A photograph of a bicycle at sunset. The sky is filled with warm orange and yellow hues. The bicycle's frame, wheels, and handlebars are visible in the foreground. A purple diagonal bar runs from the bottom right corner across the slide.

Android

IDE & UI BASICS

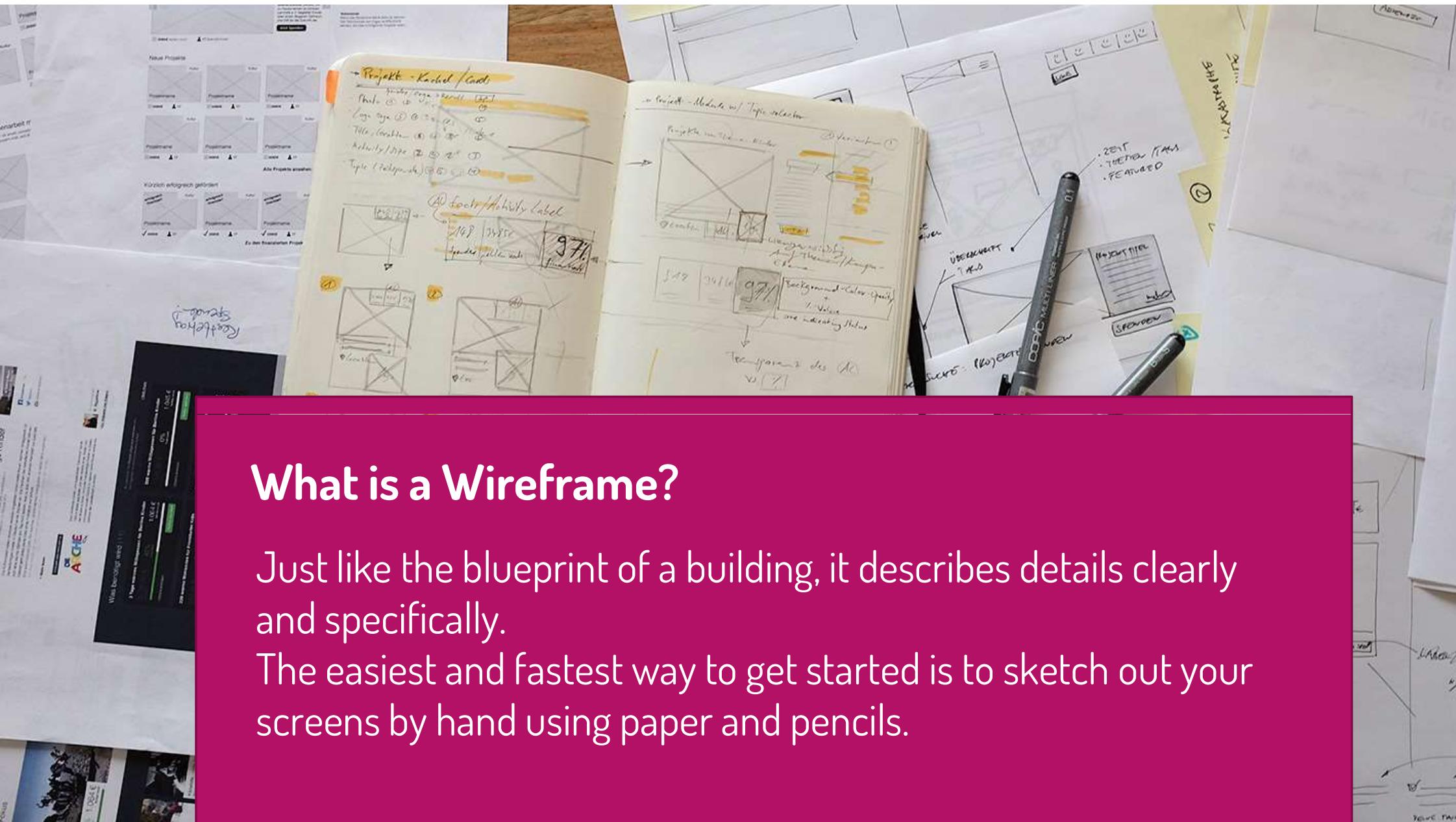
