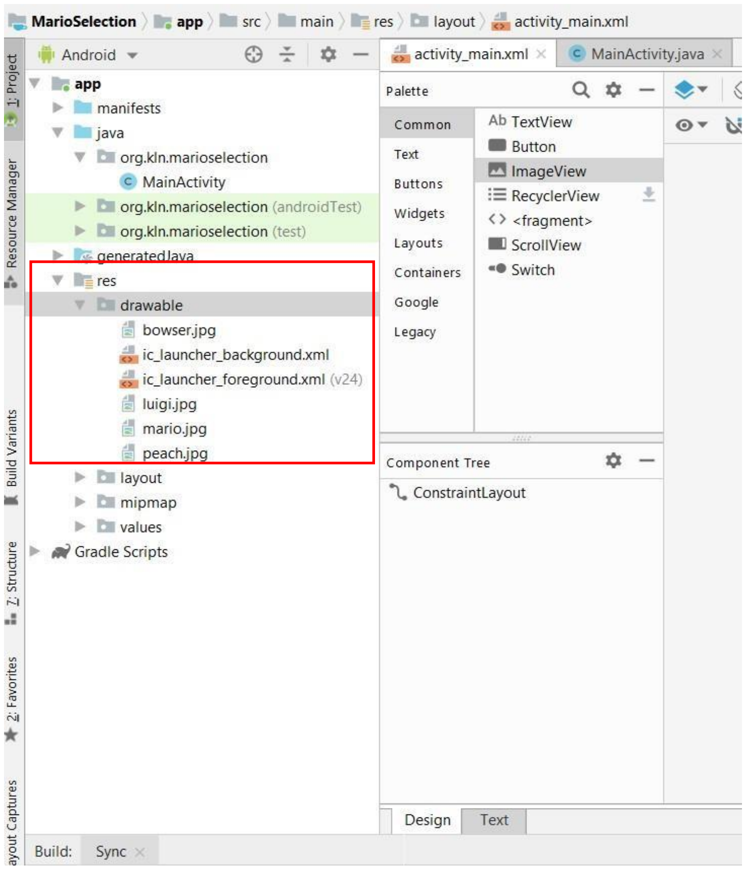
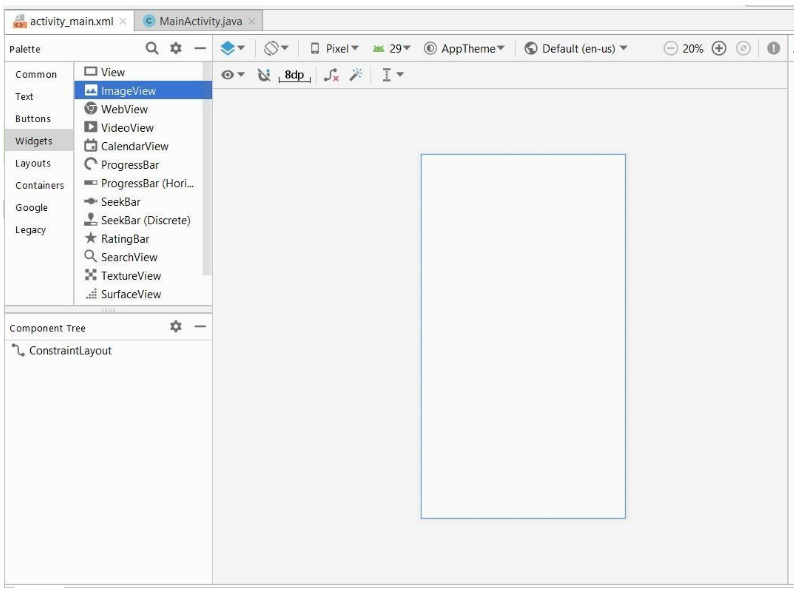
Practice Guide - 01

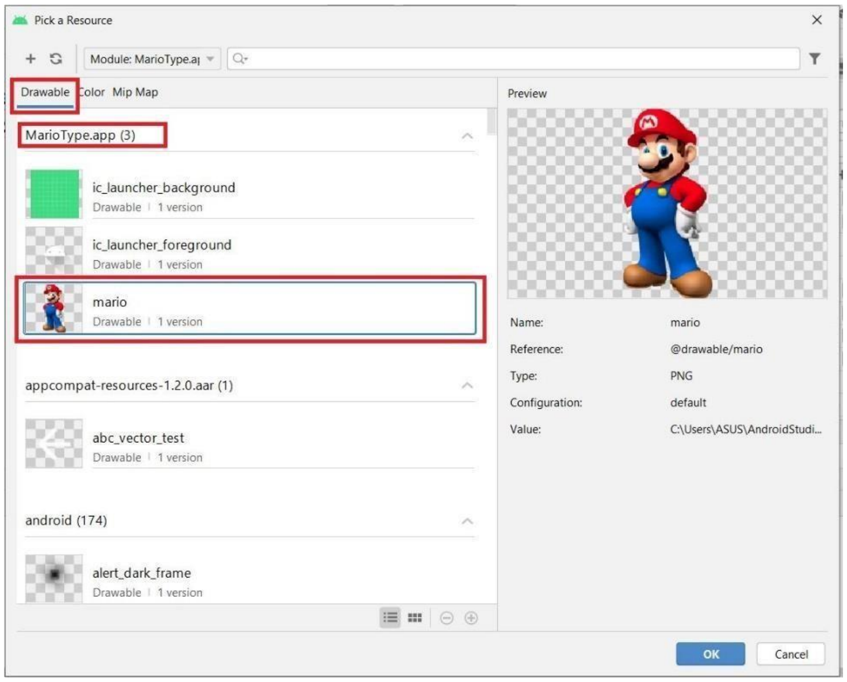
- Open android studio and create a new project with a "Empty Activity".
- Give a name for the project, package name and save it in an appropriate folder. After the project has successfully build open the "Android" project view and navigate to the "res" folder. Click on the "drawable" folder under the "res" folder. Copy four images from your computer to the "drawable" folder.



