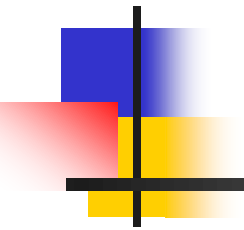
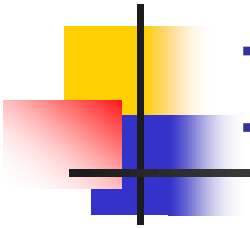
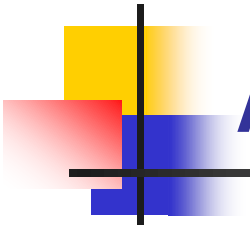


Développement mobile



IOS ??

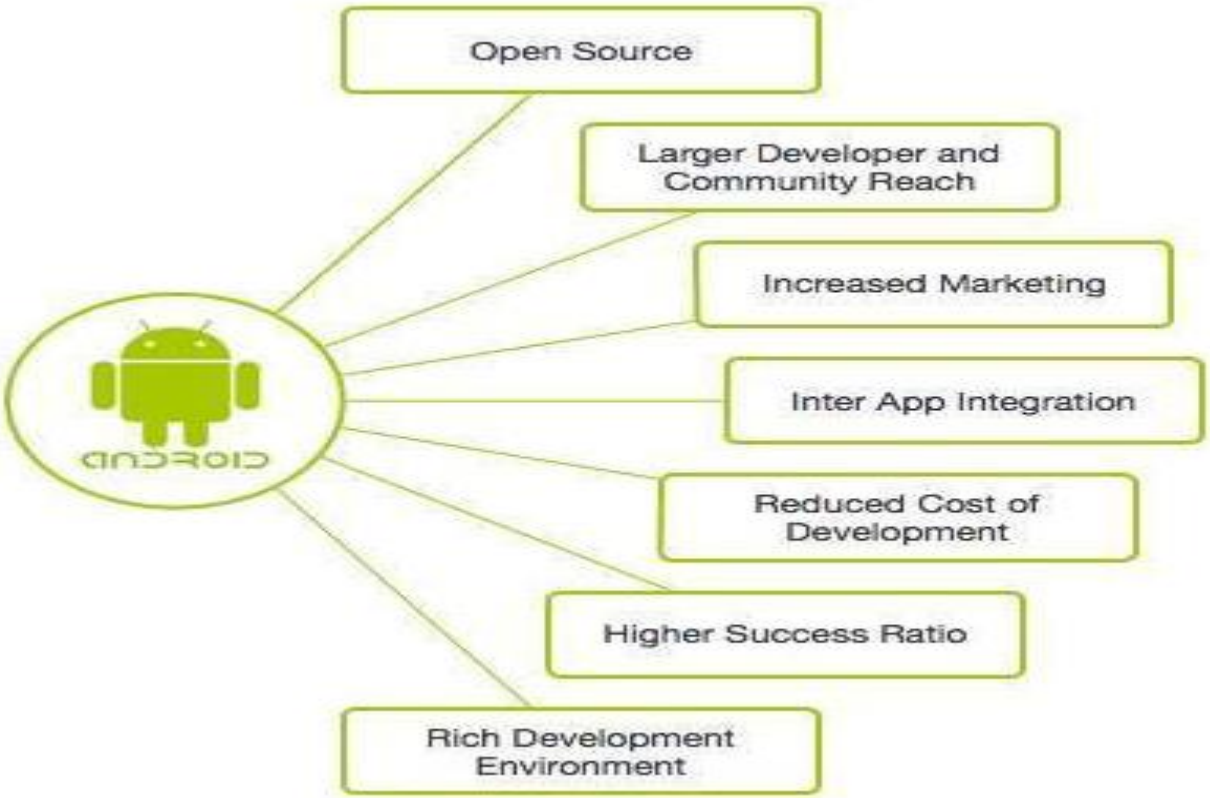
- Applications pour iphone, ipad ou ipod touch
- Un IDE apple dédié : Xcode
- Langage dédié à la programmation des produits apple : Objective-C
- Necessité :
  - Disposer d'un ordinteur MAC
  - Disposer d'un téléphone Apple

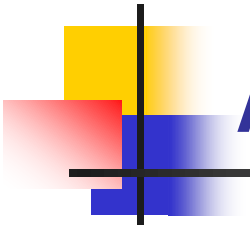


Android ??

- Un système d'exploitation open source basé sur du Linux
- Différents boutiques en ligne : Google Play, SlideME, Opera Mobile Store, Mobango, F-droid, Amazone Store
- Chaque jours plus d'un million d'utilisateurs android
- Différents types d'applications : Musique, News, Voyages, Affaires, …

Android ??