CENTRE FOR COMPETENCE FOR MOBILE APP DEVELOPMENT , NIC KERALA

SYAMKRISHNA BG
NATIONAL
INFORMATICS
CENTRE **NIC**

Process in UI Design

Identify the UI Pattern → Wire framing
→ MockUps → Prototyping → Final UI → Code

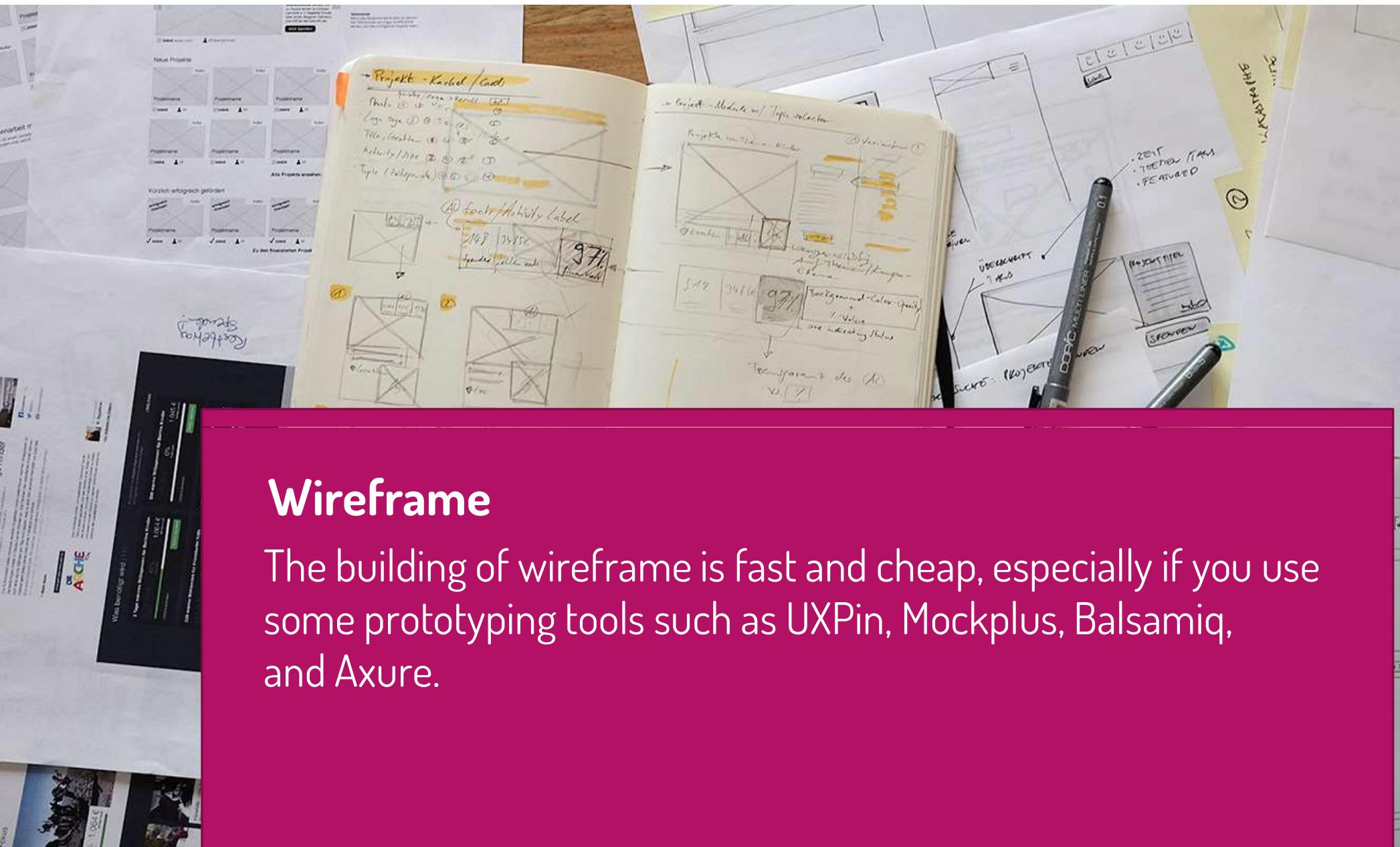
**Define the UI Workflow → Wire framing
→ MockUps → Prototyping → Final UI → Code**



What is a Wireframe?

