



ANDROID

Widgets et événements

exercice_events

1

Les zones de texte :

```
<EditText android:text=""
    android:id="@+id/EditText01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</EditText>
```

```
EditText edit = new EditText(this);
edit.setText("Mon texte");
gabarit.addView(edit);
```

Interception d'événements:

```
edit.addTextChangedListener(new TextWatcher() {
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        // faire quelque chose
    }
});
```

Dans votre application, l'utilisateur aura peut être besoin de saisir quelques données de types différents.

Dans ce cas, il faut faciliter le travail à l'utilisateur en faisant apparaître un clavier en fonction du type de champ à remplir (adresse mail, numéro de téléphone etc...).

Pour cela dans votre **"EditText"**, il faut simplement spécifier l'option **android:inputType**. Voici les différentes valeurs disponibles :

text (par défaut) : clavier normal
textCapCharacters : Clavier tout en majuscule
textCapWords : Première lettre automatiquement en majuscule
textCapSentences : Phrase en majuscule
textAutoCorrect : Activer la correction automatique
textAutoComplete : Activer la correction automatique forcée
textMultiLine : Texte sur plusieurs lignes

3

textNoSuggestions : Pas de suggestion de correction
textUri : Saisie d'une url web
textEmailAddress : Adresse mail
textEmailSubject : Sujet de mail
textShortMessage : Active le raccourci smiley sur le clavier
textPersonName : Saisie du nom d'une personne (affichage de speech to text en bas à gauche du clavier)
textPostalAddress : Saisie d'une adresse postale (affichage de speech to text en bas à gauche du clavier)
textPassword : Saisie de mot de passe
textVisiblePassword : Saisie d'un mot de passe visible
textWebEditText : Text Web (activation de raccourci tabulation et speech to text)
number / numberSigned / numberDecimal / phone / datetime / date / time : Clavier numérique

4

Les boutons

```
<Button
    android:id="@+id/Button01"
    android:text="Calculer !"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>
```

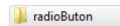
La gestion des événements de *click* se font par l'intermédiaire d'un listener:

```
Button b = (Button)findViewById(R.id.Button01);

b.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(v.getContext(), "Stop !", Toast.LENGTH_LONG).show();
    }
});
```

5

RADIOBUTTON



```
<RadioGroup
    android:id="@+id/radioSex"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/radioMale"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_male"
        android:checked="true" />

    <RadioButton
        android:id="@+id/radioFemale"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_female" />

</RadioGroup>
```

6

```

package com.example.radiobuton;
import android.app.Activity;import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
public class MainActivity extends Activity {
private RadioGroup radioSexGroup;
private RadioButton radioSexButton;
private Button btnDisplay;
@Override public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ajouterEcouterBouton();
}

```

7

```

public void ajouterEcouterBouton() {

    //android:checked="true" pour avoir le bouton choisi par défaut

    radioSexGroup = (RadioGroup) findViewById(R.id.radioSex);
    btnDisplay = (Button) findViewById(R.id.btnDisplay);

    btnDisplay.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {

            // obtenir le id du bouton radio choisie
            int idChoisie = radioSexGroup.getCheckedRadioButtonId();

            // trouver le bouton radio via le id
            radioSexBouton = (RadioButton) findViewById(idChoisie);

            Toast.makeText( MainActivity.this,
            radioSexBouton.getText(), Toast.LENGTH_SHORT).show();

}); }

```

8



CHECKBOX



```
<?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" >

    <CheckBox
        android:id="@+id/chkIos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/chk_ios" />

    <CheckBox
        android:id="@+id/chkAndroid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/chk_android"
        android:checked="true" />
```

10

```

package com.example.checkbox;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends Activity {

    private CheckBox chkIos, chkAndroid, chkWindows;
    private Button btnDisplay;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        ajouterEcouleurChkIos();
        ajouterEcouleurBouton();
    }

```

11

```

public void ajouterEcouleurChkIos() {

    chkIos = (CheckBox) findViewById(R.id.chkIos);

    chkIos.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {

            if (chkIos.isChecked()) {
                Toast.makeText(getApplicationContext(),
                    "Oups! Essaie Android", Toast.LENGTH_LONG).show();
            }

        }

    });
}

