Create Drw Res File(giving round back+color to text view)

Drawable>new>DrawableResFile

Change root element from selector to shape(xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke android:width="4dp" android:color="@color/teal_700"/>
    <solid android:color="@color/cardview_dark_background"/>
    <corners android:radius="32dp"/>
</shape>
```

Then apply background.
android:background="@drawable/back"

Adding img to both sides via Code

```
Drawable img= ResourcesCompat.getDrawable(getResources()
        ,R.drawable.border,getTheme());
Objects.requireNonNull(img);
img.setBounds(0,0,img.getIntrinsicWidth(),img.getIntrinsicHeigh
txtViewWelcomeMsg.setCompoundDrawables(img,null,img,null);
txtViewWelcomeMsg.setCompoundDrawablePadding(8);
```

Changing VISIBILITY And BtnTEXT

```
if(btnShowTextOrImage.getText().equals("Show Text")){
    txtViewWelcomeMsg.setVisibility(View.VISIBLE);
    imgViewSample.setVisibility(View.INVISIBLE);
    btnShowTextOrImage.setText(getResources().
            getText(R.string.txtShowImage));
}
```

Touch Listener

1.

```
public class CustomTouchListener implements View.OnTouchListener {
    GestureDetectorCompat gestureDetectorCompat;
    Context context;

    public class CustomGestureListener extends
    GestureDetector.SimpleOnGestureListener{

    }
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        return gestureDetectorCompat.onTouchEvent(motionEvent);//imp
    }
}
```

2.over ride methods inside CustomGestureListener class(always select onDown method)

3.Constructor outerclass

```
public CustomTouchListener(Context context) {
    this.context = context;
    gestureDetectorCompat = new GestureDetectorCompat(context,
            new CustomGestureListener());
}
```

4.On down return true

```
@Override
public boolean onDown(MotionEvent e) {
    return true;//should be true for activity to occur
}
```

5.Add methods to call(use 2nd option to create function def and change onDoubleClick ->public)

```
@Override
public boolean onDoubleTap(MotionEvent e) {
    onDoubleClick();
    return super.onDoubleTap(e);
}
```

```
@Override
public boolean onFling(MotionEvent e1, MotionEvent e2,
                       float velocityX, float velocityY) {
    final int SWIPE_DIST_THRESHOLD=10;
    final int SWIPE_VEL_THRESHOLD=50;
    float distX=e2.getX()-e1.getX();
    float distY=e2.getY()-e1.getY();

    if(Math.abs(distX)>Math.abs(distY)&&
            Math.abs(distX)>SWIPE_DIST_THRESHOLD
    && Math.abs(velocityX)>SWIPE_VEL_THRESHOLD){
        //horizontal swipe
        if(distX>0){onRightSwipe();}
        else{onLeftSwipe();}
    }
    else if(Math.abs(distY)>Math.abs(distX)&&
            Math.abs(distY)>SWIPE_DIST_THRESHOLD
            && Math.abs(velocityY)>SWIPE_VEL_THRESHOLD){
        //vert swipe
        if(distY>0){onDownSwipe();}
        else{onUpSwipe();}
    }
    return super.onFling(e1, e2, velocityX, velocityY);
}
```

6.Inside Main Activity(Override methods after this step)

```
txtViewWelcomeMsg.setOnTouchListener(
        new CustomTouchListener(MainActivity.this){
});
```

7.Swipe Right and Left, Top/Bottom use horz_grav_mark

```
@Override
public void onLeftSwipe() {
    super.onLeftSwipe();
    int vertGravity=txtViewWelcomeMsg.getGravity()&
            Gravity.VERTICAL_GRAVITY_MASK;
    txtViewWelcomeMsg.setGravity(vertGravity |Gravity.LEFT );
}
```

8.For Strike Through

```
@Override
public void onLongClick() {
    super.onLongClick();
    txtViewWelcomeMsg.setPaintFlags(
            txtViewWelcomeMsg.getPaintFlags() ^
                    Paint.STRIKE_THRU_TEXT_FLAG
    );
}
```

9.Making Text Bigger

```
if(!bigger){
    txtViewWelcomeMsg.setTextSize(
            txtViewWelcomeMsg.getTextSize()/
                    getResources().getDisplayMetrics().density+10
    );
}
```

10.Change Txt Color

