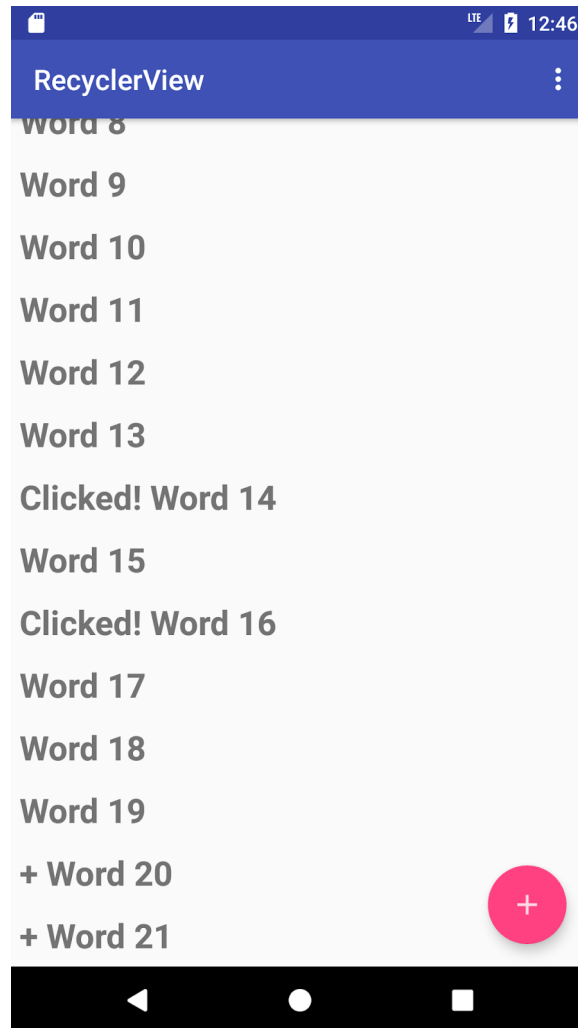


# GIỚI THIỆU ỨNG DỤNG

Trong bài này, chúng ta sẽ sử dụng RecyclerView để xây dựng ứng dụng **RecyclerView**: hiển thị một danh sách các từ (nhiều phần tử và có thể cuộn – scrollable).

Một số chức năng của ứng dụng:

- Nhấn vào 01 từ và đánh dấu đã click
- Nhấn vào Floating action button (FAB) để thêm 01 từ mới.



# Task 1: Tạo project và tập dữ liệu (dataset)

## 1.1. Tạo project and layout

1. Name: **RecyclerView**
2. Template: **Basic Activity**
3. Minimum API Level: 24 (Android 7.0)
4. Chạy ứng dụng.

## 1.2. Tạo tập dữ liệu

Trong bước này, chúng ta sẽ tạo một danh sách gồm 20 chuỗi (sử dụng LinkedList). Ví dụ: ["Word 1", "Word 2", "Word 3", .... ]

1. Mở tập tin **MainActivity.java** và thêm một biến private kiểu danh sách liên kết **mWordList**.

```
public class MainActivity extends AppCompatActivity {  
    private final LinkedList<String> mWordList = new LinkedList<>();  
    // ... Rest of MainActivity code ...  
}
```

2. Viết code tạo danh sách các chuỗi cho biến **mWordList** trong phương thức **onCreate()**:

```
// Thêm chuỗi vào danh sách. Chuỗi có dạng "Word" + <thứ tự>  
for (int i = 0; i < 20; i++) {  
    mWordList.addLast("Word " + i);  
}
```

## 1.3. Thay đổi icon của FAB

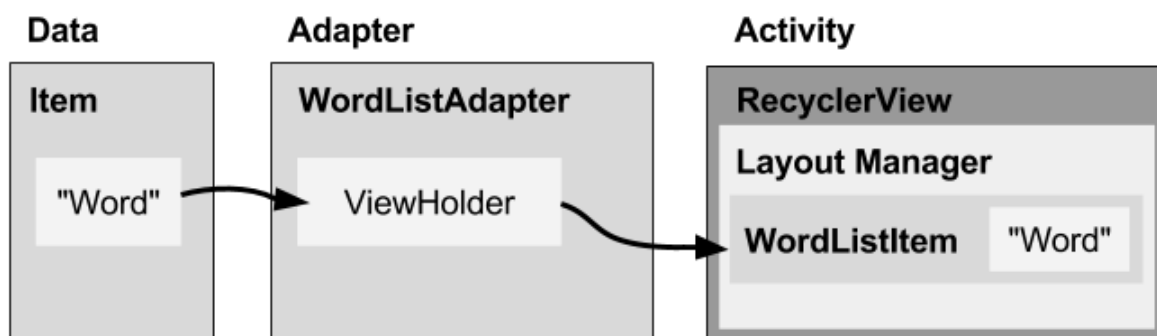
1. Nhấn chuột phải vào thư mục **drawable** (*res > drawable*)
2. Chọn **New > Image Asset**
3. Chọn **Action Bar and Tab Items** trong drop-down menu ở trên cùng của hộp thoại Configure Image Asset
4. Thay đổi **ic\_action\_name** trong ô **Name** thành **ic\_add\_for\_fab**.
5. Nhấn vào hình clip art để chọn hình (+)
6. Chọn **HOLO\_DARK** trong drop-down menu **Theme**. Nhấn **Next**.
7. Nhấn **Finish** trong hộp thoại Confirm Icon Path.

## Task 2: Tạo 01 RecyclerView

Trong task này, chúng ta sẽ hiển thị dữ liệu lên RecyclerView:

- Dữ liệu hiển thị: sử dụng biến `mWordList`
- 01 RecyclerView chứa một danh sách phần tử (scrolling list)
- [RecyclerView.LayoutManager](#) xử lý phân cấp và bố cục của các thành phần View. RecyclerView yêu cầu một layout manager tương ứng để quản lý việc sắp xếp danh sách phần tử. Bố cục có thể dọc - vertical, ngang - horizontal, hoặc lưới - grid
- [RecyclerView.Adapter](#) kết nối dữ liệu với RecyclerView. Chúng ta sẽ tạo một adapter để thêm mới, cập nhật các chuỗi trong các view
- Bên trong adapter, chúng ta sẽ tạo 01 ViewHolder chứa thông tin của View hiển thị 01 phần tử từ layout của phần tử

Sơ đồ bên dưới mô tả mối quan hệ giữa dữ liệu (data), adapter, ViewHolder, và layout manager



Các bước cài đặt:

1. Thêm 01 RecyclerView vào MainActivity XML layout (content\_main.xml)
2. Tạo 01 tập tin XML layout (wordlist\_item.xml) cho 01 phần tử (WordListItem) trong danh sách
3. Tạo 01 adapter (WordListAdapter) với 01 ViewHolder (WordViewHolder). Cài đặt phương thức để lấy dữ liệu và đưa vào ViewHolder
4. Trong phương thức onCreate() của MainActivity, tạo 01 RecyclerView và khởi tạo nó với adapter và 01 layout manager chuẩn.

## 2.1. Chỉnh sửa layout trong tập tin content\_main.xml

Thêm 01 RecyclerView vào XML layout:

1. Mở tập tin **content\_main.xml**
2. Chọn tab **Text**
3. Thay thế toàn bộ TextView element bằng đoạn XML sau:

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

## 2.2. Tạo layout cho 01 phần tử trong danh sách

*Adapter cần layout cho 01 phần tử trong danh sách. Tất cả các phần tử sẽ sử dụng chung layout. Chúng ta cần mô tả layout này trong một tập tin riêng biệt, bởi vì adapter sẽ sử dụng layout này.*

Tạo một layout đơn giản sử dụng 01 vertical LinearLayout với 01 TextView:

1. Nhấn chuột phải vào thư mục **app > res > layout** và chọn **New > Layout resource file**
2. Nhập **tên tập tin: wordlist\_item**, **Root element: LinearLayout** và nhấn OK
3. Mở tập tin **wordlist\_item.xml**, nhấn vào tab **Text**
4. Thiết lập LinearLayout với các thuộc tính sau:

LinearLayout attribute	Value
<b>android:layout_width</b>	"match_parent"
<b>android:layout_height</b>	"wrap_content"
<b>android:orientation</b>	"vertical"
<b>android:padding</b>	"6dp"

5. Thêm 01 TextView để hiển thị chuỗi vào LinearLayout với các thuộc tính sau:

Attribute	Value
<b>android:id</b>	"@+id/word"
<b>android:layout_width</b>	"match_parent"
<b>android:layout_height</b>	"wrap_content"
<b>android:textSize</b>	"24sp"
<b>android:textStyle</b>	"bold"

## 2.3. Tạo 01 style từ các thuộc tính của TextView

Việc sử dụng các style để cho phép các thành phần chia sẻ các nhóm các thuộc tính hiển thị. Trích xuất thông tin style cho `TextView` `word` trong tập tin `wordlist_item.xml`:

1. Mở tập tin **wordlist\_item.xml**
2. Nhấn chuột phải vào `TextView` trong `wordlist_item.xml`, và chọn **Refactor > Extract > Style**. Hộp thoại **Extract Android Style** xuất hiện
3. Nhập tên style là **word\_title** và chọn tất cả các tùy chọn khác. Chọn vào tùy chọn **Launch 'Use Style Where Possible'** option. Sau đó nhấn **OK**
4. Khi được hỏi, chọn áp dụng style cho toàn project (**Whole Project**)
5. `word_title` style được định nghĩa trong tập tin **values > styles.xml**.
6. Thêm thuộc tính style vào `TextView` trong tập tin **wordlist\_item.xml** như bên dưới:

```
<TextView
    android:id="@+id/word"
    style="@style/word_title" />
```

## 2.4. Tạo 01 adapter

Android sử dụng các adapter (`Adapter` class) để kết nối dữ liệu với các phần tử `View` trong một danh sách.

Adapter sử dụng 01 `ViewHolder` để mô tả 01 phần tử `View` và vị trí của nó trong `RecyclerView`.

Các bước tạo 01 adapter làm cầu nối giữa dữ liệu trong danh sách các chuỗi và `RecyclerView`:

1. Nhấn chuột phải vào **java/<tên package>** và chọn **New > Java Class**.
2. Tên class: **WordListAdapter**.
3. Lớp `WordListAdapter` kế thừa một `RecyclerView` adapter tổng quát để sử dụng một `View` holder được định nghĩa bên trong lớp `WordListAdapter` (*Lớp `WordViewHolder` chưa được định nghĩa nên sẽ hiển thị lỗi*)

```
public class WordListAdapter extends
    RecyclerView.Adapter<WordListAdapter.WordViewHolder> {
}
```

4. Cài đặt tất cả phương thức của lớp **WordListAdapter** (**ALT + ENTER** → *Implement Methods...*)

## 2.5. Tạo ViewHolder cho adapter

1. Trong lớp **WordListAdapter**, thêm mới 01 lớp **WordViewHolder**:

```
class WordViewHolder extends RecyclerView.ViewHolder {  
}
```

2. Thêm các biến vào lớp **WordViewHolder** cho **TextView** và adapter:

```
public final TextView wordItemView;  
final WordListAdapter mAdapter;
```

3. Trong lớp **WordViewHolder**, thêm 01 phương thức khởi tạo (*khởi tạo ViewHolder TextView từ word XML resource, và thiết lập adapter*):

```
public WordViewHolder(View itemView, WordListAdapter adapter) {  
    super(itemView);  
    wordItemView = itemView.findViewById(R.id.word);  
    this.mAdapter = adapter;  
}
```

4. Chạy ứng dụng để đảm bảo không có lỗi nào trong quá trình viết code.

## 2.6. Lưu trữ dữ liệu trong adapter

1. Tạo một biến danh sách liên kết các chuỗi **mWordList** trong lớp **WordListAdapter** để lưu trữ dữ liệu

```
private final LinkedList<String> mWordList;
```

2. Ghi đè phương thức **getItemCount()** trả về kích thước của **mWordList**:

```
@Override  
public int getItemCount() {  
    return mWordList.size();  
}
```

3. **WordListAdapter** cần 01 constructor để khởi tạo danh sách chuỗi từ dữ liệu. Để tạo 01 View cho 01 phần tử trong danh sách thì **WordListAdapter** cần nạp (inflate) XML layout của 01 phần tử trong danh sách → sử dụng **LayoutInflater** để đọc 01 mô tả XML layout và chuyển nó thành các phần tử View tương ứng.

Tạo một biến cho bộ nạp (inflator) trong **WordListAdapter**:

```
private LayoutInflater mInflater;
```

4. Cài đặt phương thức constructor của **WordListAdapter**:

```
public WordListAdapter(Context context, LinkedList<String> wordList) {  
    mInflater = LayoutInflater.from(context);  
    this.mWordList = wordList;  
}
```

5. Cài đặt phương thức `onCreateViewHolder()`:

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.inflate(R.layout.wordlist_item, parent, false);
    return new ViewHolder(itemView, this);
}
```

6. Cài đặt phương thức `onBindViewHolder()` (kết nối dữ liệu và view holder):

```
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    String mCurrent = mWordList.get(position);
    holder.wordItemView.setText(mCurrent);
}
```

7. Chạy ứng dụng để đảm bảo không có lỗi nào trong quá trình viết code.

## 2.7. Tạo RecyclerView trong Activity

1. Mở tập tin **MainActivity.java**

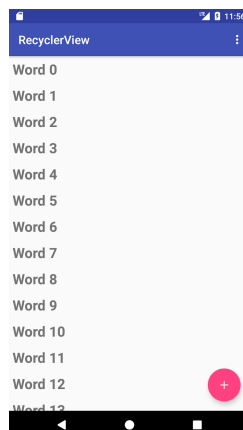
2. Thêm các biến thành viên cho **RecyclerView** và adapter.

```
private RecyclerView mRecyclerView;
private WordListAdapter mAdapter;
```

3. Trong phương thức `onCreate()` của **MainActivity**, viết mã nguồn để tạo **RecyclerView** kết nối với 01 adapter và dữ liệu. (Viết đoạn này sau khi khởi tạo **mWordList**)

```
// Lấy tham chiếu đến RecyclerView.
mRecyclerView = findViewById(R.id.recyclerview);
// Tạo 01 adapter và cung cấp dữ liệu để hiển thị.
mAdapter = new WordListAdapter(this, mWordList);
// Kết nối adapter với RecyclerView.
mRecyclerView.setAdapter(mAdapter);
// Thiết lập 01 layout manager mặc định cho RecyclerView
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

4. Chạy ứng dụng



## Task 3: Tạo tương tác với danh sách

Danh sách phần tử trong 01 `RecyclerView` có thể được chỉnh sửa một cách dynamic.

### 3.1. Cài đặt cho các phần tử phản hồi khi được click vào

1. Mở tập tin **WordListAdapter.java**.
2. Cài đặt thêm interface `View.OnClickListener` vào lớp `WordViewHolder`

```
class WordViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
}
```

3. Ghi đè phương thức `onClick()` của interface `View.OnClickListener`

```
// Lấy vị trí của phần tử được click vào  
int mPosition = getLayoutPosition();  
// Truy cập vào phần tử trong mWordList.  
String element = mWordList.get(mPosition);  
// Thay đổi nội dung phần tử trong mWordList (Thêm chuỗi Clicked vào trước)  
mWordList.set(mPosition, "Clicked! " + element);  
// Thông báo cho adapter biết dữ liệu đã được thay đổi để cập nhật lại dữ liệu hiển thị trên  
// RecyclerView  
mAdapter.notifyDataSetChanged();
```

5. Thêm dòng code bên dưới vào phương thức constructor của `WordViewHolder` để kết nối `onClickListener` với `View` (sau dòng `this.mAdapter = adapter`):

```
itemView.setOnClickListener(this);
```

6. Chạy ứng dụng và click vào các phần tử để xem thay đổi.

### 3.2. Cài đặt xử lý cho FAB

Trong bước này, chúng ta sẽ cài đặt xử lý cho FAB để:

- Thêm 1 chuỗi vào cuối danh sách
- Thông báo cho adapter biết dữ liệu đã được thay đổi
- Cuộn đến phần tử vừa được thêm vào

Thực hiện theo các bước sau:

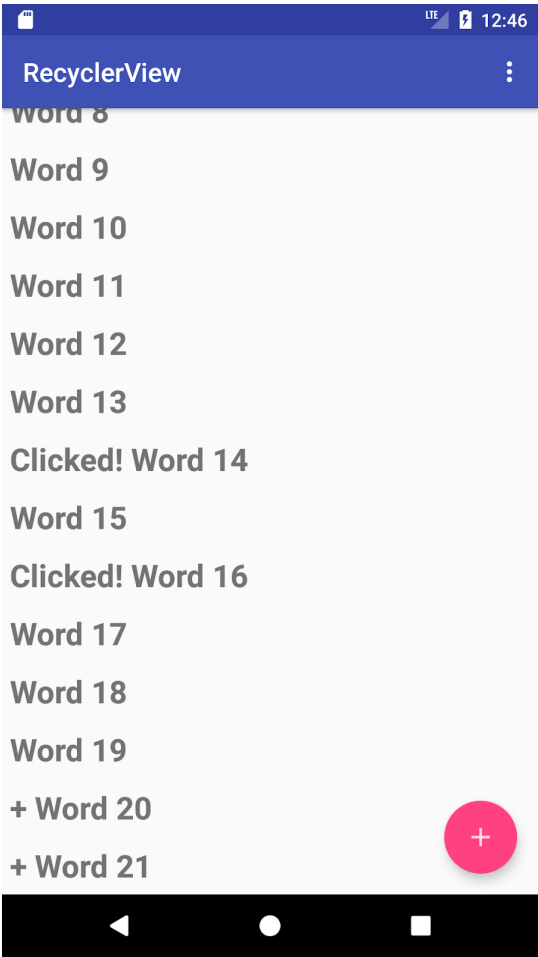
1. Cài đặt phương thức xử lý sự kiện click `onClick()` của `FloatingActionButton` trong tập tin **MainActivity.java**:

```
@Override  
public void onClick(View view) {  
    int wordListSize = mWordList.size();
```



```
// Thêm 01 từ mới vào danh sách (định dạng "+ Word" + <thứ tự>)
mWordList.addLast("+ Word " + wordListSize);
// Thông báo cho adapter biết dữ liệu đã được thay đổi
mRecyclerView.getAdapter().notifyItemInserted(wordListSize);
// Cuộn đến cuối danh sách.
mRecyclerView.smoothScrollToPosition(wordListSize);
}
```

- 2. Chạy ứng dụng.
- 3. Nhấn vào FAB để thêm các phần tử.



*Lưu ý: khi xoay ngang màn hình thì điều gì sẽ xảy ra? Hãy tìm cách khắc phục vấn đề này.*