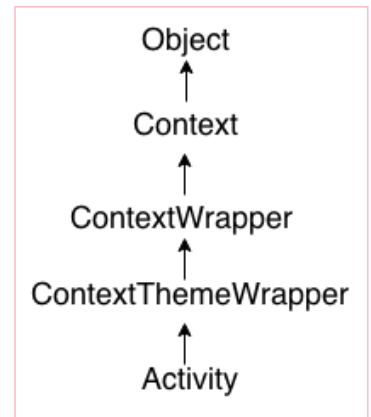


## Android Activity Lifecycle

**Android Activity Lifecycle** is controlled by 7 methods of `android.app.Activity` class. The android Activity is the subclass of `ContextThemeWrapper` class.

An activity is the single screen in android. It is like window or frame of Java.

By the help of activity, you can place all your UI components or widgets in a single screen.



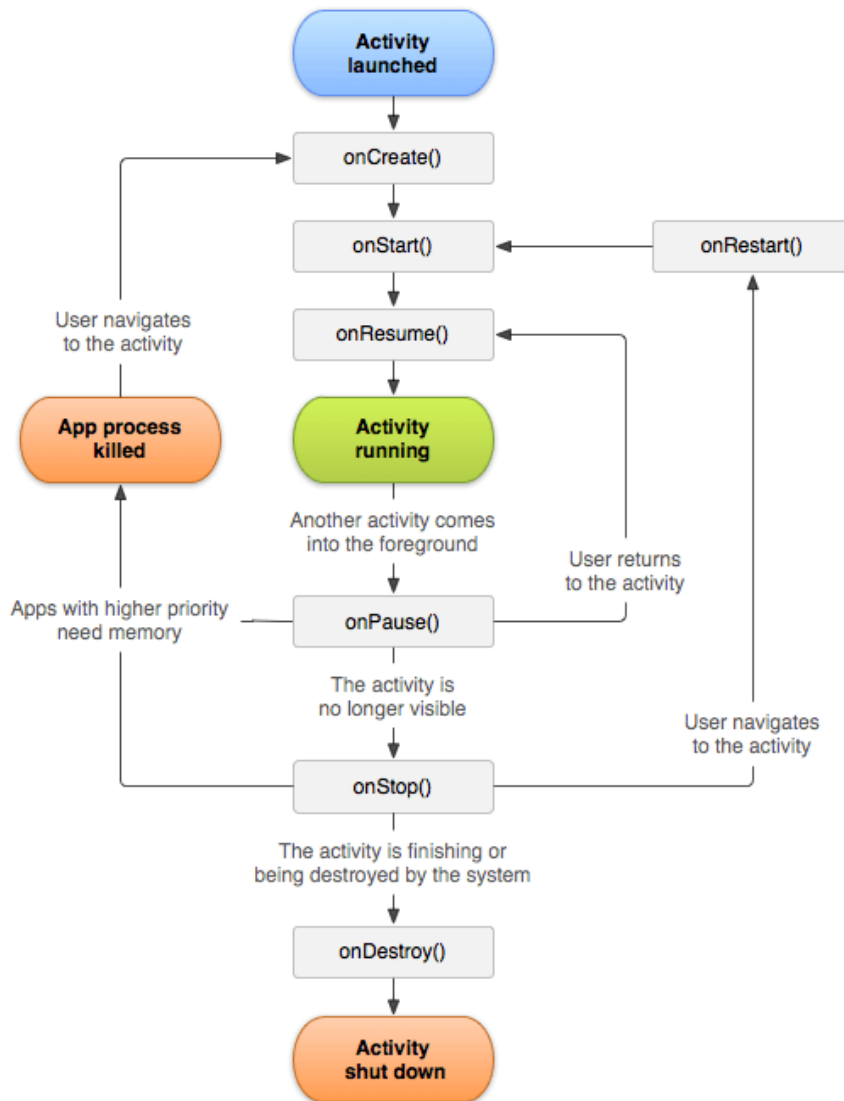
The 7 lifecycle method of Activity describes how activity will behave at different states.

## Android Activity Lifecycle methods

Let's see the 7 lifecycle methods of android activity.

Method	Description
<b>onCreate</b>	called when activity is first created.
<b>onStart</b>	called when activity is becoming visible to the user.
<b>onResume</b>	called when activity will start interacting with the user.
<b>onPause</b>	called when activity is not visible to the user.

<b>onStop</b>	called when activity is no longer visible to the user.
<b>onRestart</b>	called after your activity is stopped, prior to start.
<b>onDestroy</b>	called before the activity is destroyed.



 [Advertisement](#)



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="example.javatpoint.com.activitylifecycle.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

## Android Activity Lifecycle Example

It provides the details about the invocation of life cycle methods of activity. In this example, we are displaying the content on the logcat.

File: *MainActivity.java*

```
package example.javatpoint.com.activitylifecycle;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle", "onCreate invoked");
    }

    @Override
```



```
protected void onStart() {  
    super.onStart();  
    Log.d("lifecycle","onStart invoked");  
}  
  
@Override  
protected void onResume() {  
    super.onResume();  
    Log.d("lifecycle","onResume invoked");  
}  
  
@Override  
protected void onPause() {  
    super.onPause();  
    Log.d("lifecycle","onPause invoked");  
}  
  
@Override  
protected void onStop() {  
    super.onStop();  
    Log.d("lifecycle","onStop invoked");  
}  
  
@Override  
protected void onRestart() {  
    super.onRestart();  
    Log.d("lifecycle","onRestart invoked");  
}  
  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Log.d("lifecycle","onDestroy invoked");  
}  
}
```

### Output:

You will not see any output on the emulator or device. You need to open logcat.