```
if (txtViewWelcomeMsg.getCurrentTextColor() !=
        ResourcesCompat.getColor(getResources(),R.color.teal_200,getTheme())){
    txtViewWelcomeMsg.setTextColor(
            ResourcesCompat.getColor(getResources(),R.color.teal_200,getTheme()));
} else { txtViewWelcomeMsg.setTextColor(Color.rgb(255,255,255));}
```

11.Changing Image Scale

```
if(imgViewSample.getScaleType()!= ImageView.ScaleType.FIT_XY){
    imgViewSample.setScaleType(ImageView.ScaleType.FIT_XY);
}
else
    imgViewSample.setScaleType(ImageView.ScaleType.FIT_CENTER);
```

12.Changing image

```
Drawable birdImage = ResourcesCompat.getDrawable(
        getResources(),R.drawable.bird,getTheme());
Objects.requireNonNull(birdImage);
if(imgViewSample.getDrawable().getConstantState() !=
        birdImage.getConstantState())
    imgViewSample.setImageResource(R.drawable.bird);
else
    imgViewSample.setImageResource(R.drawable.fire);
```

Recycler View Demo

Tablayout

```
tuneTabs.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override
    public void onTabSelected(TabLayout.Tab tab) {
        Toast.makeText(MainActivity.this, "Sel "+tuneTabs.getSelectedTabPosition(), Toast.LENGTH_SHORT).show();
    }
```

Adding values to arrays

```
List<String> TuneNames =
        new ArrayList<>(Arrays.asList("B
```

Sublisitng Array

```
movieTunes = allTunes.subList(0,3);//3 is excluded
tVTunes = allTunes.subList(3,5);
```

RecyclerView Creation

1. **Create adapter class**
2. **Create inner class extends viewHolder**

```
public class TuneViewHolder extends RecyclerView.ViewHolder{
    ImageView imageViewTune;
    TextView textViewTune;
    View tuneItemView;

    public TuneViewHolder(@NonNull View itemView) {
        super(itemView);
    }
}
```

3. **Extend outer class with inner class and implement methods**

```
public class TuneAdapter extends RecyclerView.Adapter
        <TuneAdapter.TuneViewHolder> {
```

4. **Specify the data list and create constr , getters and setters**

```
List<Tune> tuneList;
public TuneAdapter(List<Tune> tuneList) {
    this.tuneList = tuneList;
}
public List<Tune> getTuneList() {
    return tuneList;
}
public void setTuneList(List<Tune> tuneList) {
    this.tuneList = tuneList;
}
```

5.OnCreate

```
public TuneViewHolder onCreateViewHolder
        (@NonNull ViewGroup parent, int viewType) {//inflate
    View itemView= LayoutInflater.from(parent.getContext()).
            inflate(R.layout.layout_tune,parent,false);
    //create viewHolder Obj
    TuneViewHolder viewHolder = new TuneViewHolder(itemView);

    viewHolder.imageViewTune = itemView.findViewById(R.id.imgViewTune);
    viewHolder.textViewTune = itemView.findViewById(R.id.txtViewTune);
    viewHolder.tuneItemView = itemView;

    return viewHolder;
}
```

6.OnBind

```
@Override
public void onBindViewHolder(@NonNull TuneViewHolder holder, int position) {
    holder.imageViewTune.setImageResource(tuneList.get(position).getTunePic());
    holder.textViewTune.setText(tuneList.get(position).getTuneName());

}
```

7. setting grid layout and adapter in ManACtivity

```
TuneAdapter tuneAdapter = new TuneAdapter(allTunes);//deflt alltunes
GridLayoutManager gridLayoutManager = new GridLayoutManager(this,2);
recyclerViewTunes.setLayoutManager(gridLayoutManager);
recyclerViewTunes.setAdapter(tuneAdapter);
```

8.changing data on each Tabs(addNotifyDataset methd in setter)

```
public void onTabSelected(TabLayout.Tab tab) {
    switch (tuneTabs.getSelectedTabPosition()){
        case 0:{
            tuneAdapter.setTuneList(allTunes);
        }
        break;
```

9.Changing color of itemView when clicked(insude createvwHldr)

```
viewHolder.tuneItemView.setOnClickListener((View view)-> {
    if(viewHolder.tuneItemView.getBackground() instanceof ColorDrawable
    && ((ColorDrawable) viewHolder.tuneItemView.getBackground()).getColor()
            != Color.LTGRAY)
        viewHolder.tuneItemView.setBackgroundColor(Color.LTGRAY);
    else
        viewHolder.tuneItemView.setBackgroundColor(
                Color.parseColor("#FAFAFA"));
```

10.To reset dirty views add this to onBind

