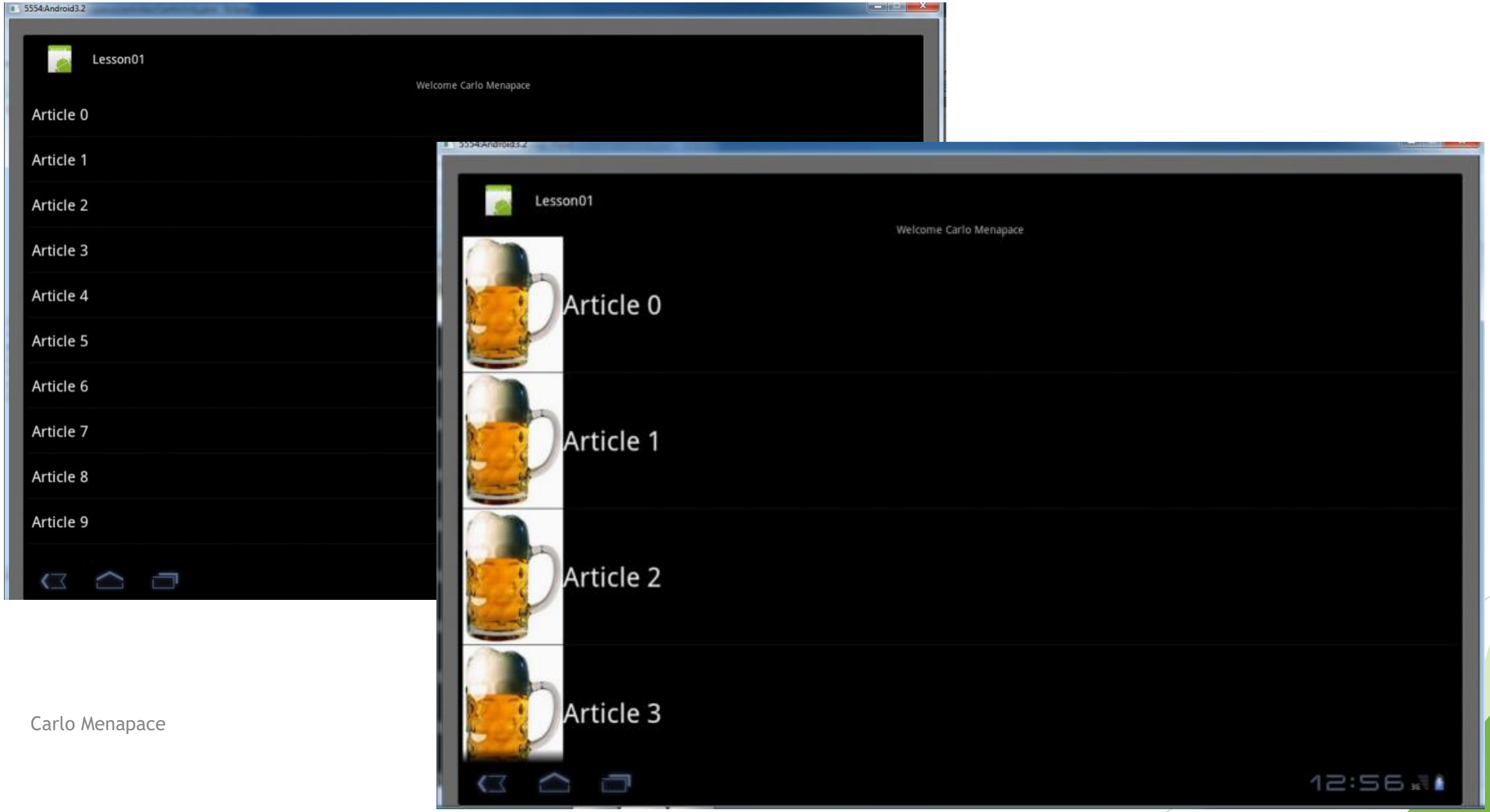
Programmazione di sistemi mobile e tablet

Android development

Carlo Menapace

carlo87@gmail.com

TODAY'S ROADMAP



5554:Android3.2

Lesson01

Welcome Carlo Menapace

Article 0

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Article 7

Article 8

Article 9

5554:Android3.2

Lesson01

Welcome Carlo Menapace

Article 0

Article 1

Article 2

Article 3

12:56

INTENT

You can start another activity by calling startActivity(), passing it an Intent that describes the activity you want to start. The intent specifies either the exact activity you want to start or describes the type of action you want to perform.

