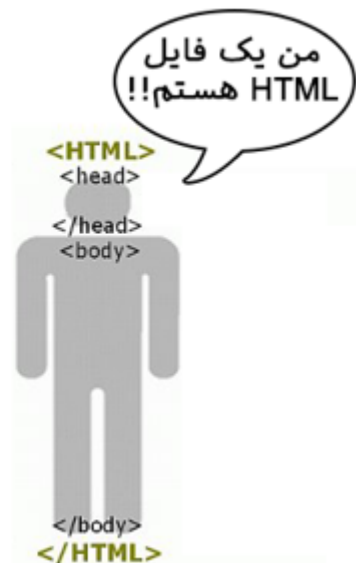
</head>

<body>

تمام کدهای اصلی و کدهایی که توسط مرورگر به صفحه ی وب تبدیل و به کاربر نمایش داده می شوند باید در این قسمت نوشته شود.

</body>

</html>



نکته: رعایت نوشتن تگ های «تشریفاتی» (html و head و body و...) در برنامه نویسی Basic4Android چندان اهمیتی ندارد و صرفاً برای خالی نبودن عریضه توضیح داده شده اند!

مثال: در این کد استاتوس صفحه (در لیست سربرج های مرورگر) برابر «خوش آمدید» و متن صفحه برابر عبارت صورتی رنگ با فونت tahoma «سلام دنیا» است (تگ استاتوس <title> است).



```
<html>
<head><title>خوش آمدید</title></head>
<body>
<font face="tahoma" color="pink">سلام دنیا</font>
</body>
</html>
```

همان طور که قبلاً گفته شد فایل متنی خود را برای تغییر فرمت باید rename کرده و نام آن را از sample.txt به sample.html تغییر دهید. حال از طریق Open with و انتخاب یک مرورگر می توانید نتیجه را مشاهده کنید.

نکته ی مهم: در صورتی که از عبارات فارسی در کد خود استفاده کرده اید بایستی انکود متن را از ASCII به UTF-8 تغییر دهید. زیرا در ASCII که انکود پیشفرض است تنها از ۱۲۸ کاراکتر پشتیبانی می شود و حروف فارسی جزء این کاراکترها نیستند. بنابراین متون فارسی به صورت نامفهوم نمایش داده می شوند. البته سیستم عامل ویندوز این قابلیت را دارد که به طور پیش فرض اگر در متنی در notepad کاراکتری نوشته شود که جزء ۱۲۸ کاراکتر ASCII نباشد قبل از ذخیره به کاربر هشدار دهد. برای تغییر انکود متن، باید به هنگام Save یا Save As... در پنجره ی ذخیره و در قسمت Encoding از منوی باز شونده گزینه ی UTF-8 را انتخاب کرد.

درج پاراگراف: برای درج پاراگراف از تگ مزدوج <p>Content</p> استفاده می شود. برای راست به چپ کردن پاراگراف از attribute زیر استفاده می شود:

```
<p dir="rtl">Content</p>
```

نکته: می توان در تگ <body>، خصوصیت dir="rtl" را قرار داد تا کل فایل rtl شود و نیازی به rtl کردن تک تک تگ ها نباشد.

درج تصویر: برای درج تصویر از تگ مجرد فوق استفاده می شود:

```

```



نکته: اگر تصویر موردنظر در اینترنت آپلود شده است آدرس آن را وارد کنید. اما در صورتی که عکس در پروژه ی Basic4Android وارد شده است تنها نوشتن نام فایل آن کفایت می کند.

نکته: با توجه به این که اندازه ی صفحه نمایش دستگاه های مختلف با هم متفاوت است می توانید طول تصویر (به دنبال آن عرض هم اتوماتیک تغییر می کند) را متناسب با اندازه ی صفحه تنظیم کنید. برای این کار از attribute عرض (width) به همراه درصد استفاده می شود. به مثال زیر دقت کنید:

```

```

در این مثال طول تصویر به اندازه ی هفتاد درصد طول صفحه نمایش در می آید.

در آخر باید گفت که HTML دستورات بسیار بیشتری دارد که در این جا تنها بخشی از آن ها که به ساخت اپلیکیشن با Basic4Android ارتباط داشتند توضیح داده شدند.

راه دیگر ساخت فایل *.html همانطور که در قسمت های قبلی هم گفته شد این است که در یک ویرایشگر وبلاگی مثل بلاگفا متن موردنظر را نوشته و با استفاده از امکانات بصری آن (رنگ و...) تغییرات لازم را ایجاد کرده و سپس کد آن را گرفته و در یک فایل notepad وارد کنید و...