```

12

```

public void ajouterEcouleurBouton() {

    chkIos = (CheckBox) findViewById(R.id.chkIos);
    chkAndroid = (CheckBox) findViewById(R.id.chkAndroid);
    chkWindows = (CheckBox) findViewById(R.id.chkWindows);
    btnDisplay = (Button) findViewById(R.id.btnDisplay);

    btnDisplay.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {

            StringBuffer result = new StringBuffer();
            result.append("IPhone check : ").append(chkIos.isChecked());
            result.append("\nAndroid check : ").append(chkAndroid.isChecked());
            result.append("\nWindows Mobile check : ").append(chkWindows.isChecked());

            Toast.makeText(getApplicationContext(), result.toString(),
                Toast.LENGTH_LONG).show();

        }
    });
}

```

13

CheckBox

☐ IPhone

☒ Android

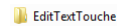
☒ Windows Mobile

Afficher

IPhone check : false
Android check : true
Windows Mobile check : true

14

EDITTEXT : TEST TOUCHE CLAVIER



```
<?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:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <requestFocus />
    </EditText>

</LinearLayout>
```

15

```
package com.textedit.android;
import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity {
    private EditText edittext;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        ajouterEcouteurTouche();
    }
}
```

16


```

public void ajouterEcouteurTouche() {

    edittext = (EditText) findViewById(R.id.editText);

    // ajouter un ecouteur
    edittext.setOnKeyListener(new OnKeyListener() {
        public boolean onKey(View v, int keyCode, KeyEvent event)
        {

            // Si la touche a été pressée et c'est «enter»
            if ((event.getAction() == KeyEvent.ACTION_DOWN)
                && (keyCode == KeyEvent.KEYCODE_ENTER)) {

                Toast.makeText(MainActivity.this,
                    edittext.getText().toString(),
                    Toast.LENGTH_LONG).show();
                return true;
            }
        }
    });
}

```

17

```

//Si la touche a été pressée et c'est «9»

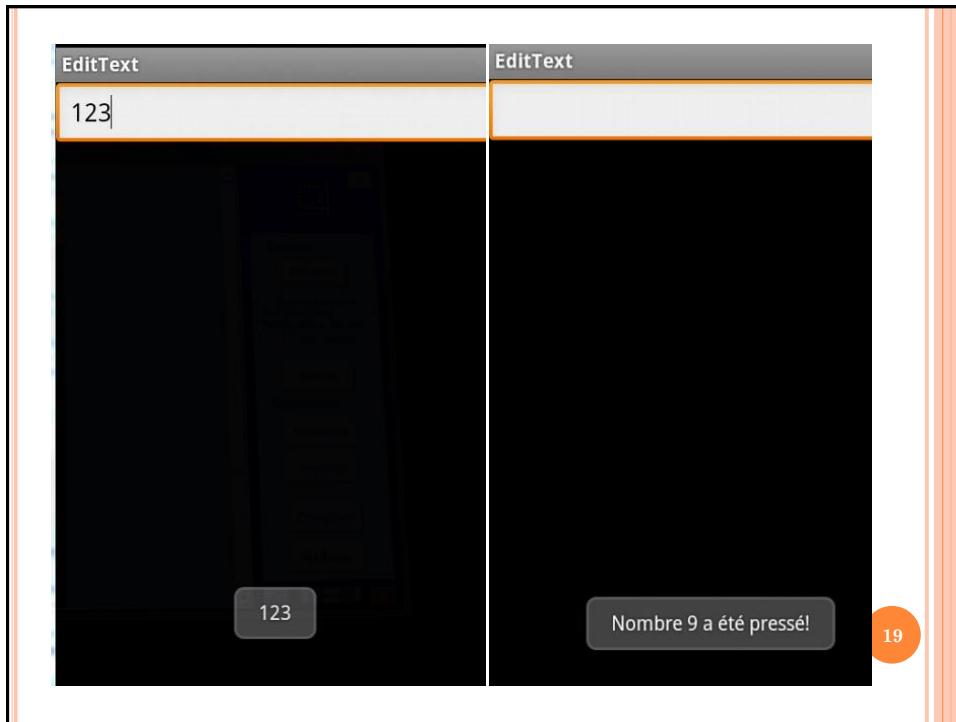
} else if ((event.getAction() == KeyEvent.ACTION_DOWN)
    && (keyCode == KeyEvent.KEYCODE_9)) {

    Toast.makeText(MainActivity.this,
        "Nombre 9 a été pressé!", Toast.LENGTH_LONG).show();
    return true;
}

return false;
}});}}