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

An intent can also carry a small amounts of data to be used by the new activity.

```
Intent.putExtra(parameterIdentifier, parameter);
```

Best paractices tell us that the parameter identifier **must** be composed by

PACKAGE NAME + OUR PARAMETER IDENTIFIER

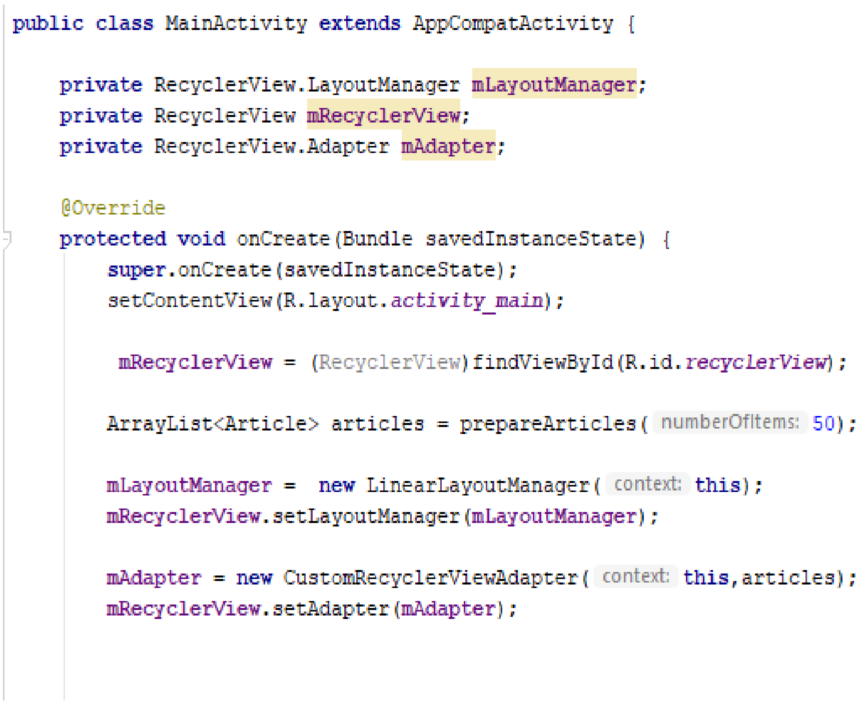
LISTVIEW – ArrayAdapter

```
list.setAdapter(new ItemsComplexAdapter (getApplicationContext(), R.layout.listRowLayout, listOfArticles));

private class ItemsComplexAdapter extends ArrayAdapter<Article> {

private ArrayList<Article> mArticles;
private Context  mContext = null;
private int mlistRowLayout;

public ItemsComplexAdapter(Context context, int layoutId, ArrayList<Article> articles) {
        super(context, layoutId, articles);
        mArticles = articles;
        mContext = context;
        mlistRowLayout = layoutId;

         }
@Override
public View getView(int position, View view, ViewGroup parent) {
        LayoutInflater vi = (LayoutInflater)mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        view = vi.inflate(mlistRowLayout , null);

        Article article = mArticles .get(position);
}}
```