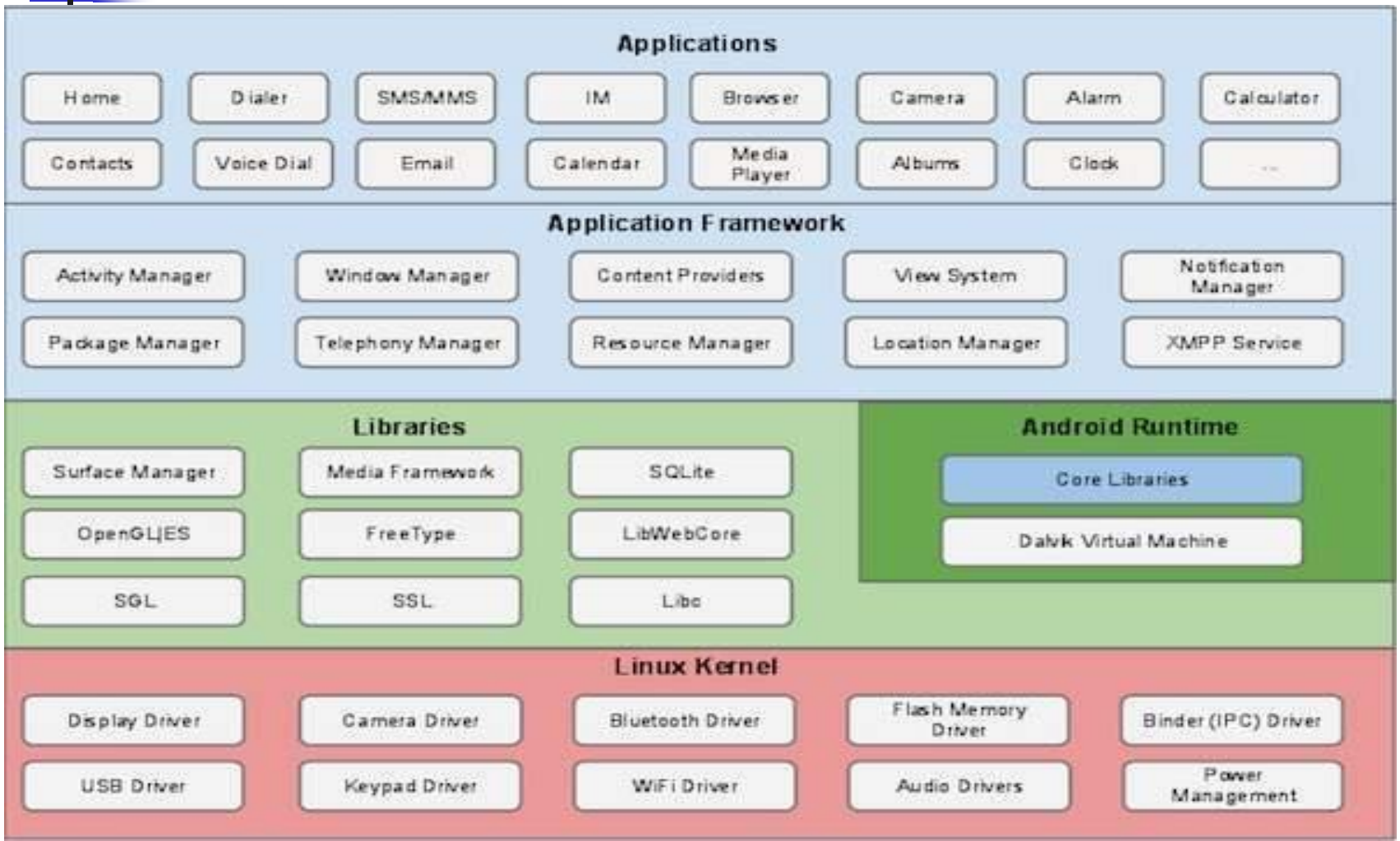

Open Source
Larger Developer and Community Reach
Increased Marketing
Inter App Integration
Reduced Cost of Development
Higher Success Ratio
Rich Development Environment
android

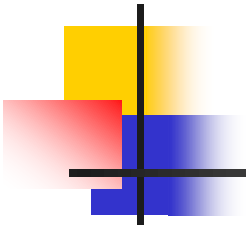


Android ??

- Android Software developement Kit (SDK) a été développé en 2007
- SDK est basé sur du Java

Architecture d'android



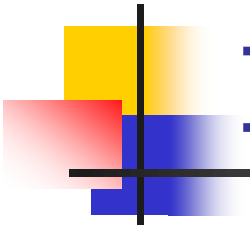


Différents composants

- *Activité* : gère l'interface utilisateur
- *Services* : gère les process d'arrière plans
- *Broadcast receivers* : Communication entre OS et APP
- *Content Providers* : gère les données
- *Layouts* : vue affichée à l'utilisateur
- *Intent* : transfert de messages
- *Resources* : images, les chaines de caracètres
- *Manifest file* : fichier de configuration

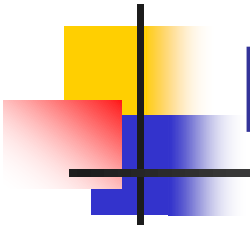
Différents composants

- *Gradle* : gère et construit les projets android
- Fichier buidl.gradle :
  - **compileSdkVersion** le numéro de version d'android sdk utilisée pour compiler le projet
  - **buildToolsVersion** la nom complet de la version d'android sdk utilisée pour compiler le projet
  - **applicationId** l'identifiant unique de l'application
  - **minSdkVersion** la version minimum d'android supportée
  - **targetSdkVersion** la version d'android pour laquelle l'application a été compilée
  - **versionCode** le numéro de version
  - **versionName** le nom complet de la version



