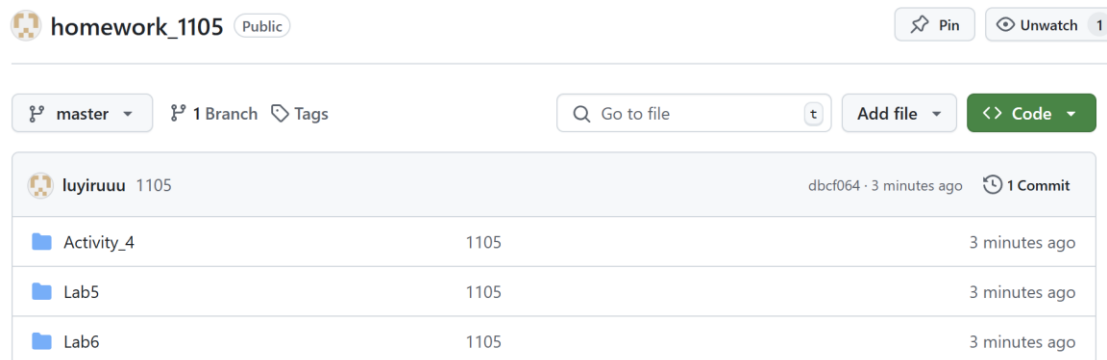


1. Git Hub 連結: https://github.com/luyiruuu/homework_1105.git
2. Git Hub 截圖:



3. 心得:

這次的課程著重於使用 AndroidStudio 中的提示訊息元件，如 Toast 和 AlertDialog，這些元件可以在應用程式中有效地向使用者傳達各種訊息。首先，Toast 是一種快速並短暫顯示訊息的元件，常用於回饋使用者操作或除錯用途。優點是操作簡單，且訊息顯示數秒後會自動消失，不會影響畫面中的其他互動。基本的 Toast 使用 `makeText` 方法建立，傳入 Context、訊息文字和總共要持續的時間，再透過 `show` 方法顯示出來。此外也介紹了客製化 Toast，讓我知道要如何設計個性化的提示訊息，可以自行加入圖片、調整位置。我們使用 `setGravity` 方法來設定顯示位置，讓使用者可以自行決定 Toast 顯示的地方，讓使用者體驗更加友善。

而 AlertDialog 是一種對話方塊，可以顯示訊息，並提供按鈕讓使用者回應。相較於 Toast，AlertDialog 功能更豐富，可放置文字、圖片，甚至是列表。常見的用法是在按鈕點擊事件中執行對應的處理，例如 `setPositiveButton`、`setNegativeButton`、`setNeutralButton` 等方法。

在寫作業的時候，我發現由於 Kotlin 支援更多高階語法，像是 Lambda 表達式和延遲初始化，因此 Kotlin 寫起來較為簡潔。但在使用 Java 語法的時候，需要考慮比較多的語法限制。例如，Kotlin 的 `setOnClickListener` 可以直接使用 Lambda，寫成 `showToast("預設 Toast")`，而在 Java 中，則需要使用匿名類別來實現。此外，Kotlin 提供的 Null 檢查機制簡化了開發工作，避免空指標的 Exception。Kotlin 可以使用 `?.` 直接處理可能為空的變數，而 Java 必須手動檢查 null。

4. Chatgpt:

- 在 AlertDialog 中嵌入自定義的圖片，會有什麼限制嗎?是否可以嵌入影片?

在 AlertDialog 中嵌入自定義圖片是可行的，但有一些限制：

- **記憶體使用限制**：嵌入過大或解析度過高的圖片可能會導致記憶體佔用增加，影響效能。尤其在低端設備上，過大的圖片可能會導致應用崩潰。
- **視覺空間限制**：AlertDialog 是針對短期通知設計的，其彈窗大小通常受到裝置螢幕大小限制。過大的圖片可能會使視覺過於擁擠，建議控制圖片大小，確保訊息清晰易讀。

至於影片的嵌入則更加複雜：

- **播放控制**：AlertDialog 沒有內建的影片控制功能，若需播放影片，需使用自訂的 Layout，並在其中加入 `VideoView` 或其他可播放影片的元件。
- **效能和體驗**：在彈窗中播放影片可能造成使用者分心，且會顯著增加記憶體和處理器負擔。因此，若要播放影片，通常建議直接使用 `DialogFragment` 或全屏 `Activity` 來實現，以避免影響使用者體驗。

● 是否可以在 Toast 或 AlertDialog 中整合語音提示或語音回應功能？

在 Toast 和 AlertDialog 中整合語音提示或回應功能是可行的，但需要注意以下幾點：

- **語音提示**：可以使用 Android 的 `TextToSpeech` (TTS) API 配合 Toast 或 AlertDialog 使用。當 Toast 或 AlertDialog 彈出時，同步觸發語音提示，例如「您的設定已成功」或「是否確認刪除？」等。這對於視力不佳的使用者特別有幫助。
- **語音回應**：Toast 本身是無法直接處理使用者回應的，因為它只是簡單顯示訊息且不能互動。而在 AlertDialog 中，若要添加語音回應，可以使用 Android 的 `SpeechRecognizer` API，實現語音指令的回應，如在 AlertDialog 中加入「確定」或「取消」的語音指令。但這樣的設計會稍微增加實作複雜度，並可能需要更多的權限申請。