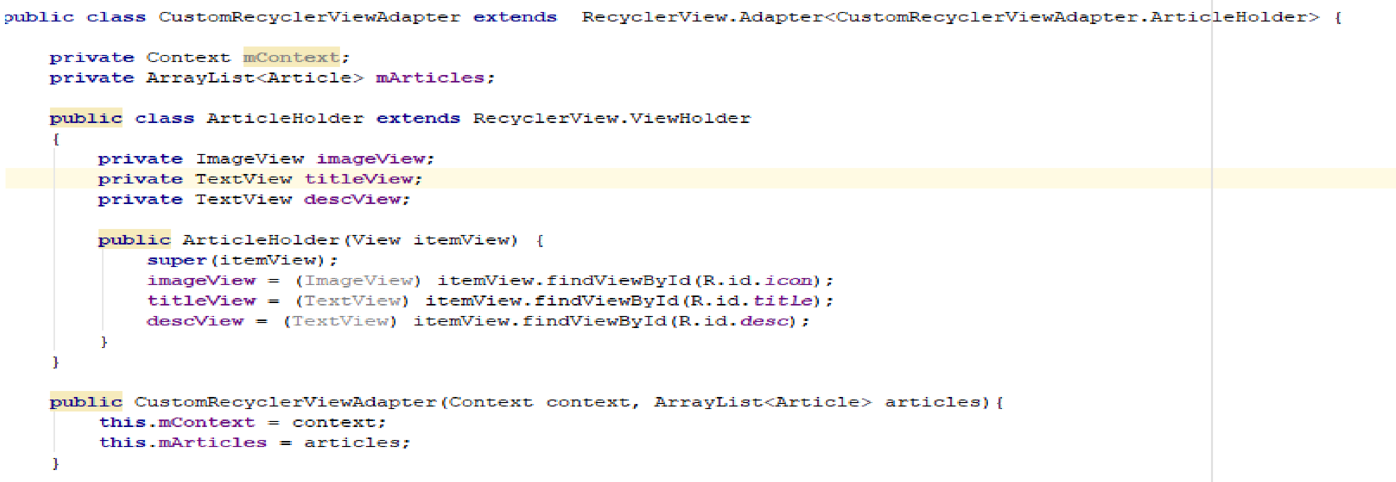
RECYCLERVIEW

Android development says "A flexible view for providing a limited window into a large data set."

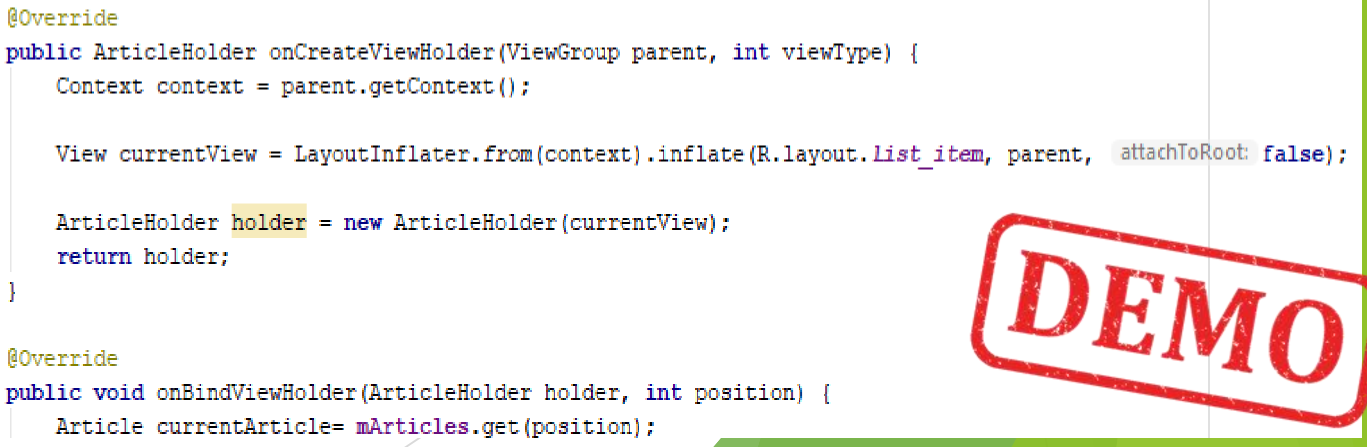
It is quite similar to an ArrayAdapter implementation but it provides a limited amounts of items based on the available space on the activity.



```java
public class MainActivity extends AppCompatActivity {

    private RecyclerView.LayoutManager mLayoutManager;
    private RecyclerView mRecyclerView;
    private RecyclerView.Adapter mAdapter;

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

         mRecyclerView = (RecyclerView)findViewById(R.id.recyclerView);

        ArrayList<Article> articles = prepareArticles( numberOfItems: 50);

        mLayoutManager =  new LinearLayoutManager( context: this);
        mRecyclerView.setLayoutManager(mLayoutManager);

        mAdapter = new CustomRecyclerViewAdapter( context: this,articles);
        mRecyclerView.setAdapter(mAdapter);
```



```java
public class CustomRecyclerViewAdapter extends  RecyclerView.Adapter<CustomRecyclerViewAdapter.ArticleHolder> {

    private Context mContext;
    private ArrayList<Article> mArticles;

    public class ArticleHolder extends RecyclerView.ViewHolder
    {
        private ImageView imageView;
        private TextView titleView;
        private TextView descView;

        public ArticleHolder(View itemView) {
            super(itemView);
            imageView = (ImageView) itemView.findViewById(R.id.icon);
            titleView = (TextView) itemView.findViewById(R.id.title);
            descView = (TextView) itemView.findViewById(R.id.desc);
        }
    }

    public CustomRecyclerViewAdapter(Context context, ArrayList<Article> articles){
        this.mContext = context;
        this.mArticles = articles;
    }
```



```java
@Override
public ArticleHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    Context context = parent.getContext();

    View currentView = LayoutInflater.from(context).inflate(R.layout.list_item, parent,  attachToRoot: false);

    ArticleHolder holder = new ArticleHolder(currentView);
    return holder;
}

@Override
public void onBindViewHolder(ArticleHolder holder, int position) {
    Article currentArticle= mArticles.get(position);
```

