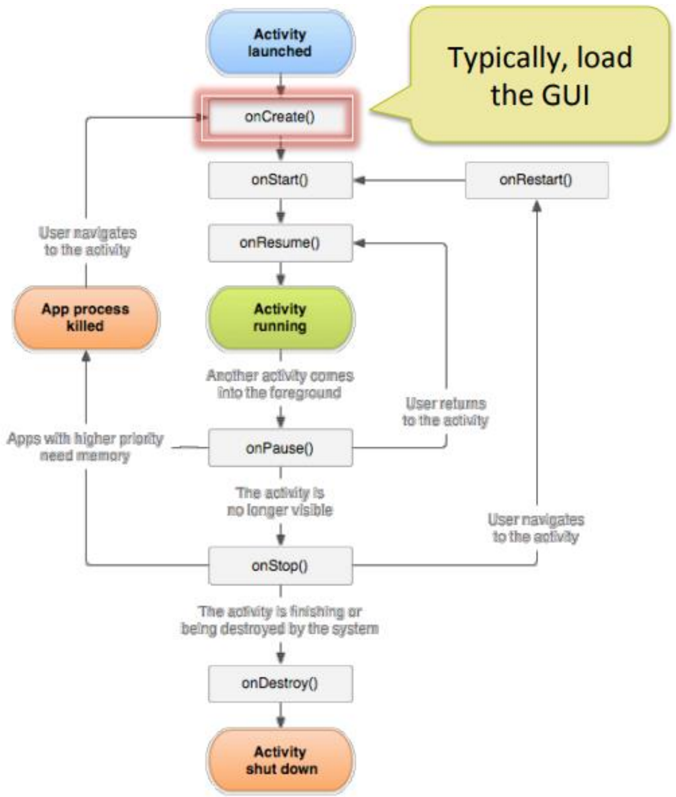
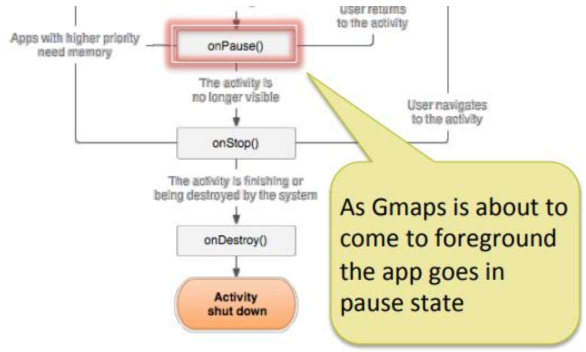
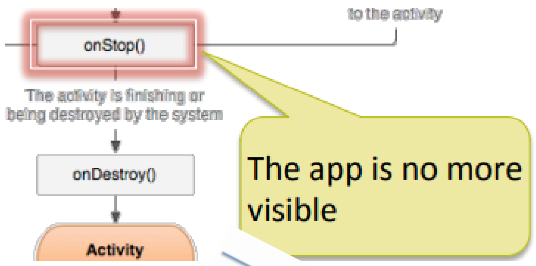

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

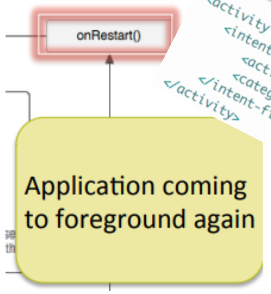
Typically, load the GUI


Apps with higher priority need memory
onPause()
user returns to the activity
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