ضمیمه ۲: معرفی و آموزش کار با کتابخانه ی AriaLib:

کتابخانه ی AriaLib کتابخانه ای ایرانی است که توسط یکی از مدیران وبسایت www.answercenter.ir ساخته شده است. کارهایی که با این کتابخانه می توان انجام داد به شرح زیر است:

- اشتراک متن
- اشتراک متن در یک برنامه خاص (مثل لاین و وایبر و ...)
- اشتراک عکس
- اشتراک عکس در یک برنامه خاص (مثل لاین و وایبر و ...)
- اشتراک عکس از دیتابیس
- اشتراک عکس از دیتابیس در یک برنامه خاص (مثل لاین و وایبر و ...)
- اشتراک هر نوع فایل (مثل: jpg, png, mp4, apk و فایل های ویدیویی، برنامه ها و هر نوع فایل دیگر)
- اشتراک هر نوع فایل در یک برنامه خاص (مثل: jpg, png, mp4, apk و فایل های ویدیویی، برنامه ها و هر نوع فایل دیگر)
- کپی متن در کلیپبرد
- گرفتن متن از کلیپبرد
- چک کردن وجود متن در کلیپبرد
- چک کردن نصب بودن یک برنامه در گوشی
- چک کردن اینترنت (قطع و وصل)
- باز کردن پروفایل اینستاگرام با ID
- باز کردن پروفایل لاین با ID
- امکان به اشتراک گذاری برنامه و یا هر برنامه ی نصب شده بر روی گوشی
- اضافه کردن فایل های تصویری و صوتی بلافاصله به گالری و موزیک پلیر



: AriaMarkets

- باز کردن صفحه برنامه در کافه بازار، ایران اپس، کندو، مایکت، پارس هاب
- باز کردن صفحه نظر دهی به برنامه در کافه بازار، ایران اپس، کندو، مایکت، پارس هاب
- باز کردن صفحه برنامه های توسعه دهنده در کافه بازار، ایران اپس، کندو، مایکت، پارس هاب
- و امکانات اضافی مارکت ها مثل :
- باز کردن صفحه برنامه و دانلود خودکار (مایکت)
- تعیین امتیاز در ایران اپس از ۱ تا ۵ و بعد باز شدن صفحه نظر
- جستجو (ایران اپس)
- لیست دسته بندی (ایران اپس)
- ارتباط با توسعه دهنده (پارس هاب)

دریافت آخرین نسخه ی این کتابخانه از این تاپیک: [لینک](#)

آموزش کار با این کتابخانه (کار با سایر کتابخانه نیز روندی مشابه دارد):
 برحسب نیاز خود متغیری از نوع AriaLib، AriaMarket یا AriaPhotoshp (برای
 ویرایش تصاویر) تعریف کنید. سپس در قسمت دلخواه و در یک سطر
 جدید نام متغیر را نوشته و سپس کاراکتر نقطه را تایپ کنید. پس از این کار
 لیستی باز می شود که امکان استفاده از امکانات کتابخانه را فراهم می کند.

همچنین آموزش این کتابخانه در این تاپیک: [لینک](#)

پایان.



ادامه ی راه؟؟

مطالبی که در این کتاب گفته شد تنها در حکم استارت و دست گرمی بودند برای کسانی که نمی دانستند برنامه نویسی اندروید را چگونه باید شروع کرد.

مطالبی مثل دیتابیس های آفلاین و آنلاین برای کتاب ها، دانلود و نمایش فیلم و صدا، استفاده ی همزمان از زبان php برای بخش های سمت سرور برنامه های آنلاین، JSON، ساخت کتابخانه، طراحی متریال، پرداخت درون برنامه ای مارکت ها، محافظت از دیتابیس های اپلیکیشن در برابر سرقت و بسیاری از مطالب مهم دیگر در این کتاب بررسی نشده اند.

پس از این که به مهارت های سطح این کتاب رسیدید، می توانید از مطالب پیشرفته تر موجود در سایت های www.learn-net.ir و www.answercenter.ir استفاده کنید.

بهترین مرجع (که خود من هم از آن استفاده می کنم) بخش جستجوی این دو انجمن است.

همچنین در صورتی که پرسشی داشتید و پاسخ خود را نیافتید آن را در قسمت پرسش و پاسخ انجمن های فوق مطرح نمایید تا من و دوستانم در اسرع وقت پاسخگوی شما باشیم.

در پناه حق

خدانگهدار (:

