# 3. Linking Activity

- Android Intent is an abstract description of an operation to be performed.
- It can be used with,
  - An startActivity() to launch an Activity
  - An sendBroadcast() to send it to any interested BroadcastReceiver components
  - An startService(Intent) to communicate with a background Service

# Linking Activity (con…)

- Intent Objects
  - An Intent object is a bundle of information which is used by the component that receives the intent.
  - Intent object can contain the following components
    - Action
      - A string naming the action to be performed.
      - The action in an Intent object can be set by the setAction() method and read by getAction().

# Linking Activity (con…)

- Data
  - Adds a data specification to an intent filter.
  - The setData() method specifies data only as a URI, setType() specifies it only as a MIME type, and setDataAndType() specifies it as both a URI and a MIME type. The URI is read by getData() and the type by getType().

# Linking Activity (con…)

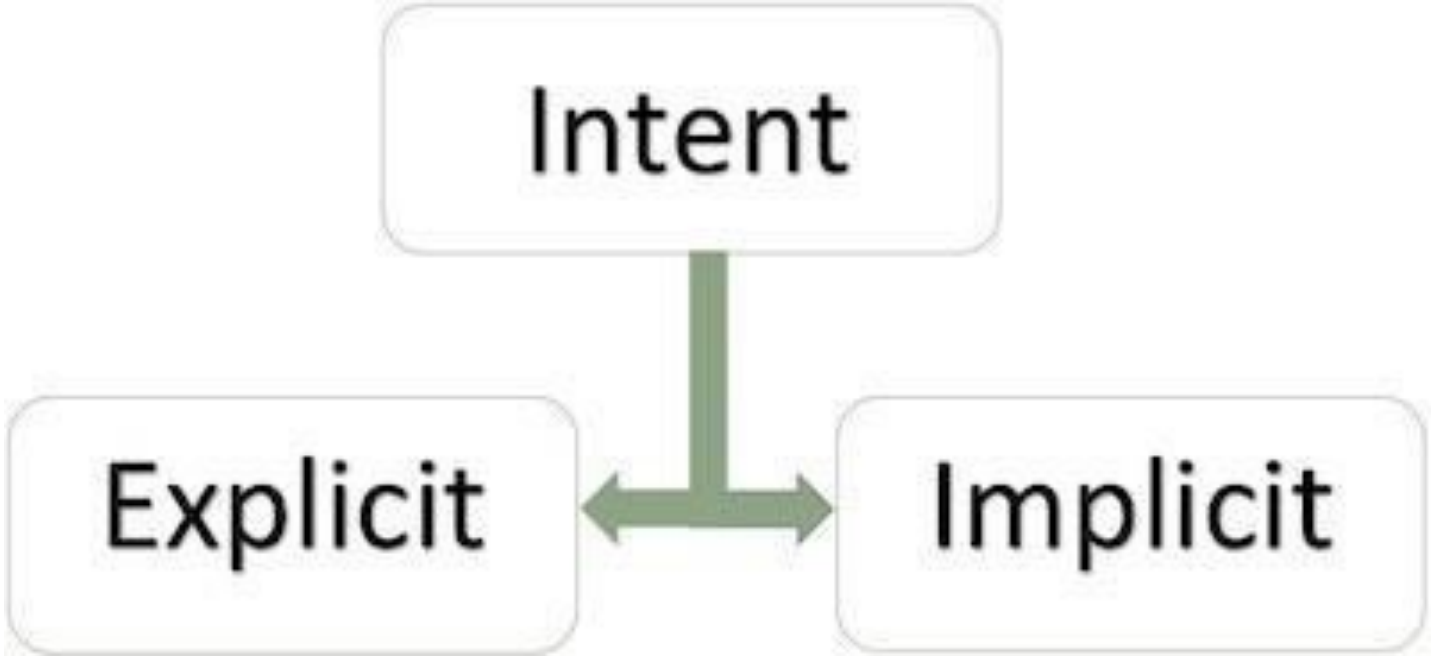
- Category
  - The category is an optional part of Intent object and it's a string containing additional information about the kind of component that should handle the intent.
  - The addCategory() method places a category in an Intent object, removeCategory() deletes a category previously added

# Linking Activity (con…)

- Extras
  - This will be in key-value pairs for additional information that should be delivered to the component handling the intent.
  - The extras can be set and read using the putExtras() and getExtras() methods respectively.
- Flags
  - The flags are optional part of Intent object
  - It helps to instruct the Android system how to launch an activity, and how to treat it after it's launched etc.

