Programming Python on Android device

Using Python is one of the fastest ways to get started and test some simple code on Android. Further, when you are familiar with it, you can upgrade your phone with unique features only you have and can even build a full APK.

There are many reasons why you are tempted to become an Android application developer. You think that building an application will give you a 'decent' economy or even become a 'world-changing' person, which is a strong driving force. Or others simply wish that learning code, writing programs, building applications, tools to satisfy their passion is a goal throughout. Android is an open and accessible operating system, so it is right to start implementing your goals here.

The problem is that learning programming on Android is not entirely an easy task. In the past, when you wanted to run a simple 'Hello World' program, you need to download Android Studio, Android SDK and Java JDK. You need to install links, create APKs and add permissions to your phone. And then when all seems ready, you continue to control some issues such as displaying the program on the screen. There are so many things to 'hinder' you quickly get close to your goals.

That's why many programmers now choose Python. Python is an alternative solution to help you overcome these difficulties. Python programming is very special, it's simple, 'elegant' and extremely suitable for beginners. Another outstanding thing is that you can start building scripts and test them on your Android device almost instantly.

In short, using Python is one of the fastest ways to get started and test some simple code on Android. Further, when you are familiar with it, you can upgrade your phone with the unique features you only have and can even build a full APK.

QPython for Android

So how do we start with Python? If your desire is to learn Python on a PC, download the latest version of Python 2 or Python3 with the IDE (Integrated Development Environment). You can see how to install Python for your computer here.

But in this article, Quantrimang's concern is not here. To start with Python on Android, you need to be in your device QPython (used to run Python2) or QPython3 (running Python3 code).

Python is a continuous improvement and development project, so to ensure your code runs as smoothly as possible, you should download the updated Python version. In this article, I use Python 3.6.1.

There are a few issues when updating from Python ver2 to ver3. You will have to make some modifications when you want to run Python2 code in a higher version, some commonly used libraries are broken. So, if you're a newbie in Python programming, start with Python3 to get the most up-to-date knowledge. But this doesn't mean that we say you don't need to know Python2. In the future, there will certainly be but in case you need to revert to version 2 to work with supported libraries in the old version.

The main library we will use is the following Kivy and fortunately it is supported in Python3.

Write simple code with some variables and input.

After downloading and installing QPython3, you have an environment to start programming. You can load scripts from here and later to be useful when you create your own native apps, for example, you can create some basic tools for doing math and tools. learning tests, or tools for storing and retrieving data . It's completely in your hands.

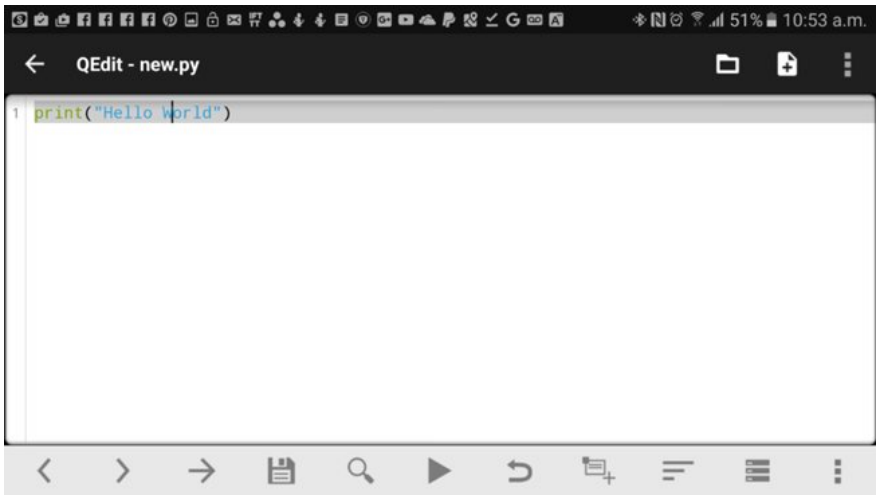
So let's learn how to build those tools here. First start with the 'Hello World' application.

To do this, open QPython3 and select 'Editor'. This is where you can write code or edit other scripts. Personally, editing will be easier if you have a bluetooth keyboard and mouse when working here.

