Pause:

It pauses the activity for whatever time u want
Usually used in splash screens

```
new Handler().postDelayed(new Runnable()
{
        @Override
        public void run()
        {
            //Your Work
            //u can add intent here
        }
  }, 1000);
//   1sec pause
```

For aligment of textView with lot of text like a paragraph,
android:justificationMode="inter_word"

Menu:

If we wanna create a menu(three dots on upper right corner) then,

Create res->new directory called menu->create a resource file called main_menu

main_menu:

```
<menu xlms:android="http:// ........">
    <item
       android:title="Sign Out"
       android:id="@+id/signout"/>

    <item
       android:title="Settings"
       android:id="@+id/settings"/>
</menu>
```

In the activity u wanna show the menu:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId()==R.id.signout) {
        FirebaseAuth.getInstance().signOut();
        finish();
    }
    else if (item.getItemId()==R.id.settings) {
        Toast.makeText(MainActivity.this, "You have selected the Settings!", Toast.LENGTH_LONG).show();
    }
    return true;
}
```

Password:

android:inputType="textPassword"
can be added in EditTextView if we want text to look like we are entering a password(like ****)

Theme:
u can add theme to the activiy in manifest like
android:theme="@style/Theme.Design.NoActionBar"

```
        <activity android:name=".DetailActivity" android:theme="@style/Theme.Design.NoActionBar"/>
        <activity android:name=".DelhiListActivity" android:theme="@style/Theme.Design.NoActionBar"/>
        <activity android:name=".RegisterActivity" android:theme="@style/Theme.Design.NoActionBar"/>
        <activity android:name=".LoginActivity" android:theme="@style/Theme.Design.NoActionBar"/>
        <activity android:name=".SplashActivity" android:theme="@style/Theme.Design.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

Transparent background:

If u want the background of the view to be transparent, like to show the background of parent
android:background="@android:color/transperent"

If u wanna **check something is not null,** like if u wanna check whether the data from the editText is not null then
if(!TextUtils.isEmpty(variable)){

Translations:
You can Right click on the styles->translations->select language add translations

For **scrolling** apps include the whole xml inside ScrollView

Retriving data from EditText:
```
String name = enterName.getText().toString();
```

XML attribute:
```
inputType  -> can be text,textCapWords (strating letter will be caps)
hint, etc
```

CheckBox:
```
 private CheckBox creamCheckbox;
 boolean hasCream;
 hasCream= creamCheckbox.isChecked();
```

In **RelativeLayout** you **can place one view over the other**
You can't do that in LinearLayout

Decalring Views or anything is usually outside OnCreate()
Assigning a view to a instance variable by using id must be inside OnCreate()

Coverting to string:
eg:
```
nCofee.setText(String.valueOf(n));
```

For accessing a string from strings.xml:

R.string.resource_name

In XML:@string/resource_name

Here even though reference name looks like a string but its a int

For playing audio onClicking the view:

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
         word currentBlock=words.get(position);   //getting the current object using position
         int play=currentBlock.getPlayId();      //getting the audio id
         MediaPlayer mediaPlayer=MediaPlayer.create(PhrasesActivity.this,play);
         mediaPlayer.start();
    }
});
```

For accessing a color from colors.xml:

R.color.color_name

for setting the color:
eg:
should Orange first be in colrs.xml and have import something

```
Money.setTextColor(ContextCompat.getColor(MainActivity.this,R.color.Orange));
```

Another Easy Method of coloring without using colors.xml:

```
Any_view_reference_variable.setTextColor(RED);
```
you have to press Color. at that time RED or other color pops select it

Displays a message in bottom of screen for a while(LONG OR SHORT):

Toast:
Param must be MainActivity,string to display,Duration

```
Toast.makeText(MainActivity.this,"You are Awesome",Toast.LENGTH_SHORT).show();
```

Snackbar: Same as Toast but with more features
param: any view,text,duration | .setAction is the new stuff here

```
        Snackbar.make(Money,"You are Awesome",Snackbar.LENGTH_SHORT).setAction("More", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //this happens if we click More in Snackbar in screen
                //Log.d("Snack", "showInfo: Snackbar More");
                Log.d("Snack","Hey");
                //log is visible for developers only in Logcat Debug
            }
        }).show();
```

To display Money_val in dollar:

```
NumberFormat number=NumberFormat.getCurrencyInstance();
number.format(Money_val)
```

log is visible for developers only in Logcat Debug
params: tag,msg (put string for them)

```
Log.d("Snack","Hey");
```

Visibilty:
It can VISIBLE,INVISIBLE,GONE

```
hobbies.setVisibilty(View.VISIBLE)
hobbies.getVisibility()
```

hobbies is the instance variable of text view which was invisible/gone

Hide Keyboard:

```
//The below stuff is used to access the few functions of android device in which your app is running

