



## Contents

- Basic Knowledge
- Release Dates of Versions
- Software Language of Android Studio
- Install and Update an Application to Google Play





## Basic Knowledge

The key features that Android Studio offers to code developers are:

- A drag-and-drop rich editor that simplifies screen designs.
- Test tools and frameworks that can control application performance, availability, and interoperability in different versions.
- Easy and secure APK signing.
- Ability to add Google services to the application without the need for additional engagement.

A typical Android application consists of activity, template and resource files. Templates indicate how the application will look.

The activities indicate what the application will do and how it will respond to user behavior.

Every new version of Android uses a dessert name. Google puts their names in alphabetical order.

- Android 1.5, Cupcake: 27 April 2009(First Version)
- Android 6.0-6.0.1, Marshmallow: 5 October 2015
- Android 7.0-7.1.2, Nougat: 22 August 2016
- Android 8.0-8.1, Oreo: 21 August 2017
- Android 9.0 P , Pie 6 August 2018(Last Version)

The Android Emulator simulates Android devices on your computer, so you can test your application on various devices without need for each device.

## Android Studio Software Language

Java is an open-source, object-oriented, ground-independent, highly efficient, multifunctional, high-level, step-by-step language developed by James Gosling, a Sun Microsystems engineer.

```

public class Classname
{
public variabletype variablename;

//constructor Classname() { }

//constructor Classname(variabletype newvalue) {variabletype = newvalue; }

//metot void Classname() { }

}

String yazi = et.getText().toString();

if(yazi.contentEquals("4")){

tv.setText(«True Answer"); }

else if(yazi.contentEquals("5")){

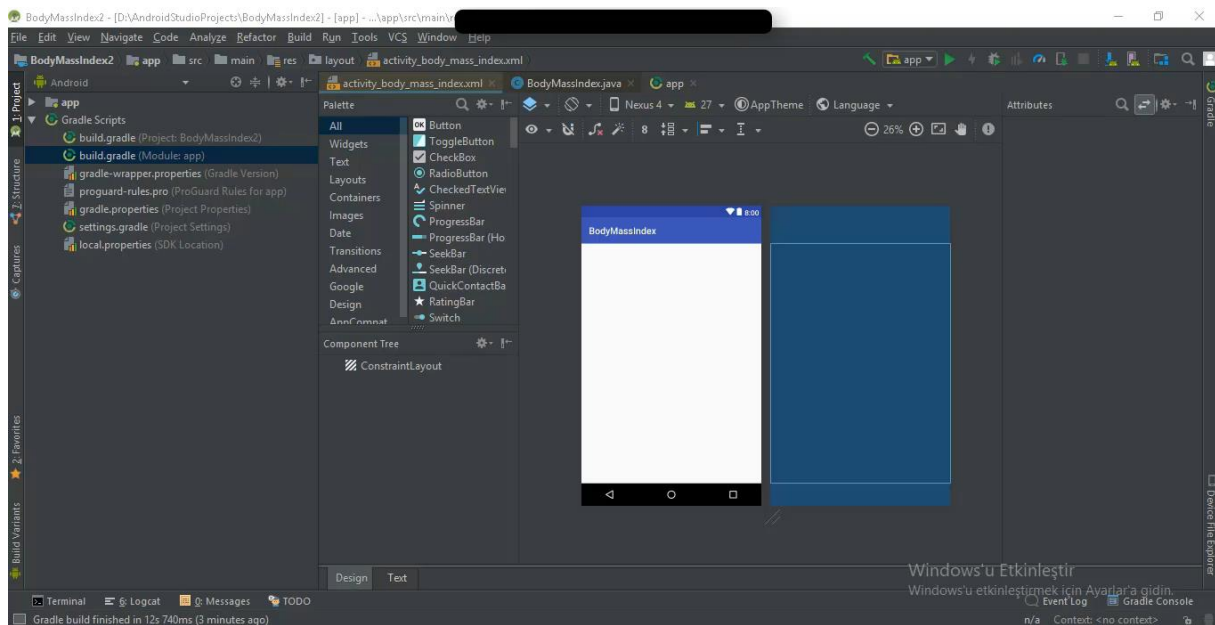
tv.setText(«So Close Mate"); }

else else if(yazi.contentEquals("10")){ tv.setText(«You are Far a way"); }

{ tv.setText(«False Answer");

}