- Go to the "activity_main.xml" of your project and select the design tab. From the "Palette" section click on the "Widgets" and select the "ImageView" widget. Drag and drop an "ImageView" to the mobile layout.
- Next the IDE will pop open a "Pick a resource" popup indicating to select an image for the ImageView widget.

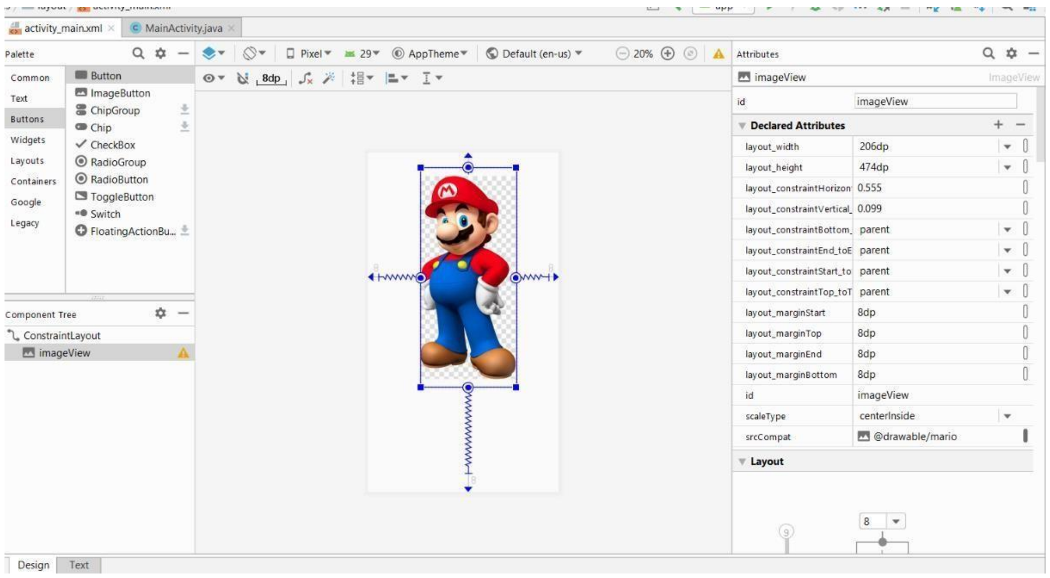


- Click on the “Project_name.app” section of the "Drawable" tab and select an image you added under the drawable folder and click ok.
- Resize the image to the size you want and position the widget according to the parents’ view.



Name: mario
Reference: @drawable/mario
Type: PNG
Configuration: default
Value: C:\Users\ASUS\AndroidStudi...

- You can try out different widget attributes and center the image as you want. Provide an “Id” to the imageview as “mario_images” (You can change this according to the images your choosing).