```

18



ÉVÈNEMENT SUR LA «VIEW»

```
import android.view.View.OnTouchListener;
import android.view.View.OnClickListener;
import android.app.Activity;
import android.os.Bundle;
import android.view.MotionEvent; //Pour le toucher
import android.view.View;
import android.widget.Button;

// Notre activité détectera les touches et les clics sur les vues qui se sont inscrites
public class MainActivity extends Activity implements View.OnTouchListener,
View.OnClickListener {
    private Button b = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        b = (Button) findViewById(R.id.boutton);
        b.setOnTouchListener(this);
        b.setOnClickListener(this);
    }
}
```

20

```
@Override
public boolean onTouch(View v, MotionEvent event) {
    /* Réagir au toucher */
    return true;
}
```

```
@Override
public void onClick(View v) {
    /* Réagir au clic */
}
}
```

À chaque fois qu'on appuiera sur un bouton, quel qu'il soit, on rentrera dans la même méthode, et on exécutera donc le même code...

21

Heureusement, la vue passée dans la méthode `onClick(View)` permet de différencier les boutons.

En effet, il est possible de récupérer l'identifiant de la vue (l'identifiant défini en XML et qu'on retrouve dans le fichier `R`) sur laquelle le clic a été effectué.

Ainsi, nous pouvons réagir différemment en fonction de cet identifiant :

```
public void onClick(View v) {
    switch(v.getId()) {

        case R.id.bouton1:
            ....
            break;

        case R.id.bouton2:
            ....
            break;

    }
}
```

22

EXERCICE DE PRATIQUE (FORMATIF)

Bonjour

Vous êtes bien dans l'inscription du Collège Ahuntsic
Enregistrement d'un utilisateur

Nom :

Prénom :

SEXE : Homme ☐ Femme ☐

Fonction : Enseignant

Type de travail : Journalier ☐ Temps partiel ☐
Temps plein ☐ Occasionnel ☐

Commentaires :

Envoyer Effacer

Voir plus tard (spinner)

Concevoir le formulaire suivant (sauf fonction). Le texte en haut doit apparaître. Lorsqu'on clique sur «Envoyer», vous devez afficher toutes les informations dans un Toast, dans le format :

Nom : Tremblay
Prénom : Carole
Sexe : Femme
.....

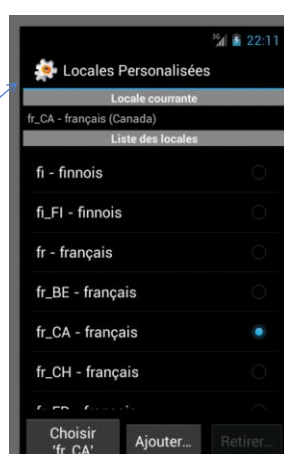
Lorsqu'on clique sur «Effacer» tous les champs seront remis aux valeurs de défaut. Vous devez faire 2 versions :

- 1) Chaque bouton à son propre écouteur d'événement;
- 2) Une seul écouteur d'événement, à partir duquel on détermine quel bouton a été cliqué.

23

DATEPICKER (PROJET DATEPICKER)

Changer la langue, pour le format de date en Français :



24

CONCEPTS

Date et heure

Révision dates en Java.

Il suffit de récupérer un objet de type `Calendar` à l'aide de la méthode de classe `Calendar.getInstance()`. Cette méthode retourne un `Calendar` qui contiendra les informations sur la date et l'heure, au moment de la création de l'objet.

25

Il est ensuite possible de récupérer des informations à partir de la méthode `int get(int champ)` avec `champ` qui prend une valeur telle que :

`Calendar.YEAR` pour l'année ;

`Calendar.MONTH` pour le mois. Attention, le premier mois est de rang 0, alors que le premier jour du mois est bien de rang 1 !

`Calendar.DAY_OF_MONTH` pour le jour dans le mois ;

`Calendar.HOUR_OF_DAY` pour l'heure ;

`Calendar.MINUTE` pour les minutes ;

`Calendar.SECOND` pour les secondes.

26

```
// Contient la date et l'heure au moment de sa création
```

```
Calendar calendrier = Calendar.getInstance();
```

```
// On peut ainsi lui récupérer des attributs
```

```
int mois = calendrier.get(Calendar.MONTH);
```

Insertion de dates

Pour insérer une date, on utilise le widget `DatePicker`. Ce widget possède en particulier deux attributs XML intéressants.

Tout d'abord `android:minDate` pour indiquer quelle est la date la plus ancienne à laquelle peut remonter le calendrier, et son opposé `android:maxDate`.

