Universitas Esa Unggul

Smart, Creative and Entrepreneurial

www.esaunggul.ac.id

# *CMC 101 TOPIK DALAM PEMROGRAMAN*

## PERTEMUAN 5

## PROGRAM STUDI MAGISTER ILMU KOMPUTER

## FAKULTAS ILMU KOMPUTER

# TOPIK DALAM PEMROGRAMAN SCRIPTING LAYER menggunakan Python

Pertemuan 5

# TUJUAN PERKULIAHAN

- Mahasiswa mampu membuat design solusi persoalan dengan Scrypting Layer

- Instruksi aritmatika
- Interaksi dengan device (android)
- Penggunaan bbrp fungsi android secara interaksi

Survey by Eclipse

# THE TOP 6 PROGRAMMING LANGUAGES FOR IOT PROJECTS

# What we mean when we say 'IoT devices’

The IoT architectural environment into three major sections:

- the sensors that create the data,
- the hubs or local gateways that organize it, and
- the geographically distant, centralized servers that collect the data.

# Java

The top choice of the Eclipse survey and another survey by embedded-computing.com was Java, a result that's not surprising for a language still known for being "write once, run anywhere." The original project was aimed at set-top boxes, one of the first domains for non-desktop computing.

# C

The language is still the first choice for many programmers who write for the lowest layer of software, the one closest to the hardware.

# Python

It started as a scripting language to glue together real code, but it's increasingly used as the main language for many developers.

When small devices have enough memory and computational power, the developers are free to choose the language that makes their life easier and that is more and more often turning out to be Python.

# JavaScript

Much of this work is focused on the servers and gateways or hubs that gather the information and then store it. The smaller smart hubs and sensors that run Linux can usually run Node.js.

# Swift

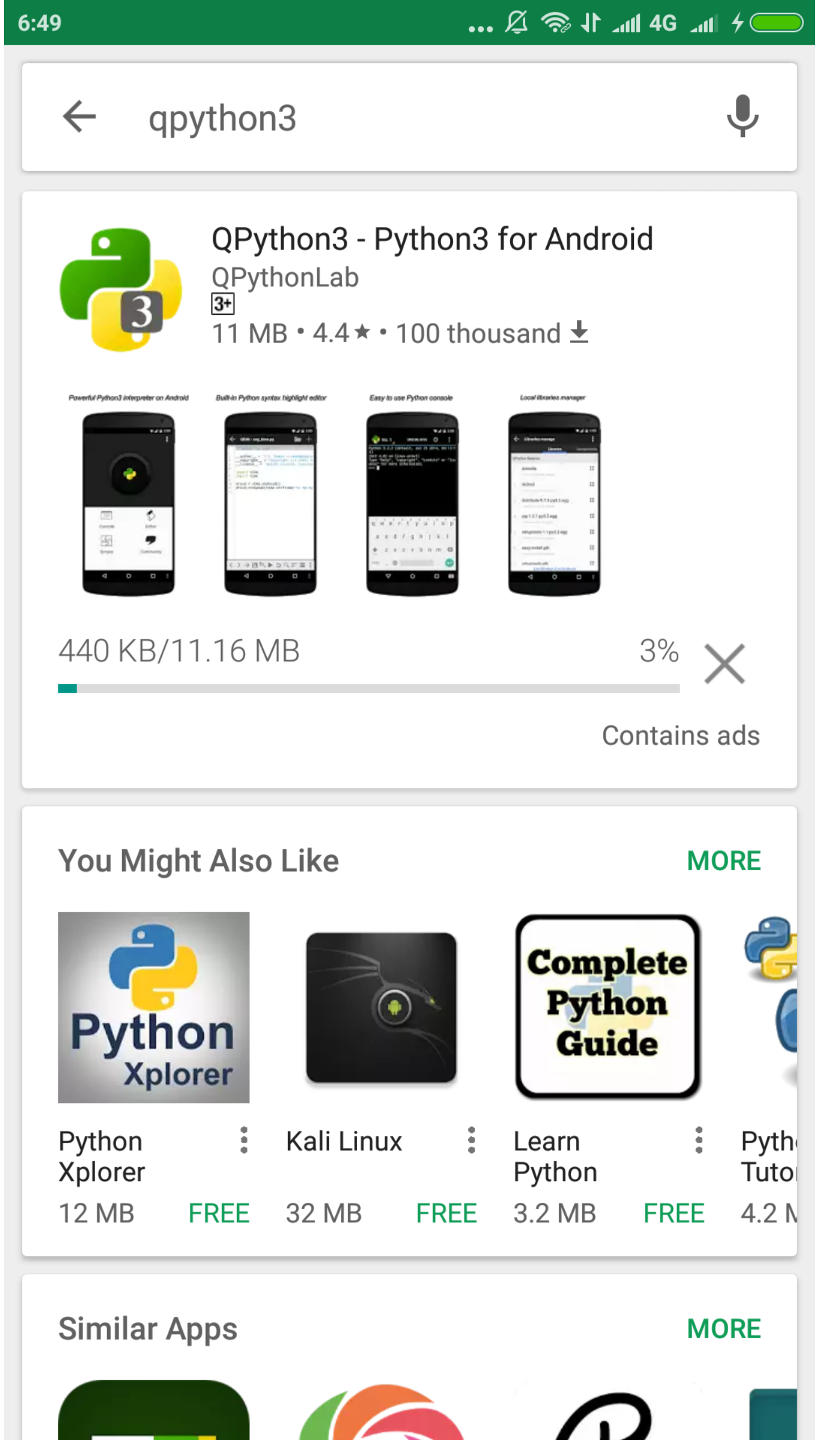
While Swift is still mainly used to build applications for Apple's iOS and macOS devices, the preponderance of these machines means that it's often part of the IoT stack.