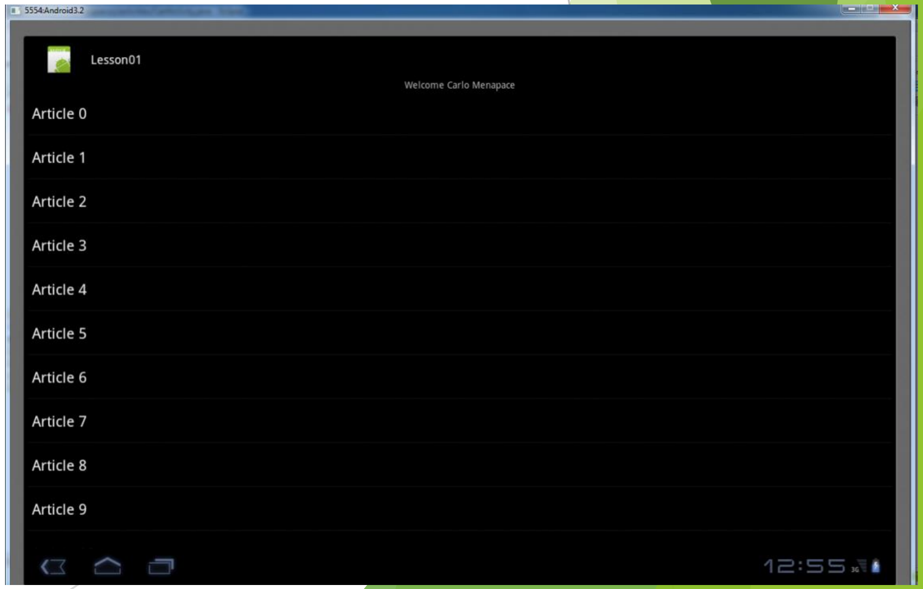
DEMO

TODAY'S ROADMAP

- HANDS ON!

  You have to create an Android Application composed by **2 Activities** that interact passing between them some parameters. The first activity (let's call it A) accepts in input a **String** and an **Integer.** When we start the second Activity (let's call it B) we are going to use those parameters in order to:

  - **Say Welcome ☺**
  - Create a list of **Items** with the given number.