Installation et Configuration

- Installer JDK et JRE
- Installer Android Studio
- Créer une machine virtuelle si …
- Créer un nouveau projet
- Lancer

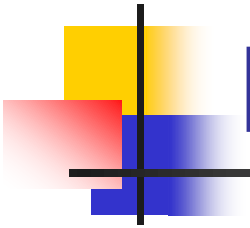


Manifest file

- Les activités, les services, les contents sont tous déclarés ici
- Broadcast receiver peut être déclarer dynamiquement
- Des permissions pour par exemple acceder au service réseau

Manifest file

```
<application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <uses-feature android:name="android.hardware.camera" />
        <activity
            android:name="NameActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action  android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="MyPreferenceActivity" >
        </activity>
</application>
```

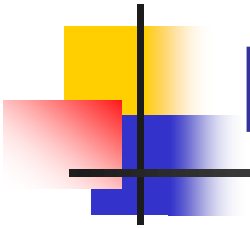


Ressources

/res/drawables	Images .png .jpeg
/res/values	Les strings, les colors, les styles
/res/layout	Les fichiers xml pour UI
/res/menu	Les menus

Ressources

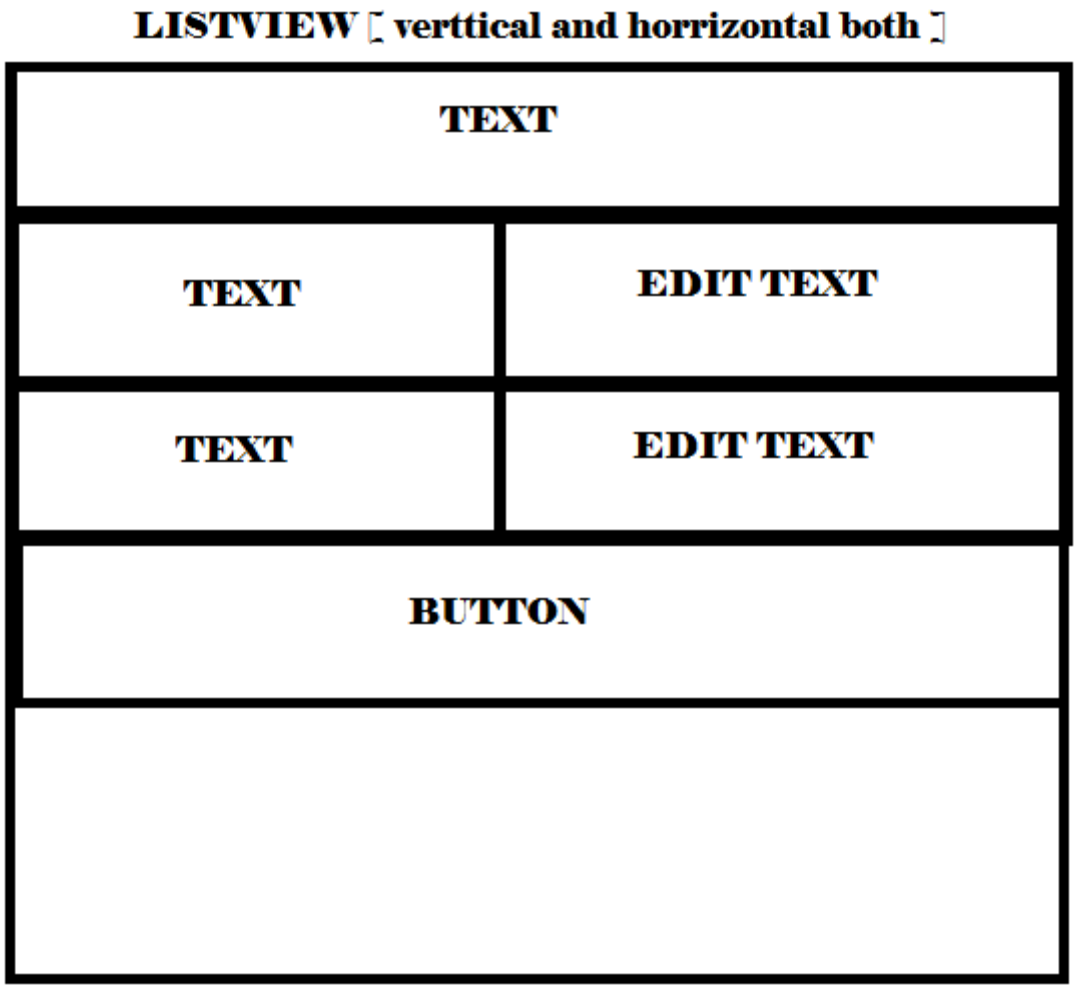
```
<resources>
    <string name="app_name">Nom_App</string>
    <string name="action_settings">Settings</string>
<string-array name="operationsystems">
        <item>Ubuntu</item>
        <item>Android</item>
        <item>Microsoft Windows</item>
    </string-array>
    <color name="red">#ffff0000</color>
</resources>
```

