

Is there a way to run Python on Android?

Asked 16 years, 3 months ago Modified 1 year, 3 months ago

Viewed 775k times  Part of [Mobile Development](#) Collective



2084



This question's answers are a [community effort](#).

Edit existing answers to improve this post. It is not currently accepting new answers or interactions.

We are working on an [S60](#) version and this platform has a nice Python API..

However, there is nothing official about Python on Android, but since [Jython](#) exists, is there a way to let the snake and the robot work together??

MD

android

python

jython

ase

android-scripting

Share Follow

edited Nov 8, 2018 at 19:09



Ümañg Bűrman

10.1k ● 4 ● 25 ● 43

asked Sep 19, 2008 at 13:21



Bite code

595k ● 116 ● 309 ● 334

Comments disabled on deleted / locked posts / reviews

23 Answers

Sorted by:

Highest score (default)



1189



One way is to use [Kivy](#):

Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.

Kivy runs on Linux, Windows, OS X, Android and iOS. You can run the same [python] code on all supported platforms.

[Kivy Showcase app](#)

Share Improve this answer

Follow

edited Aug 31, 2014 at 22:09



[Peter Mortensen](#)

31.6k ● 22 ● 109 ● 133

answered Nov 18, 2011 at 21:49



[JohnMudd](#)

13.8k ● 2 ● 30 ● 24

79 If you use Kivy, here is a tool to help package your project into an APK: github.com/kivy/python-for-android – [gdw2](#) Jan 9, 2012 at 4:42

21 I've been playing around with Kivy this past week attempting to write a game. Their main developers were very quick to answer questions on IRC however if you've programmed a

GUI before Kivy will make you say WTF quite a bit. Some examples of undocumented things that were weird for me: All widgets get every on_touch_down event even if the event occurred outside their region, No widget has a draw() method, almost everything happens via a custom observer pattern on custom Properties they made up (note these share the name with Python's property, but are not the same) – [Trey Stout](#) Jan 30, 2012 at 20:41

9 @Trey: What about non-GUI-related things? Are there any limitations regarding usage of Python's modules? What about access to Android-specific features, such as ability to read messages, add notification, work in the background, make a photo, read contacts list, determine GPS location etc.? – [Tadeck](#) Mar 24, 2012 at 4:20

8 And now there is iOS support too! – [rubik](#) Jun 30, 2012 at 7:00

3 To answer the old question of @Tadeck: Kivy has direct access to Android-specific features via [pyjnius](#) and (for a more pythonic wrapper) [plyer](#). You can also use just about any pure python module, or modules with compiled components if you write (or there already exists) a python-for-android recipe - we already have recipes for many popular modules such as numpy, sqlalchemy and even django. – [inclement](#) Aug 11, 2015 at 17:29 ✎



366

There is also the new [Android Scripting Environment](#) (ASE/SL4A) project. It looks awesome, and it has some integration with native Android components.



Note: no longer under "active development", but some forks may be.





Share Improve this answer

Follow

edited Feb 19, 2018 at 4:49



Shadow

9,417 ● 4 ● 49 ● 59

answered Jun 10, 2009 at 5:24



Heat Miser

19.7k ● 7 ● 37 ● 38

-
- 71 True, but they have to have ASE installed, it's not a solution that lets you write an Android app in python without anything already installed (normal user will be all "wtf is this ASE thing?) – [Stuart Axon](#) Nov 26, 2010 at 18:30
-
- 22 Further, ASE is a restricted environment; you cannot write full-blown Android apps even if ASE is pre-installed. See stackoverflow.com/questions/2076381 – [Sridhar Ratnakumar](#) Jan 23, 2011 at 21:36
-
- 11 I think it was renamed to SL4A. – [Vanuan](#) Dec 4, 2012 at 21:25
-
- 10 You can write apps, package them, and even sell them on Play Store if you like, with SL4A now days. It's come along well since the comments above were posted. If you want Python on Android, then PY4A, which runs on SL4A is probably the best choice. – [Carl Younger](#) Mar 9, 2013 at 17:53
-



231



Yes! : [Android Scripting Environment](#)

