Coroutines and background image processing in Android

이미지를 URL 로 가져와서 프로세싱하고 ImageView 에 띄우는 간단한 앱을 만들어보자.

```
Android Studio TIPS :
* Shift 두 번 누르면 파일 검색 창 뜸.
```

1. Gradle Dependency 추가하기

```
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.0'
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.0'
```

2. Filter.kt 파일 붙여넣기

DevTides/AndroidCoroutinesImageProcessing
https://github.com/DevTides/AndroidCoroutinesImageProcessing/blob/master/app/src/main/java/com/devtides/imageprocessingcoroutines/Filter.kt

→ 컬러 사진을 흑백 사진으로 변환해주는 클래스.

3. 레이아웃 파일에 ImageView 와 ProgressBar 넣어주기

4. MainActivity 에 코루틴 스코프 선언하기

```
class MainActivity : AppCompatActivity() {

    private val IMAGE_URL = "https:// ... "
    private val coroutineScope = CoroutineScope(Dispatchers.Main)
```

5. MainActivity 에 비트맵 처리하는 함수 넣기

```
//IO 디스패처로 이 작업을 처리하겠다.
    private fun getOriginalBitmap() =
        //download the image
        URL(IMAGE_URL).openStream().use {
            BitmapFactory.decodeStream(it)
        }

    private fun loadImage(bitmap : Bitmap){
        progressBar.visibility = View.GONE
        imageView.setImageBitmap(bitmap)
        imageView.visibility = View.VISIBLE
    }
```

5. onCreate 함수 안에다가 코루틴 스코프 launch 하기

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

        //Main 스레드에서 스코프를 런칭한다.
        coroutineScope.launch {
            val originalDeferred = coroutineScope.async(Dispatchers.IO){
                getOriginalBitmap()
            }
            var bitmap = originalDeferred.await()

...
```

→ 원래의 비트맵을 IO 디스패처로 가져와서 결과를 받는다

6. 받아온 비트맵을 Filter.apply(Bitmap) 함수를 통해서 흑백으로 프로세싱 한다. 마찬가지로 코루틴 안에서 처리한다.

```
        coroutineScope.launch {
            val originalDeferred = coroutineScope.async(Dispatchers.IO){
                getOriginalBitmap()
            }
            var bitmap = originalDeferred.await()

      //오리지널 비트맵에 필터를 가하는 async - await
            val filteredDeferred = coroutineScope.async(Dispatchers.Default){
                Filter.apply(bitmap)
            }
            bitmap = filteredDeferred.await()
```

```
            loadImage(bitmap)
        }
```

결과 :