If you want your things to interact with an iPhone or an iPad, you're probably going to want to build the app in Swift (or perhaps its predecessor, Objective C).

# PHP

This language may be the first choice of bloggers and website prototypers, but it is also surprisingly popular in the IoT. After the big languages and their cousins such as C#, PHP is the one language that is mentioned the most often by developers in the Eclipse survey; 11.2 percent say they are including PHP code in their stack.

# PYTHON



qpython3

QPython3 - Python3 for Android

QPythonLab

3+

11 MB • 4.4★ • 100 thousand

Powerful Python3 interpreter on Android
Built-in Python syntax highlight editor
Easy to use Python console
Local libraries manager

440 KB/11.16 MB 3%

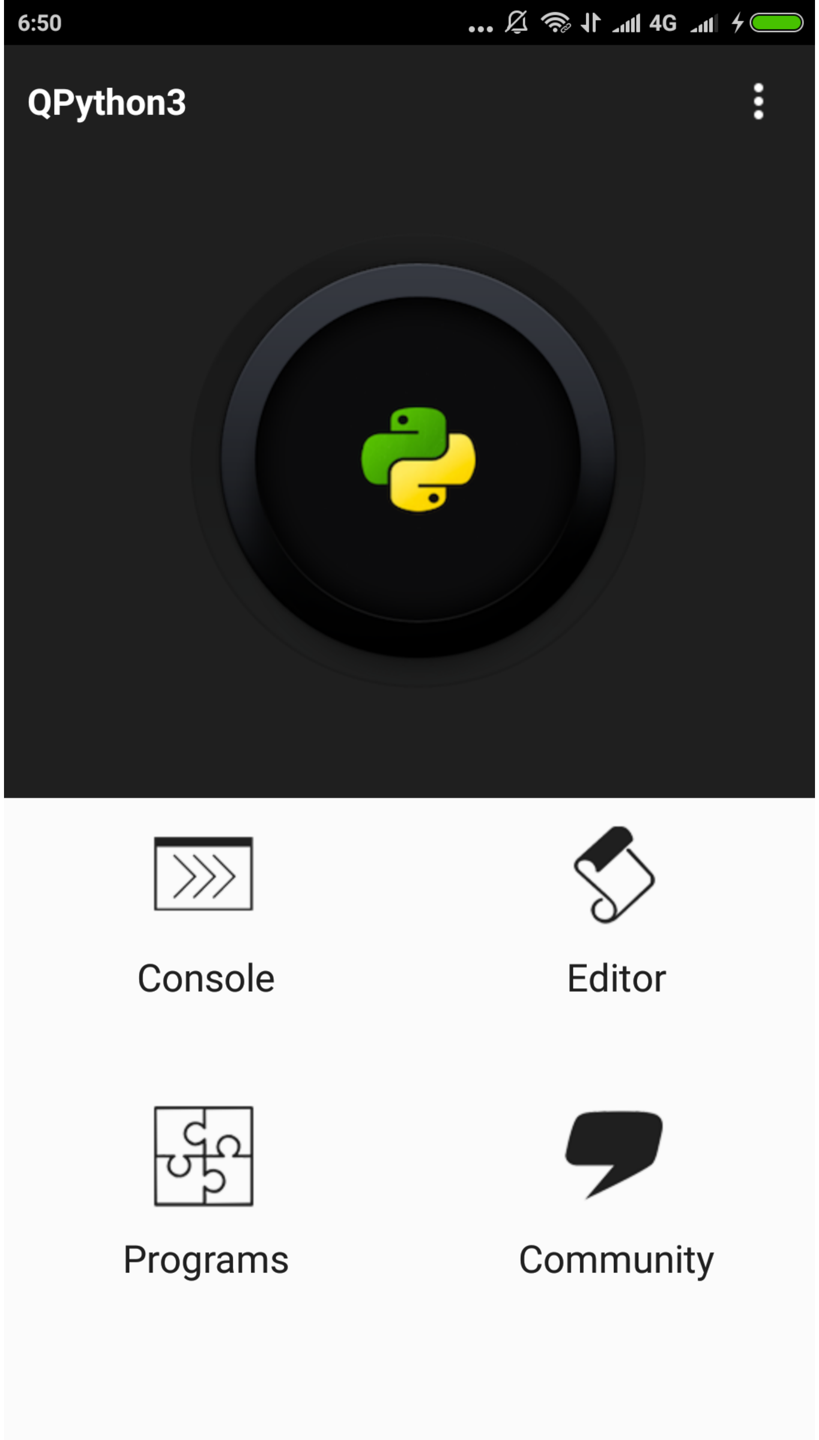
Contains ads

You Might Also Like MORE

Python
Xplorer
Complete
Python
Guide

| | | | |
|---|---|---|---|
| Python Xplorer | Kali Linux | Learn Python | Pyth<br>Tuto |
| 12 MB FREE | 32 MB FREE | 3.2 MB FREE | 4.2 M |

Similar Apps MORE


6:50
4G
QPython3
Console
Editor
Programs
Community

*“Python is a programming language that lets you work quickly and integrate systems more effectively”* (Python.org)