An example [via Matt Cutts](#) via SL4A -- "here's a barcode scanner written in six lines of Python code:

```
import android
droid = android.Android()
code = droid.scanBarcode()
```



```
isbn = int(code['result']['SCAN_RESULT'])
url = "http://books.google.com?q=%d" % isbn
droid.startActivity('android.intent.action.VIEW', url)
```

Share Improve this answer

edited Jan 15, 2019 at 2:29

Follow



Morse

9,045 ● 7 ● 40 ● 69

answered Jun 10, 2009 at 5:13



unmounted

34.3k ● 16 ● 63 ● 61

48 s/YES/meh. maybe/ ...it's extremely limited. anything graphical or multi touch? a big NO. – [gcb](#) Oct 23, 2010 at 9:09

4 @gcb you can't use the normal android widget set, but you can use "webviews" (which is what the native gmail application uses, for example). – [gdw2](#) Mar 16, 2012 at 16:10

30 golfed: `import android as a;d=a.Android();d.startActivity('android.intent.action.VIEW','http://books.google.com?q=%d'%int(d.scanBarcode()['result']['SCAN_RESULT']))` – [Alex L](#) Dec 28, 2012 at 6:20

15 @gdw2, surely the native Gmail app only uses WebViews to parse emails though, not for the actual UI. That makes the comparison a bit absurd. – [Veselin Romić](#) Jul 21, 2014 at 21:55



Pygame Subset for Android

94

[Pygame](#) is a 2D game engine for Python (on desktop) that is popular with new programmers. The [Pygame Subset for Android](#) describes itself as...



...a port of a subset of Pygame functionality to the Android platform. The goal of the project is to allow the creation of Android-specific games, and to ease the porting of games from PC-like platforms to Android.

The examples include a complete game packaged as an APK, which is pretty interesting.

Share Improve this answer

Follow

edited Nov 12, 2019 at 23:54



[Carl Younger](#)

3,070 ● 25 ● 38

answered Jan 28, 2011 at 12:18



[muriloq](#)

2,712 ● 5 ● 29 ● 32

-
- 5 Several aspects were broken on my Droid X (buttons, or touchscreen, can't remember), so I didn't get very far with this route. – [gdw2](#) Oct 5, 2011 at 21:56
-



As a [Python](#) lover and Android programmer, I'm sad to say this is not a good way to go. There are two problems:

92



One problem is that there is a lot more than just a programming language to the Android development tools. A lot of the Android graphics involve XML files to configure the display, similar to HTML. The built-in java objects are integrated with this XML layout, and it's a lot easier than writing your code to go from logic to bitmap.



The other problem is that the G1 (and probably other Android devices for the near future) are not that fast. 200 MHz processors and RAM is very limited. Even in Java, you have to do a decent amount of rewriting-to-avoid-more-object-creation if you want to make your app perfectly smooth. Python is going to be too slow for a while still on mobile devices.

Share Improve this answer

edited Feb 7, 2020 at 18:44

Follow

answered Nov 1, 2008 at 20:29



[lacker](#)

5,550 ● 6 ● 37 ● 39

52 There is not a single java word on an Android phone, it's compiled to byte code during the packaging process. Speed is not the issue : Google could provide tools producing the right byte code from a python code (like for Jython). BTW, Dalvik is not the Java VM so this is not about Java VS Python. – [Bite code](#) Nov 2, 2008 at 18:20

- 11 JAVA bytecode still needs to be processed by a JVM, and the Java language requires a garbage collector anyway. Actual speed could only come from C++. – [LtWorf](#) Apr 29, 2014 at 11:40
-
- 21 @Touzen 2 more years later, we are running Intel Atom processor or octa-core processors now. ;) – [Rohan Kandwal](#) Apr 10, 2015 at 6:13
-
- 3 Due to the difference in the power they are able to draw, phones will always be an order of magnitude slower than desktops. However, don't give up on Python for the mobile device, because typically only a fraction of the code is responsible for the processor-intensive work, and this fraction can be optimised by rewriting it in another language. – [Evgeni Sergeev](#) Dec 5, 2015 at 6:05
-
- 4 By the way Google pushes Kotlin these days - which is a pretty unknown language to most users - it is clear that it's possible to support other languages than Java (I'm a Java dev and a Python fan). So I think the answer is rather a political one than a practical one. – [Alex](#) Jul 11, 2018 at 12:55 ✎
-



Scripting Layer for Android

82

[SL4A](#) does what you want. You can easily install it directly onto your device from their site, and do not need root.