As Gmaps is about to come to foreground the app goes in pause state



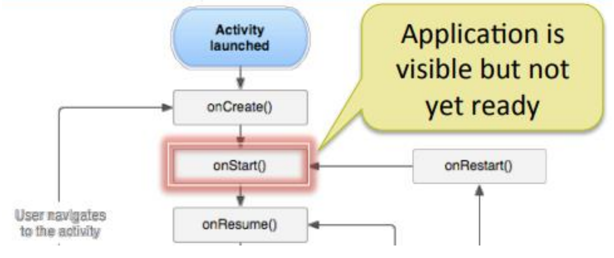
The app is no more visible



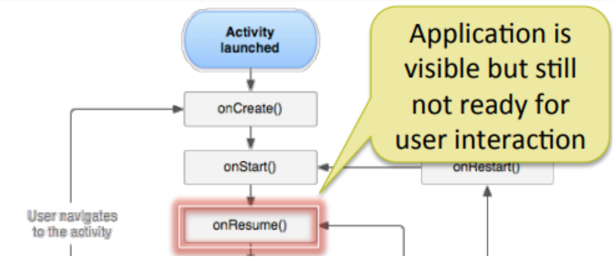
Application coming to foreground again

Declaring the activitie

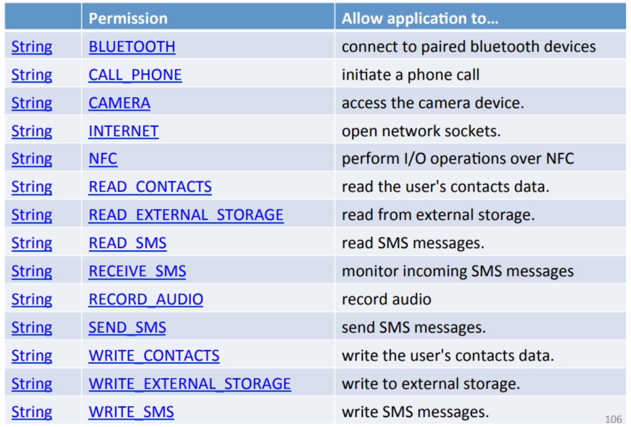
```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```



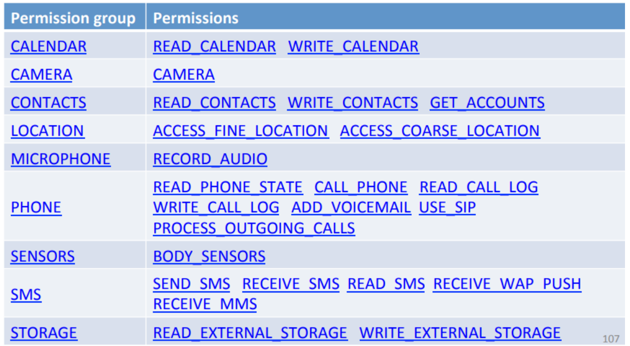
Application is visible but not yet ready



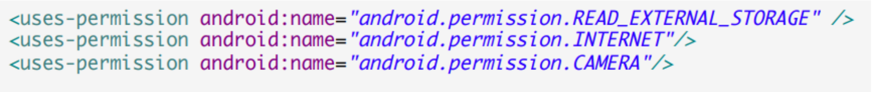
Application is visible but still not ready for user interaction



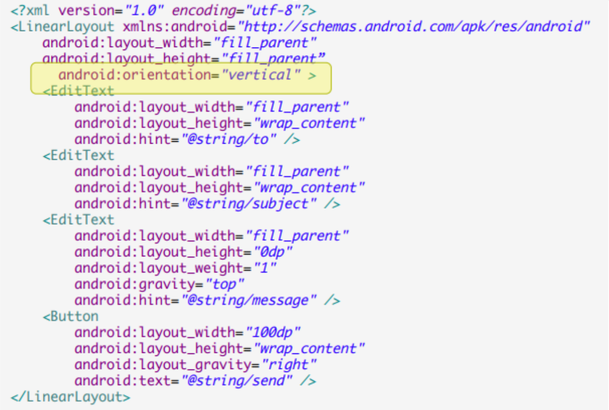
	Permission	Allow application to...
String	BLUETOOTH	connect to paired bluetooth devices
String	CALL_PHONE	initiate a phone call
String	CAMERA	access the camera device.
String	INTERNET	open network sockets.
String	NFC	perform I/O operations over NFC
String	READ_CONTACTS	read the user's contacts data.
String	READ_EXTERNAL_STORAGE	read from external storage.
String	READ_SMS	read SMS messages.
String	RECEIVE_SMS	monitor incoming SMS messages
String	RECORD_AUDIO	record audio
String	SEND_SMS	send SMS messages.
String	WRITE_CONTACTS	write the user's contacts data.
String	WRITE_EXTERNAL_STORAGE	write to external storage.
String	WRITE_SMS	write SMS messages.



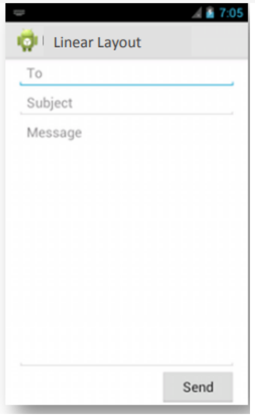
Permission group	Permissions
CALENDAR	READ_CALENDAR WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE CALL_PHONE READ_CALL_LOG WRITE_CALL_LOG ADD_VOICEMAIL USE_SIP PROCESS_OUTGOING_CALLS
SENSORS	BODY_SENSORS
SMS	SEND_SMS RECEIVE_SMS READ_SMS RECEIVE_WAP_PUSH RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE



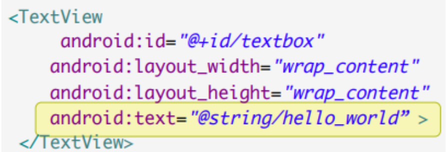
```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CAMERA"/>
```



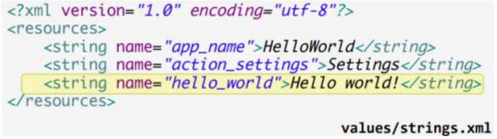
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



Pour l'internationalisation , il faut utiliser le translation, càd utiliser values/string,xml pour définir les strings de chaque langage supporté par notre application, Les strings sont stockés dans un fichier string,xml