Attributes

imageView — ImageView

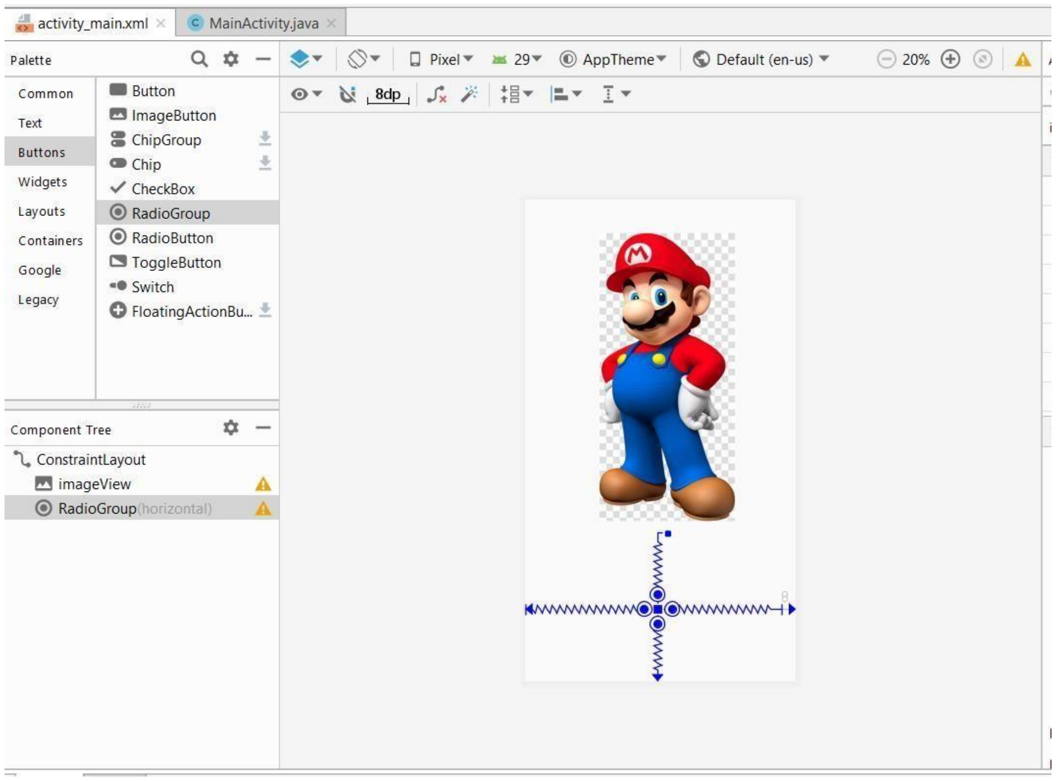
id	imageView

Declared Attributes

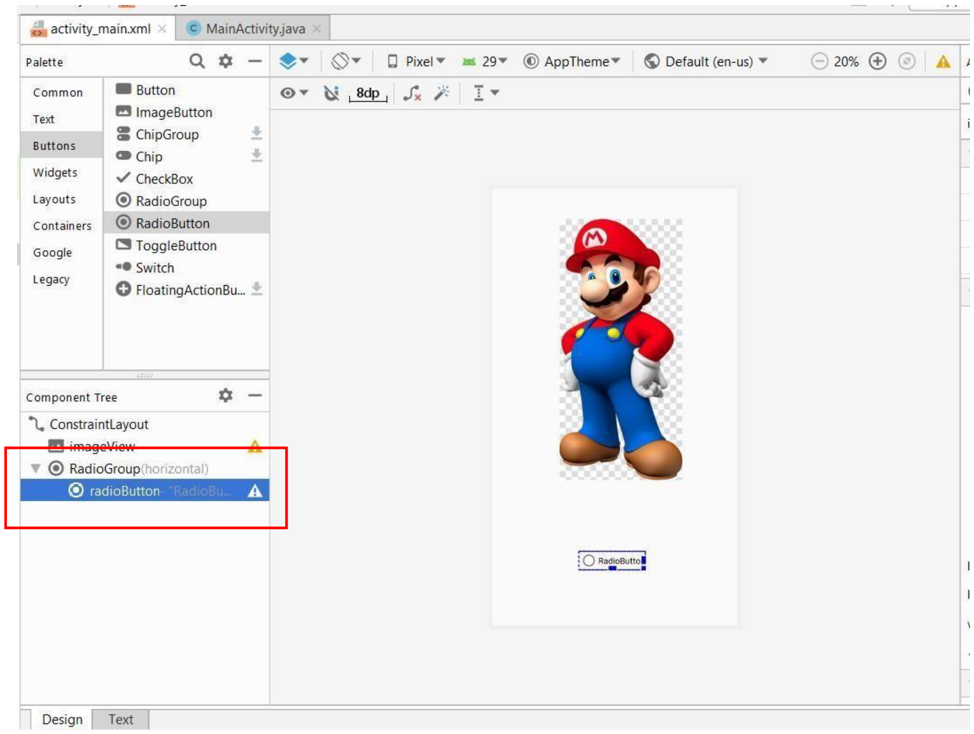
layout_width	206dp
layout_height	474dp
layout_constraintHorizon	0.555
layout_constraintVertical_	0.099
layout_constraintBottom_	parent
layout_constraintEnd_toE	parent
layout_constraintStart_to	parent
layout_constraintTop_toT	parent
layout_marginStart	8dp
layout_marginTop	8dp
layout_marginEnd	8dp
layout_marginBottom	8dp
id	imageView
scaleType	centerInside
srcCompat	@drawable/mario

Layout

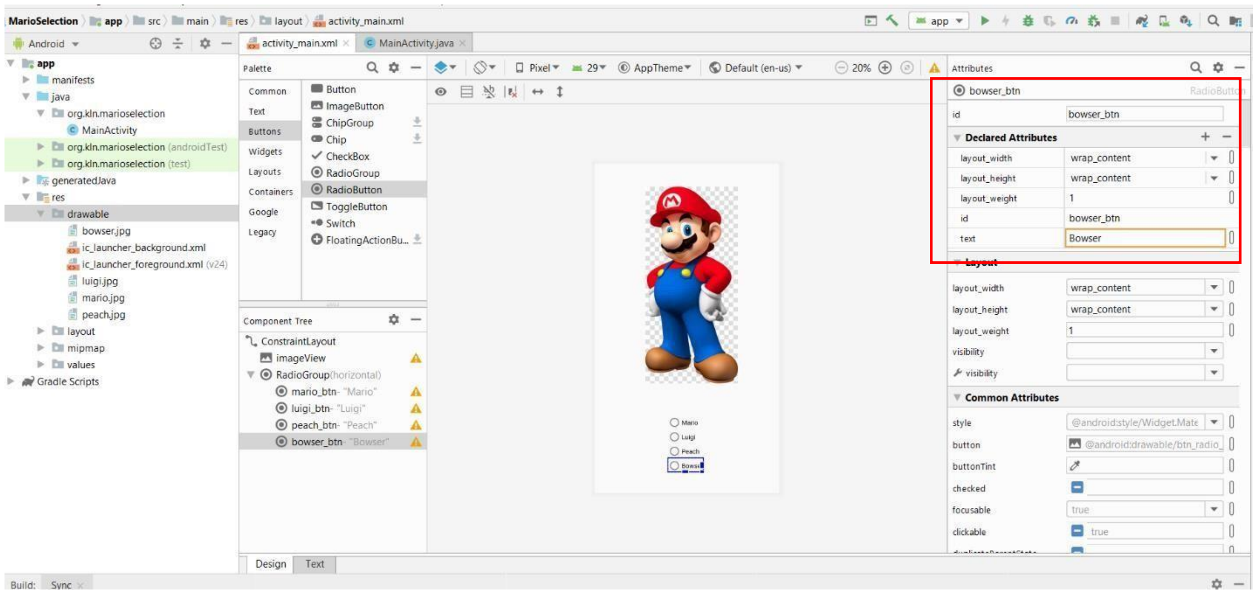
- Next click on the “Buttons” Palette and select the “RadioGroup” widget. Drag and drop the “RadioGroup” into the layout. Position the widget under the ImageView.