```
holder.tuneItemView.setBackgroundColor(Color.parseColor("#FAFAFA"));
```

Music Player

1.In adaper

```
int currentPlayIndex = -1;
```

2.In bind

```
if(position==currentPlayIndex)
    holder.imgViewPlayPause.setImageResource(R.drawable.pause);
else
    holder.imgViewPlayPause.setImageResource(R.drawable.play);

holder.imgViewPlayPause.setOnClickListener((View view)-> {
    if(position==currentPlayIndex)
        currentPlayIndex=-1;
    else
        currentPlayIndex=holder.getAdapterPosition();
    notifyDataSetChanged();
});
```

3. Linear Layout

```
TuneAdapter2 tuneAdapter=new TuneAdapter2(allTunes);
LinearLayoutManager layoutManager = new LinearLayoutManager(this);
recyclerViewTunes.setLayoutManager(layoutManager);
recyclerViewTunes.setAdapter(tuneAdapter);
```

4.Prevent dirty views

```
public void setTuneList(List<Tune> tuneList) {
    this.tuneList = tuneList;
    currentPlayIndex = -1;
    notifyDataSetChanged();
}
```

Binding

```
binding = ActivityMainBinding.inflate(getLayoutInflater());
```

Menu Functions(xml->res->menu)

```
public boolean onOptionsItemSelected(MenuItem item) {
```

Give include an id(mainContent)+ layout an id(mainLayout)

Setting init background and button Color

```
ConstraintLayout mainLayout = binding.mainContent.mainLayout;
mainLayout.setBackgroundColor(Color.parseColor("#FAFAFA"));
binding.fab.setBackgroundTintList(ColorStateList.valueOf(Color.LTGRAY));

binding.fab.setOnClickListener((View view)-> {
    ColorDrawable colorDrawable
            = (ColorDrawable) mainLayout.getBackground();
    int colorId = colorDrawable.getColor();//need the init color setup
    if(colorId != Color.LTGRAY)
    {
        mainLayout.setBackgroundColor(Color.LTGRAY);
        binding.fab.setBackgroundTintList(
                ColorStateList.valueOf(Color.parseColor("#FAFAFA")));
    }
```

SnackBar

```
Snackbar.make(view,"ShowText",Snackbar.LENGTH_LONG)
                .setAction("Button Text",(View v)->{}).show();
```

Defining View Model

```
public class ColorSpecViewModel extends ViewModel {
    MutableLiveData<List<ColorSpec>> ColorList = new MutableLiveData<>();

    public LiveData<List<ColorSpec>> getColors(){
        if(ColorList==null){
            ColorList=new MutableLiveData<>();}
        return ColorList;
    }
    public void loadColors(List<ColorSpec> colorList){
        ColorList.setValue(colorList);
//        Handler handler = new Handler();
//        handler.postDelayed(()-> {
//            ColorList.setValue(colorList);
//        },1000);
    }
}
```

Inside Main Activity(loading data to List)

```
List<ColorSpec> colorSpecs = new ArrayList<>();
List<String> colorDesc = new ArrayList<>(
        Arrays.asList("BLACK","ORANGE","PURPLE"));
List<Integer> colorVals = new ArrayList<>(
        Arrays.asList(Color.BLACK,Color.rgb(255,165,0),R.color.purple_700)
);
for(int i=0;i<colorDesc.size();i++){
    ColorSpec colorSpec = new ColorSpec
            (colorDesc.get(i),colorVals.get(i));
    colorSpecs.add(colorSpec);
}
```

Inside Main (Laoding data to view Model)

```
colorSpecViewModel = new ViewModelProvider(this).get(ColorSpecViewModel.class);
colorSpecViewModel.loadColors(colorSpecs);
```

CustomAdapter-COlorList

1.new->xml->Layoutxml,then(layoutcoloritem),create constr

```
public class ColorSpecAdapter extends BaseAdapter {
    List<ColorSpec> colorList;
    LayoutColoritemBinding binding;
```

implement the methods,then

```
@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    if(view==null){
        binding = LayoutColoritemBinding.inflate(
                LayoutInflater.from(viewGroup.getContext()),
                viewGroup,false);
    }
    binding.textViewColorItem.setText(colorList.get(i).getColorDesc());
    binding.textViewColorItem.setTextColor(
            colorList.get(i).getColorVal());
    return binding.getRoot();
}
```

Setting up Adapter from Fragment1