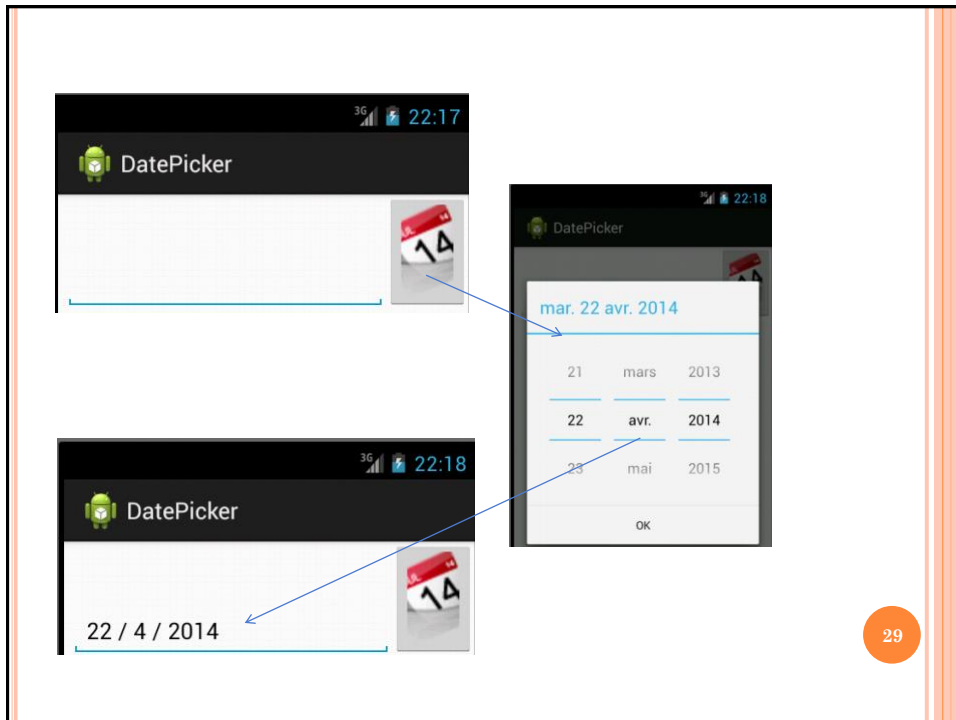
27

```
<DatePicker
    android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:startYear="2012"
    android:endYear="2032" />
```

```
mDatePicker = (DatePicker) findViewById(R.id.datePicker);
mDatePicker.updateDate(mDatePicker.getYear(), 0, 1); //0-mois,1-jour
```



28



29

CODE :ACTIVITY_MAIN.XML

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
>
<EditText
    android:id="@+id/editText"
    android:layout_width="250dp"
    android:layout_marginTop="50dp"
    android:layout_marginLeft="5dp"
    android:layout_height="wrap_content"
    android:editable="false" >
</EditText>

```

30

```

<ImageButton
    android:id="@+id/imageButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editText"
    android:layout_toRightOf="@+id/editText"
    android:contentDescription="@string/date_choisie"
    android:cropToPadding="true" → Si true, l'image sera recadrée pour
    android:src="@drawable/icone_calendrier" /> s'adapter à son remplissage.

</RelativeLayout>

```

31

MainActivity

```

import java.util.Calendar;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;

public class MainActivity extends Activity implements
OnClickListener {
    private ImageButton ib;
    private Calendar cal;
    private int day;
    private int month;
    private int year;
    private EditText et;

```

32


```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ib = (ImageButton) findViewById(R.id.imageButton1);
    cal = Calendar.getInstance();
    day = cal.get(Calendar.DAY_OF_MONTH);
    month = cal.get(Calendar.MONTH);
    year = cal.get(Calendar.YEAR);
    et = (EditText) findViewById(R.id.editText);
    ib.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    //appelle onCreateDialog
    showDialog(0); //on verra avec les fragments
}

```

33

```

@Override
@Deprecated
protected Dialog onCreateDialog(int id){

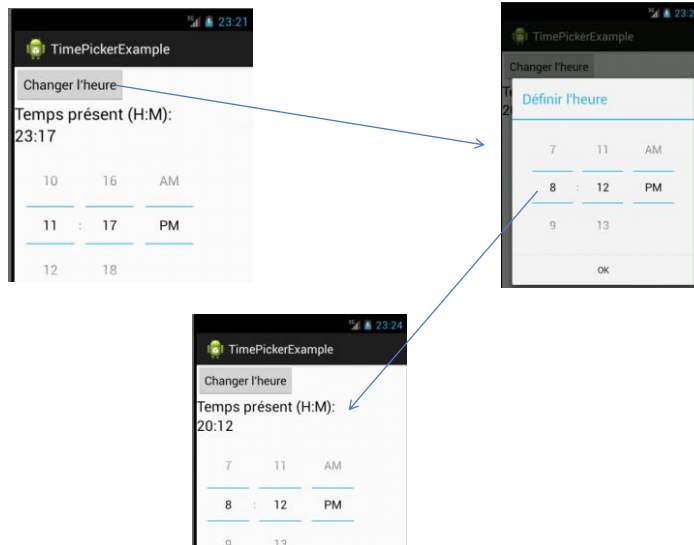
    return new DatePickerDialog(this, datePickerListener, year,
        month, day);
}

