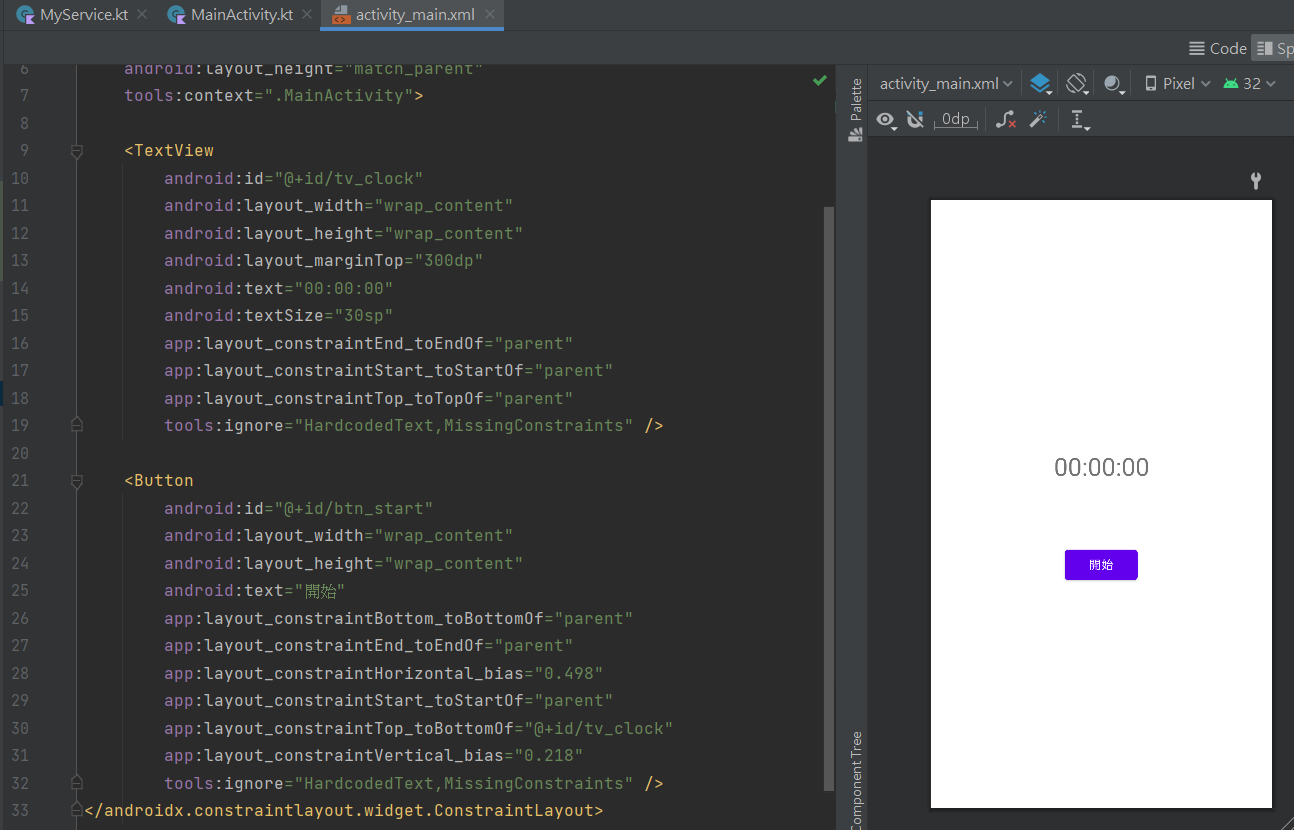
計時器



<TextView  
 android:id="@+id/tv\_clock"

//元件名為tv\_clock  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"

//元件寬高擴展到最大化  
 android:layout\_marginTop="300dp"

//距離上緣基準點300dp  
 android:text="00:00:00"  
 android:textSize="30sp"

//顯示文字00:00:00大小為30sp  
 app:layout\_constraintEnd\_toEndOf="parent"

//物件的下緣與另外一個物件的下緣對其  
 app:layout\_constraintStart\_toStartOf="parent"