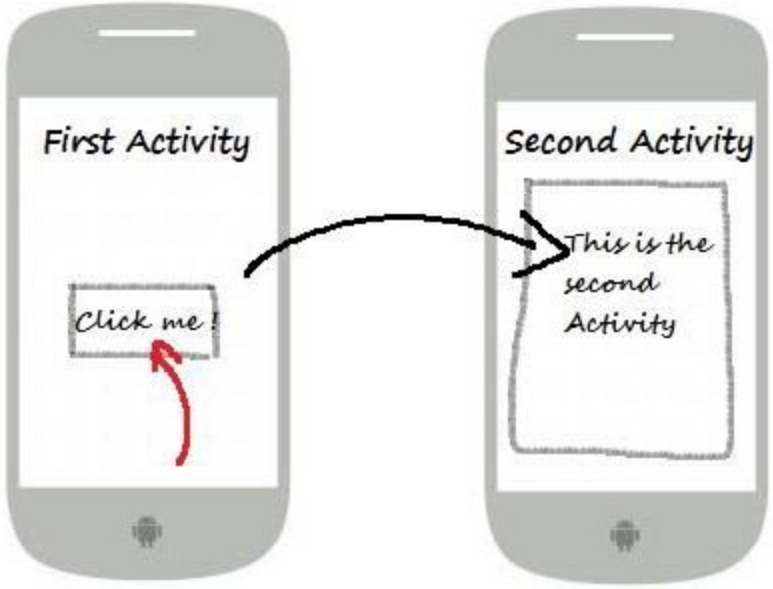
# Linking Activity (con…)

- Types of Intents
  - There are following two types of intents supported by Android



# Linking Activity (con…)

- Explicit Intents
  - It going to be connected internal world of application, i.e. to connect one activity to another activity, below image is connecting first activity to second activity by clicking the button.



# Linking Activity (con…)

- Explicit intents explicitly define the component which should be called by the Android system, by using the Java class as identifier.
- Explicit intents are typically used within an application as the classes in an application are controlled by the application developer.
- Example
- To start new activity we have to create Intent object and pass source activity and destination activity as shown below −

```
Intent i = new Intent(FirstActivity.this, SecondAcitivity.class);
startActivity(i);
```

# Linking Activity (con…)

- Implicit Intents
  - It going to connect with out side application such as call, mail, phone, see any website ..etc. In implicit intent we have to pass an action using setAction() as shown below example.
  - Implicit intents are often used to activate components in other applications.
- Example

```
Intent i=new Intent(Intent.ACTION_VIEW,
Uri.parse(www.google.com));
startActivity(i);
```

# 4. Passing Data

- Activity is used to represent the data to user and allows user interaction.
- In an android application, we can have multiple activities and that can interact with each other.
- During activity interaction we might required to pass data from one activity to other.

# Passing Data (con…)

- Data is passed as extras and are key/value pairs.
- The key is always a String and the value you can use the primitive data types int, float, chars, etc.
- Syntax for sending data

```
Intent intent = new Intent(context,
YourActivityClass.class);

intent.putExtra(KEY, <your value here>);
startActivity(intent);
```

# Passing Data (con...)

- Syntax for retrieving data

```
Intent i = getIntent();
String stringData= i.getStringExtra(KEY);
int numberData = i.getIntExtra(KEY,
defaultValue);
boolean booleanData = i.getBooleanExtra(KEY,
    defaultValue);
char charData = i.getCharExtra(KEY,
    defaultValue);
```

# Example

```
public class MainActivity extends Activity implements
    OnClickListener {
    Button btn;
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn = (Button) findViewById(R.id.btnPassData);
        btn.setOnClickListener(this);
 }
```

# Example (con…)

```
 @Override
public void onClick(View view)
{
Intent i = new
Intent(getApplicationContext(), SecondActivity.class);
    i.putExtra("message",     "Hello     From
MainActivity");
startActivity(i);
}
}
```

# Example (con…)

```
public class SecondActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_another);
        Intent i = getIntent();
        String msg = intent.getStringExtra("message");
        Toast toast = Toast.makeText(this, msg,
     Toast.LENGTH_LONG);
        toast.show();
    }
}
```