InputMethodManager manager=getSystemService(Context.INPUT_METHOD_SERVICE);

//Once you do this android studio will suggest you to cast it,just cast it

manager.hideSoftInputFromWindow(view.getWindowToken(),0)
```

or

```
InputMethodManager manager= (InputMethodManager) v.getContext().getSystemService
                            (Context.INPUT_METHOD_SERVICE);
manager.hideSoftInputFromWindow(v.getWindowToken(),0);
```

or u can add it in utils

```
public static void hideSoftKeyboard(View view) {
        InputMethodManager imm =
     (InputMethodManager) view.getContext().getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(view.getWindowToken(), 0);

}


calling: Utils.hideSoftKeyboard(view)
```

Data binding:

```
Add this in build.gradle
buildFeatures{
    dataBinding true
}

Then put XML code inside
<layout
......>
......
</layout>
You can easily do it by pressing alt+enter then dataBinding will pop up, press it

Then create instance variable, Here binding is instance variable
private ActivityMainBinding binding;

Then inside main function(that is protected on create function which is always there)
binding=DataBindingUtil.setContentView(this,R.layout.activity_main);
                                    or
                                MainActivity.this
```

Using Data binding
eg:
binding.id_of_the_view.setText();

or

Add this in build.gradle
```
buildFeatures{
     viewBinding true
}
```

```
//if activity name is SignUpActivity
ActivitySignUpBinding binding;

@Override
protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     binding = ActivitySignUpBinding.inflate(getLayoutInflater());
     setContentView(binding.getRoot());
}
```

or

```
//if activity name is SignInActivity
ActivitySignInBinding binding;

@Override
protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     binding= ActivitySignInBinding.inflate(getLayoutInflater());
     setContentView(binding.getRoot());
}
```

Creating new activity:
Right click on java->new->Empty activity

To move to new activity (Intent):

a. To make it display:

these are inside first java file(MainActivity.java)

```
                              From                  to
Intent intent = new Intent(MainActivity.this,Next_activity_name.class);
startActivity(intent);
```

You have to put this on may be on click of button or it depends on the app that you are building

b. If u just want to go to next page then u can do this:

```
startActivity(new Intent(MainActivity.this,Next_activity_name.class));  //instead of those two line
```

c. To pass some value from 1st java file to second:

```
intent.putExtra(key,value);
```
this must added before calling startActivity(intent)

d. For receiving that passed value:

Inside second java file

```
String value = getIntent().getStringExtra(key);
```

Here value is the instance variable

Bundle: U can put anything inside it
Just storing everything sent from intent to bundle

a. For storing one value sent from intent, Here extra is instance variable

```
Bundle extra = getIntent().getStringExtra(key);
```

b. For receiving all value sent from intents,

```
Bundle extra = getIntent().getStringExtras();
```

c. For accessing it:

```
extra.getString(key);
extra.getInt(key);
```
It depends on type so getString,getInt,.....

For passing something from 2nd layout to first:

You have go to 2nd layout by using (In 1st layout)

```
startActivityForResult(intent, REQUEST_CODE);   //not startActivity(intent); Because it will know it is expecting
```
something from 2nd layout

override onActivityResult
You can do that by right click+override

Then inside 2nd layout,

```
Intent replyIntent = getIntent();
replyIntent.putExtra("message_back", "From Second Activity"); //data that must be sent back to the called activity
setResult(RESULT_OK, replyIntent);                          //RESULT_OK is inbuilt thing which say everything is fine
finish();                                                   //it will remove 2nd layout which was stacked over 1st layout
```

And inside the overrided function:REQUEST_CODE is set by us(private final int REQUEST_CODE = 2;) inside main

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == REQUEST_CODE && resultCode == RESULT_OK) {
            assert data != null;
            String message = data.getStringExtra("message_back");

            Toast.makeText(MainActivity.this, message,Toast.LENGTH_LONG).show();
        }


    }
```

Common Intents: Just google it
For accessing other apps in the android like maps,email,cam,etc.

Example:From Cofee Order app

```
 String name=nameET.getText().toString();
 String summary=createOrderSummary(name,price,hasCream,hasChoclate);

 Intent intent = new Intent(Intent.ACTION_SENDTO);
 intent.setData(Uri.parse("mailto:")); // only email apps should handle this
 intent.putExtra(Intent.EXTRA_SUBJECT,"Oder from "+name);
 intent.putExtra(Intent.EXTRA_TEXT,summary);
 if (intent.resolveActivity(getPackageManager()) != null) {
      startActivity(intent);
 }
```

There many ways of **ArrayAdapter**, we used the below in mivok app
For other kind google it

ListView + ArrayAdapter:

```
                                id of ListView which
                                 we wanna populate
ListView listView = findViewById(R.id.list);                              in-built layout with one textview
ArrayAdapter <String> itemsAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, words);
listView.setAdapter(itemsAdapter);
```