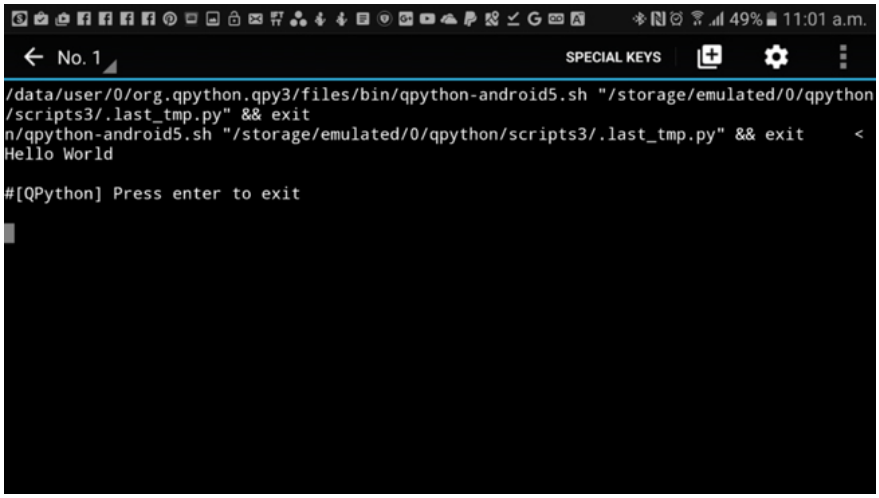
Now run the command:

```
print('Hello World')
```

Then save the script with the suffix '.py'. You save by clicking on the floppy disk icon at the bottom. Note that the 'print' command must be lowercase.



Run the program by clicking on the arrow icon and the words 'Hello World' will appear on the screen with a lot of other jargon. This is where your scripts will run, called the console. Later you can add graphical features to this screen.



Next try with variables. Variable is the name of a region in memory used to store information and data. Unlike other languages, when programming with Python you don't need to define variables. Follow the following example:

```
Name = 'Adam'
print('Hello ' + Name)
```

This is a code snippet that names the variable Name and assigns the value 'Adam' to greet the user with their name.

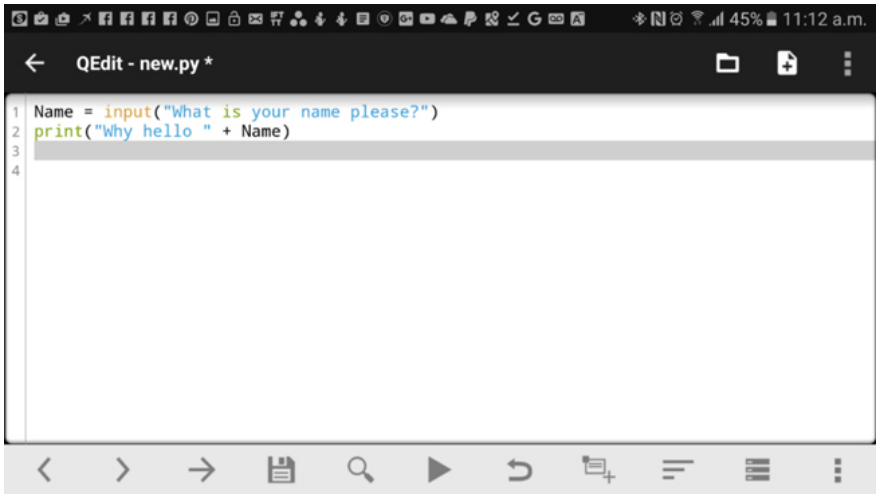
It can also be done easily with the example:

```
Number=7
print('The number is ' + Number)
```

The practical point of the variable here is that it allows us to change the elements in the code. You can write Number = Number + 1 so that the values ??increase. Similarly, create a small app like this:

```
Name = input('What is your name please?')
print('Why hello ' + Name)
```

The input of the above command allows to retrieve data from the user. In this case, you are using the user input to determine the *Name* variable . Note that the variables are case sensitive. Python commands are written in lowercase, so declaring variables in uppercase is easier to distinguish.