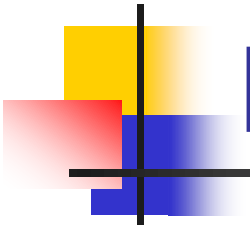


Les layouts

- FrameLayout : affiche un élément dans une interface
- LinearLayout : organisation en ligne ou en colonne
- RelativeLayout : positionner un objet par rapport à un autre (plus complexe)
- GridLayout : positionnement en grille
- Scrollviews : glissement de l'interface

Linearlayout





Les views

- Button
- TextView
- ImageView
- CheckBox
- EditText
- DatePicker
- RadioButton
- Toast
- ImageButton

Les views

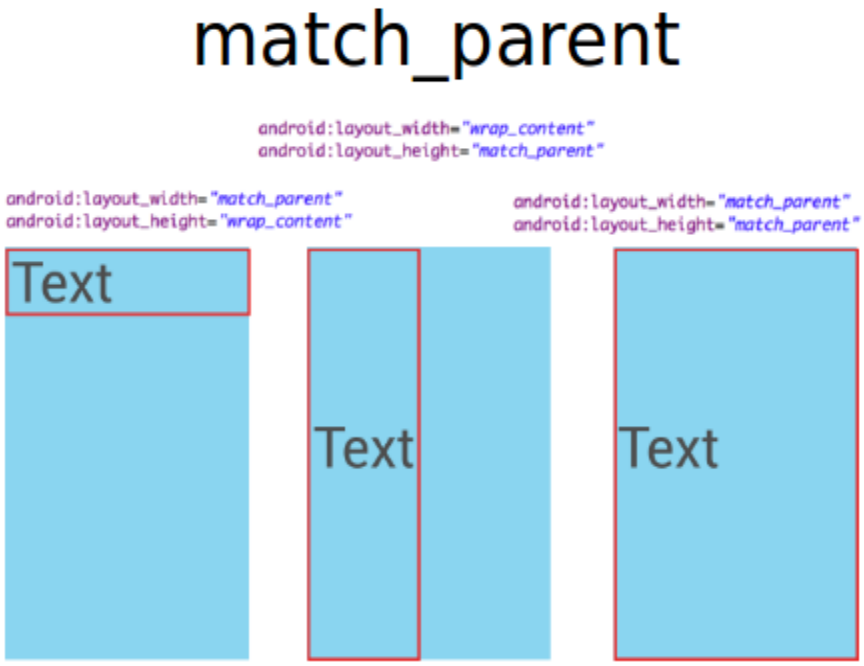
- Attributs :
  - android:layout_width
  - android:layout_height

