



Android sweatshop II

7.7.16



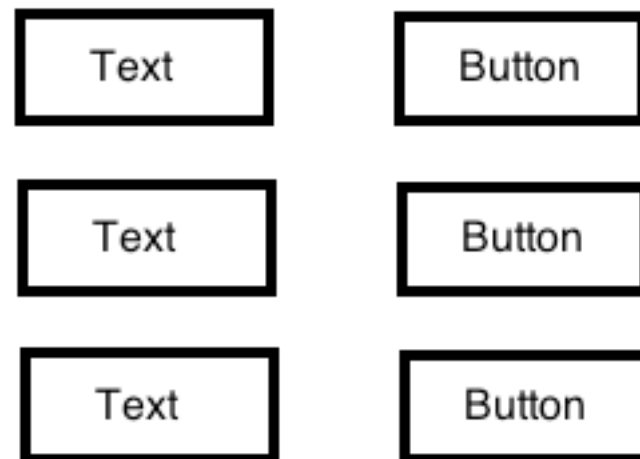
Last episode

- Activity lifecycle
- Logcat
- Layouts
 - MATCH_PARENT vs WRAP_CONTENT
- Views
 - OnClickListener



Home work

- A client asked for a grid like app, each TextView should hold a separate counter
- Bonus - use three horizontal LinearLayout for each row, and a vertical LinearLayout for their parent





Home work

- Challenge - how to align them ?
- Hint: `LinearLayout` + `LinearLayouts`
- Any one wanna try ?



Home work UI





Homework solution

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

    textCounter1 = (TextView) findViewById(R.id.text_counter_1);
    textCounter2 = (TextView) findViewById(R.id.text_counter_2);
    textCounter3 = (TextView) findViewById(R.id.text_counter_3);

    button1 = (Button) findViewById(R.id.button_counter_1);
    button2 = (Button) findViewById(R.id.button_counter_2);
    button3 = (Button) findViewById(R.id.button_counter_3);