So using a few lines of code has brought some interesting things especially for your Android device. Another example script about your age details like this:

```
 Age = int(input('How old are you?'))

print('In ', 100 – Age, ' years, you will be 100! That's around ', (100 -Age) * 3
```

This small program tells you how many more days you will be 100 years old. Here use a few operators (multiply '*', except '-'). The int declaration at the beginning indicates that the input entered must be an integer.

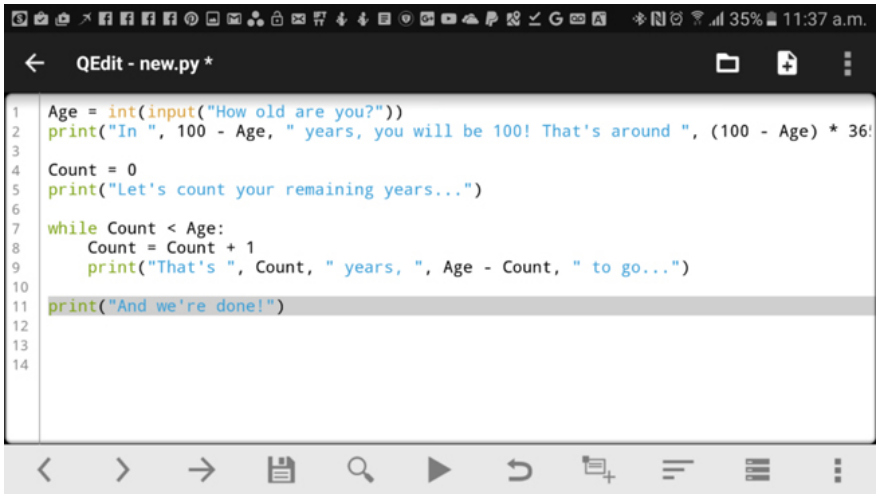
WHILE loops and IF statements

The WHILE loop in Python is used to run a code repeatedly when the given condition returns TRUE. Add the following lines to the above age script:

```
Count = 0
print('Let's count your remaining years…')

while Count  Age:
 Count = Count + 1
 print('That's ', Count, ' years, ', Age – Count, ' to go!')

print('And we're done!)
```



Notice that the next two lines are indented meaning they are part of the loop. If you have learned C, C ++ or Java, you will know that these programming languages ??use {} to identify code blocks. In Python, it is different, the blocks will be identified through indentation. That's why Python indentation is so important, if you reverse the line the program will report an error immediately.

Along with the loop, **the IF statement in Python programming** is also a very important part. The IF statement is used to execute conditional instructions, if the correct command executes the command, if it is false, the command does not execute. For example:

```
if Age > 50:
 print('You're over half way!')
```

Also Python has **an IF . ELSE** . Execute IF if the condition is true, if it is false, execute the ELSE.

```
if Age > 50:
 print('You're over half way!')
 else:
 print('Ah, still young!')
```

The ELIF command is also very useful. ELIF is the abbreviation of the ELSE IF, which allows us to test many conditions. If the condition is false, it will check the condition of the next ELIF block and so on. If all conditions are wrong it will execute the ELSE block.

```
if Age > 50:
 print('You're over half way!')
 elif Age  50:
 print('Ah, still young!')
 else:
 print('You're exactly halfway!')
```

Here Python will announce 'You're exactly halfway!' when the user is exactly 50 years old (no more than 50, not less than 50).

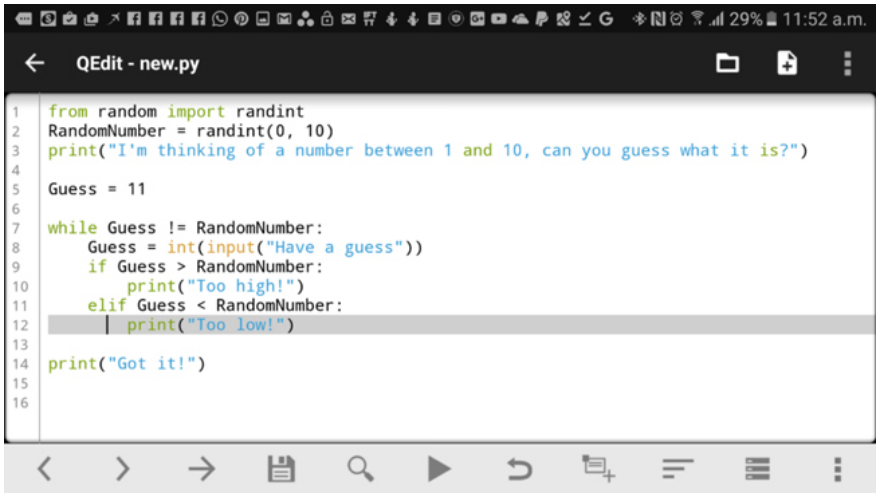
Use the Library and create a simple game.