```
<TextView
    android:id="@+id/textbox"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/hello_world” >
 </TextView>
```

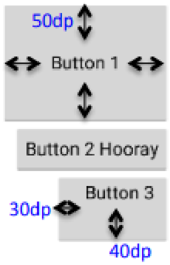


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
</resources>
```

values/strings.xml

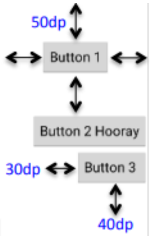
Padding

```
<LinearLayout ...
    android:orientation=”vertical” >

  <Button ... android:text="Button 1"
              android:padding=”50dp" />

  <Button ... android:text="Button 2 Hooray” /

  <Button ... android:text="Button 3 "
               android:paddingLeft="30dp"
               android:paddingBottom="40dp" />
```



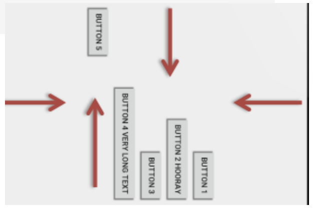
Marging

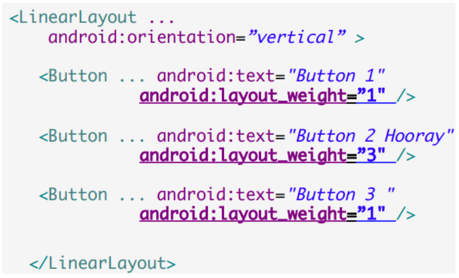
```
<LinearLayout ...
    android:orientation="vertical" >

  <Button ... android:text="Button 1"
              android:layout_margin="50dp" />

  <Button ... android:text="Button 2 Hooray" />

  <Button ... android:text="Button 3 "
               android:layout_marginLeft="30dp"
               android:layout_marginBottom="40dp" /
```



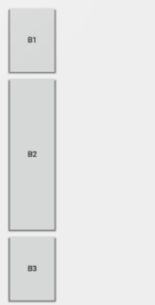
Linear Layout

```
<LinearLayout ...
    android:orientation=”vertical"
    android:gravity=”center|right" >

   <Button ... android:text="Button 1" />

   <Button ... android:text="Button 2 Hooray" />

   <Button ... android:text="Button 3 " />


   <Button ... android:text="Button 5"
               android:layout_gravity=”left" />

</LinearLayout>
```