    button1.setOnClickListener(this);
    button2.setOnClickListener(this);
    button3.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.button_counter_1:
            counter1++;
            textCounter1.setText(String.valueOf(counter1));
            break;
        case R.id.button_counter_2:
            counter2++;
            textCounter2.setText(String.valueOf(counter2));
            break;
        case R.id.button_counter_3:
            counter3++;
            textCounter3.setText(String.valueOf(counter3));
            break;
    }
}
```



Home work

activity_main.xml

- RelativeLayout (why?)
 - LinearLayout
 - LinearLayout
 - TextView
 - Button
 - LinearLayout
 - TextView
 - Button
 - LinearLayout
 - TextView
 - Button

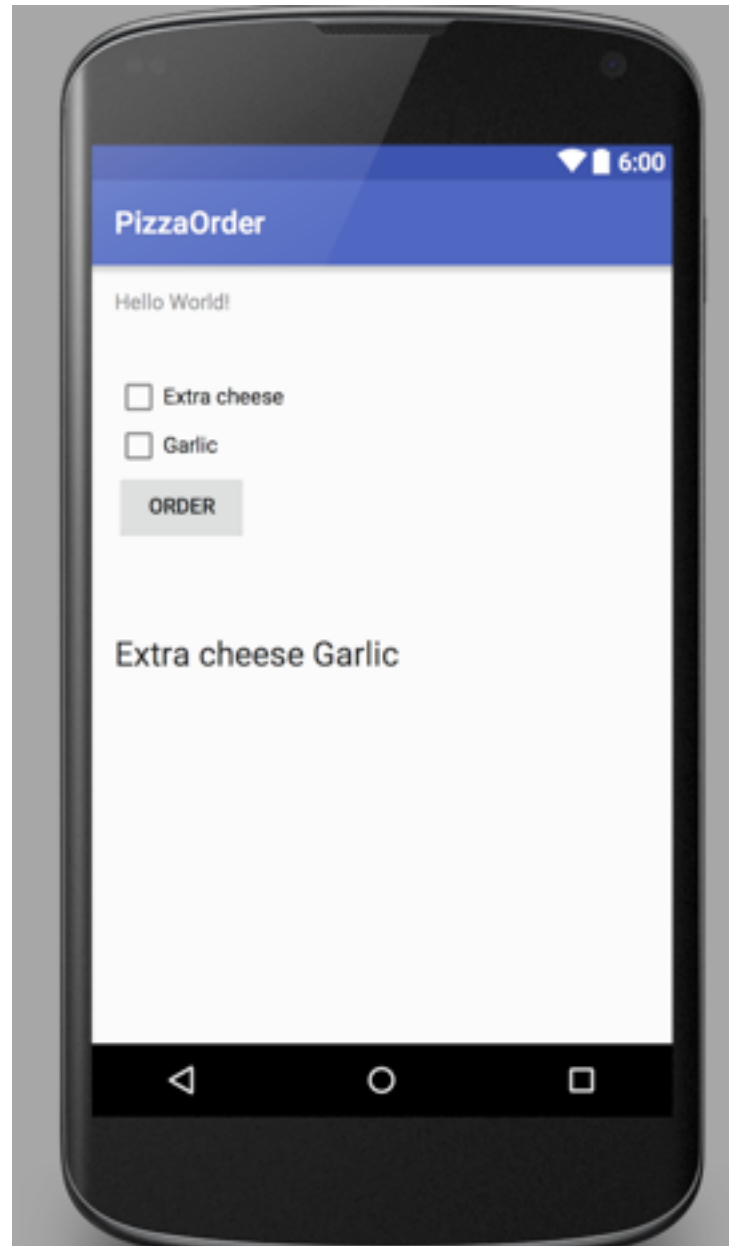


Content table

- Pizza app (from Android For Dummies)
 - More views and layouts
 - Bitamp
 - Passing data to another activity
 - Intent
- TODO app
 - ListView
 - Array adapter



PIZZA !





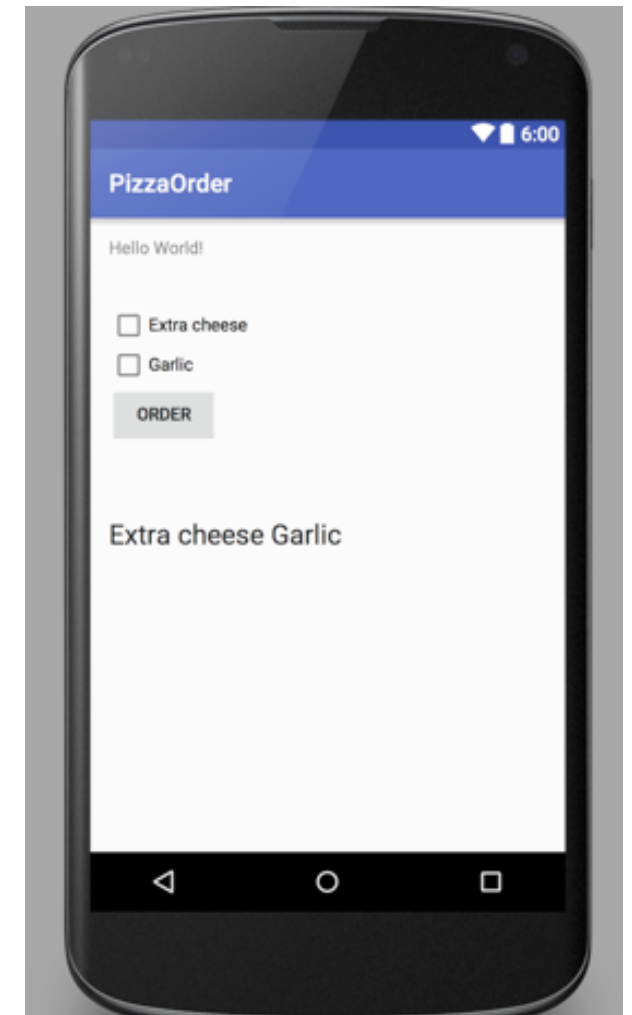
More examples

<p>01_05_01</p> <p>Hello world!</p> <p><input type="checkbox"/> Pepperoni</p> <p><input checked="" type="checkbox"/> Extra cheese</p> <p>Show</p> <p>Extra cheese</p>	<p>01_05_01</p> <p>Hello world!</p> <p><input checked="" type="checkbox"/> Pepperoni</p> <p><input type="checkbox"/> Extra cheese</p> <p>Show</p> <p>Pepperoni</p>