- Then click on “RadioButton” widget and drag it on to the layout and include it inside the “RadioGroup”. Under the “Component Tree” panel you will be able to see if the RadioButton is under the RadioGroup. If not drag the RadioButton into the RadioGroup using the Component tree panel itself.



- Give the radio button a "Text" and "Id" from the widget attribute section.
- Add three more "RadioButtons" under the "RadioGroup" and provide suitable "Text" and "Id" values.



bowser_btn

id	bowser_btn

Declared Attributes

layout_width	wrap_content
layout_height	wrap_content
layout_weight	1
id	bowser_btn
text	Bowser

Layout

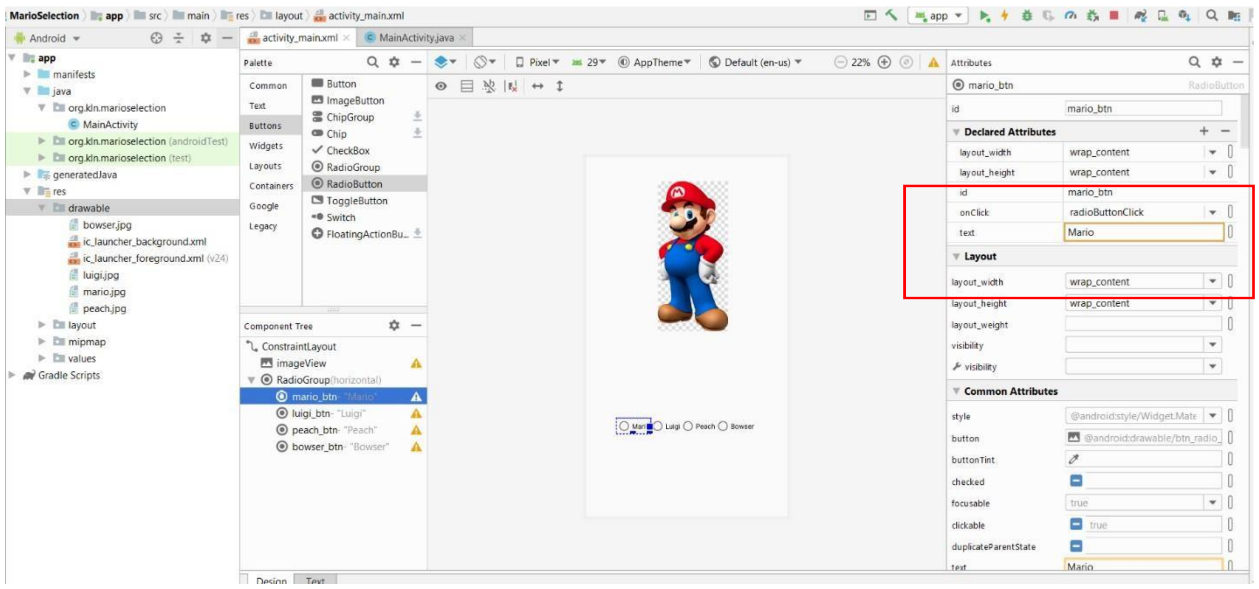
layout_width	wrap_content
layout_height	wrap_content
layout_weight	1
visibility	
visibility	

Common Attributes

style	@android:style/Widget.Mate
button	@android:drawable/btn_radio_
buttonTint	
checked	
focusable	true
clickable	true

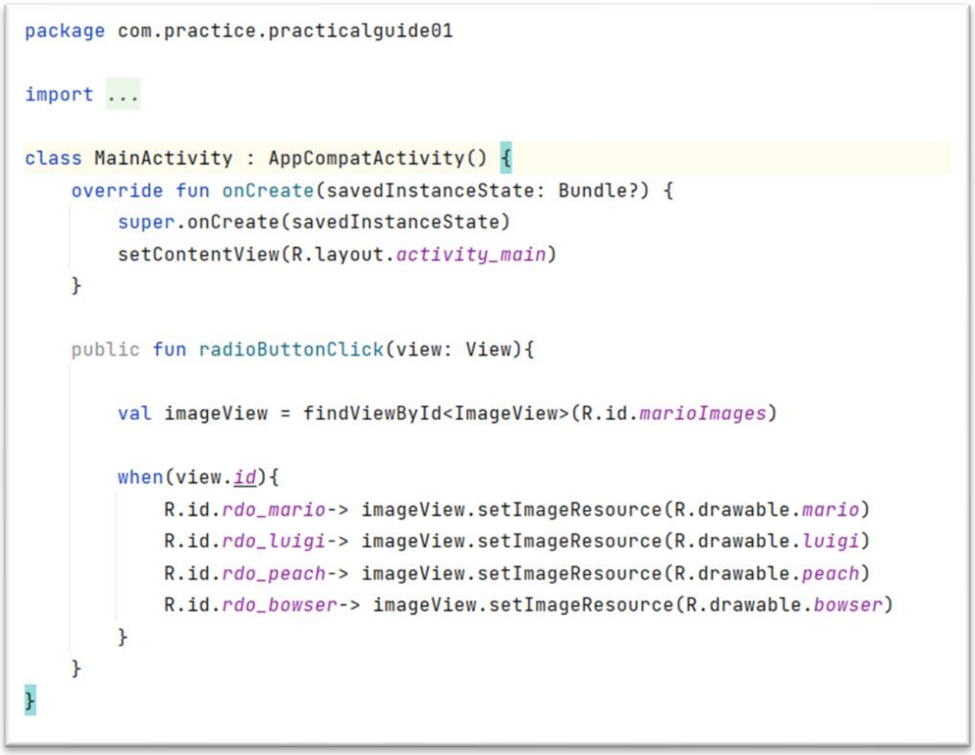
- Now you can go to the "MainActivity.kt" class and write methods to switch between images based on the radio button click.
- Write a public method named "radioButtonClick" and pass a "View" parameter. Navigate to the "activity_main.xml" and add this method to each of the "RadioButton" widgets under the "onClick" attribute.