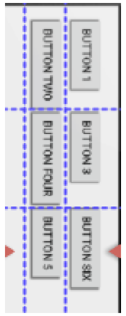


```
<LinearLayout ...
    android:orientation=”vertical” >

  <Button ... android:text="Button 1"
              android:layout_weight=”1" />

  <Button ... android:text="Button 2 Hooray"
              android:layout_weight=”3" />

  <Button ... android:text="Button 3 "
              android:layout_weight=”1" />


 </LinearLayout>
```



Grid Layout

```
<GridLayout ...
    android:columnCount="3"
    android:rowCount="2"
    android:orientation=”vertical" >

    <Button ... android:text="Button 1" />
    <Button ... android:text="Button two" />
    <Button ... android:text="Button 3" />
    <Button ... android:text="Button four" />

    <Button ... android:text="Button 5"
                android:layout_row=”1”
                android:layout_column=”2"/>

    <Button ... android:text="Button six"
                android:layout_row=”0”
                android:layout_column=”2"/>

</GridLayout>
```



Nested Layouts

Combinaison de plusieurs linear layout dans un linear layout pour une interface plus compliquée

Relative Layouts

- Properties for x/y relative to another widget:

`layout_[below | above | toLeftOf | toRightOf]`

Positions this view [below, above…] the given view ID.

`layout_align[Baseline | Bottom| Left | Right | Top]`

Positions this view so that it is aligned the given view ID.

- Properties for x/y relative to layout container (the activity):

`layout_alignParent[Top | Bottom | Left | Right]`

Set these flags to a boolean value of "true" to enable them

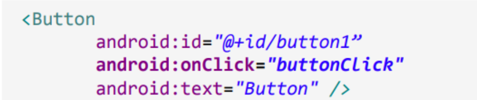
`layout_center[Horizontal | Vertical | InParent]`

Set these flags to "true" to center the control within its parent in a dimension

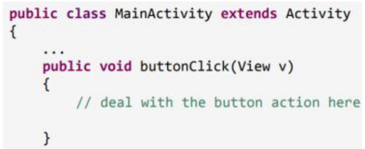
```
<RelativeLayout ...>

    <Button ... android:id="@+id/button1” android:text="B1"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"/>

    <Button ... android:id="@+id/button2” android:text="B2"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/button1” />

    <Button... android:id="@+id/button3” android:text="B3"
        android:layout_alignLeft="@+id/button1"
        android:layout_below="@+id/button2” />

    <Button... android:id="@+id/button4” android:text="B4"
        android:layout_alignBaseline="@+id/button3"
        android:layout_alignBottom="@+id/button3"
        android:layout_alignParentRight="true”/>

    <TextView... android:id="@+id/textView1 »
          android:layout_centerInParent="true"
          android:text="I’m a TextView" />

    <Button... android:id="@+id/button5” android:text="B5"
        android:layout_alignLeft="@+id/button3"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="48dp”/>

</RelativeLayout>
```

Button

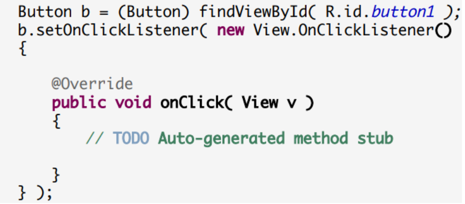
android:clickable="***bool***”	set to false to disable the button
android:id="@+id/***theID***”	unique ID for use in Java code
android:onClick="***function***”	function to call in activity when clicked (must be public, void, and take a View arg)
android:text="***text***”	text to put in the button (USE STRINGS!)



```
<Button
        android:id="@+id/button1”
        android:onClick="buttonClick"
        android:text="Button" />
```



```
public class MainActivity extends Activity
{
    ...
    public void buttonClick(View v)
    {
        // deal with the button action here

    }
```



```
Button b = (Button) findViewById( R.id.button1 );
b.setOnClickListener( new View.OnClickListener()
{

    @Override
    public void onClick( View v )
    {
        // TODO Auto-generated method stub

    }
} );
```