Lesson01

Welcome Carlo Menapace

Article 0

Article 1

Article 2

Article 3

Article 4

Article 5

Article 6

Article 7

Article 8

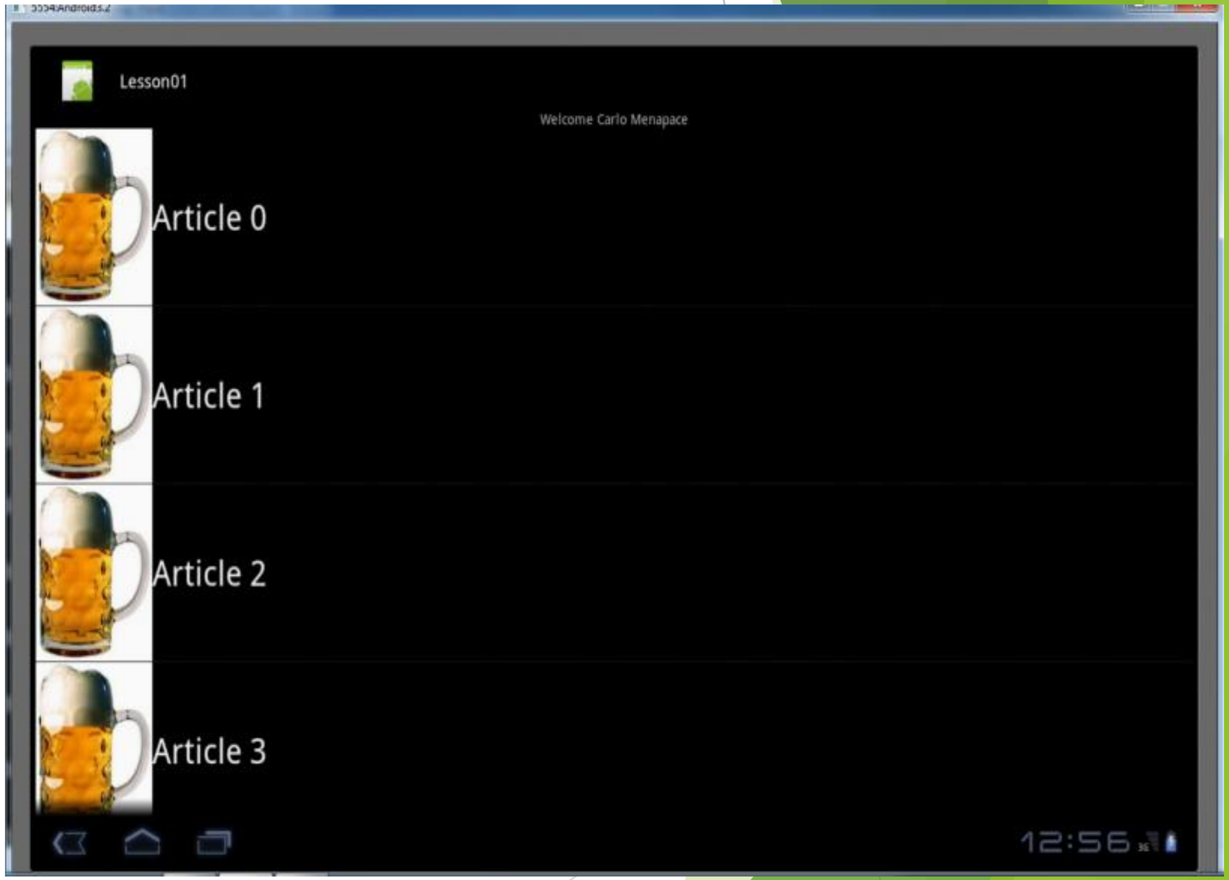
Article 9

ADVANCED EXERCISE

- HANDS ON +1

  Crate a simple scrollable list and show its content inside a listView

<ins>Suggestion</ins>: Create a class named Article with attributes image and description.



Lesson01

Welcome Carlo Menapace

Article 0

Article 1

Article 2

Article 3

12:56

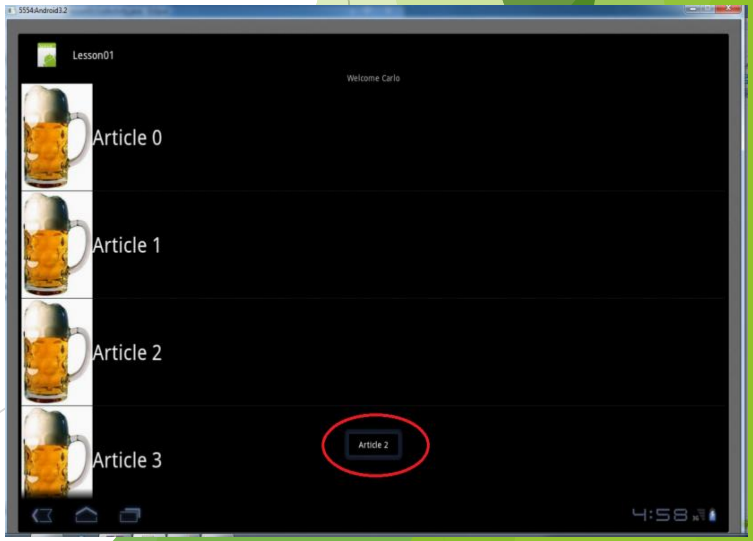
WOULD YOU LIKE A TOAST?

A toast notification is a message that pops up on the surface of the window. It only fills the amount of space required for the message and the user's current activity remains visible and interactive. The notification automatically fades in and out, and does not accept interaction events.

We can print on our device whatever (STRING) we want in this way:

```
Toast.makeText(getApplicationContext(), stringWeWantToPrint, Toast.LENGHT_LONG).show();
```

The result of a Toast is something like this:



5554:Android3.2

Lesson01

Welcome Carlo

Article 0

Article 1

Article 2

Article 3

Article 2

4:58