- You could change the “RadioGroup” orientation to “horizontal” if you want or you could keep the default orientation to “vertical”.



Declared Attributes	
layout_width	wrap_content
layout_height	wrap_content
id	mario_btn
onClick	radioButtonClick
text	Mario

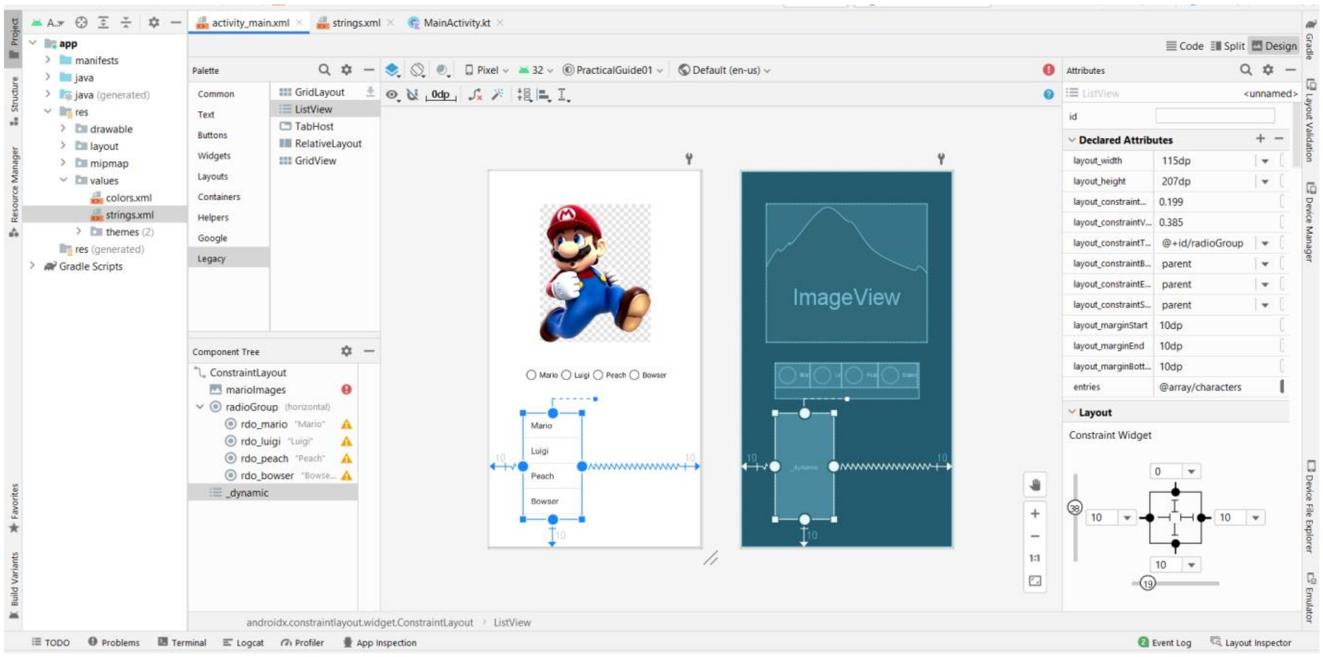
- Under the “radioButtonClick” method write the logic to determine which radio button was clicked. For this you can use the view argument that was passed. View contains the Id of the widget that was clicked.
- Use this Id and the radio button Id to compare what button was clicked. According to that you can set the image of the “ImageView” to match the button click.



```
package com.practice.practicalguide01

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    public fun radioButtonClick(view: View){

        val imageView = findViewById<ImageView>(R.id.marioImages)

        when(view.id){
            R.id.rdo_mario-> imageView.setImageResource(R.drawable.mario)
            R.id.rdo_luigi-> imageView.setImageResource(R.drawable.luigi)
            R.id.rdo_peach-> imageView.setImageResource(R.drawable.peach)
            R.id.rdo_bowser-> imageView.setImageResource(R.drawable.bowser)
        }
    }
}
```

