



殷志忠 Chih-Chung Yin
Google / Oracle 原廠認證講師
teach.yin@gmail.com



ORACLE®
Certified Professional
Java SE 8 Programmer

ORACLE®
Certified Expert
Java EE 6 Web
Component Developer

ORACLE®
Certified Professional
Java EE 7–
Application Developer

ViewPager2

新增 XML layout
question_layout.xml

新增 PagerActivity
activity_pager.xml
PagerActivity.kt

TabLayout

新增 XML layout
tab_fragment1_layout.xml
tab_fragment2_layout.xml

新增 TabActivity
activity_tab.xml
TabActivity.kt

Handler (週期任務、換圖片)

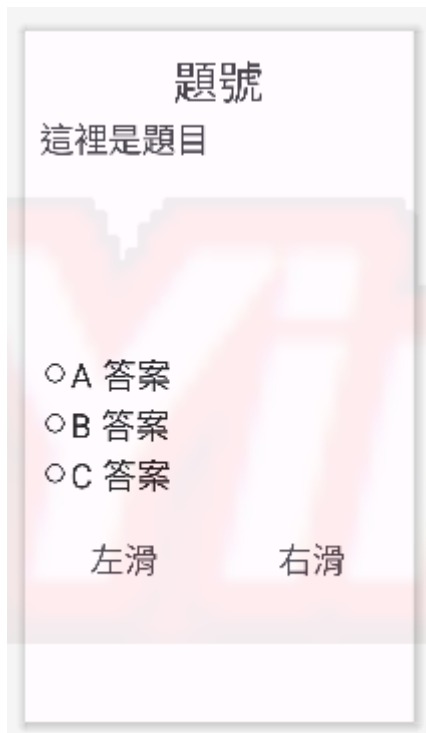
新增 AnimationActivity
activity_animation.xml
AnimationActivity.kt

ViewPager2

<https://developer.android.com/guide/navigation/navigation-swipe-view-2?hl=zh-tw>

新增 XML layout

question_layout.xml



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/tv_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="48sp"
        android:text="題號"
        android:layout_gravity="center"/>

    <TextView
        android:id="@+id/tv_question"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:textSize="36sp"
        android:text="這裡是題目"
        android:layout_gravity="center"/>

    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="center">
```

```
<RadioButton
```

```
    android:id="@+id/tv_op_a"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="36sp"  
    android:text="A 答案"/>
```

```
<RadioButton
```

```
    android:id="@+id/tv_op_b"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="36sp"  
    android:text="B 答案"/>
```

```
<RadioButton
```

```
    android:id="@+id/tv_op_c"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="36sp"  
    android:text="C 答案"/>
```

```
</RadioGroup>
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="36dp"  
    android:textSize="36sp"  
    android:layout_gravity="center"  
    android:text="左滑          右滑"/>
```

```
</LinearLayout>
```

新增 PagerActivity

activity_pager.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.viewpager2.widget.ViewPager2 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"  
    android:id="@+id/pager"  
    tools:context=".PagerActivity">  
  
</androidx.viewpager2.widget.ViewPager2>
```