<p>01_05_01</p> <p>Hello world!</p> <p><input checked="" type="checkbox"/> Pepperoni</p> <p><input checked="" type="checkbox"/> Extra cheese</p> <p>Show</p> <p>Pepperoni Extra cheese</p>	<p>01_05_01</p> <p>Hello world!</p> <p><input type="checkbox"/> Pepperoni</p> <p><input type="checkbox"/> Extra cheese</p> <p>Show</p> <p>Plain</p>



PIZZA - what does it do ?

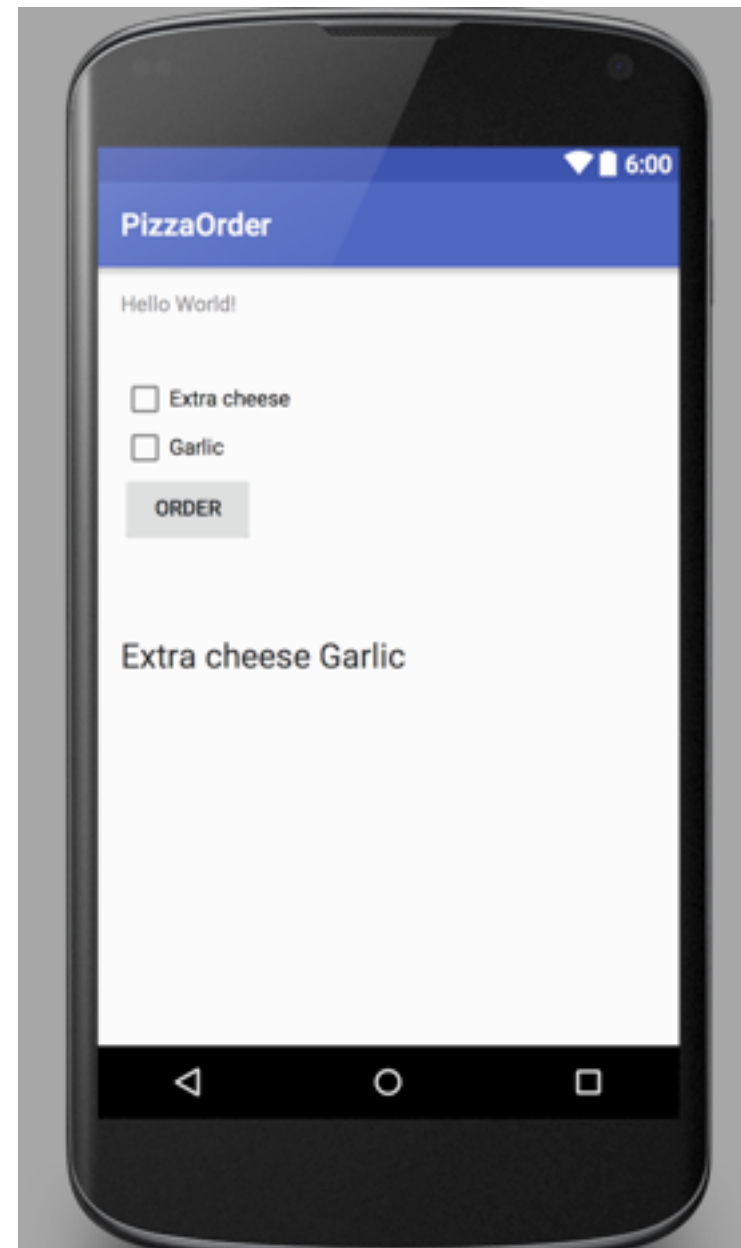
- Press on Order
- Check which checkboxes are press
- Set the order string inside the textview





Pizza instructions - layout designer

- Drag checkboxes
- Drag buttons
- Drag a large TextView
- Remove the HelloWorld
- Change the id's to match





PIZZA - MainActivity

- add Class members - all of the views
 - private CheckBox ...
- set them using `findViewById(R.id.view_id)`
 - `garlicCheckBox = findViewById...`



