

Java and Python on Android

Accessing Java classes in Python

Both the SDK and the NDK are written in Java and C++.

The usage of Java classes in a Python program is allowed using Jython, a Python interpreter written in Java.

An other option is using Pyjnius, which is what Python-For-Android does :

```
from jnius import autoclass

Stack = autoclass('java.util.Stack')
stack = Stack()
stack.push('hello')
stack.push('world')

print stack.pop() # --> 'world'
print stack.pop() # --> 'hello'
```

Python-For-Android internally mix Cython, JNI (Java Native Interface) and Pyjnius.

The cost in terms of efficiency is minimal, and provides a more flexible interface.

While Pyjnius sounds to have no limitation, Kivy does not officially supports the whole SDK.

Basic Hardware Interface such as GPS or Bluetooth API are currently not available in Kivy.

Table of concepts

How do you...	Java	Python
...define a job for the DVM?	Using the Activity class.	Using the App class. <ul style="list-style-type: none">• onCreate(Bundle) -> build()• onPause(Bundle) -> on_pause()• and so on...
...interact with an other application?	Using a Bundle object.	Actually, there is no way of doing this.
...handle touch events?	Overriding the onTouchEvent method.	Overriding respectively : <ul style="list-style-type: none">• on_touch_down• on_touch_move• on_touch_up You also can grab a certain touch to a Widget.
...handle multi-touch events?	Using the touch-event unique id .	
...make 2D drawing?	You need to extend the View class and override the onDraw(Canvas c) method.	Every Widget object as a canvas property, which you can treat in different ways : <ul style="list-style-type: none">• Using Kv-Language, expecially for static things, in comfortable a separated file.• Using the Python with statement.• Or using it as a python variable (not so widely used).
...extend a widget?	By extending the Widget class and overriding the onDraw() method.	You have to extend a class, like in Java, then you will modify the canvas property as mentioned before.
...make 3D drawing?	Of course, you will need to pass through OpenGL ES, which APIs are provided by the Application Framework. In both languages however you will not have the same set of API you will have on a desktop app.	
	Just use the classes of the OpenGL package in the NDK. While it not differs a lot from JOGL or Java3D, you will have a different set of APIs.	Kivy have its own OpenGL wrapper, which offers common APIs on every platform. 3D drawing is currently in a "beta" phase. However, it does not have the same set of API of PyOpenGL.