Using the newly introduced code in the article is enough for you to create a simple small game. Before doing that, join Quantrimang to learn how to use libraries in Python.

Python comes with libraries called 'Python Standard Library' - the standard Python library, without installing any additional programs. The game we will get acquainted with below will be the number guessing form like 'higher or lower' - higher or lower. To do this, we need to create a random number and no command in Python is possible. Write the following command:

```
from random import randint
```

We were then able to use randint (lowest, highest) functions with two parameters, the lowest and the highest.



Complete the game with the following code:

```
from random import randint
RandomNumber = randint(0, 10)

print('I'm thinking of a number between 1 and 10, can you guess what it is?')


Guess = 11

while Guess != RandomNumber:
 Guess = int(input('Have a guess…'))
 if Guess > RandomNumber:
 print('Too high!')
 if Guess  RandomNumber:
 print('Too low!')

print('Got it!')
```

Although this is not an Android application, it can't stop you from creating scripts like this, you can even share it with friends or colleagues if they also use QPython3.

So, by using the Python Standard Library, you can write files, download everything from the website, and have many more interesting things right on your device.

Of course there are many more things to learn. For example, you can create a very simple Class like this:

```
def counter(Name):
 length = len(Name)
 return length;

NamePlease = input("Name length counter! Enter your name ")
print(counter(NamePlease))
```

Hay List is presented like this:

```
List = ['Apples', 'Oranges', 'Pears']
```

There are lots of Python resources you need to learn. You can study Python with Quantrimang here.

Use Python Android Scripting Layer

If you want to create a real Android application in Python, you will have a few options, depending on your idea and purpose how to use that application.

If you only want an app to access your device's native features, then do it with the SL4A library or Python Android Scripting Layer. This library allows you to do functions such as displaying dialogs, emotional reading or camera access.

The program below will open the camera and save your photos:

```
import sl4a

droid = sl4a.Android()
droid.cameraInteractiveCapturePicture('/sdcard/qpython.jpg')
```

Or do you want to open a website by:

```
from android import Android

droid = Android()
droid.webViewShow('https://www.quantrimang.com')
```

You can even launch to display the interface of the HTML file stored on the device. This will be a great way to view GUI elements (Graphical User Interface).

```
droid.webViewShow('file:///sdcard/ index.html')
```

One more thing you can do is create a file to display dynamic HTML based on the script you create. Combine this function with Tasker (a tool to automate all tasks on Android devices) to create potential surprises.

Kivy Library

If you want to go further in this area, you need to use Kivy. Kivy allows you to create Android apps with full functionality, multitouch, graphics and more. This is also a way to help you turn your scripts in Python into

APKs that can be installed on Android devices and distributed via CH Play. Great, Kivy is a cross-platform library so you can create apps for many other platforms when you use them.

You can display UI elements (User Interface) such as buttons or graphics. A simple example:

```
from kivy.app import App
from kivy.uix.button import Button

class HelloWorld(App):
 def build(self):
 btn = Button(text='Hello World')
 return btn

HelloWorld().run()
```

Conclude

Overall, Python is not a perfect choice for developing professional applications, but this is a great language for you to create scripts and build personal utility tools for Android devices. mine. Indeed, everything is more comfortable when working with Python on phones with QPython3. This is the method that is considered to be the easiest for those who enter code writing on mobile devices.

So what are you waiting for without trying? A world of rich Android application programming is waiting. Good luck!

See more:

1. What is Python? Why choose Python?
2. The best tools for Android developers
3. 5 free application building platforms do not need code
4. These programming languages ??for the best mobile application development