MainActivity - onClickListener

- Set our Order button OnClickListener
- Check checkboxes ;)
- Update string



Improvements



TODO app

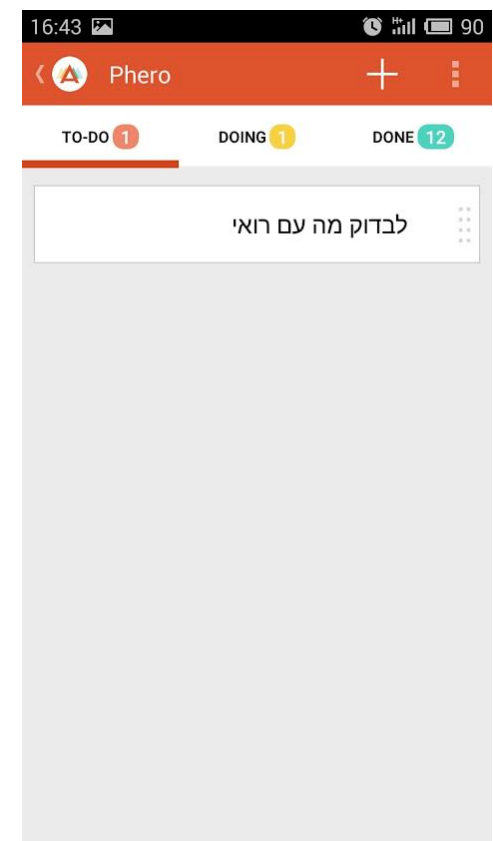
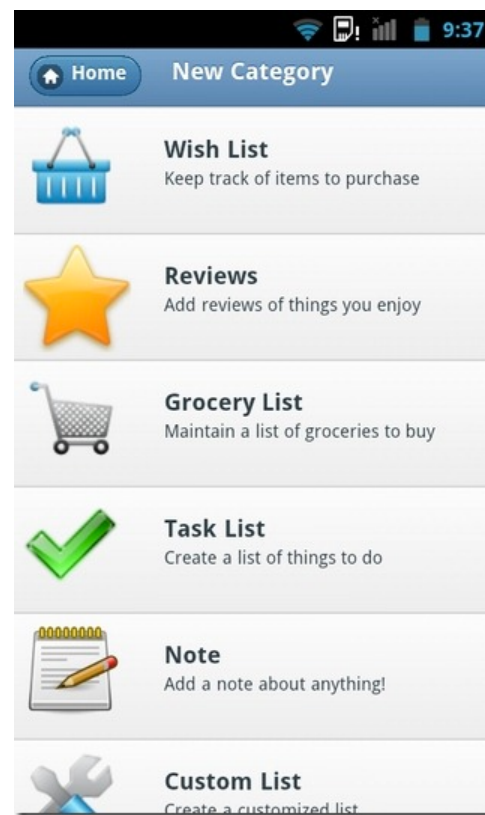
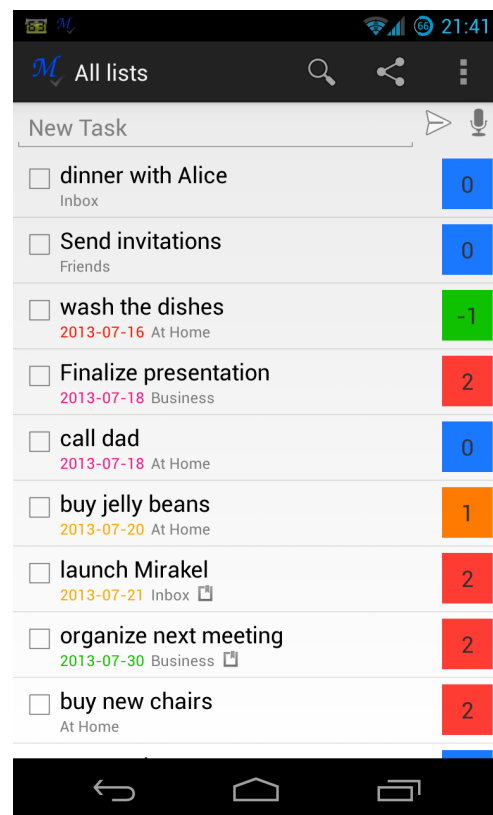


TODO lists are useful

- We're gonna learn more about Android UI
- Activities and intents
- Later: fragments, action bars, preferences, sql

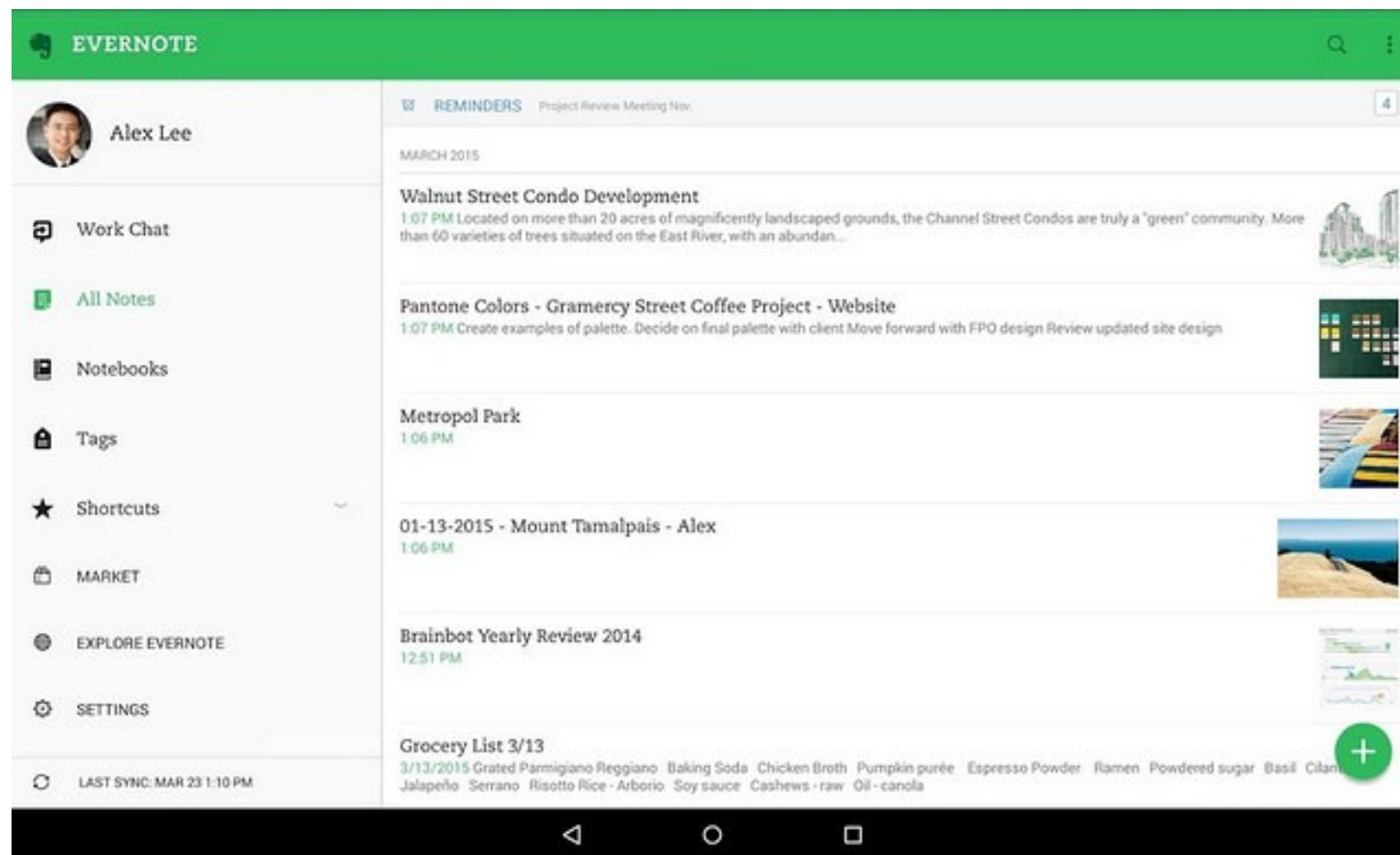


simple





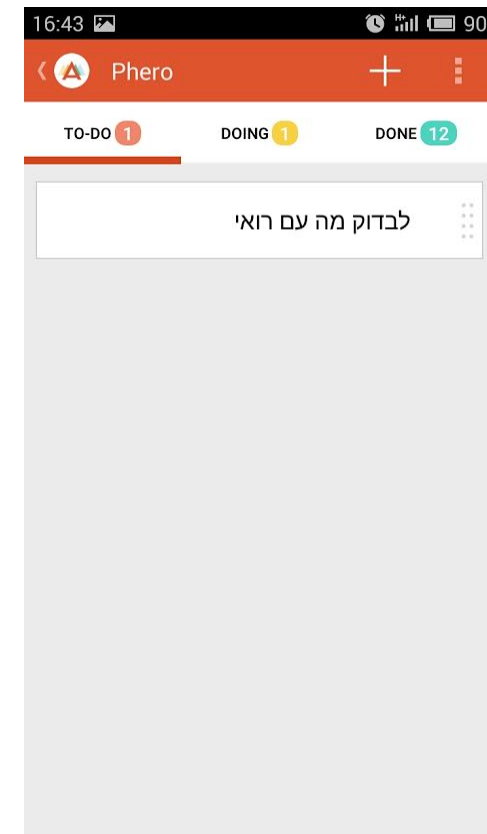
Complex





NoteListActivity

- Design the layout
 - ListView
- Add TODO item button
 - notify the adapter list changed
- Later: onClick() of item starts EditActivity





Adapter class as an Android design pattern

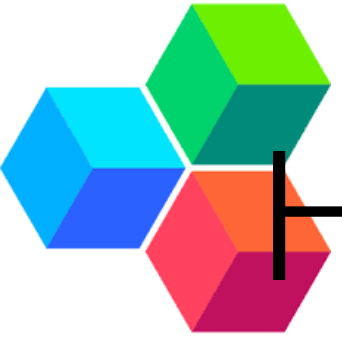