Here **list** is the id of the list view we want to populate
listView is the instance variable and words is list(array or arraylist)
android.R.layout.simple_list_item_1 this is the pre-defined layout provided by android for other layout google R.layout
or u can put your own layout but if has more than one view then u have to use custom adapter

U can use listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){.....rest will be done by android studio itself
When we wanna do do something when we click a item on listview ,use the above
In that position parameter will the index of the list

GridView + ArrayAdapter:
in XML file add-> android:numcolumns="no.of_columns" then

```
                                id of ListView           <>
GridView gridView =findViewById(R.id.list);              or
ArrayAdapter <String> itemsAdapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, words);
gridView.setAdapter(itemsAdapter);
```

Here gridView is the instance varable and words is list(array or arraylist or array of objects etc)
android.R.layout.simple_list_item_1 this is the pre-defined layout provided by android
we can add custum layout instead of android.R.layout.simple_list_item_1 also
eg:
R.layout.layout_name

xml file to use in array adapter:
xmlns:tools="http://schemas.android.com/tools"
This will be in xml file of main activity, so u can copy paste to other activity also
Under this we can add things like
tools:text="one"
This will just be an place holder
This will be visible only on design not in actual app

Custom ArrayAdapter: (is made when we have to display more than just a single view,like in mivok app)
For more info check mivok app
eg:

```
//it's a .java file
public class WordAdapter extends ArrayAdapter<word> {

    //our own constructer
    public WordAdapter(Context Activity, ArrayList<word> words ){
        super(Activity,0,words);
    }

    //overriding getView so that it takes more than 1 view like in this case 2 textviesws
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
        View freeView = convertView;

        //Checking id it's null, if null means no free views(scrap views) are available
        //So we have to do it from scratch
        if(freeView==null){
            freeView= LayoutInflater.from(getContext()).inflate(R.layout.list_item,parent,false);
        }
        //object(in this case) word located in this position in the list(in this words)
        word currentBlock=getItem(position);

        //freeView(i.e, convertView) is free so we have setText to display
        //currentBlock will have what to display so
```

```
        //connecting freeView views to instance to set values
        TextView mivokTextView=freeView.findViewById(R.id.MiwoktextView);
        TextView GeneraltextView=freeView.findViewById(R.id.GeneraltextView);

        //setting values for views of freeViews from current block
        mivokTextView.setText(currentBlock.getMiwokTranslation());
        GeneraltextView.setText(currentBlock.getDefaultTranslations());

        return freeView;
    }
}
```

In main java code which we want display:

```
WordAdapter adapter = new WordAdapter(this,words);
ListView listView = findViewById(R.id.list);        //list is the id given to the ListView
listView.setAdapter(adapter);
```

MediaPlayer: Playing music/audio
Check this link
https://www.tutorialspoint.com/android/android_mediaplayer.htm
To play music if it is present in raw folder
If u it from internet create() is bit differnt
eg:

```
        Button play = findViewById(R.id.playButton);
        Button pause = findViewById(R.id.pauseButton);

        MediaPlayer mediaPlayer = MediaPlayer.create(this,R.raw.roman);

        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mediaPlayer.start();
            }
        });

        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mediaPlayer.pause();
            }
        });
```

mp3's are placed in raw folder
right click on res->new->Android Resource Directory (name it raw and set value to raw) to create raw

Playing music when onclick for listview,

```
ListView listView = findViewById(R.id.list);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        MediaPlayer mediaPlayer=MediaPlayer.create(NumbersActivity.this,R.raw.number_one);
        mediaPlayer.start();
    }
});
```

You can ovveride activity cycle methods as onStart(),onResume(),onPause(),onDestroy()
e.g:

```
//it will realease resources if the app is in stop phase
    @Override
    protected void onStop() {
        super.onStop();
        releaseMediaPlayer();
    }
```

```
//if it is done playing audio release it so that android resources will be free
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
   @Override
   public void onCompletion(MediaPlayer mp) {
       releaseMediaPlayer();
   }
});
```

```
//release the resources
private void releaseMediaPlayer(){
   if(mediaPlayer==null){
       mediaPlayer.release();
       mediaPlayer=null;
   }
}
```

Shake Animation:

create a res directory of type anim
then add a new animation resource file inside anim directory and paste this for shake animation

```
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="150"
    android:fromDegrees="-10"
    android:pivotX="30%"
    android:pivotY="30%"
    android:repeatCount="1"
    android:repeatMode="reverse"
    android:toDegrees="10"
    />
```