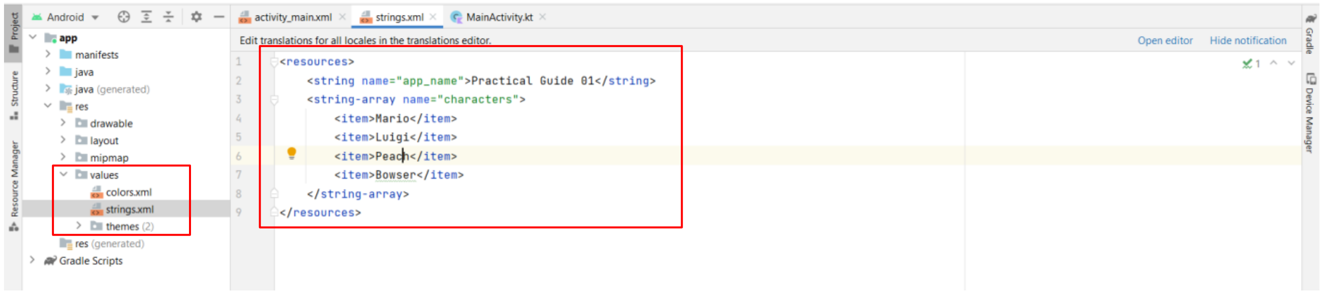
Let's work with a ListView now.

- Navigate to the "activity_main.xml" file. There you will find the "TextView" then drag and drop the text view and change the text as "simple_ListView". Then under the palette, click on the legacy. There you will find the "ListView" widget. Drag and drop the "ListView" widget to the bottom of the layout. Now adjust the positioning of the widget as before.



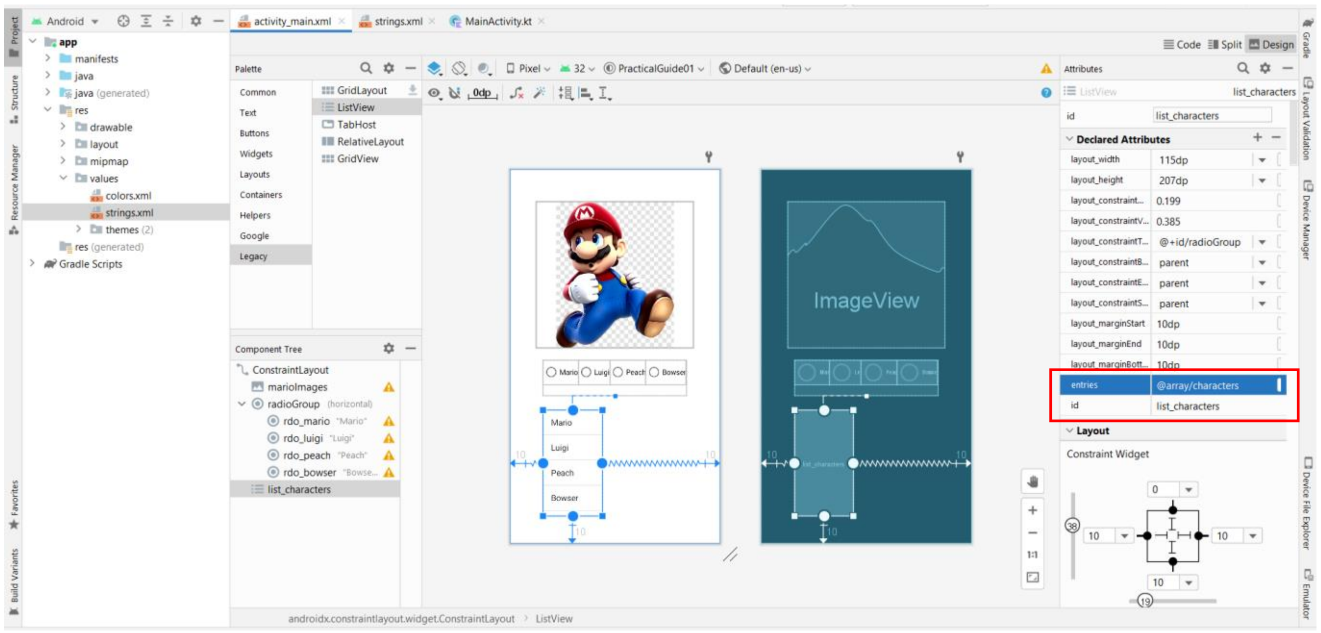
layout_width	115dp
layout_height	207dp
layout_constraint...	0.199
layout_constraintV...	0.385
layout_constraintT...	@+id/radioGroup
layout_constraintB...	parent
layout_constraintE...	parent
layout_constraintS...	parent
layout_marginStart	10dp
layout_marginEnd	10dp
layout_marginBott...	10dp
entries	@array/characters

- To populate the ListView we need to create an array.
- If the items in the array are known we can create it in the "string.xml" file under the "res" ->"values" folder. Use the "string-array" element and assign its name as "characters". These are static lists.



```
<resources>
    <string name="app_name">Practical Guide 01</string>
    <string-array name="characters">
        <item>Mario</item>
        <item>Luigi</item>
        <item>Peach</item>
        <item>Bowser</item>
    </string-array>
</resources>
```

- Next go to the “activity_main.xml” and using the “Text” tab click the “ListView” widget. Under the attributes section you can insert the “entries” attribute as “@array/characters”.
- To access the ListView you need to provide an id for your list as “list_characters”. Then call the list from the “MainActivity.kt” class.