It supports a range of languages. Python is the most mature. By default, it uses Python 2.6, but there is a [3.2 port](#) you can use instead. I have used that port for all kinds of things on a Galaxy S2 and it worked fine.



API

SL4A provides a port of their `android` library for each supported language. The library provides an interface to the underlying Android API through a single `Android` object.

```
from android import Android

droid = Android()
droid.ttsSpeak('hello world') # example using the text
```

Each language has pretty much the same API. You can even use the JavaScript API inside webviews.

```
let droid = new Android();
droid.ttsSpeak("hello from js");
```

User Interfaces

For user interfaces, you have three options:

- You can easily use the generic, native dialogues and menus through the API. This is good for confirmation dialogues and other basic user inputs.
- You can also open a webview from inside a Python script, then use HTML5 for the user interface. When you use webviews from Python, you can pass messages back and forth, between the webview and the Python process that spawned it. The UI will not be native, but it is still a good option to have.

- There is *some* support for native Android user interfaces, but I am not sure how well it works; I just haven't ever used it.

You can mix options, so you can have a webview for the main interface, and still use native dialogues.

QPython

There is a third party project named [QPython](#). It builds on SL4A, and throws in some other useful stuff.

QPython gives you a nicer UI to manage your installation, and includes a little, touchscreen code editor, a Python shell, and a PIP shell for package management. They also have a Python 3 port. Both versions are available from the Play Store, free of charge. QPython also bundles libraries from a bunch of Python on Android projects, including Kivy, so it is not just SL4A.

Note that QPython still develop their fork of SL4A (though, not much to be honest). The main SL4A project itself is pretty much dead.

Useful Links

- SL4A Project (now on GitHub):
<https://github.com/damonkohler/sl4a>
- SL4A Python 3 Port:
<https://code.google.com/p/python-for->

android/wiki/Python3

- QPython Project: <http://qpython.com>
- Learn SL4A (Tutorialspoint):
<https://www.tutorialspoint.com/sl4a/index.htm>

Share Improve this answer

edited Oct 12, 2019 at 2:28

Follow

answered Mar 19, 2012 at 15:45



Carl Younger

3,070 ● 25 ● 38

-
- 2 Can you use any of those to run python script from terminal or tasker? I can't find a way :(– [Pitto](#) Aug 26, 2015 at 11:27
-
- 2 You can launch an SL4A script from Tasker. There are some examples on [this page](#) that have snippets of Python being launched from Tasker. It is really a whole different question though. – [Carl Younger](#) Sep 1, 2015 at 19:05 ✎
-



Cross-Compilation & Ignifuga

82

My blog has [instructions and a patch](#) for cross compiling Python 2.7.2 for Android.



I've also open sourced [Ignifuga](#), my 2D Game Engine. It's Python/SDL based, and it cross compiles for Android.



Even if you don't use it for games, you might get useful ideas from the code or builder utility (named *Schafer*, after Tim... you know who).

Share Improve this answer

edited Nov 12, 2019 at 23:59

Follow



Carl Younger

3,070 ● 25 ● 38

answered Oct 12, 2011 at 13:49



[gabomdq](#)

1,750 ● 13 ● 10

5 Impressive. +1 for this. Not accepted because you can't possibly write anything for the public with this. – [Bite code](#)
Oct 13, 2011 at 14:14

2 If you mean you can not do graphic apps with it, you most definitely can, of course, more work is needed. I actually use this port combined with SDL 1.3, it's not trivial to go from the python interpreter to an interactive app, but it can be done.
– [gabomdq](#) Oct 14, 2011 at 2:38



Termux

68

You can use the [Termux](#) app, [F-Droid link](#), which provides a POSIX environment for Android, to install Python.



Note that `apt install python` will install Python3 on Termux. For Python2, you need to use `apt install python2`.