```



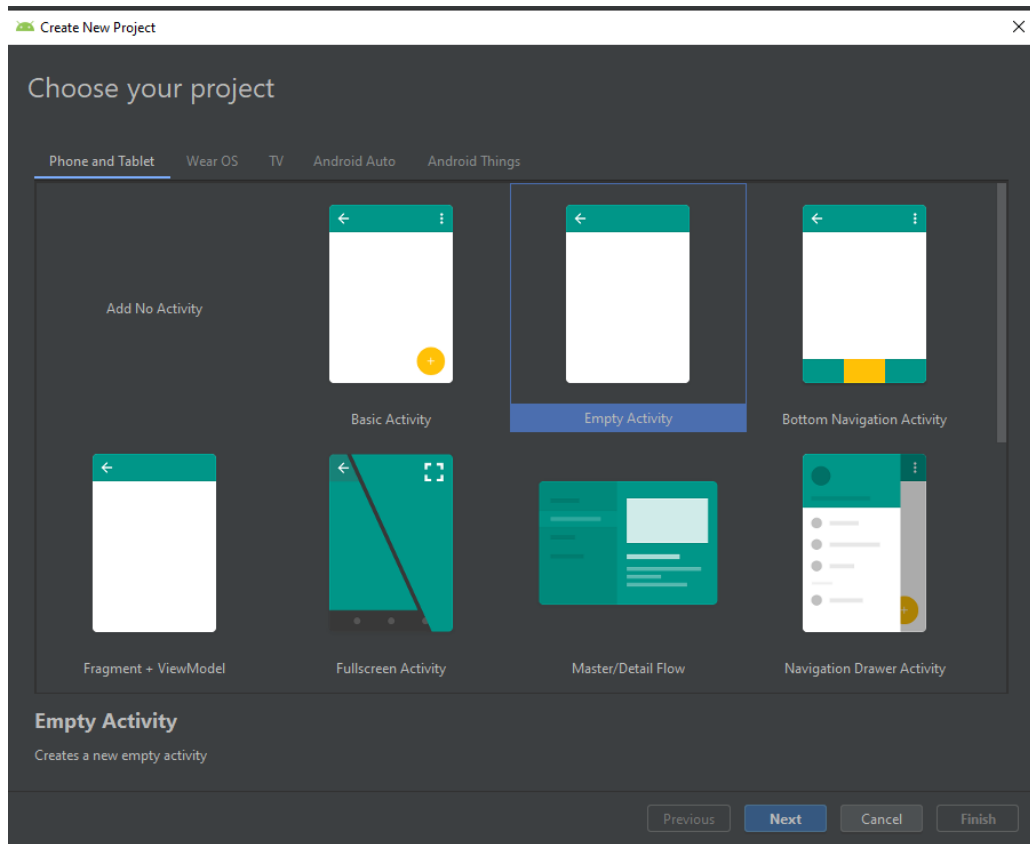
## Install and Uptade an Application to Google Play

After you have written and packaged our Android application, the process of bringing together users now occurs.

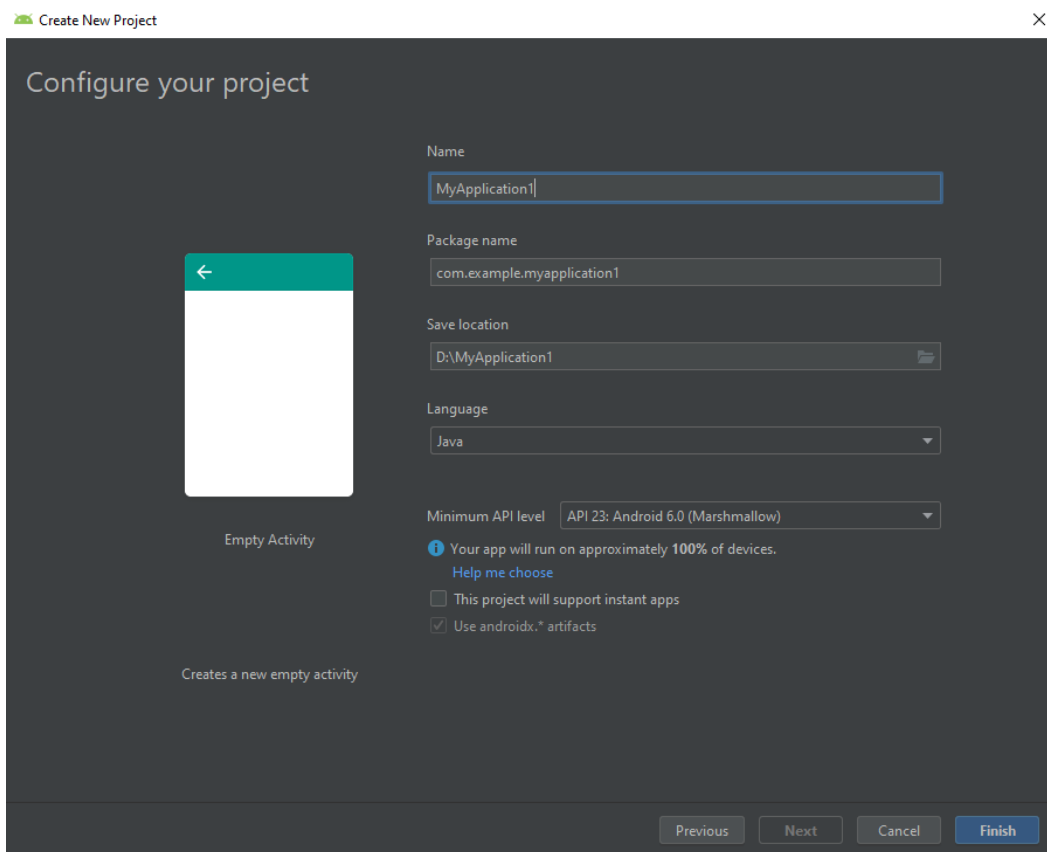
We need to use the Google Play Developer Console, which is also available from Google, to send the app to the Google Play app store available on every Android device.

