Exploring User Interface Screen Elements

* What is Android View with Example.

=> In Android Development, View is a Fundamental building block of the user interface.

Views are used to create the visual components that users interact with in an android application.

A View is an Object that used to display element on screen.

A View is provides a way for the user to interact with the Android Application Screen.

Example: TextView, Button, EditTextView, CheckBox, Radio Buttons and Seekbar etc.

1 Textview: Textview is used to display the text on the screen.

This Text does not perform any type of action.

- Attributes:
  - id - Textview Id
  - Text - Textview Text
  - textSize - Textview size
  - textColor - Textview text color
  - Fontfamily - Text font style etc.

- Events:

a Click Event: onClick(View v): used Perform the click event on the textview.

Using onClickListener we can Perform this event.

b TextChanged Listener: If you want to perform an action when this event is used.

Method name: beforeTextChanged
onTextChanged
afterTextChanged.

- Example:

```
public class MainActivity extends
        AppCompatActivity
{
    TextView t;
    t=FindViewById(R.id.tet);

    t.setOnClickListener(new
        View.OnClickListener()
    {
        public void onClick(View v)
        {
            t.setTextColor(Color.RED);
        }
    });
}
```

2. Button: Button is used to perform action when it is clicked.

Button is used to represents a clickable Button.

- Attributes:
  - id : Button Id
  - text : Button Text
  - Fontfamily : Button Text Style
  - Colour : Button Color.

- Events : Button is used to Perform click events.

Using onClickListener, we can perform click event.

method : onclick.

- Example :

```
public class MainActivity extends
        AppCompatActivity
{
    Button btn;
    btn = FindViewById(R.id.button);

    btn.SetOnClickListener(new
        View.OnClickListener()
    {
      public void onCick(View v)
      {
          btn.setText("Thank You");
      }
    });

}
```

3 EditText: This view is used to allow users to enter and modify text.

It is commanly use to take user input in android.

- Attributes:
  - id: EditText Id
  - hint: hint for User input
  - text: Edittext Predefined text
  - TextSize: Text size
  - TextColor: Text Color

- Events:

a TextChange Event: Use to change the text.

Using addTextChangedListener we can perform this event.

Method: beforeTextChanged
onTextChanged
AfterTextChanged

b FocusChange Event: Detect the EditText Focus gain or losses

Listener - onFocusChange

Page No.
Date : / /

Method - onFocusChange

- Example:

```
public class MainActivity extends
        AppCompatActivity
{
    Button btn;
    EditText et;
    String name;

    tv=FindViewById(R.id.textview);
    btn=FindViewById(R.id.button);
    btn.setOnClickListener(new
        View.onClickListener()
    {
        public void onClick(View v)
        {
            name=et.getText().toString();
            tv.setText("Welcome "+name);
        }
    });
}
```

4 CheckBox : It is used to allows user to toggle between two state - checked and unchecked.

- Attributes:
  - id : CheckBox id
  - checked : Set checkBox state
  - text : Display text
  - textSize : Text size set
  - textColor : Text color change.

- Events:

a OnCheckedChange Event: Used to change state of the check box.

Listener : setOnCheckedChange

Method : onCheckedChanged

b Click Event: Used to handle clicks on the Check Box.

Listener : setOnClickListener

Method : onClick

- Example:

```
public class MainActivity extends
    AppCompatActivity
{
  TextView tet;
  CheckBox red, abc, xyz;
  Button btn;
```

```
tet = FindViewById(R.id.qview);
red = FindViewById(R.id.rb);
abC = FindViewById(R.id.ab);
XYZ = findViewById(R.id.xb);
btn = FindViewById(R.id.but);

btn.setonClickListener(new
    View.onClickListener()
{
  public void onClick(View v)
{
   if(abc.isChecked())
       tet.setText("No");
   else if(red.isChecked())
       tet.setText("Yes");
   else
       tet.setText("No");
  }
});
}
```

5 RadioButton: RadioButton allows to select single option at a one time.

- Attributes:
  - id: Set RadioButton Id
  - checked: set inital state
  - text: set Text
  - textSize: Change Text size.

- Events :

a OnCheckedChange Event : Used to change the state of radio button.

Listener : setonCheckedChange

Method : onCheckedChanged

b Click Event : Handle the click of radio button.

Listener : setonClicked

Method : onClick.