ImageButton

android:clickable="***bool***”	set to false to disable the button
android:id="@+id/***theID***”	unique ID for use in Java code
android:onClick="***function***”	function to call in activity when clicked (must be public, void, and take a View arg)
android:src=”***@drawable/img***”	Image for the button, it must correspond to an image resource

put image file in project folder **app/src/main/res/drawable**

use @drawable/foo to refer to foo.png

EditText

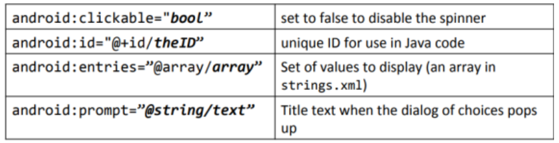
android:hint=“text”	Grey text to show before user input
android:inputType=”***type***”	The type of input to be typed (number, mail…)
android:id="@+id/***theID***”	unique ID for use in Java code
android:lines=”***int***”	Number of visible lines
android:maxLines=”***int***”	Max number of lines that the user can enter

RadioButton (Boutton à coché)

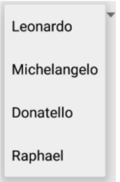
android:clickable="***bool***”	set to false to disable the button
android:checked="***bool***”	set to true to have it checked at the beginning
android:id="@+id/***theID***”	unique ID for use in Java code
android:onClick="***function***”	function to call in activity when clicked (must be public, void, and take a View arg)
android:text=”***text***”	Text to place close to the button

```
<LinearLayout ...
        android:orientation="vertical"
        android:gravity="center|top">
    <RadioGroup ...
            android:orientation="horizontal">
        <RadioButton ... android:id="@+id/lions"
                         android:text="Lions"
                         android:onClick="radioClick" />
        <RadioButton ... android:id="@+id/tigers"
                         android:text="Tigers"
                         android:checked="true"
                         android:onClick="radioClick" />
        <RadioButton ... android:id="@+id/bears"
                         android:text="Bears, oh my!"
                         android:onClick="radioClick" />
    </RadioGroup>
</LinearLayout>
public void radioClick(View view) {
    // check which radio button was clicked
    if (view.getId() == R.id.lions) {
        // ...
    } else if (view.getId() == R.id.tigers) {
        // ...
    } else {
        // bears ...
    }
}
```

Spinner



android:clickable="*bool*"	set to false to disable the spinner
android:id="@+id/*theID*"	unique ID for use in Java code
android:entries="@array/*array*"	Set of values to display (an array in strings.xml)
android:prompt="*@string/text*"	Title text when the dialog of choices pops up

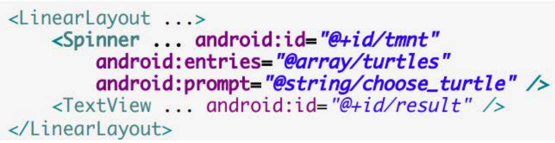


Leonardo

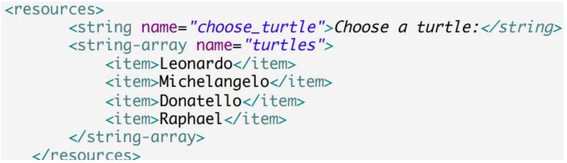
Michelangelo

Donatello

Raphael



```
<LinearLayout ...>
    <Spinner ... android:id="@+id/tmnt"
        android:entries="@array/turtles"
        android:prompt="@string/choose_turtle" />
    <TextView ... android:id="@+id/result" />
</LinearLayout>
```