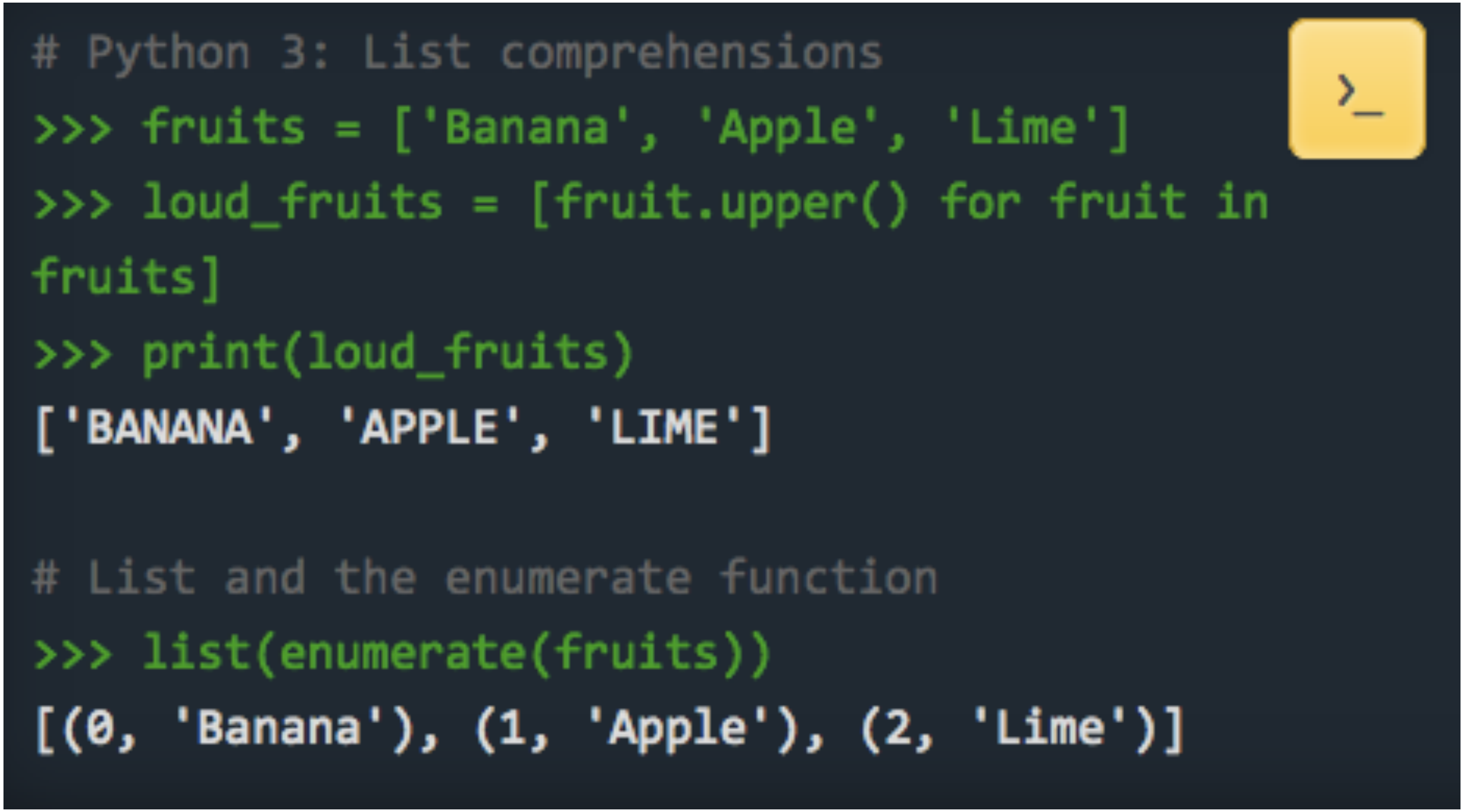
# # Python 3: Simple arithmetic

```
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

# # Python 3: Simple output (with Unicode)

```
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```



```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

# # For loop on a list

```
>>> numbers = [2, 4, 6, 8]
>>> product = 1
>>> for number in numbers:
... product = product * number
...
>>> print('The product is:', product)
The product is: 384
```

# ANDROID

# Hello world

```
import sl4a
droid = sl4a.Android()
droid.makeToast(‘Hello, Android!’)
Print(‘Hello World!’)
```

# Say time