- The Adapter pattern allows two incompatible classes to work together by converting the interface of one class into an interface expected by the clients.





Coding time !

- Start with `NoteListActivity`



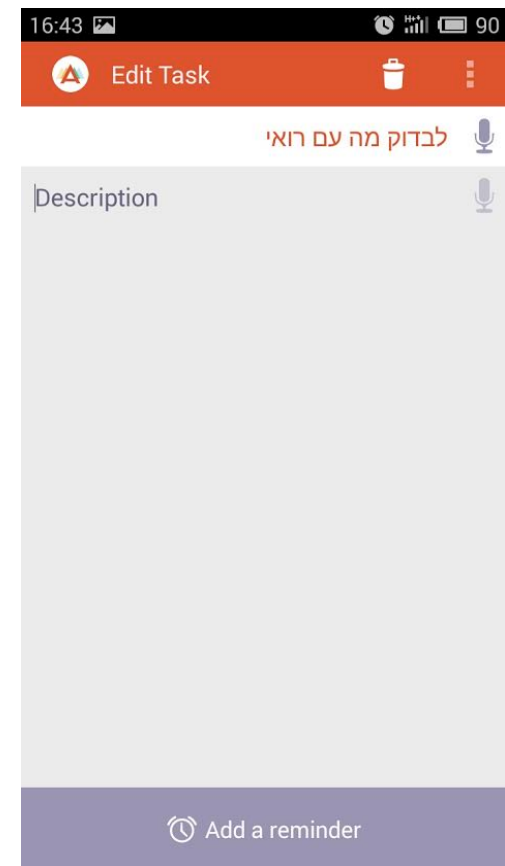
How do we add the content of the note ?

- We need another activity !



EditNoteActivity

- **Title** of note
- **Body** content
- Edit button





Starting another activity

- How do we move from screen to screen ?



Intent - special Android component

- The most common use of intents is to **bind** your application **components**. Intents are used to **start**, **stop** and transition the **activities** within an application.



Explicitly Starting New Activities