wrap_content

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ...
    />
```

Les views

- Attributs :
  - android:layout_width
  - android:layout_height



Les views

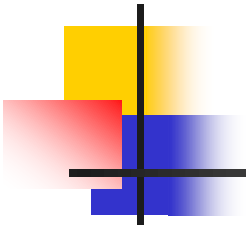
- Pour Image View :
  - android:layout_height : Définir la hauteur de l'élément.
  - android:id : Identifiant de l'élément (utilisé pour l'interaction avec l'élément).
  - android:layout_width : Largeur de l'élément.
  - android:src : Image source utilisée pour l'ImageView.
  - android:layout_marginTop : Marge externe du haut. Les différentes tailles utilisées sont déclarées dans le fichier dimens.xml (dossier values) et utilisées en suivant la syntaxe @dimen/nom_de_la_variable

Les views

- Pour Button:
  - android:layout_gravity : Position de l'élément (center, left, right…).
  - android:text : Texte à afficher dans l'élément. Toute les chaines de caractères utilisées sont déclarées dans le ficher strings.xml et utilisées à l'aide de la syntaxe @string/nom_de_la_chaine

Les views

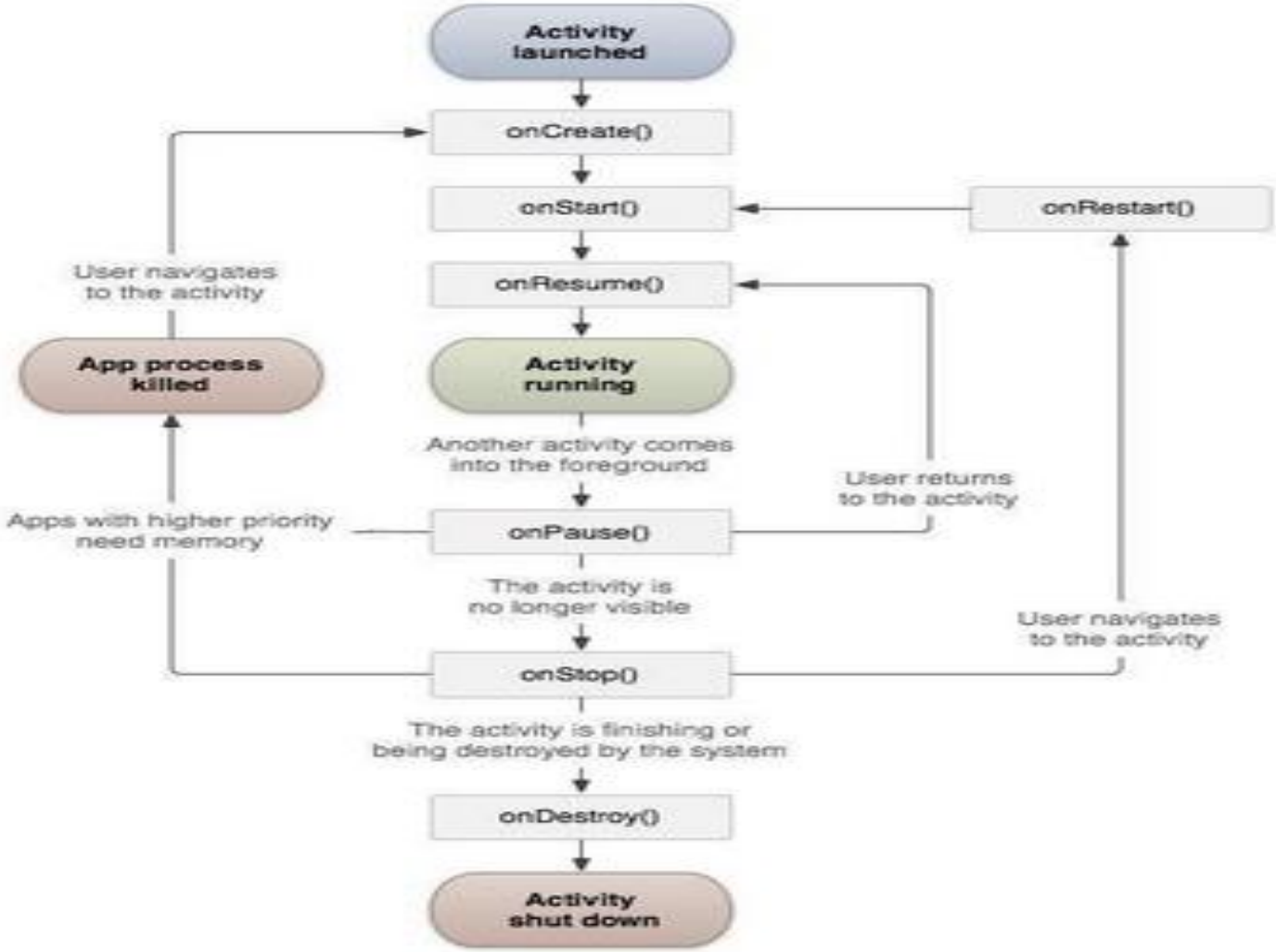
- Pour TextView:
  - android:textColor : Couleur du texte. Toute les couleurs utilisées sont déclarées dans le fichier colors.xml et utilisées à l'aide de la syntaxe @color/nom_de_la_couleur.
  - android:paddingTop : Marge interne du haut.
  - android:textSize : Définie la taille du text.

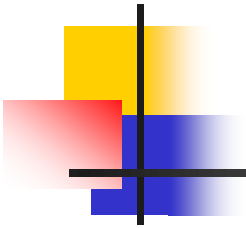


Les activités

- Une activité est la composante principale pour une application Android.
- Elle représente l'implémentation et les interactions des interfaces.
- Elle est associé à une vue au minimum

Cycle de vie activité





Les services

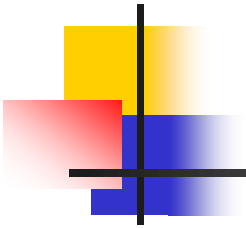
- Un service, à la différence d'une activité, ne possède pas de vue mais permet l'exécution d'un algorithme
- Il ne s'arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée.
- Il s'execute en background sans besoin d'interagir avec l'utilisateur

Les services

- Il peut être lancé à différents moments :
  - Au démarrage du téléphone.
  - Au moment d'un événement (arrivée d'un appel, SMS, mail, etc…).
  - Lancement de votre application.
  - Action particulière dans votre application.

Les services

- Il peut être lancé à différents moments :
  - Au démarrage du téléphone.
  - Au moment d'un événement (arrivée d'un appel, SMS, mail, etc…).
  - Lancement de votre application.
  - Action particulière dans votre application.

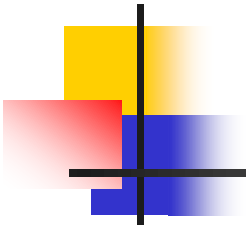


Les intents

- Les Intents permettent de communiquer entre les différentes activités de l'application, mais aussi du téléphone.
- Ainsi une activité peut en lancer une autre soit en passant un intent vide, soit en y passant des paramètres
- Les Intent Filters jouent le rôle de filtre.

Les broadcast receiver

- Un Broadcast Receiver permet d'écouter ce qui se passe sur le système ou sur votre application
- Il déclenche une action que vous aurez prédéfinie
- Avec ce mécanisme on peut lancé les services

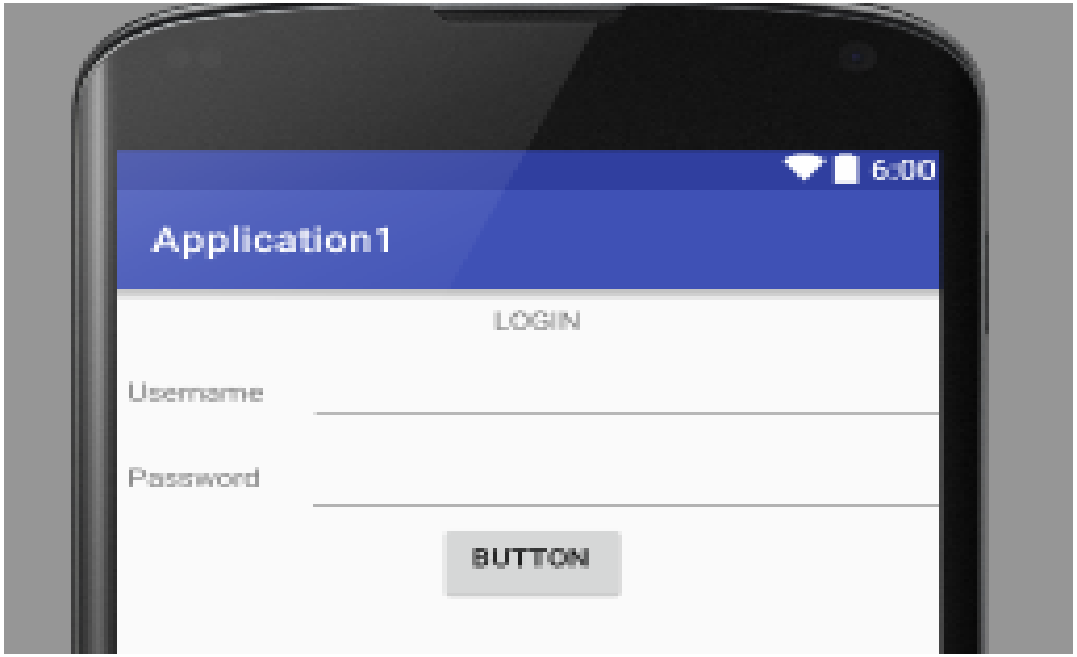


Les content providers

- Les content providers permettent d'accéder à un ensemble de données depuis une application
- Vous pouvez ainsi accéder aux contacts, à l'agenda, aux photos, etc.
- Vous pouvez également définir vos propres content providers.

Application 1

- Objectifs : insérer des views



Application 1

- Objectifs : insérer des views
  - Mettez les layout_width et layout_height à fill_parent pour la racile du liyaout
  - Mettez les layout_width et layout_height à wrap_content pour les autres
  - Utilisez layout_gravity
  - À voir marginleft à texte à password

Application 1

- Objectifs : insérer
  - Un bouton
    - Ajouter un bouton dans le layout