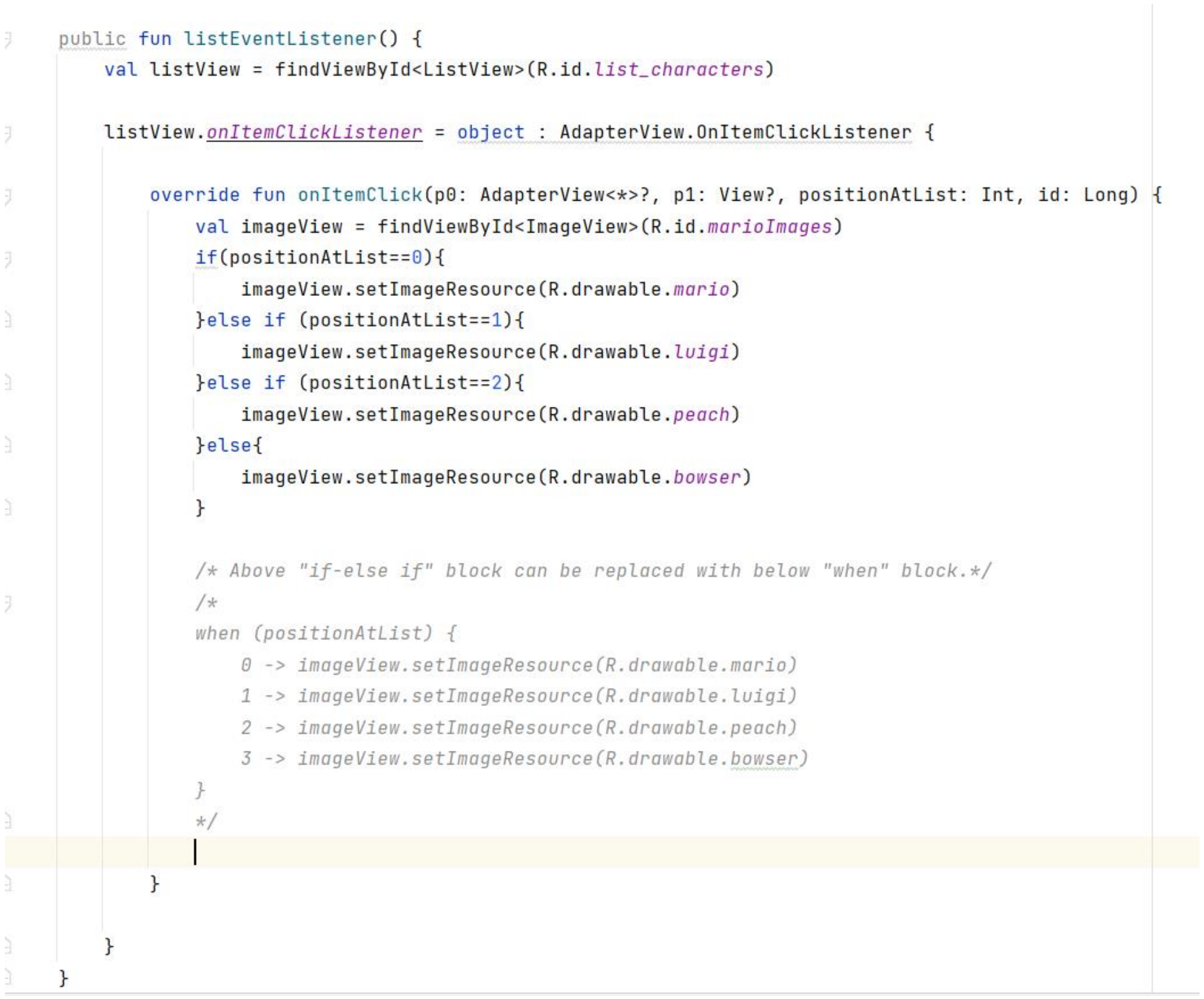


id	list_characters
layout_width	115dp
layout_height	207dp
layout_constraint...	0.199
layout_constraintV...	0.385
layout_constraintT...	@+id/radioGroup
layout_constraintB...	parent
layout_constraintE...	parent
layout_constraintS...	parent
layout_marginStart	10dp
layout_marginEnd	10dp
layout_marginBott...	10dp
entries	@array/characters
id	list_characters

- We can create a separate method to call the ListView listener event. First the ListView widget object needs to be created.

```kotlin
val listView = findViewById<ListView>(R.id.list_characters)
```

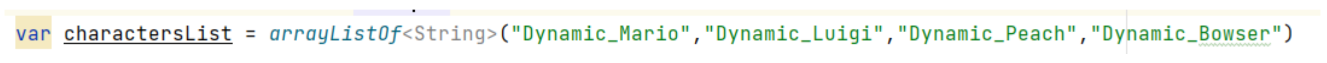
- Next an event listener will be attached to this listView object reference.
- As the parameters of the OnItemClickListener() an anonymous function under the “AdapterView” class will be called. Under the “OnItemClickListener” interface “onItemClick” method has to be overridden. Hence inside this method the listener logic will be written.
- Finally, call the “listEventListener” from the “onCreate” method.



```
public fun listEventListener() {
    val listView = findViewById<ListView>(R.id.list_characters)

    listView.onItemClickListener = object : AdapterView.OnItemClickListener {

        override fun onItemClick(p0: AdapterView<*>?, p1: View?, positionAtList: Int, id: Long) {
            val imageView = findViewById<ImageView>(R.id.marioImages)
            if(positionAtList==0){
                imageView.setImageResource(R.drawable.mario)
            }else if (positionAtList==1){
                imageView.setImageResource(R.drawable.luigi)
            }else if (positionAtList==2){
                imageView.setImageResource(R.drawable.peach)
            }else{
                imageView.setImageResource(R.drawable.bowser)
            }

            /* Above "if-else if" block can be replaced with below "when" block.*/
            /*
            when (positionAtList) {
                0 -> imageView.setImageResource(R.drawable.mario)
                1 -> imageView.setImageResource(R.drawable.luigi)
                2 -> imageView.setImageResource(R.drawable.peach)
                3 -> imageView.setImageResource(R.drawable.bowser)
            }
            */

        }

    }
}
```