```
Intent intent = new Intent(MyActivity.this, MyOtherActivity.class);  
startActivity(intent);
```

- Of course we can add variables to this intent



Implicit Intents and Late Runtime Binding

```
if (somethingWeird && itDontLookGood) {  
    Intent intent = new Intent(Intent.ACTION_DIAL,  
                               Uri.parse("tel:555-2368"));  
    startActivity(intent);  
}
```



Returning results from activity

```
Intent result = new Intent(null, data);
result.putExtra(IS_INPUT_CORRECT, inputCorrect);
result.putExtra(SELECTED_PISTOL, selectedPistol);

setResult(RESULT_OK, result);

finish();
```

receive on the activity end

```
@Override
public void onActivityResult(int requestCode,
                             int resultCode,
                             Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    switch(requestCode) {
        case (SHOW_SUB_ACTIVITY_ONE) : {
            if (resultCode == Activity.RESULT_OK) {
                Uri horse = data.getData();
```



Practice

- Start EditNoteActivity with the intent as in the example
 - Explicit intent
 - startActivityForResult(intent)
 - add to it a test string “test” (putExtra)
- Add an edit box and a go back button and return the result back the NoteActivity
 - set OnClickListener for the go back button
 - init a new Intent
 - setResult(intent)
 - finish



What's left for the note?

- Long click on a note to delete
 - Dialog Interface
- Saving the data



Alert dialog

```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(  
    context);  
  
// set title  
alertDialogBuilder.setTitle("Your Title");  
  
// set dialog message  
alertDialogBuilder  
    .setMessage("Click yes to exit!")  
    .setCancelable(false)  
    .setPositiveButton("Yes",new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog,int id) {  
            // if this button is clicked, close  
            // current activity  
            MainActivity.this.finish();  
        }  
    })  
    .setNegativeButton("No",new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog,int id) {  
            // if this button is clicked, just close  
            // the dialog box and do nothing  
            dialog.cancel();  
        }  
    })  
    .create();  
  
// create alert dialog  
AlertDialog alertDialog = alertDialogBuilder.create();  
  
// show it  
alertDialog.show();
```




Practice

- Start a dialog on a long click of the item list view, it should ask either to delete it or not. On delete, delete.



Android Techniques for Saving Data

The data persistence techniques in Android provide options for balancing speed, efficiency, and robustness:

- ❑ **Shared Preferences** When storing the UI state, user preferences, or application settings, you want a lightweight mechanism to store a known set of values. Shared Preferences let you save groups of key/value pairs of primitive data as named preferences.
- ❑ **Files** It's not pretty, but sometimes writing to, and reading from, files directly is the only way to go. Android lets you create and load files on the device's internal or external media.
- ❑ **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has total control.
- ❑ **Content Providers** Rather than a storage mechanism in their own right, Content Providers let you expose a well-defined interface for using and sharing private data. You can control access to Content Providers using the standard permission system.



Creating and Saving Preferences