```
<Button
    android:id="@+id/main_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:text="@string/button"
/>
```

Application 1

- Objectifs : insérer
  - Un bouton
    - Ajouter un bouton dans le layout
    - Modifier le fichier strings.xml
    - Récupérer l'id du bouton au niveau de l'activity
    - Ajouter le listener sur le bouton :

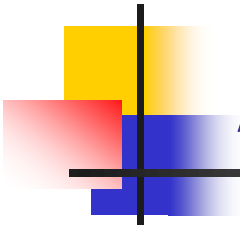
Application 1

- Objectifs : insérer
  - Un bouton

```
mainButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           // AlertDialog
        }
    });
```

Application 1

- Objectifs : insérer
  - L'image android devant le bouton
  - Pour cela créer un LinearLayout et mettez y un bouton et une l'image
    - Source de l'image @drawable/ic_launcher
  - Ajouter un texte éditable en bas
- Objectif : Une fois qu'on clique sur le bouton , le nom entré dans le édit text est affiché avec un message



Application 1

- Objectifs : les intents
- Dans une activité mettre :

```
Intent intent = new Intent();

intent.setClass(this, Other_Activity.class);

intent.putExtra("EXTRA_ID", "SOME DATAS");

startActivity(intent);
```

Application 1

- Objectifs : les intents
- Dans l'autre activité mettre :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
         Bundle extras = getIntent().getExtras();
         if (extras != null) {
                String datas= extras.getString("EXTRA_ID");
                if (datas!= null) {
                        // faire
                 }
        }
```

Application 1

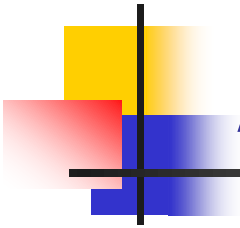
- Objectifs : les listviews
  - Mettre ceci dans le layout :

```
<ListView
        android:id="@+id/list"
        android:layout_height="wrap_content"
        android:layout_width="match_parent">
    </ListView>
```