private DatePickerDialog.OnDateSetListener
    datePickerListener = new
        DatePickerDialog.OnDateSetListener() {
    public void onDateSet(DatePicker view, int selectedYear,
        int selectedMonth, int selectedDay) {
        et.setText(selectedDay + " / " + (selectedMonth + 1) + " / "
            + selectedYear);
    }
};
}

```

34

TIMEPICKER (PROJET TIMEPICKEREXAMPLE)



35

Insertion d'horaires

Pour choisir un horaire, on utilise TimePicker, qui fonctionne comme DatePicker .

Il n'est pas possible de définir un horaire maximal et un horaire minimal cette fois, il est possible de définir l'heure avec **void setCurrentHour(Integer hour)**, de la récupérer avec **Integer getCurrentHour()**, et de définir les minutes avec **void setCurrentMinute(Integer minute)**, puis de les récupérer avec **Integer getCurrentMinute()**.

36

Le Listener pour le changement d'horaire est cette fois géré par **void setTimeChangedListener(TimePicker.OnTimeChangedListener)**.

```
<TimePicker
    android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true" />
```

Mettre au format 24 heures
et non pas anglais (AM,
PM).

void
setIs24HourView(Boolean).

```
mTimePicker = (TimePicker) findViewById(R.id.timePicker);
mTimePicker.setIs24HourView(true);
mTimePicker.setTimeChangedListener(new TimePicker.OnTimeChange
dListener() {
    @Override
    public void onTimeChanged(TimePicker view, int heure, int minute) {
        Toast.makeText(MainActivity.this, "C'est vous qui voyez, il est donc " + St
ring.valueOf(heure) + ":" + String.valueOf(minute), Toast.LENGTH_SHOR
T).show();
    }
});
```

37

MAIN.XML

```
<?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" >

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_label" />

    <TextView
        android:id="@+id/time_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/time_txt"
        android:textAppearance="?android:attr/textAppearanceLarge" />
```

38

```

<TextView
    android:id="@+id/txtTime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textAppearance="?android:attr/textAppearanceLarge"
/>

<TimePicker
    android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

</LinearLayout>

```

39

```

import java.util.Calendar;
import android.app.Activity;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends Activity {

    private TextView textViewTime;
    private TimePicker timePicker;
    private Button button;

    private int hour;
    private int minute;

    static final int TIME_DIALOG_ID = 999;

```

40

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    tempsPresentDansView();
    BoutonListener();
}

```

41

```

// afficher le temps présent
public void tempsPresentDansView() {

    textViewTime = (TextView) findViewById(R.id.txtTime);
    timePicker = (TimePicker) findViewById(R.id.timePicker);

    final Calendar c = Calendar.getInstance();
    hour = c.get(Calendar.HOUR_OF_DAY);
    minute = c.get(Calendar.MINUTE);

    // placer le temps présent dans un text textview
    textViewTime.setText(new
    StringBuilder().append(padding_str(hour)).append(":").append(
    padding_str(minute)));

    // placer le temps présent dans le timepicker
    timePicker.setCurrentHour(hour);
    timePicker.setCurrentMinute(minute);

}

```

42

```

public void BoutonListener() {

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

    button.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // Ok à voir avec les fragments plus tard
            showDialog(TIME_DIALOG_ID);

        }

    });

}

```

43

```

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case TIME_DIALOG_ID:
            // timepicker avec le temps présent
            //Syntaxe:
            //TimePickerDialog(Context context,
            //TimePickerDialog.OnTimeSetListener callBack, int hourOfDay,
            //int minute, boolean is24HourView)
            return new TimePickerDialog(this, timePickerListener,
            hour, minute,false);

        }

    return null;
    }

```

44

```

private TimePickerDialog.OnTimeSetListener
timePickerListener = new
TimePickerDialog.OnTimeSetListener() {
public void onTimeSet(TimePicker view, int selectedHour, int
selectedMinute) {
hour = selectedHour;
minute = selectedMinute;

// temps présent dans le textview
textViewTime.setText(new
StringBuilder().append(padding_str(hour)).append(":").append(
padding_str(minute)));

//temps présent dans le timepicker
timePicker.setCurrentHour(hour);
timePicker.setCurrentMinute(minute);

}
};