PagerActivity.kt

```
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentActivity
import androidx.viewpager2.adapter.FragmentStateAdapter
import androidx.viewpager2.widget.ViewPager2

// 全域變數 (寫在類別外面, 任何程式都可取用)
const val POSITION_KEY = "POSITION_KEY"
// 資料類別
data class Q(
    val number: Int,
    val question: String,
    val op_a: String,
    val op_b: String,
    val op_c: String
)
// ArrayList 陣列
val qArrayList = arrayListOf<Q>(
    Q(1, "1+1=?", "1", "2", "3"),
    Q(2, "1+2=?", "1", "2", "3"),
    Q(3, "2-1=?", "1", "2", "3"),
)
class PagerActivity : AppCompatActivity() {
    // lateinit 稍後會設定初值, 確認未來不會是空值
    private lateinit var viewPager: ViewPager2

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_pager)

        viewPager = findViewById<ViewPager2>(R.id.pager)
        viewPager.adapter = MyPagerAdapter(this) // 設定 ViewPager2 資料適配器
    }
    // 資料適配器
    class MyPagerAdapter(fa: FragmentActivity) : FragmentStateAdapter(fa) {
        // 有多少項資料
        override fun getItemCount(): Int {
            return qArrayList.size
        }
        // 產生 片段畫面
        override fun createFragment(position: Int): Fragment {
            // 建立片段
            val qf = QuestionFragment()
            // 寄放資料
            qf.arguments = Bundle().apply {
                // 放 整數資料 目前位置
                putInt(POSITION_KEY, position)
            }
            return qf
        } // MyPagerAdapter
    } // Activity
}
```

```

// 片段類別 不能是 Activity 內部類別(inner class)
class QuestionFragment : Fragment() {
    // 建立片段畫面
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflater(畫面佈局 載入器)
        return inflater.inflate(R.layout.question_layout, container, false)
    }

    // 建立片段畫面之後
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        // 若有寄放資料
        arguments?.takeIf { it.containsKey(POSITION_KEY) }?.apply {
            val tv_number = view.findViewById<TextView>(R.id.tv_number)
            val tv_question = view.findViewById<TextView>(R.id.tv_question)
            val tv_op_a = view.findViewById<TextView>(R.id.tv_op_a)
            val tv_op_b = view.findViewById<TextView>(R.id.tv_op_b)
            val tv_op_c = view.findViewById<TextView>(R.id.tv_op_c)
            // 取得 寄放整數資料 目前位置
            val position = getInt(POSITION_KEY)
            // 從陣列取得題目資訊
            val q = qArrayList.get(position)
            // 設定要顯示的文字內容
            tv_number.text = q.number.toString()
            tv_question.text = q.question
            tv_op_a.text = q.op_a
            tv_op_b.text = q.op_b
            tv_op_c.text = q.op_c
        }
    }
} // Fragment

```

TabLayout



新增 XML layout

tab_fragment1_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textSize="36sp"
        android:gravity="center"
        android:text="這裡是片段1"/>
</LinearLayout>
```

tab_fragment2_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textSize="36sp"
        android:gravity="center"
        android:text="這裡是片段2"/>
</LinearLayout>
```

新增 TabActivity

activity_tab.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".TabActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="36sp"
        android:gravity="center"
        android:text="示範"/>

    <!-- TabLayout 必須在 ViewPager2 之前 -->
    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tab_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabMode="scrollable" />

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/tab_pager"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

    </androidx.viewpager2.widget.ViewPager2>
</LinearLayout>
```

TabActivity.kt

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentActivity
import androidx.viewpager2.adapter.FragmentStateAdapter
import androidx.viewpager2.widget.ViewPager2
import com.google.android.material.tabs.TabLayout
import com.google.android.material.tabs.TabLayoutMediator

class TabActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_tab)

        init() // 初始化
    }
}
```

```

private fun init() {
    val tab_layout = findViewById<TabLayout>(R.id.tab_layout)
    val tab_pager = findViewById<ViewPager2>(R.id.tab_pager)

    // 設定 ViewPager2 資料適配器
    tab_pager.adapter = MyPagerAdapter(this)
    // 連結 tab_layout 與 tab_pager
    TabLayoutMediator(tab_layout, tab_pager) { tab, position ->
        val s = "第" + position + "頁"
        tab.text = s // 設定 tab 標題字串
    }.attach()
}

// inner class(內部類別) 能使用外部類別私有成員
// 資料適配器
inner class MyPagerAdapter(fa: FragmentActivity) : FragmentStateAdapter(fa) {
    // 有多少項資料
    override fun getItemCount(): Int {
        return 2 // 有2頁
    }

    // 產生 片段畫面
    override fun createFragment(position: Int): Fragment {
        // 建立片段
        lateinit var f: Fragment
        when(position) {
            0 -> {
                f = TabFragment1()
            }
            1 -> {
                f = TabFragment2()
            }
        }
        return f
    }
} // MyPagerAdapter
} // Activity

```

```

// 片段類別 不能是 內部類別(inner class)
class TabFragment1: Fragment() {
    // 建立片段畫面
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflater(畫面佈局 載入器)
        return inflater.inflate(R.layout.tab_fragment1_layout, container, false)
    }
}