# MANUAL FOR THE FIRST PROJECT

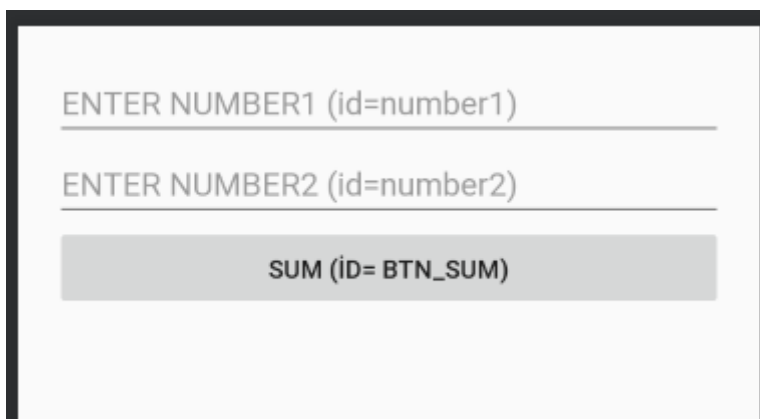
First we have to create a new project.



You can only use English and a few selected Special characters ( like " \_ " this one ) for the Name and Save Location of your project.



We are going to do the program's design page as seen under this sentence.  
(xml page)



We have to add "layout(vertical)" first to do this, after adding that then we need to add 2 numbers from the section called "Text" and after that we need to add a "Button" from the section called "common" under it in to the "layout(vertical)" section and a "TextView" from the Text menu (you can find the "layout(vertical)" section on down left screen you have to add all items inside of it).

After doing those you can change every specs of the items you've added like their id numbers (the names of the added things that only the programmer sees), the space between the items (Margin) and it's content (the words that are inside of the item) etc. If you want to make your design look exactly like

on the shown picture above and work perfectly with the code below you have to change it's id numbers, content and add hints (to be able to make those gray sentences, also those gray sentences will be gone when you type something in those places) as seen on the above picture. You can also find the id numbers and contents of the items you have to change for the each item on the above picture. To be honest learning programming is kinda hard by only reading but that's why we are here, right ? ;)

## Program's Java code(You can copy and paste it to the page called MainActivity.java).

```
package com.example.sum;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    EditText java_number1,java_number2;
    Button java_btn_sum;
    TextView java_result;
    Double java_sum;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        java_number1=findViewById(R.id.number1);
        java_number2=findViewById(R.id.number2);
        java_btn_sum=findViewById(R.id.btn_islem);
        java_result=findViewById(R.id.result);

        java_btn_sum.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                java_sum=Double.parseDouble(java_number1.getText().toString()+
                Double.parseDouble(java_number2.getText().toString()));
                java_result.setText("Sum="+java_sum);
            }
        });
    }
}
```

```
    }  
  });  
  
}  
}
```

At last we are going to run the program. Run key shortcut (Shift+F10)