Application 1

- Objectifs : les listviews
  - Dans l'activité :
    - Récupérer l'id de la liste
    - Définir la liste des éléments

```
String[] values = new String[] { "element1", "element2",
"element3","element4" };
ArrayList<String> list = new ArrayList<String>();
    for (int i = 0; i < values.length; ++i) {
      list.add(values[i]);
    }
```



Application 1

- Objectifs : les listviews
  - Définir le arrayAdapter

```
ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1
, list)
```

  - Assigner l'adaptateur

```
listView.setAdapter(adapter)
```

Application 1

- Objectifs : les listviews

```
listView.setOnItemClickListener(new OnItemClickListener() {
        @Override
            public void onItemClick(AdapterView<?> parent, View view,  int position, long id) {
             int itemPosition     = position;
             String  itemValue    = (String) listView.getItemAtPosition(position);
               Toast.makeText(getApplicationContext(),
               "Position :"+itemPosition+"  ListItem : " +itemValue , Toast.LENGTH_LONG)
               .show();
            }

        });
```

Application 1

- Objectifs : partage
- Dans le fichier menu_main.xml ajouter :

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:omgandroid="http://schemas.android.com/apk/res-auto">
    <!-- Share item -->
    <item
        android:id="@+id/menu_item_share"
        android:title="Share"
        omgandroid:actionProviderClass=
"android.support.v7.widget.ShareActionProvider" />
</menu>
```

Application 1

- Objectifs : partage
- Supprimer la classe OnOptionsItemSelected
- Dans la classe OnCreateOptionsMenu de l'activité on rajoute :

```
// Access the Share Item defined in menu XML
    MenuItem shareItem = menu.findItem(R.id.menu_item_share);
            if (shareItem != null) {
        mShareActionProvider= (ShareActionProvider)
MenuItemCompat.getActionProvider(shareItem);
    }
```

Application 1

- Objectifs : partage
- Dans la classe createOptionsMenu de l'activité on rajoute

```
if (mShareActionProvider != null) {
        Intent shareIntent = new Intent(Intent.ACTION_SEND);
        shareIntent.setType("text/plain");
        shareIntent.putExtra(Intent.EXTRA_SUBJECT, "Android Development");
        shareIntent.putExtra(Intent.EXTRA_TEXT, mainTextView.getText());
         mShareActionProvider.setShareIntent(shareIntent);
    }
```

Application 1

- Objectifs : broadcast receiver
- Créer cette classe :

```
public class MyReceiver extends BroadcastReceiver {
   @Override
   public void onReceive(Context context, Intent intent) {
      Toast.makeText(context, « Evenement détécté.",
Toast.LENGTH_LONG).show();
   }
}
```

Application 1

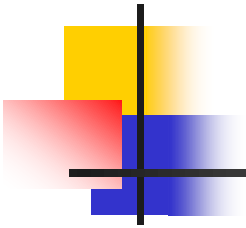
- Objectifs : broadcast receiver
- Au niveau du maniefst ajoutez :

```
<receiver android:name="MyReceiver">
    <intent-filter>
      <action android:name="android.intent.action.BOOT_COMPLETED">
      </action>
    </intent-filter>
  </receiver>
```

Menu

- Création dans le menu_main.xml

```
<menu xmlns:androclass="http://schemas.android.com/apk/res/android" >
    <item  android:id="@+id/item1"   android:title="Item 1"/>
    <item  android:id="@+id/item2"  android:title="Item 2"/>
    <item  android:id="@+id/item3"  android:title="Item 3"/>
</menu>
```



Menu

- Mettez ceci dans l'activité

```
public boolean onCreateOptionsMenu(Menu menu) {
        // Attacher le menu à l'activité.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
}
```

Menu

- Définir les actions pour chaque activité

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.item1:
          Toast.makeText(getApplicationContext(),"Item 1 Selected",Toast.LENGTH_LONG).show();
        return true;
       case R.id.item2:
            Toast.makeText(getApplicationContext(),"Item 2 Selected",Toast.LENGTH_LONG).show();
          return true;
          default:
           return super.onOptionsItemSelected(item);
    }
```

Service

- Ajouter ces deux boutons dans l'interface

```
<Button
    android:id="@+id/buttonStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="19dp"
    android:text="Start Service" />
```

```
<Button
    android:id="@+id/buttonStop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/butt
    android:layout_marginBottom="35dp"
    android:text="Stop Service" />
```

Service

- Définir les actions pour chaque activité

```
public class MyService extends Service {
 MediaPlayer myPlayer;
public IBinder onBind(Intent intent) {     return null;   }
 public void onCreate() {    Toast.makeText(this, "Service Created", Toast.LENGTH_LONG).show();
  myPlayer = MediaPlayer.create(this, R.raw.sun);
  myPlayer.setLooping(false);
 }
 public void onStart(Intent intent, int startid) {
  Toast.makeText(this, "Service Started", Toast.LENGTH_LONG).show();
  myPlayer.start();
 }
```