- Some demos: <https://www.youtube.com/watch?v=fqgsl72mASE>
- The GitHub project: <https://github.com/termux>

Share Improve this answer

edited Sep 3, 2023 at 20:35

Follow



Morrison Chang

12.1k ● 3 ● 43 ● 84

answered Jan 20, 2017 at 11:36



Adrian Stanculescu

1,048 ● 2 ● 13 ● 22



Kivy

60



I wanted to add to what @JohnMudd has written about Kivy. It has been years since the situation he described, and Kivy has evolved substantially.



The biggest selling point of Kivy, in my opinion, is its cross-platform compatibility. You can code and test everything using any desktop environment (Windows/*nix etc.), then package your app for a range of different platforms, including Android, iOS, MacOS and Windows (though apps often lack the native look and feel).

With Kivy's own [KV](#) language, you can code and build the GUI interface easily (it's just like Java XML, but rather than TextView etc., KV has its own `ui.widgets` for a similar translation), which is in my opinion quite easy to adopt.

Currently [Buildozer](#) and [python-for-android](#) are the most recommended tools to build and package your apps. I have tried them both and can firmly say that they make

building Android apps with Python a breeze. Their guides are well documented too.

iOS is another big selling point of Kivy. You can use the same code base with few changes required via [kivy-ios](#) Homebrew tools, although Xcode is required for the build, before running on their devices (AFAIK the iOS Simulator in Xcode currently doesn't work for the x86-architecture build). There are also some dependency issues which must be manually compiled and fiddled around with in Xcode to have a successful build, but they wouldn't be too difficult to resolve and people in [Kivy Google Group](#) are really helpful too.

With all that being said, users with good Python knowledge should have no problem picking up the basics quickly.

If you are using Kivy for more serious projects, you may find existing modules unsatisfactory. There are some workable solutions though. With the (work in progress) [pyjnius](#) for Android, and [pyobjus](#), users can now access Java/Objective-C classes to control some of the native APIs.

Share Improve this answer

Follow

edited Nov 13, 2019 at 0:18



[Carl Younger](#)

3,070 ● 25 ● 38

answered Jan 13, 2015 at 2:08



[Anzel](#)

20.5k ● 5 ● 53 ● 53



Making Android apps with Python has come a long way in the past couple of years.

41



One new way is using the [FireMonkey \(FMX\) framework for Python](#). FMX is a GUI framework for developing apps for many platforms, with Android being one of them.



You can use [PythonFMXBuilder](#) to build your Android App once you're finished programming it.

Share Improve this answer

Follow

edited May 31, 2023 at 16:39



Shaun Roselt

3,213 ● 5 ● 26 ● 53

answered Sep 19, 2008 at 13:26



Ilya Kochetov

18.4k ● 6 ● 47 ● 62



40



Using SL4A (which has already been mentioned by itself in other answers) you can [run](#) a full-blown [web2py](#) instance (other [python web frameworks](#) are likely candidates as well). SL4A doesn't allow you to do native UI components (buttons, scroll bars, and the like), but it does support [WebViews](#). A WebView is basically nothing more than a striped down web browser pointed at a fixed address. I believe the native Gmail app uses a WebView instead of going the regular widget route.



This route would have some interesting features:

- In the case of most python web frameworks, you could actually develop and test without using an android device or android emulator.
- Whatever Python code you end up writing for the phone could also be put on a public webserver with very little (if any) modification.
- You could take advantage of all of the crazy web stuff out there: query, HTML5, CSS3, etc.

Share Improve this answer

Follow

edited Jan 26, 2016 at 5:48



ΦΧορῆ ὤτ Περεύπα Ψ

48.2k ● 17 ● 74 ● 101

answered Jan 6, 2012 at 14:34



gdw2

7,976 ● 4 ● 48 ● 49

3 Cherrypy works well, with ws4py websocket support. Bottle is also fine on SL4A. – [Carl Younger](#) Feb 13, 2013 at 1:28



QPython

39

I use [the QPython app](#). It's free and includes a code editor, an interactive interpreter and a package manager, allowing you to create and execute Python programs directly on your device.