```

45

```

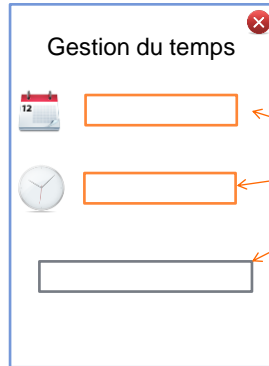
private static String padding_str(int c) {
if (c >= 10)
return String.valueOf(c);
else
return "0" + String.valueOf(c);
}
}

```

46

Exercice : Créer la vue suivante

www.iconfinder.com



Fermer l'activité

- Images sont des «ImageButton»
- Utiliser un DatePicker et TimePicker, via Dialog
- Afficher le choix dans les champs.
- Afficher dans ce champ si l'année choisie est bissextile et selon l'heure si on est le matin, après-midi ou nuit.

47

Cette fois-ci pour des raisons de maintenance, il faut utiliser

//Création du DatePickerDialog

```
DatePickerDialog datePickerDialog = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
```

```
@Override
```

```
public void onDateSet(DatePicker view, int myYear, int myMonth, int myDay)
```

```
{
```

```
.....
```

```
}
```

```
}, theYear, theMonth, theDay);
```

```
//Affichage du DatePickerDialog
```

```
datePickerDialog.show();
```

DatePickerDialog

```
DatePickerDialog (Context context,
DatePickerDialog.OnDateSetListener listener,
int year,
int month,
int dayOfMonth)
```

Creates a new date picker dialog for the specified date using the parent context's default date picker dialog theme.

Parameters	
context	Context: the parent context
listener	DatePickerDialog.OnDateSetListener: the listener to call when the user sets the date
year	int: the initially selected year
month	int: the initially selected month (0-11 for compatibility with <code>android</code>)
dayOfMonth	int: the initially selected day of month (1-31, depending on month)

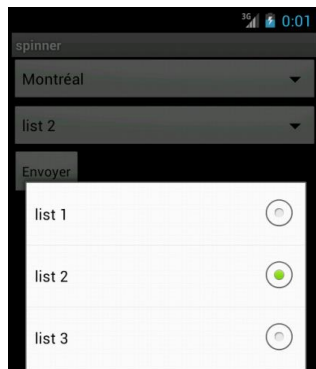
Même principe pour le TimePickerDialog

48

SPINNER (LISTE DÉROULANTE) (PROJET SPINNER)



49



50

MAIN.XML

```
<?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">

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:entries="@array/table_villes"
        android:prompt="@string/choix_ville" />

    <Spinner
        android:id="@+id/spinner2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```

51

```
<Button
    android:id="@+id/btnSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Envoyer" />
```

```
</LinearLayout>
```

52

DANS LE DOSSIER RES/VALUES

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">spinner</string>
    <string name="choix_ville">Choisir une ville</string>

    <string-array name="table_villes">
        <item>Québec</item>
        <item>Montréal</item>
        <item>Drummondville</item>
        <item>Laval</item>
        <item>Longueuil</item>
        <item>Trois-Rivières</item>
        <item>St. Lazare</item>
        <item>Brossard</item>
    </string-array>

</resources>
```

53

MainActivity

```
import java.util.ArrayList;
import java.util.List;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends Activity implements
OnItemSelectedListener {

    private Spinner spinner1, spinner2;
    private Button btnSubmit;
```

54

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    addItemOnSpinner2();
    addListenerOnButton();

}

```

55

```
public abstract void onItemSelected (AdapterView<?> parent, View view, int position, long id) Added in API level 1
```

Callback method to be invoked when an item in this view has been selected. This callback is invoked only when the newly selected position is different from the previously selected position or if there was no selected item.

Implementers can call getItemAtPosition(position) if they need to access the data associated with the selected item.

Parameters

<i>parent</i>	The AdapterView where the selection happened
<i>view</i>	The view within the AdapterView that was clicked
<i>position</i>	The position of the view in the adapter
<i>id</i>	The row id of the item that is selected

```

public void onItemSelected(AdapterView<?> parent, View view, int
pos, long id) {

    Toast.makeText(parent.getContext(),
        " Selectionné : \n" + parent.getItemAtPosition(pos).toString(),
        Toast.LENGTH_LONG).show();