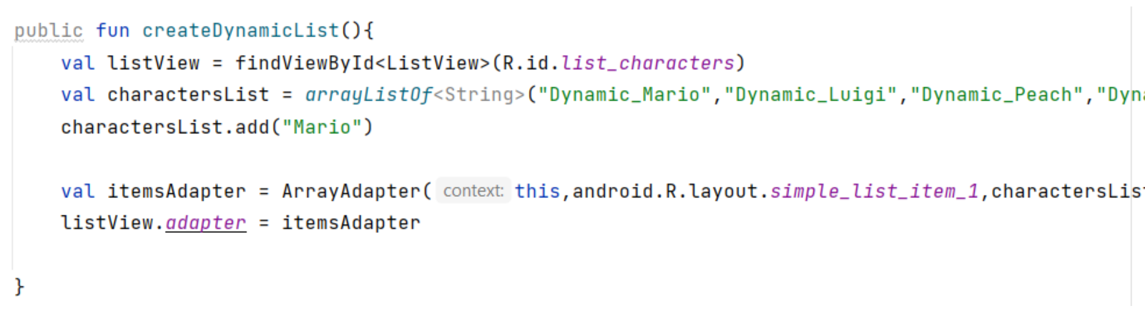
Creating Dynamic Lists

- Write a new method in the “MainActivity.kt” class as “createDynamicList”.
- In the method initialize an ArrayList with the reference as “charactersList”. Include any characters to the arraylist.



```
var charactersList = arrayListOf<String>("Dynamic_Mario","Dynamic_Luigi","Dynamic_Peach","Dynamic_Bowser")
```

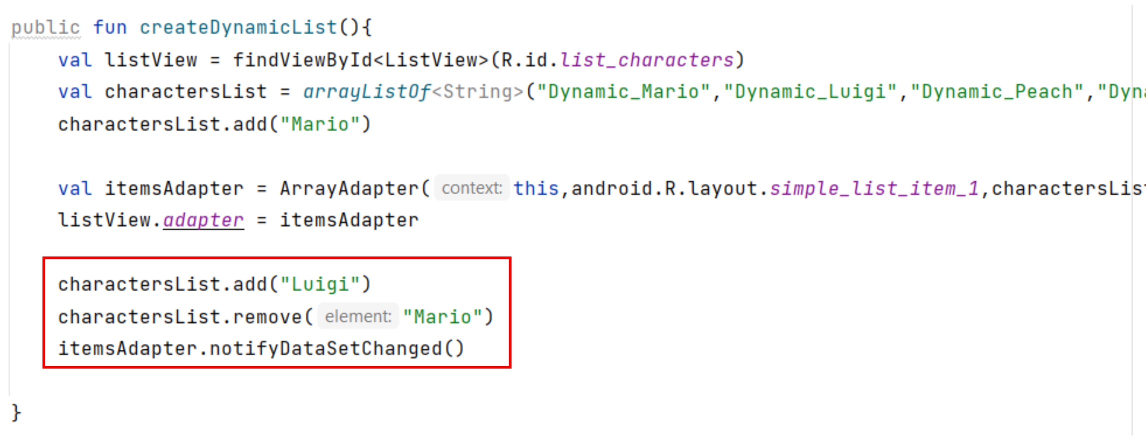
- Then include the array list values to the ListView using the “ArrayAdapter” class. Finally set the array adapter object created by setting the reference to the listView object.



```
public fun createDynamicList(){
    val listView = findViewById<ListView>(R.id.list_characters)
    val charactersList = arrayListOf<String>("Dynamic_Mario","Dynamic_Luigi","Dynamic_Peach","Dynamic_Bowser")
    charactersList.add("Mario")

    val itemsAdapter = ArrayAdapter( context: this,android.R.layout.simple_list_item_1,charactersList)
    listView.adapter = itemsAdapter

}
```

- Finally, then call the “createDynamicList” from the “onCreate” method.

- If you want to add or remove an item from the list you have to change the array list first. Then you need to call “notifyDataSetChanged” method on the array adapter.



```
public fun createDynamicList(){
    val listView = findViewById<ListView>(R.id.list_characters)
    val charactersList = arrayListOf<String>("Dynamic_Mario","Dynamic_Luigi","Dynamic_Peach","Dynamic_Bowser")
    charactersList.add("Mario")

    val itemsAdapter = ArrayAdapter( context: this,android.R.layout.simple_list_item_1,charactersList)
    listView.adapter = itemsAdapter

    charactersList.add("Luigi")
    charactersList.remove( element: "Mario")
    itemsAdapter.notifyDataSetChanged()

}
```

- Try adding new characters and check the list view.

TRY:
Try to add characters using “Plain Text” component in the palette and after button click. Then check how the list changes.