Just like the blueprint of a building, it describes details clearly and specifically.

The easiest and fastest way to get started is to sketch out your screens by hand using paper and pencils.



Wireframe

The building of wireframe is fast and cheap, especially if you use some prototyping tools such as UXPin, Mockplus, Balsamiq, and Axure.

Select the type of leave

CL RH EL OTHERS

You have Selected Casual Leave

Select Leave Period

From 11/02/2018 FULL DAY To 14/02/2018 FORENOON

You are applying for 3.5 days

Nature of Leave to Apply
Personal Grounds

Station Leave Required

Ground on which leave is applied for

Address During the Leave Period

Reporting Officer Details

Reporting to officer of
Same Office

Designation
Personal Grounds

Name of Officer
Personal Grounds

Approving Authority Details

Reporting to officer of
Same Office

Designation
Personal Grounds

What is a MockUp?

Wireframes are the skeleton. Prototypes demonstrate the behavior. Mockups are the skin..

It is very important to have the knowledge of the platform for which you are doing the mockup. Slicing of images will play very important role when you prepare the prototype

A sample Mock Up

The mock-up displays four screens of the SPARK OnMobile app:

- OTP Confirmation Screen:** Shows a placeholder for an OTP code and a "Continue" button. It includes a note about sending an SMS to the registered number and terms of service.
- Verify your details Screen:** Requests the user to provide their PEN and Mobile Number. It also includes a note about agreeing to terms and conditions.
- Welcome Screen:** Displays the user's profile picture, name (SYAMKRISHNA BG), designation (Junior Engineer, Gr II), and department (Public Works Department). It also shows a welcome message and a note about leave credits.
- Leave management Screen:** A dashboard showing leave statistics (CL: 1.0, RH: 2.0, EL: 400, HPL: 81) and past applications. It includes buttons for "Apply Leave" and "Update Leave".

Each screen includes the SPARK OnMobile logo and the National Informatics Centre (NIC) branding at the bottom.

Mock Up Tools



Sketch



Photoshop



Adobe Illustrator

And many

What is a Prototype?

The requirement of a prototype is higher than wireframe, it must be a high fidelity prototype which is interactive and fit the final user interface as much as possible.

And Finally

UI Design is Not UX Design

Web 2 Mobile

Comparison & Transformation
Onboarding for a web developer

Example Activity

```
public class ExampleActivity extends Activity
{
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.demo);
        Button button = (Button) findViewById(R.id.corky);

    }
}
```

To start with, you need basic Knowledge on
MVC
OOPS
XML
JAVA

Very important
For your app to be able to use activities, you must declare the activities, and certain of their attributes, in the manifest.

UI Action Elements

Android Controls

UI Basic Elements

	TextView
	EditText
	Button
	RadioButton
	Checkbox
	RatingBar
	ProgressBar
	SeekBar
	Switch
	ToggleButton
	Spinner

	AutoComplete TextView
	MultiAutoComplete TextView
	CheckedTextView
	TextSwitcher
	ImageSwitcher
	AdapterViewFlipper

	ImageButton
	ImageView
	VideoView
	TextClock
	DatePicker
	TimePicker
	Chronometer

UI Basic Elements In Action

UI

Basic Layouts

Linear Layout

1 VIEW File path : res/layout/demo.xml

```
<?xml version="1.0" encoding="utf-8"?>

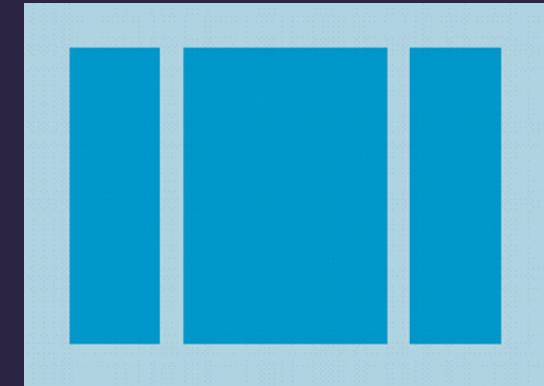
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="One" />
    <TextView android:id="@+id/text2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Two" />
    <TextView android:id="@+id/text3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Three" />

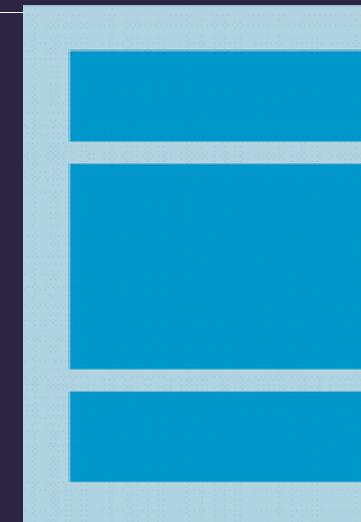
</LinearLayout>
```