```
public static final String MYPREFS = "mySharedPreferences";

protected void savePreferences() {
    // Create or retrieve the shared preference object.
    int mode = Activity.MODE_PRIVATE;
    SharedPreferences mySharedPreferences = getSharedPreferences(MYPREFS,
                                                                mode);

    // Retrieve an editor to modify the shared preferences.
    SharedPreferences.Editor editor = mySharedPreferences.edit();

    // Store new primitive types in the shared preferences object.
    editor.putBoolean("isTrue", true);
    editor.putFloat("lastFloat", 1f);
    editor.putInt("wholeNumber", 2);
    editor.putLong("aNumber", 31);
    editor.putString("textEntryValue", "Not Empty");

    // Commit the changes.
    editor.commit();
}
```



Retrieving Shared Preferences

```
int mode = Activity.MODE_PRIVATE;
```

```
SharedPreferences mySharedPreferences = getSharedPreferences(MYPREFS,  
                                                         mode);
```

```
// Retrieve the saved values.
```

```
boolean isTrue = mySharedPreferences.getBoolean("isTrue", false);  
float lastFloat = mySharedPreferences.getFloat("lastFloat", 0f);  
int wholeNumber = mySharedPreferences.getInt("wholeNumber", 1);  
long aNumber = mySharedPreferences.getLong("aNumber", 0);  
String stringPreference;  
stringPreference = mySharedPreferences.getString("textEntryValue",  
                                                "");
```



דוגמה של שמירת נתונים

יצירה של אובייקט שאחראי על שמירה וטעינה של משתנים מהזכרון

```
SharedPreferences sharedPreferences;  
MyPREFERENCES = "grandpa"  
sharedPreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE)  
b1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String n = ed1.getText().toString();  
        String ph = ed2.getText().toString();  
        String e = ed3.getText().toString();  
        שמירה  
        SharedPreferences.Editor editor = sharedPreferences.edit();  
        editor.putString(Name, n);  
        editor.putString(Phone, ph);  
        editor.putString(Email, e);  
        editor.commit();  
        Toast.makeText(MainActivity.this, "Thanks", Toast.LENGTH_LONG).show();  
    }  
})
```

דוגמה של טעינת נתונים

```
int value = sharedPreferences.getString(Name, "");
```

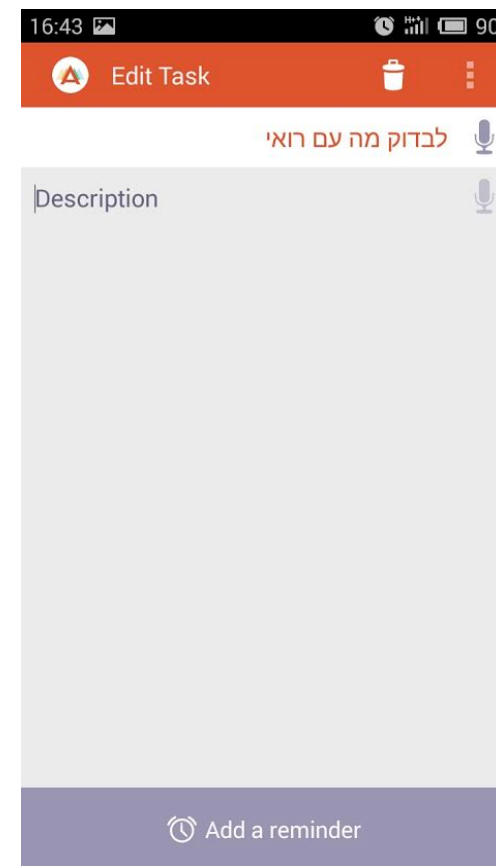
מפתח

ערך דיפולטי במידה ולא שמרנו שום דבר במפתח מסוים



Practice

- Create a shared preferences
- Save a string
- Save all title notes
- Save all note details





Homework