Share Improve this answer

Follow

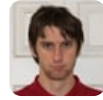
edited Nov 13, 2019 at 0:23



Carl Younger

3,070 ● 25 ● 38

answered Jun 12, 2013 at 19:46



kyle k

5,482 ● 10 ● 34 ● 46

3 Now qpython added Django support! That's all that I need!
Amazing! – [swdev](#) Oct 7, 2013 at 14:36



[Here](#) are some tools listed in official python website

38



There is an app called **QPython3** in playstore which can be used for both editing and running python script.

[Playstore link](#)



Another app called **Termux** in which you can install python using command

```
pkg install python
```

[Playstore Link](#)

If you want develop apps , there is [Python Android Scripting Layer \(SL4A\)](#) .

The Scripting Layer for Android, SL4A, is an open source application that allows programs written in a range of interpreted languages to run on Android. It also provides a high level API that allows these

programs to interact with the Android device, making it easy to do stuff like accessing sensor data, sending an SMS, rendering user interfaces and so on.

You can also check [PySide for Android](#), which is actually Python bindings for the Qt 4.

There's a platform called [PyMob](#) where apps can be written purely in Python and the compiler tool-flow (PyMob) converts them in native source codes for various platforms.

Also check [python-for-android](#)

python-for-android is an open source build tool to let you package Python code into standalone android APKs. These can be passed around, installed, or uploaded to marketplaces such as the Play Store just like any other Android app. This tool was originally developed for the Kivy cross-platform graphical framework, but now supports multiple bootstraps and can be easily extended to package other types of Python apps for Android.

Try [Chaquopy](#) A Python SDK for Android

Andddd... [BeeWare](#)

BeeWare allows you to write your app in Python and release it on multiple platforms. No need to rewrite the app in multiple programming languages. It means no issues with build tools, environments, compatibility, etc.

Share Improve this answer
Follow

edited Feb 13, 2020 at 5:55

answered Apr 15, 2017 at 14:09



pz64_

2,252 ● 2 ● 23 ● 44

2 using Termux you can install python 2 as well \$ apt install python2 – M.Hefny Jul 16, 2017 at 1:53



29



From the [Python for android](#) site:

Python for android is a project to create your own Python distribution including the modules you want, and create an apk including python, libs, and your application.

Share Improve this answer
Follow

answered Jan 9, 2012 at 4:46



gdw2

7,976 ● 4 ● 48 ● 49

1 @e-satis: Thanks for your comment. I don't see which answer I'm duplicating. I can only guess that you think PFA is

the same as Kivy. Though it's hosted under the Kivy project, you don't even have to use Kivy to use PFA. – [gdw2](#) Jan 9, 2012 at 22:46

Check @tito's deleted answer at the bottom point to PFA. Plus, there is little interest in running PFA without kivy since it's the only toolkit you got. – [Bite code](#) Jan 10, 2012 at 0:55

It's not because kivy is currently the only toolkit available (some people are working to integrate others) than it's a duplicate. Please consider the others options as-it, and not merge all into one post :) – [tito](#) Mar 16, 2012 at 14:34



Chaquopy

29



[Chaquopy](#) is a plugin for Android Studio's Gradle-based build system. It focuses on [close integration](#) with the standard Android development tools.



- It provides complete APIs to call [Java from Python](#) or [Python from Java](#), allowing the developer to use whichever language is best for each component of their app.
- It can automatically download [PyPI packages](#) and build them into an app, including selected native packages such as NumPy.
- It enables full access to all Android APIs from Python, including the native user interface toolkit ([example pure-Python activity](#)).

This used to be a commercial product, but it's now free and open-source.

(I am the creator of this product.)

Share Improve this answer

edited Feb 3, 2023 at 9:20

Follow

answered Feb 16, 2018 at 17:12



mhsmith

8,053 ● 3 ● 47 ● 62

-
- 1 Just want to add that I was very impressed by this option. It works really well. Pricing isn't terrible (free for open source apps, \$50 one time payment for closed source apps). I have decided against running python on Android because it bloats app size A LOT, but if that isn't a concern I really recommend this option – [JAnton](#) Jul 9, 2021 at 1:20
-



Yet another attempt: <https://code.google.com/p/android-python27/>

22



This one embed directly the Python interpreter in your app apk.