2 View Output wireframe

android:orientation="horizontal"



android:orientation="vertical"



3

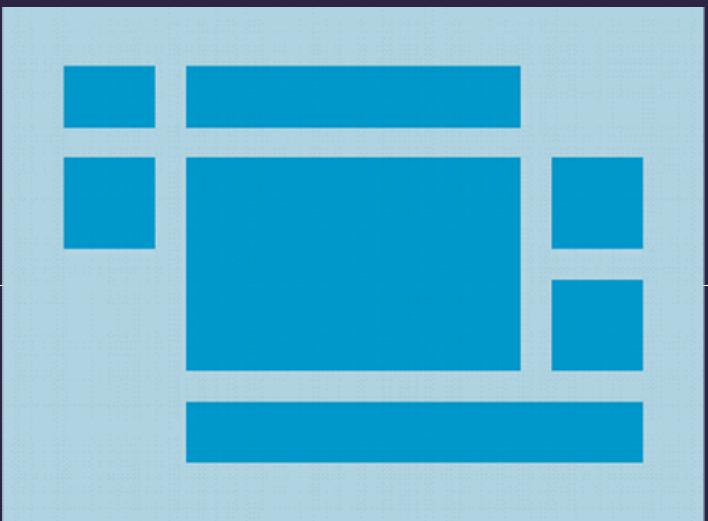
CODE

File path : src/demoActivity.java

```
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.demo);
```

TableLayout

Displays child View elements in rows and columns.