}

```

56

//ajouter les éléments dans le spinner de façon dynamique

```
public void addItemOnSpinner2() {
```

```
spinner2 = (Spinner) findViewById(R.id.spinner2);
```

```
List<String> list = new ArrayList<String>();
```

```
list.add("list 1");
```

```
list.add("list 2");
```

```
list.add("list 3");
```

//Matière a approfondir plus tard

```
ArrayAdapter<String> dataAdapter = new
```

```
ArrayAdapter<String>(this,  
android.R.layout.simple_spinner_item,list);
```

```
dataAdapter.setDropDownViewResource(android.R.layout.simple_s  
spinner_dropdown_item);
```

```
spinner2.setAdapter(dataAdapter);
```

```
}
```

```
public void setDropDownViewResource (int resource)
Sets the layout resource to create the drop down views.
Parameters
resource the layout resource defining the drop down views
See Also
getDropDownView(int, android.view.View, android.view.ViewGroup)
```

57

//obtenir la valeur sélectionnée

```
public void addListenerOnButton() {
```

```
spinner1 = (Spinner) findViewById(R.id.spinner1);
```

```
spinner2 = (Spinner) findViewById(R.id.spinner2);
```

```
btnSubmit = (Button) findViewById(R.id.btnSubmit);
```

```
btnSubmit.setOnClickListener(new OnClickListener() {
```

```
@Override
```

```
public void onClick(View v) {
```

```
Toast.makeText(MainActivity.this,
```

```
"OnClickListener : " +
```

```
"\nSpinner 1 : " + String.valueOf(spinner1.getSelectedItem()) +
```

```
"\nSpinner 2 : " + String.valueOf(spinner2.getSelectedItem()),
```

```
Toast.LENGTH_SHORT).show();
```

```
}};}
```

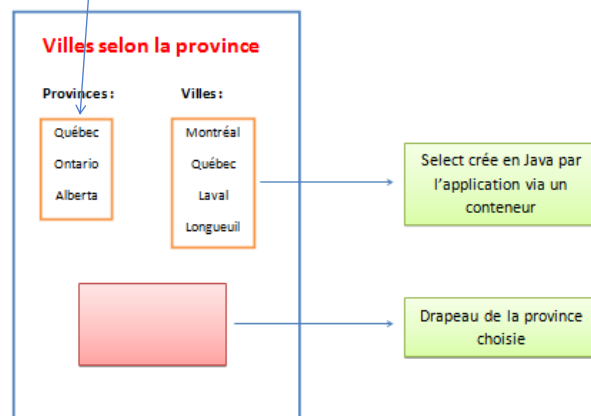
```
public Object getSelectedItem ()
Returns
The data corresponding to the currently selected item, or null if there is nothing selected.
```

Retourne la représentation en format de String.

58

Exercice de laboratoire sur les «Spinner»

Afficher un **Spinner** avec la liste des provinces et selon la province choisie un **Spinner** de villes de cette province apparaîtra. Les provinces seront données `<string-array name="table_provinces">`, tandis que les villes de la province choisie seront prises d'un conteneur de Java. Choisir parmi les conteneurs que vous avez appris dans votre cours de Java, celui qui s'adapte le mieux à notre problème. Pour les drapeaux, voir le site : http://fr.wikipedia.org/wiki/Liste_des_drapeaux_canadiens#Provinciaux



59

```
<ImageView
    android:id="@+id/image"
    android:layout_width="50dip"
    android:layout_height="fill_parent"
    android:src="@drawable/icon"
    android:layout_marginLeft="3dip"
    android:scaleType="center"/>
```

```
</LinearLayout>
```

Solution 1: Si créé par XML

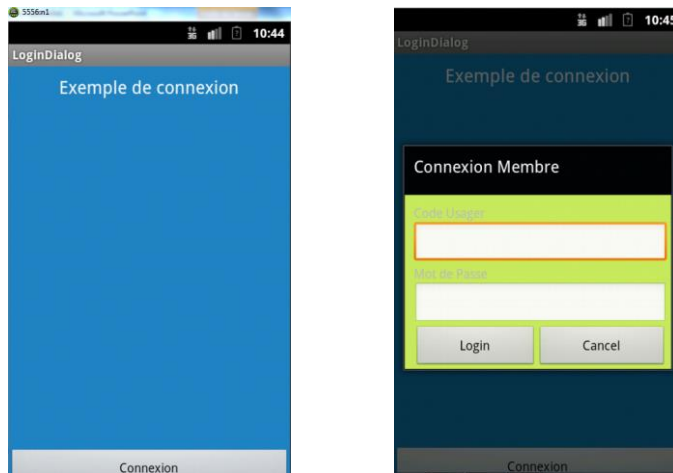
```
ImageView img= (ImageView) findViewById(R.id.image);
img.setImageResource(R.drawable.mon_image);
```

Solution 2: Si crée par ImageView d'une classe de Java