- Example :

```
public class MainActivity extends
    AppCompatActivity
{
    TextView text, tet;
    RadioGroup rg;
    RadioButton male, female;

    text = FindViewById(R.id.bs);
    tet = FindViewById(R.id.tv);
    rg = FindViewById(R.id.radio);
```

```
male = findViewById(R.id.m);
Female = findViewById(R.id.fe);
if(male.isChecked())
{
    tet.setText("Male");
}
else if(female.isChecked())
{
    tet.setText("Female");
}
}
```

6 SeekBar : SeekBar is used to allow user to select a value within a specified range.

- Attributes:
  - id : Set the Seekbar id
  - max : Set the Maximum value
  - progress : Set the Inital Progress

- Events:

a OnSeekBarChange Event: Allow user to change the progress of seekbar.

Listener - setOnSeekBarChange

Method - onProgressChanged
- onStartTrackingTouch
- onstopTrackingTouch

Date : / /

Example:

```
public class MainActivity extends
    AppCompatActivity
{
    SeekBar sb;
    TextView tv;

    sb=FindViewById(R.id.bar);
    tv=FindViewById(R.id.text);

    sb.setOnSeekBarChangeListener
     (new SeekBar.OnSeekBarChange
      Listener()
    {
      public void onProgressChanged
       (SeekBar seekbar, int progress,
        boolean FromUser)
      {
        tv.setText("Progress"+
            progress);
      }
    });
}

}
```

Page No.
Date : / /

7 Menus: Menus are use to provides a way to offer multiple option to the user.

Option menu is appers in the Application's action bar or Tool bar.

- Example:

```
public class MainActivity extends
    AppCompatActivity
{
  public boolean onCreateOption
      Menu(Menu menu)
  {
    MenuInflater m = getMenuIn
        flater();
      m.inflate(R.menu.menu, menu);
      return true;
  }
  public boolean onOptionItem
      Selected(MenuItem item)
  {
    switch(item.getItemId())
    {
      case R.id.mins:
        Toast.makeText(this, "AT",
          Toast.LENGTH_SHORT).show();
        return true;
```

```
        case R.id.mitoc:
            Toast.makeText(this, "AP",
                Toast.LENGTH_SHORT).show();
            return true;
        }
        return true;
    }
}
```

8 Alert Dialog : This Dialog is a pop-up box that can be used to show some important information.

- Example :

```
public class MainActivity extends
    AppCompatActivity
{
    Button abutton = FindViewById
            (R.id.btn);

    abutton.setOnClickListener
        (new View.OnClickListener()
    {
        public void onClick(View view)
        {
            showAlertDialog();
        }
    });
```

```
Void showAlertDialog()
{
    AlertDialog.Builder builder =
        new AlertDialog.Builder
            (this);

    builder.setTitle("Alert Dialog").
        setMessage("Simple Alert
            Dialog");

    builder.setPositiveButton("ok",
        new DialogInterface.onClick
        Listerner()
    {
        void onClick(DialogInterface
            dialoginterface, int i)
        {

        }
    });

    builder.setNegativeButton("canc
        new DialogInterface.OnClick
        Listerner()
    {
        void onClick(DialogInterface
            dialoginterface, int i)
        {

        }
    });
```

```
        AlertDialog alertDialog =
                builder.create();

        alertDialog.show();
    }
}
```

* Explain Styles and Themes in Android.

=> Style: Style is used to defines the visual properties of UI components.

Every UI component, we can define different style.

This style is defined in style.xml file with is available in res/value Foldar.

Example:

```
<resources>
    <style name = "MyText" >

        <item name = "android:
            textColor" >#0077cc
        </item>
```

```
    <item name="android:textSize"
        >18sp</item>

    </style>
</resources>
```

You can use styles to define properties for individual UI components.

Styles can inherit in the Themes.

=> Themes:

Themes are a collection of style that are applied to an entire activity or application.

For every activity, we can define different Themes.

This Theme is defined in style.xml file which is available in res/values foldar.

Example:

```
<resources>
```

```
<style name = "AppTheme" >
    <item name = "colorPrimary" >
        #3F51B5 </item>
    <item name = "colorPrimaryDark">
        #303F9F </item>
    </style>
</resources>
```

You can use themes to define the overall appearance of your application.

Styles and Themes can inherit From each other.