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_open"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_open_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_save"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_save_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
</TableLayout>
```



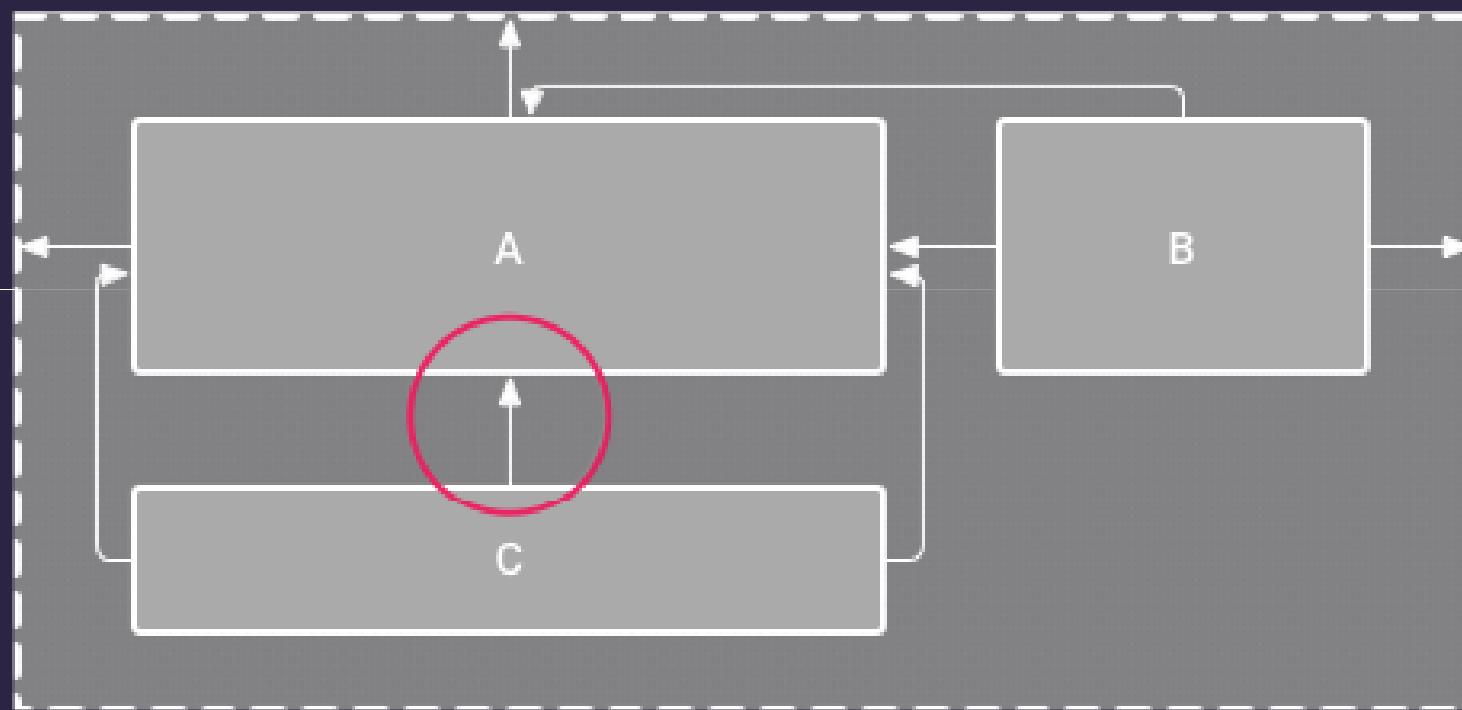
RelativeLayout



RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

Constraint Layout



ListView

Displays a scrolling single column list.



WebView

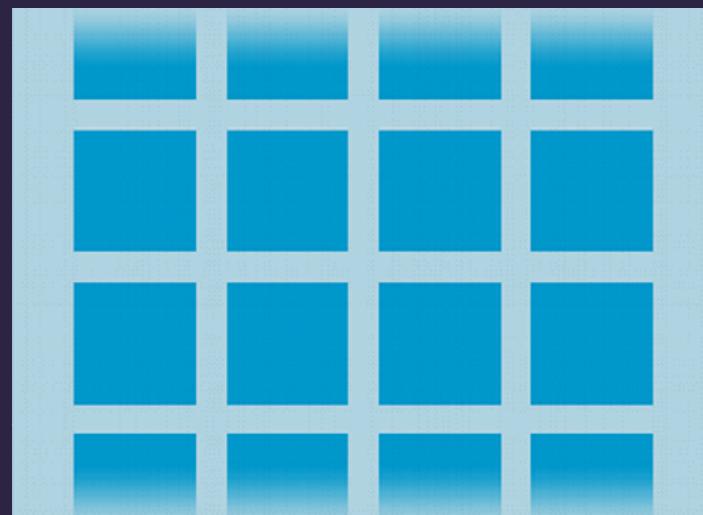
```
<html>
```

```
<!-- web page -->
```

```
</html>
```

GridView

Displays a scrolling grid of columns and rows.



For better performance in your list, you should instead build your list with RecyclerView.

```
<?xml version="1.0" encoding="utf-8"?>
<WebView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

Event listeners

Android UI Events

Example Activity with Events

```
public class ExampleActivity extends Activity implements OnClickListener
{
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.demo);
        Button button = (Button) findViewById(R.id.corky);
        button.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v)
    {
        // do something when the button is clicked ,identify the View from variable v
    }
    ....
}
```

For your app to be able to use activities, you must declare the activities, and certain of their attributes, in the manifest.

Understand the IDE

Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

sagara2 > app > src > main > res > layout > activity_dashboard_main.xml

Project 1: Project 2: Structure Captures 2: Favorites Build Variants

Palette Attributes

Common Text Buttons Widgets Layouts Containers Google Legacy Project

ConstraintLayout
Guideline (horizontal)
Guideline (vertical)
LinearLayout (horizontal)
LinearLayout (vertical)
FrameLayout
TableLayout
TableRow
Space

Component Tree

Nothing to show

Design Text

Gradle projects sagara2 sagara2 (root) :app

Thank you