```

```

class TabFragment2: Fragment() {
    // 建立片段畫面
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflater(畫面佈局 載入器)
        return inflater.inflate(R.layout.tab_fragment2_layout, container, false)
    }
}

```


Handler (週期任務、換圖片)

新增 AnimationActivity

activity_animation.xml



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".AnimationActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal"
        android:padding="16dp">

        <Button
            android:id="@+id/btn_start"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="start" />

        <Button
            android:id="@+id/btn_stop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="stop" />

        <Button
            android:id="@+id/btn_clear"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="clear" />

    </LinearLayout>

    <TextView
        android:id="@+id/tv_message"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="目前沒有訊息"
        android:textSize="28sp" />

<ImageView
    android:id="@+id/img"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center" />
</LinearLayout>

```

AnimationActivity.kt

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.view.View
import android.view.View.OnClickListener
import android.widget.Button
import android.widget.ImageView
import android.widget.TextView

class AnimationActivity : AppCompatActivity() {
    // lateinit 稍後會設定初值 不會是空值
    lateinit var img_arraylist: ArrayList<Int>

    lateinit var btn_start: Button
    lateinit var btn_stop: Button
    lateinit var btn_clear: Button

    lateinit var tv_message: TextView
    lateinit var img: ImageView

    lateinit var handler: Handler // 任務 管理員
    lateinit var task: MyTask // 任務

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_animation)

        init() // 初始化(初值設定)
    }

    fun init() {
        // 陣列 圖片id
        img_arraylist = arrayListOf<Int>(
            R.drawable.p01,
            R.drawable.p02,
            R.drawable.p03,
        )

        img = findViewById(R.id.img)
        img.setBackgroundResource(R.drawable.no_image_box) // 設定圖片

        tv_message = findViewById(R.id.tv_message)

        btn_start = findViewById(R.id.btn_start)
        btn_stop = findViewById(R.id.btn_stop)
        btn_clear = findViewById(R.id.btn_clear)
    }
}

```

```

// 設定 按鈕 監聽器
btn_start.setOnClickListener(StartOnClick())
btn_stop.setOnClickListener(StopOnClick())
btn_clear.setOnClickListener(ClearOnClick())

// 建立 任務 管理員
handler = Handler(Looper.getMainLooper())

// 建立 任務
task = MyTask()
}

// 任務
inner class MyTask : Runnable { // Runnable 任務
    var count = 0 // 換圖次數
    var i = 0 // 第幾張圖
    var isDone = false // 任務是否完成

    override fun run() {
        if (isDone) { // 如果任務完成
            return // 結束方法
        }
        btn_start.isEnabled = false // start按鈕 無作用 (不可按)
        // 顯示訊息
        tv_message.text = "i=" + i.toString() + " count=" + count
        // 顯示圖片
        val imgId = img_arraylist.get(i)
        img.setBackgroundResource(imgId)
        // 若 i 加 1 後 等於 陣列大小
        if (++i == img_arraylist.size) {
            i = 0
        }
        // 提交任務 1000毫秒(1秒) 後執行
        handler.postDelayed(this, 1000)

        // 若 count 加 1 後 大於 9
        if(++count > 9) {
            isDone = true // 任務完成
            btn_start.isEnabled = true // start按鈕 有作用 (可按)
        }
    }
}

fun reset() {
    i = 0
    count = 0
    isDone = false
}
}

// OnClick 監聽器
inner class StartOnClick : OnClickListener {
    override fun onClick(v: View?) {
        // 提交任務 1000毫秒(1秒) 後執行
        handler.postDelayed(task, 1000)
    }
}
}

```

```
// OnClickListener 監聽器
inner class StopOnClick : OnClickListener {
    override fun onClick(v: View?) {
        handler.removeCallbacks(task) // 移除任務
        btn_start.isEnabled = true // start按鈕 有作用（可按）
    }
}

// OnClickListener 監聽器
inner class ClearOnClick : OnClickListener {
    override fun onClick(v: View?) {
        tv_message.text = "目前沒有訊息"
        img.setBackgroundResource(R.drawable.no_image_box)
        btn_start.isEnabled = true // start按鈕 有作用（可按）
        task.reset()
    }
}
}
```