```
ImageView img = new ImageView(this);
img.setImageResource(R.drawable.mon_image);
```

60

BOITE DE DIALOGUE : EXEMPLE DE ÉCRAN DE CONNEXION



61

LAYOUT DE L'ACTIVITÉ

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#2186c6">
    <TextView
        android:textColor="#fff"
        android:textSize="20sp"
        android:layout_height="wrap_content"
        android:id="@+id/textView1"
        android:text="Exemple de connexion"
        android:layout_width="wrap_content"
        android:layout_gravity="center_horizontal"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:layout_weight="1"></TextView>
```

62

```

<Button
    android:layout_width="match_parent"
    android:id="@+id/btnLoginDialog"
    android:text="Connexion"
    android:layout_gravity="center_vertical"
    android:layout_height="wrap_content"></Button>
</LinearLayout>

```

63

LAYOUT DU DIALOGUE

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:background="#caee60"
    android:layout_width="300dp"
    android:paddingTop="10dp">
    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Code Usager" />
    <EditText
        android:id="@+id/txtUsername"
        android:layout_height="wrap_content"
        android:layout_width="match_parent">
    </EditText>

```

64


```

<TextView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Mot de Passe" />

<EditText
    android:id="@+id/txtPassword"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:inputType="textPassword"
    >
</EditText>

```

65

```

<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/btnLogin"
        android:text="Login" ></Button>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/btnCancel"
        android:text="Cancel" ></Button>
</LinearLayout>
</LinearLayout>

```

66

L'ACTIVITÉ

```
package com.exemple.logindialog;

import com.exemple.logindialog.R;

import android.app.Activity;
import android.app.Dialog;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginDialogActivity extends Activity implements
OnClickListener {

    Button btnLoginDialog;
```

67

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    btnLoginDialog = (Button) findViewById(R.id.btnLoginDialog);
    btnLoginDialog.setOnClickListener(this);
}
```

68

```

@Override
public void onClick(View v) {
    if (v == btnLoginDialog) {

        // Création du dialogue
        final Dialog login = new Dialog(this);
        // View du dialogue
        login.setContentView(R.layout.login_dialog);
        login.setTitle("Connexion Membre");

        // Contenu
        Button btnLogin = (Button) login.findViewById(R.id.btnLogin);
        Button btnCancel = (Button) login.findViewById(R.id.btnCancel);
        final EditText txtUsername =
            (EditText)login.findViewById(R.id.txtUsername);
        final EditText txtPassword =
            (EditText)login.findViewById(R.id.txtPassword);
    }
}

```

69

```

// Écouteurs
btnLogin.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        if(txtUsername.getText().toString().trim().length() > 0 &&
            txtPassword.getText().toString().trim().length() > 0)
        {
            //Validation ici
            Toast.makeText(LoginDialogActivity.this,
                "Ok login valide", Toast.LENGTH_LONG).show();

            //Retour à l'activité
            login.dismiss();
        }
        else
        {
            Toast.makeText(LoginDialogActivity.this,
                "Vous devez entrer un code et mot de passe", Toast.LENGTH_LONG).show();
        }
    }
});

```

70

Dismiss this dialog, removing it from the screen. This method can be invoked safely from any thread. Note that you should not override this method to do clean
Dismiss this dialog, removing it from the screen. This method can be invoked safely from any thread. Note that you should not override this method to do cle
Dismiss this dialog, removing it from the screen. This method can be invoked safely from any thread. Note that you should not override this method to do

```
btnCancel.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        login.dismiss();
    }
});

// Rendre dialog visible.
login.show();
}
}
}
```

L'enleve de l'écran. On ne doit pas réécrire cette méthode pour libérer des ressources. Il faut utiliser alors onStop()

71

Exercice formatif :

1. Vous avez créé dans l'exercice 3 un formulaire. Vous devez y ajouter un «spinner» pour la fonction.
2. Créer dans votre activité principal deux «ImageButton» un pour «Crér membre» et un autre pour «Connexion». Tous les deux doivent afficher dans un «Dialog» leurs formulaires respectifs. Prévoir dans les deux cas un «ImageButton» pour fermer le dialogue. Prévoir un autre dans l'activité principal pour fermer l'activité.

72