Service

- Définir une classe qui implements OnClickListener :

```
public class MainActivity extends Activity implements OnClickListener {
    Button buttonStart, buttonStop,buttonNext;
public void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
     buttonStart = (Button) findViewById(R.id.buttonStart);
     buttonStop = (Button) findViewById(R.id.buttonStop);
     buttonStart.setOnClickListener(this);
     buttonStop.setOnClickListener(this);
    }
}
```

Service

- Définir une classe qui implements OnClickListener :

```
public class MainActivity extends Activity implements OnClickListener {

    Button buttonStart, buttonStop,buttonNext;

        public void onClick(View src) {
            switch (src.getId()) {

            case R.id.buttonStart:
             startService(new Intent(this, MyService.class));
             break;

            case R.id.buttonStop:
             stopService(new Intent(this, MyService.class));
             break;
         }

             }
```

Service

- Définir une classe qui implements OnClickListener :

```
public class MainActivity extends Activity implements OnClickListener {

    Button buttonStart, buttonStop,buttonNext;

        public void onClick(View src) {
            switch (src.getId()) {

            case R.id.buttonStart:
             startService(new Intent(this, MyService.class));
             break;

            case R.id.buttonStop:
             stopService(new Intent(this, MyService.class));
             break;
         }

             }
```

Internal Storage

- Définir dans le layout:

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Data:" />
  <Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="save" />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="read" />
```

Internal Storage

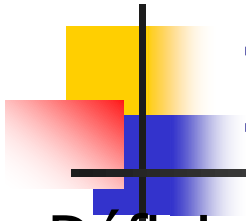
- Définir dans le layout:

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Data:" />
  <Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="save" />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="read" />
```

Internal Storage

- Définir dans l'activité avec le bouton save:

```
public void onClick(View arg0) {
    String filename=« fileName »;
    String data=editTextData.getText().toString();
    FileOutputStream fos;
    try {
        fos = openFileOutput(filename, Context.MODE_PRIVATE);
        //default mode is PRIVATE, can be APPEND etc.
        fos.write(data.getBytes());
        fos.close();
        Toast.makeText(getApplicationContext(),filename + " saved",
                       Toast.LENGTH_LONG).show();
        } catch (FileNotFoundException e) {e.printStackTrace();}
              catch (IOException e) {e.printStackTrace();}

}
```



Internal Storage

- Définir dans l'activité avec le bouton save:

MODE_PRIVATE : pouvant etre utiliser par l'application seulement

MODE_WORLD_READABLE : autoriser les autres applications à avoir un droit de lecture sur le fichier

MODE_WORLD_WRITEABLE : autoriser les autres applications à avoir un droit d'écriture sur le fichier

Internal Storage

- Définir dans l'activité avec le bouton save:

```
public void onClick(View arg0) {
    String filename= « fileName »;
    StringBuffer stringBuffer = new StringBuffer();
    try {
        //Attaching BufferedReader to the FileInputStream by the help of InputStreamReader
         BufferedReader inputReader = new BufferedReader(new InputStreamReader(openFileInput(filename)));
         String inputString;

       //Reading data line by line and storing it into the stringbuffer
       while ((inputString = inputReader.readLine()) != null) {
                    stringBuffer.append(inputString + "\n");
        }

            } catch (IOException e) {
                e.printStackTrace();
            }
       //Displaying data on the toast
       Toast.makeText(getApplicationContext(),stringBuffer.toString(),
                    Toast.LENGTH_LONG).show();

    }
```

Internal Storage

- Shared preferences
- kjjSharedPreferences preferences =

```
PreferenceManager.getDefaultSharedPreferences(this);
    SharedPreferences.Editor editor = preferences.edit();
    editor.putString("nom",nom);
    editor.apply();
```

Internal Storage

- Shared preferences
- kjj preferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);
- SharedPreferences.Editor editor = preferences.edit();

```
editor.putString("nom",nom);
editor.apply();
```

SMS

- Autoriser l'envoie de sms

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

- Envoyer sms

```
SmsManager smsManager = SmsManager.getDefault();
smsManager.sendTextMessage("709547822", null, "message", null,
null);
Toast.makeText(context, "Message Sent successfully!", );
```

EMAIL

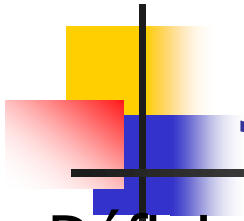
- Envoyer un sms

```
Intent email = new Intent(Intent.ACTION_SEND);
 email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
 email.putExtra(Intent.EXTRA_SUBJECT, subject);
 email.putExtra(Intent.EXTRA_TEXT, message);

email.setType("message/rfc822");

 startActivity(Intent.createChooser(email, "Choose an Email client :"));
```

JSON

- Définir les différentes classes

```
public class Person {
    private String name;
    private Address address;
    private List<PhoneNumber> phoneList;

    // get and set
}
```



JSON

- Définir les différentes classes

```
}
```