//物件的起始邊與另外一個物件的起始邊對其  
 app:layout\_constraintTop\_toTopOf="parent"

//物件的上緣與另外一個物件的上邊對其  
 tools:ignore="HardcodedText,MissingConstraints" />

//忽略xml某些警告

<Button  
 android:id="@+id/btn\_start" //元件名為btn\_start  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"

//元件寬高擴展到最大化  
 android:text="開始" //顯示文字”開始”  
 app:layout\_constraintBottom\_toBottomOf="parent"

 //將此物件的邊與另一個物件的邊對齊  
 app:layout\_constraintEnd\_toEndOf="parent"

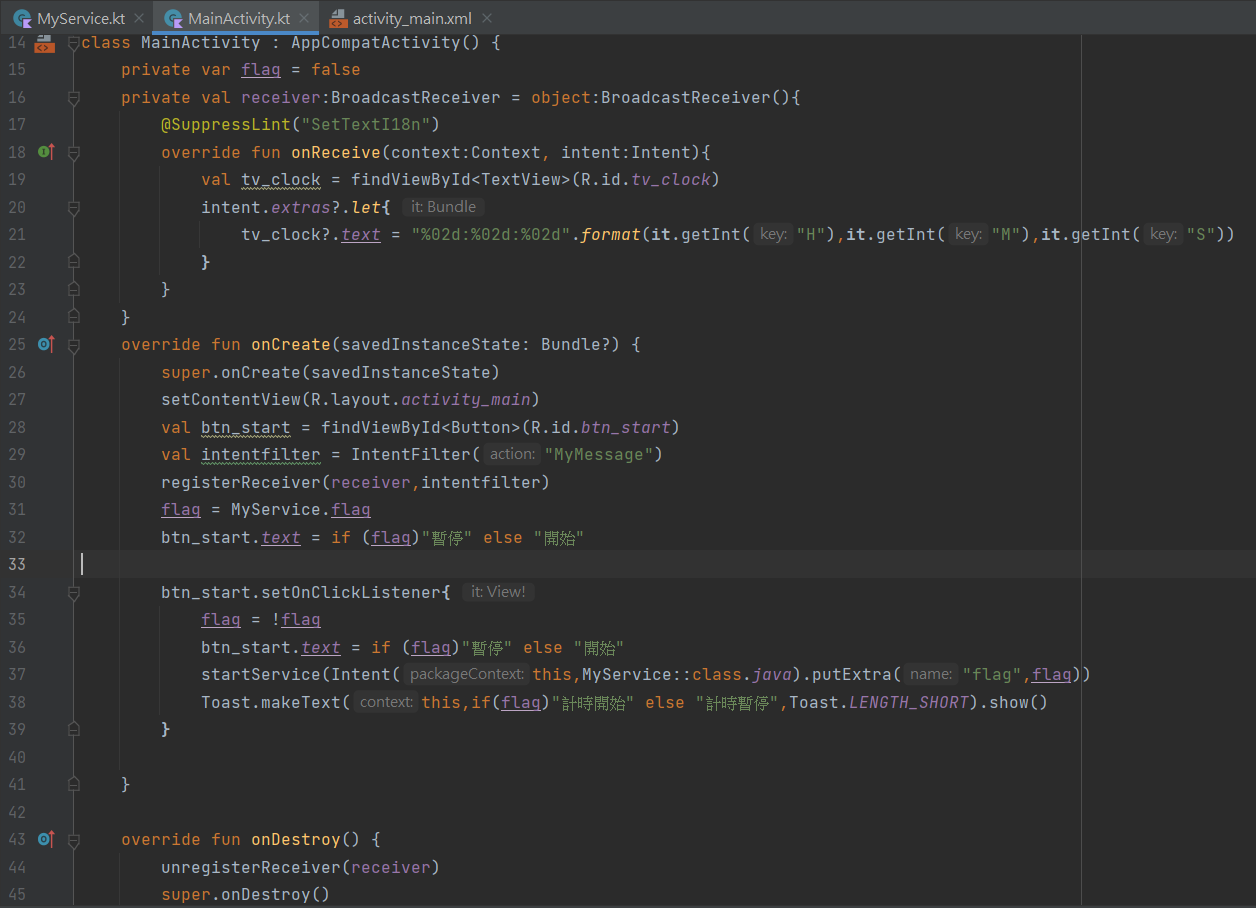
//物件的下緣與另外一個物件的下緣對其  
 app:layout\_constraintHorizontal\_bias="0.498"