```
List<ColorSpec> fragColors = new ArrayList<>();
```

then,inside onViewCreated

```
ColorSpecViewModel colorSpecViewModel = new ViewModelProvider(
        requireActivity()).get(ColorSpecViewModel.class);

colorSpecViewModel.getColors().observe(requireActivity(),
        new Observer<List<ColorSpec>>() {
    @Override
    public void onChanged(List<ColorSpec> colorSpecList) {
        fragColors = colorSpecList;
        ColorSpecAdapter adapter = new ColorSpecAdapter(fragColors);

        binding.spinnerColors.setAdapter(adapter);
    }
});
```

Passing colorData using Bundle(button Onclick)

```
Bundle bundle=new Bundle();
bundle.putInt("COLORVAL",fragColors.get(
        binding.spinnerColors.getSelectedItemPosition()).getColorVal());
NavHostFragment.findNavController(FirstFragment.this)
        .navigate(R.id.action_FirstFragment_to_SecondFragment,bundle);
```

Getting and setting the color(2nd frag->inside onViewCreated)

```
if(getArguments() !=null){
    int colorVal = getArguments().getInt("COLORVAL", Color.BLACK);
    binding.textviewSecond.setTextColor(colorVal);
}
```

Adding a csv file

Res->right click->new->androidres direc->res type=raw

Read from file,make list,parse Data

```
private void ReadDogData() {
    DogList = new ArrayList<>();

    //FileRead
    InputStream inputStream = getResources().openRawResource(R.raw.doginfo);
    BufferedReader reader = new BufferedReader(
            new InputStreamReader(inputStream));
    try {

    String line;
    while((line = reader.readLine())!=null){
    String[] eachWord = line.split(",");
    int id = Integer.parseInt(eachWord[0]);
    String picName = eachWord[1];
    String dogBreed = eachWord[2];
    String dogName = eachWord[3];
    String dogDOBStr = eachWord[4];

    int dogDrawable = getResources().getIdentifier(picName,
            "drawable",getPackageName());

        DateTimeFormatter formatter=DateTimeFormatter.ofPattern("d-MMM-yyyy");
        LocalDate dob = LocalDate.parse(dogDOBStr,formatter);

        Dog eachDog = new Dog(id,dogBreed,dogName,dogDrawable,dob);
        DogList.add(eachDog);
    }
```

Recylcer View2.

```
public class DogViewHolder extends RecyclerView.ViewHolder {

    LayoutDogitemBinding binding;

    public DogViewHolder(LayoutDogitemBinding bindingP) {
        super(bindingP.getRoot());
        binding = bindingP;
    }
}
```

2.

```
public class DogAdapter extends RecyclerView.Adapter<DogAdapter.DogViewHolder> {

    List<Dog> AdapterDogData;

    public DogAdapter(List<Dog> adapterDogData) {
        AdapterDogData = adapterDogData;
    }
```

3.Implement Methods

```
    @NonNull
    @Override
    public DogViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        LayoutInflater inflater = LayoutInflater.from(parent.getContext());
        LayoutDogitemBinding binding = LayoutDogitemBinding.inflate
                (inflater,parent,false);

        DogViewHolder dogViewHolder = new DogViewHolder(binding);
//        dogViewHolder.binding.txtViewDOB.setOnClickListener((View view)-> {
//
//        });
        dogViewHolder.binding.txtViewName.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(context.getApplicationContext(),
                ""+AdapterDogData.get(dogViewHolder.getAdapterPosition()).getDogName(),
                Toast.LENGTH_SHORT).show();
            }
        });
        return dogViewHolder;
    }
```

```
@RequiresApi(api = Build.VERSION_CODES.O)
@Override
public void onBindViewHolder(@NonNull DogViewHolder holder, int position) {
holder.binding.txtViewId.setText(String.valueOf(
        AdapterDogData.get(position).getId()    ));
holder.binding.txtViewBreed.setText(AdapterDogData.get(position).getDogBreed());
holder.binding.txtViewName.setText(AdapterDogData.get(position).getDogName());
holder.binding.imgViewDogPic.setImageResource(
        AdapterDogData.get(position).getDogPicDrawable()    );

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
holder.binding.txtViewDOB.setText(
        formatter.format(AdapterDogData.get(position).getDob())    );

}
```

Main Activity

```
binding.recyclerViewDogItems
        .setLayoutManager(new LinearLayoutManager(this));
binding.recyclerViewDogItems
        .setAdapter(new DogAdapter(DogList));
```