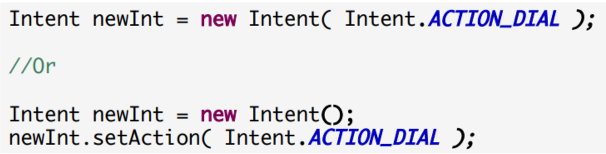


```
<resources>
      <string name="choose_turtle">Choose a turtle:</string>
      <string-array name="turtles">
          <item>Leonardo</item>
          <item>Michelangelo</item>
          <item>Donatello</item>
          <item>Raphael</item>
      </string-array>
   </resources>
```

Intents

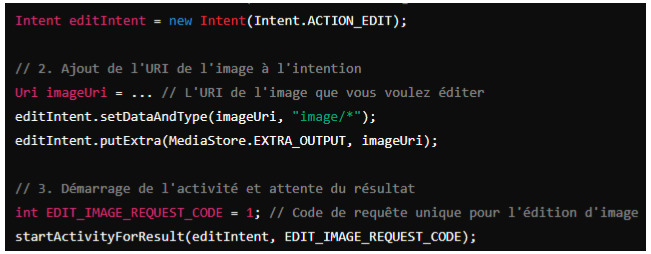
Explicit Intent when calling a specific Activity

Implicit Intent when requesting a general action to perform



```
Intent newInt = new Intent( Intent.ACTION_DIAL );

//Or

Intent newInt = new Intent();
newInt.setAction( Intent.ACTION_DIAL );
```

Exemple pour éditer une image



```
Intent editIntent = new Intent(Intent.ACTION_EDIT);

// 2. Ajout de l'URI de l'image à l'intention
Uri imageUri = ... // L'URI de l'image que vous voulez éditer
editIntent.setDataAndType(imageUri, "image/*");
editIntent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);

// 3. Démarrage de l'activité et attente du résultat
int EDIT_IMAGE_REQUEST_CODE = 1; // Code de requête unique pour l'édition d'image
startActivityForResult(editIntent, EDIT_IMAGE_REQUEST_CODE);
```

Fichier Manifest et intents

Pour qu'une application puisse répondre à des intents ACTION_SEARCH et ACTION_WEB_SEARCH, il faut configurer les filtres d'intention approprié dans le fichier AndroidManifest,xml en déclarant les activités,

Intents et camera

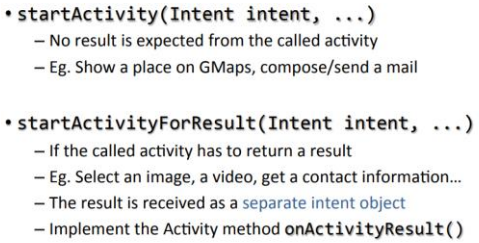
Pour réaliser l'action take picture, on peut utiliser une application externes pour prendre des photos en utilsiant les intents, ou créer une application customiser de camera,

From gallery, on utilise l'intent avec ACTION_PICK

Difference entre uses_feature et permissions

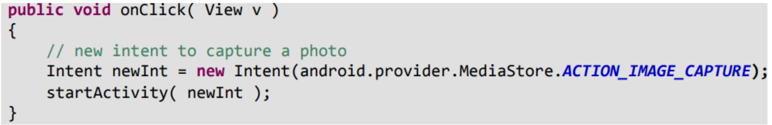
Uses-feature déclare les fonctionnalités matérielles ou logicielles requise pour le fonctionnement de l'application (l'app ne fonctionne pas sans,)

La permission demande l'autorisation d'accéder à des ressources pour faire fonctionner une certaine fonctionnalité (l'app peut fonctionner sans accorder la permission d'accéder à quelque chose),



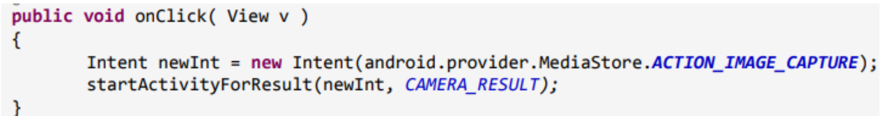
- `startActivity(Intent intent, ...)`
  - No result is expected from the called activity
  - Eg. Show a place on GMaps, compose/send a mail
- `startActivityForResult(Intent intent, ...)`
  - If the called activity has to return a result
  - Eg. Select an image, a video, get a contact information…
  - The result is received as a separate intent object
  - Implement the Activity method `onActivityResult()`