//横向傾向  
 app:layout\_constraintStart\_toStartOf="parent"

//物件的起始邊與另外一個物件的起始邊對其  
 app:layout\_constraintTop\_toBottomOf="@+id/tv\_clock"

//物件的上邊與另外一個物件的底邊對其  
 app:layout\_constraintVertical\_bias="0.218"

//縱向的倾向  
 tools:ignore="HardcodedText,MissingConstraints" />

class MainActivity : AppCompatActivity() {  
 private var flag = false

//建立BroadcastReceiver物件  
 private val receiver:BroadcastReceiver = object:BroadcastReceiver(){

//在onReceive()中加入接收廣播後要執行的動作  
 @SuppressLint("SetTextI18n")  
 override fun onReceive(context:Context, intent:Intent){

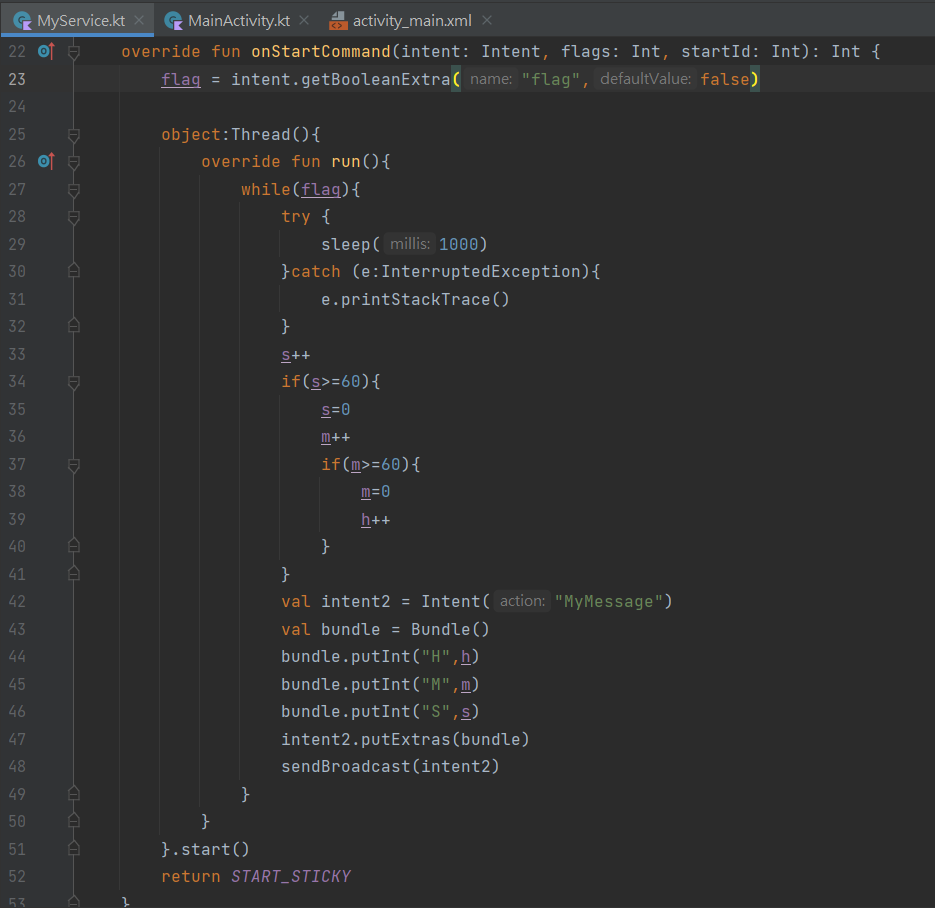
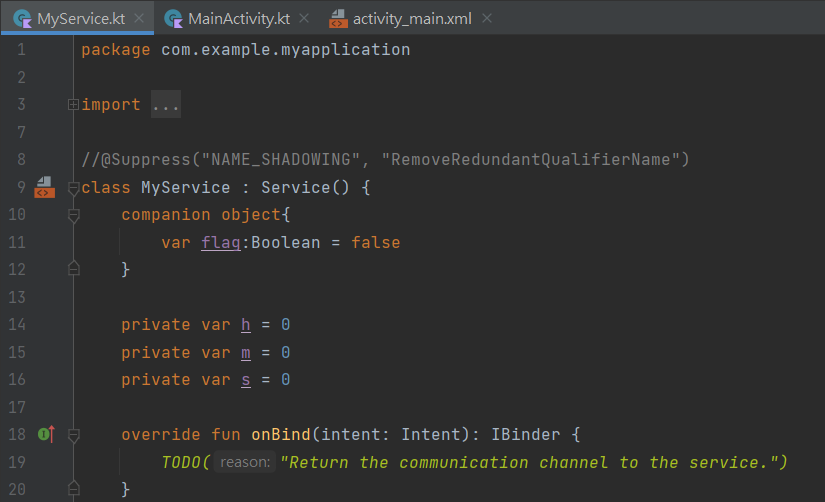
//解析Intent取得秒數資訊  
 val tv\_clock = findViewById<TextView>(R.id.*tv\_clock*)  
 intent.*extras*?.*let***{** tv\_clock?.*text* = "%02d:%02d:%02d".*format*(**it**.getInt("H"),**it**.getInt("M"),**it**.getInt("S"))  
 **}** }  
 }