```
Import sl4a
Import time

Droid = sl4a.Android()
Droid.ttsSpeak(time.strftime(%I %M %p on %A,
%B %e, %Y)
```

# Take a picture

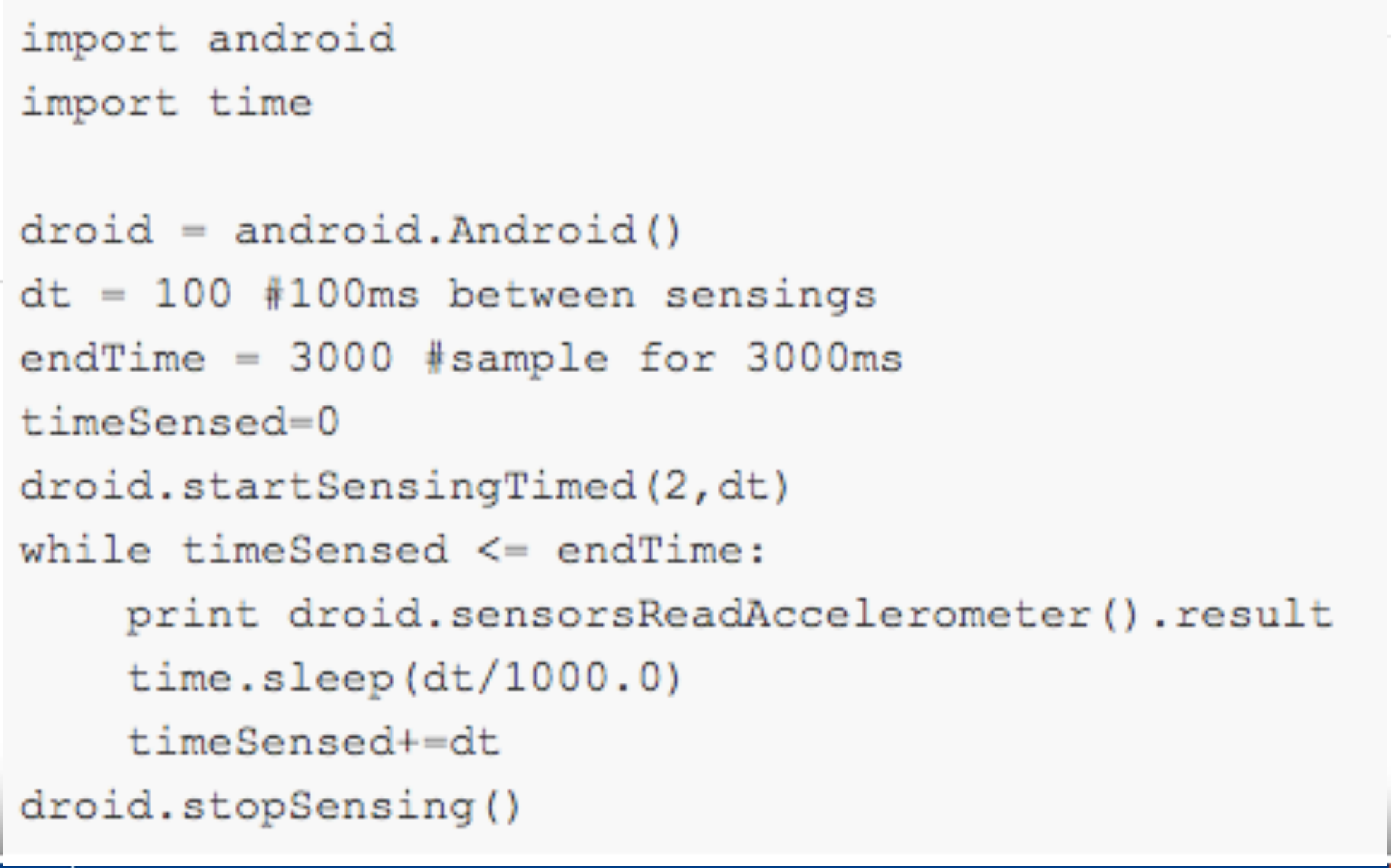
```
import sl4a

droid = sl4a.Android()
droid.cameraINteractiveCapturePicture(‘/
sdcard/qpython.jpg’)
```

# Send sms

- Import androidhelper
- Droid = androidhelper.Android()
- Droid.smsSend(“00628118111610”,test sms by python”)

# Read Accelerator



```
import android
import time

droid = android.Android()
dt = 100 #100ms between sensings
endTime = 3000 #sample for 3000ms
timeSensed=0
droid.startSensingTimed(2,dt)
while timeSensed <= endTime:
    print droid.sensorsReadAccelerometer().result
    time.sleep(dt/1000.0)
    timeSensed+=dt
droid.stopSensing()
```