Share Improve this answer



Follow

answered May 9, 2012 at 15:44



Bite code

595k ● 116 ● 309 ● 334



You can run your Python code using [sl4a](#). sl4a supports Python, [Perl](#), [JRuby](#), [Lua](#), BeanShell, JavaScript, [Tcl](#), and shell script.

17



You can learn sl4a [Python Examples](#).



Share Improve this answer

edited Jan 13, 2015 at 0:34



Follow



[Peter Mortensen](#)

31.6k ● 22 ● 109 ● 133

answered Mar 11, 2013 at 9:36



[Hitul](#)

357 ● 1 ● 5 ● 11



You can use **QPython**:

15

It has a Python Console, Editor, as well as Package Management / Installers



<http://qpython.com/>



It's an open source project with both Python 2 and Python 3 implementations. You can download the source and the Android .apk files directly from github.

QPython 2: <https://github.com/qpython-android/qpython/releases>

QPython 3: <https://github.com/qpython-android/qpython3/releases>

Share Improve this answer

answered Sep 16, 2016 at 2:41

Follow



[ode2k](#)

2,723 ● 16 ● 21



15

Take a look at [BeeWare](#). It has grown significantly. It is awarded with PSF (Python Software Foundation) Education Grant.



Beeware's aim is to be able to create native apps with Python for all supported operating systems, including Android.



Official Website: [Beeware](#)

Github Repo: <https://github.com/beeware>

Share Improve this answer

Follow

edited Jun 24, 2020 at 21:28



[bad_coder](#)

12.8k ● 20 ● 53 ● 88

answered Jan 12, 2018 at 18:56



[DrawT](#)

252 ● 4 ● 6

What is the difference to Kivy? It is stable portable python Gui based – [Carsten Thielepape](#) Feb 16, 2020 at 18:47



14

Another option if you are looking for 3.4.2 or newer (3.9.6 as of this writing) is this archive on GitHub.

[Python3-Android 3.4.2](#) or [Python3-Android 3.9.6](#)



I believe the original archive supports Python 3.4.2, the latest GRRedwings branch support 3.9.6 and the 22b version of the NDK. Older branches support other versions, but are not as easy to compile with docker.



The older version you simply clone the archive, run make and you get the .so or the .a

The newer versions follow the ReadMe, but it uses docker for consistent builds.

I currently use this to run raw Python on android devices. With a couple modifications to the build files you can also make x86 and armeabi 64 bit

Share Improve this answer

edited Aug 24, 2021 at 14:18

Follow

answered Feb 25, 2015 at 13:04



Brian S

3,214 ● 39 ● 56



12

Didn't see this posted here, but you can do it with Pyside and Qt now that Qt works on Android thanks to Necessitas.



It seems like quite a kludge at the moment but could be a viable route eventually...



http://qt-project.org/wiki/PySide_for_Android_guide

Share Improve this answer

answered Oct 6, 2014 at 14:59

Follow



radix07

1,536 ● 1 ● 17 ● 22



One more option seems to be [pyqtdeploy](#) which citing the docs is:

9



a tool that, in conjunction with other tools provided with Qt, enables the deployment of PyQt4 and PyQt5 applications written with Python v2.7 or Python v3.3 or later. It supports deployment to desktop platforms (Linux, Windows and OS X) and to mobile platforms (iOS and Android).

According to [Deploying PyQt5 application to Android via pyqtdeploy and Qt5](#) it is actively developed, although it is difficult to find examples of working Android apps or tutorial on how to cross-compile all the required libraries to Android. It is an interesting project to keep in mind though!

Share Improve this answer

Follow

edited May 23, 2017 at 12:18



Community Bot

1 • 1

answered May 18, 2015 at 1:56



Andrzej Pronobis

35.9k • 18 • 81 • 92



Check out [enaml-native](#) which takes the react-native concept and applies it to python.

5



It lets users build apps with native Android widgets and provides APIs to use android and java libraries from python.



It also integrates with android-studio and shares a few of react's nice dev features like code reloading and remote debugging.

Share Improve this answer

edited Aug 20, 2018 at 16:59

Follow

answered Jan 16, 2018 at 16:01



[frmdstryr](#)

21.3k ● 3 ● 40 ● 33