override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)

//建立Intentfilter物件來指定要接收的識別字串MyMessage  
 val btn\_start = findViewById<Button>(R.id.*btn\_start*)  
 val intentfilter = IntentFilter("MyMessage")

//註冊Receiver  
 registerReceiver(receiver,intentfilter)

//取得Service狀態  
 flag = MyService.flag  
 btn\_start.*text* = if (flag)"暫停" else "開始"  
  
 btn\_start.setOnClickListener**{** flag = !flag  
 btn\_start.*text* = if (flag)"暫停" else "開始"  
//啟動Service startService(Intent(this,MyService::class.*java*).putExtra("flag",flag))  
 Toast.makeText(this,if(flag)"計時開始" else "計時暫停",Toast.*LENGTH\_SHORT*).show()  
 **}** }  
 override fun onDestroy() {  
 unregisterReceiver(receiver) //註銷廣播  
 super.onDestroy()  
 }  
}



class MyService : Service() {  
 companion object{  
 var flag:Boolean = false //計時器狀態  
 }

//計時器數值  
 private var h = 0  
 private var m = 0  
 private var s = 0  
 override fun onBind(intent: Intent): IBinder {  
 *TODO*("Return the communication channel to the service.")  
 }  
 override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {  
 flag = intent.getBooleanExtra("flag",false)  
  
 object:Thread(){  
 override fun run(){  
 while(flag){  
 try {

//使用Thread來計算秒數，延遲1秒  
 sleep(1000)  
 }catch (e:InterruptedException){  
 e.printStackTrace()  
 }

//計數器+1  
 s++  
 if(s>=60){  
 s=0  
 m++  
 if(m>=60){  
 m=0  
 h++  
 }  
 }

//產生帶MyMessage識別字串的Intent  
 val intent2 = Intent("MyMessage")

//把累加的值(經過的秒數)放入Intent  
 val bundle = Bundle()  
 bundle.putInt("H",h)  
 bundle.putInt("M",m)  
 bundle.putInt("S",s)  
 intent2.putExtras(bundle)

//發送廣播  
 sendBroadcast(intent2)  
 }  
 }  
 }.start()  
 return *START\_STICKY* }  
}