Activer la camera sans retourner l'image

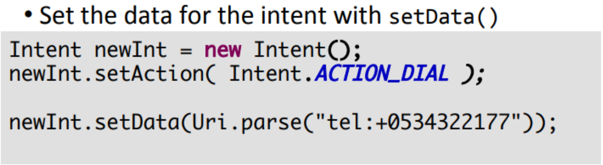


```
public void onClick( View v )
{
    // new intent to capture a photo
    Intent newInt = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
    startActivity( newInt );
}
```

Activer la camera avec retour de l'image

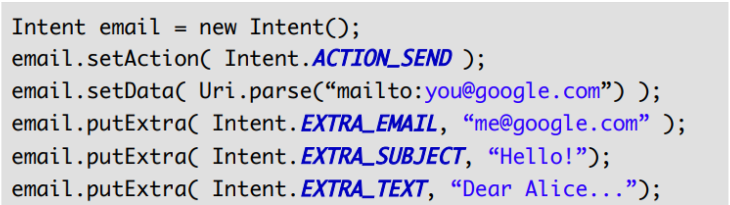


```
public void onClick( View v )
{

        Intent newInt = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(newInt, CAMERA_RESULT);
}
```

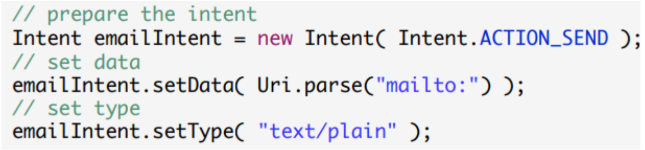


- Set the data for the intent with `setData()`

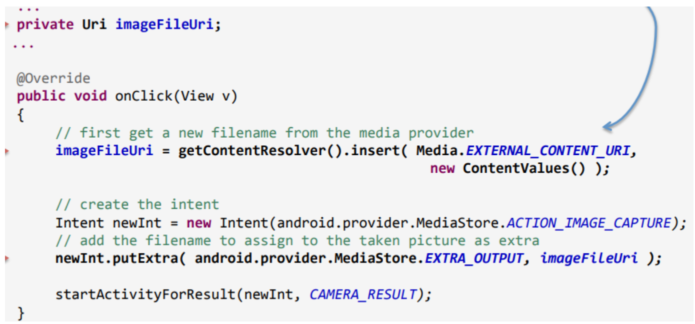
```
Intent newInt = new Intent();
newInt.setAction( Intent.ACTION_DIAL );

newInt.setData(Uri.parse("tel:+0534322177"));
```



```
Intent email = new Intent();
email.setAction( Intent.ACTION_SEND );
email.setData( Uri.parse("mailto:you@google.com") );
email.putExtra( Intent.EXTRA_EMAIL, "me@google.com" );
email.putExtra( Intent.EXTRA_SUBJECT, "Hello!");
email.putExtra( Intent.EXTRA_TEXT, "Dear Alice...");
```



```
// prepare the intent
Intent emailIntent = new Intent( Intent.ACTION_SEND );
// set data
emailIntent.setData( Uri.parse("mailto:") );
// set type
emailIntent.setType( "text/plain" );
```



```
...
private Uri imageFileUri;
...

@Override
public void onClick(View v)
{
    // first get a new filename from the media provider
    imageFileUri = getContentResolver().insert( Media.EXTERNAL_CONTENT_URI,
                                                 new ContentValues() );

    // create the intent
    Intent newInt = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
    // add the filename to assign to the taken picture as extra
    newInt.putExtra( android.provider.MediaStore.EXTRA_OUTPUT, imageFileUri );

    startActivityForResult(newInt, CAMERA_RESULT);
}
```