● 對於使用者視力不好或有特殊需求的使用者，可不可以結合語音或震動提示等輔助技術？

是的，Toast 和 AlertDialog 可以結合語音和震動提示，為視力不佳或有特殊需求的使用者提供更好的輔助體驗：

- **語音提示**：如前所述，可以透過 `TextToSpeech` 在 Toast 或 AlertDialog 彈出時提供語音訊息，幫助使用者理解提示內容。
- **震動提示**：Android 支援震動功能，可以在顯示重要訊息時配合 `Vibrator` API。例如，當彈出重要警告的 AlertDialog 時觸發震動，提示使用者此訊息需要注意。此外，也可結合不同的震動模式來區分訊息重要性。
- **無障礙設定**：Android 的無障礙功能提供 `TalkBack` 模式，讓裝置自動讀取螢幕內容，若使用者開啟此模式，Toast 和 AlertDialog 的文字會自動被讀出，無需額外設定。開發者可以針對無障礙用戶優化介面，例如增加描述性標籤 (`contentDescription`) 以提升使用者體驗。

5. Java & Kotlin 語法差異

■ 變數 Null 檢查

Java	Kotlin
<pre>if (data == null) return;</pre>	<pre>result.data?.let { intent -> // 取得回傳的飲料名稱、甜度、冰塊的值 val drink = intent.getStringExtra(name: "drink") val sugar = intent.getStringExtra(name: "sugar") val ice = intent.getStringExtra(name: "ice") // 設定 tvMeal 的文字 findViewById<TextView>(R.id.tvMeal).text = "飲料：</pre>

需要手動檢查變數是否為 null	確保 intent 不為空才會執行內部程式碼，防止 NullPointerException
------------------	------------------------------------------------

■ ActivityResult

Java	Kotlin
<pre>@Override protected void onActivityResult(int requestCode,</pre>	<pre>// 宣告 ActivityResultLauncher，負責處理從 SecActivity 回傳的結果 private val startForResult = registerForActivityResult(Activi</pre>
使用 startActivityForResult()，並在 onActivityResult() 中處理回傳的結果，程式碼比較冗長	使用 registerForActivityResult() 來取代 Java 中的 onActivityResult()

■ 擴充函數的運用

Java	Kotlin
<pre>public class MainActivity extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); enableEdgeToEdge(); setContentView(R.layout.activity_main); View mView = findViewById(R.id.main); ViewCompat.setOnApplyWindowInsetsListener(mView, (v, WindowInsetsCompat systemBars = insets.getInsets(Wi v.setPadding(systemBars.left, systemBars.top, syste return insets; });</pre>	<pre>override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) enableEdgeToEdge() setContentView(R.layout.activity_main) setupWindowInsets() setupViewPager() lifecycle { method { onCreate() } } }</pre>
enableEdgeToEdge() 必須在類別內明確定義，所有方法在 MainActivity 中實作或繼承而來	enableEdgeToEdge() 可以直接在 onCreate 中被呼叫

■ setPadding 的寫法

Java	Kotlin
<pre>View mView = findViewById(R.id.main); ViewCompat.setOnApplyWindowInsetsListener(mView, (v, WindowInsetsCompat systemBars = insets.getInsets(Wi v.setPadding(systemBars.left, systemBars.top, syste return insets; });</pre>	<pre>private fun setupWindowInsets() { ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) { val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars()) v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom) }) }</pre>
使用 v.setPadding 設定 padding，要明確寫出 return insets	使用 v.setPadding 設定 padding，並直接回傳 insets

■ offscreenPageLimit 的設定

Java	Kotlin
------	--------

<pre>viewPager2.setAdapter(adapter); // 預先載入鄰近的頁面 viewPager2.setOffscreenPageLimit(1);</pre>	<pre>private fun setupViewPager() { findViewById<ViewPager2>(R.id.view adapter = ViewPagerAdapter(su offscreenPageLimit = 1 } }</pre>
使用方法呼叫設定，必須明確指定方法	使用賦值語法設定頁數

■ setOnClickListener 的設置

Java	Kotlin
<pre>btnToast.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { showToast("預設 Toast"); } });</pre>	<pre>btnToast.setOnClickListener { showToast("預設 Toast") }</pre>
需要使用匿名類別來設置 setOnClickListener	使用 Lambda 表達式，事件監聽器 簡潔

6. 讀書會

- 組員:陸蕙如、羅文均、官奕愷
- 討論時間:11/5 22:02 ~ 22:27(共計 25 分鐘)
- 討論地點:線上(Line)
- 照片:

10:27



4G 26

想不到名字 (3)



25:16



退出