Add this to MainActivity:

```
private void shakeAnimation(){
        Animation Shake= AnimationUtils.loadAnimation(MainActivity.this,R.anim.shake);
        binding.cardView.setAnimation(Shake);
        Shake.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
                binding.questionDisp.setTextColor(Color.RED);
            }

            @Override
            public void onAnimationEnd(Animation animation) {
                binding.questionDisp.setTextColor(Color.WHITE);
            }

            @Override
            public void onAnimationRepeat(Animation animation) {

            }
        });
    }
```

For calling it:

```
shakeAnimation();
```

There are many animations like fade which come under AlphaAnimation, just google these stuffs
link:
https://developer.android.com/reference/android/view/animation/AlphaAnimation

Fade Animation:
This is inside Mainactivity itself
here we need not create a anim file
we are using AlphaAnimation class directly

```
private void fadeAnimation(){
        AlphaAnimation alphaAnimation=new AlphaAnimation(1.0f,0.0f);
        alphaAnimation.setDuration(300);
        alphaAnimation.setRepeatCount(1);
        alphaAnimation.setRepeatMode(Animation.REVERSE);

        binding.cardView.setAnimation(alphaAnimation);

        alphaAnimation.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
                binding.questionDisp.setTextColor(Color.GREEN);
            }

            @Override
            public void onAnimationEnd(Animation animation) {
                binding.questionDisp.setTextColor(Color.WHITE);
            }

            @Override
            public void onAnimationRepeat(Animation animation) {

            }
        });
    }
```

Hide ActionBar:

```
getSupportActionBar().hide();
```

enum class:

eg:

```
public enum Priority {
    HIGH,
    MEDIUM,
    LOW
}
```

To get time:

```
Calendar.getInstance().getTime()
```

CalendarView:

```
private CalendarView calendarView = view.findViewById(R.id.calendar_view);

calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
     @Override
     public void onSelectedDayChange(@NonNull CalendarView view, int year, int month, int dayOfMonth) {
           //u can use year,month,dayOfMonth
     }
});
```

If u wanna convert it to Date Type:

```
Calendar calender = Calendar.getInstance();
private Date dueDate;

CalendarView calendarView = view.findViewById(R.id.calendar_view);

calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
     @Override
     public void onSelectedDayChange(@NonNull CalendarView view, int year, int month, int dayOfMonth) {
           calender.clear();
           calender.set(year,month,dayOfMonth);
           dueDate = calender.getTime();
     }
});
```

To convert Date type variable to String:

```
public static String formatDate(Date date){
    SimpleDateFormat simpleDateFormat = (SimpleDateFormat) SimpleDateFormat.getDateInstance();
    simpleDateFormat.applyPattern("EEE,MMM d");

    return simpleDateFormat.format(date);
}
```

eg:
String formatted = formatDate(date)

We usually keep this function in Utils,

```
public class Utils {
    public static String formatDate(Date date){
        SimpleDateFormat simpleDateFormat = (SimpleDateFormat) SimpleDateFormat.getDateInstance();
        simpleDateFormat.applyPattern("EEE,MMM d");

        return simpleDateFormat.format(date);
    }
}
```

RadioGroup:

```
//for checking which radio button is checked in the radio button group
priorityRadioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
       @Override
       public void onCheckedChanged(RadioGroup group, int checkedId) {
           if(checkedId == R.id.radioButton_high){
                               priority = Priority.HIGH;
       }
       else if(checkedId == R.id.radioButton_med){
           priority = Priority.MEDIUM;
       }
       else if(checkedId == R.id.radioButton_low){
           priority = Priority.LOW;
       }
       else{
           priority = Priority.LOW;
```

```
        }
}
```

```
//For setting visibility
priorityRadioGroup.setVisibility(priorityRadioGroup.getVisibility() == View.GONE? View.VISIBLE : View.GONE);
//for setting visible/invisible
```

ScreenUtility:
Gives screen width and Height
This is usually placed in util directory

```
public class ScreenUtility {
     private Activity activity;
     private float dpWidth;
     private float dpHeight;

     public ScreenUtility(Activity activity) {
           this.activity = activity;

           Display display = activity.getWindowManager().getDefaultDisplay ( );
           DisplayMetrics outMetrics = new DisplayMetrics();
           display.getMetrics(outMetrics);
           float density = activity.getResources().getDisplayMetrics().density;

           dpHeight = outMetrics.heightPixels / density;
           dpwidth = outMetrics.widthPixels / density;
    }
```

```
    public float getDpWidth() {
       return dpWidth;
    }

    public float getDpHeight(){
       return dpHeight;
    }
}
```

Using:

```
ScreenUtility screenUtility = new ScreenUtility(getActivity());
Log.d("Width",String.valueof(screenUtility.getDpWidth()));
```

For setting different layouts for different size devices,

eg:

For devices having at least 600dp(like tablets) will take layouts from here

Create a new directory as,

layout-sw600dp

Press pakages to see it

Put layouts inside it

you can set image as background of views