

# Programmazione Android per esseri umani

*Roberto Bettazzoni*



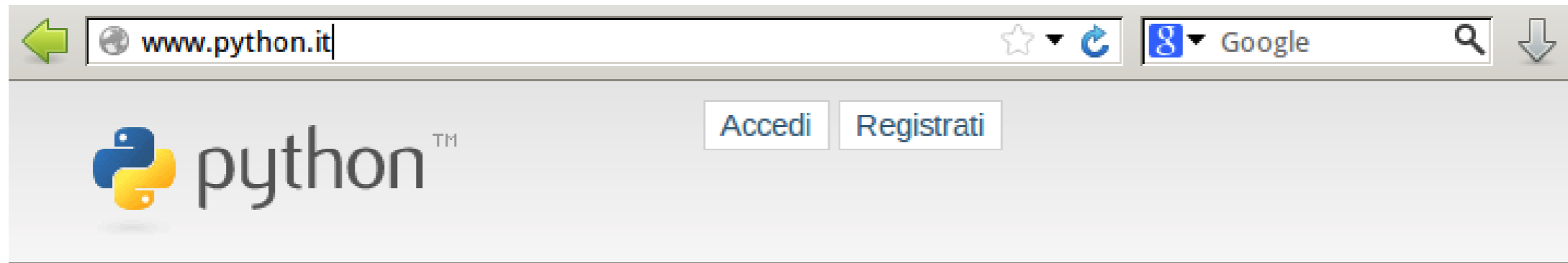
**ERLUG**



```
print "hello world"
```



www.python.it



- COS'È PYTHON >>
- DOCUMENTAZIONE >>
- DOWNLOAD >>
- NEWS >>
- COMUNITÀ >>

**Quick Links (2.7.5)**

- » Documentazione in italiano
- » Documentazione in inglese
- » Installer Windows
- » Sorgenti
- » Indice dei pacchetti

**Quick Links (3.3.2)**

- » Documentazione in inglese
- » Installer Windows
- » Sorgenti

## Python - sito ufficiale della comunità italiana

**Python è un linguaggio di programmazione dinamico orientato agli oggetti utilizzabile per molti tipi di sviluppo software. Offre un forte supporto all'integrazione con altri linguaggi e programmi, è fornito di una estesa libreria standard e può essere imparato in pochi giorni. Molti programmatori Python possono confermare un sostanziale aumento di produttività e ritengono che il linguaggio incoraggi allo sviluppo di codice di qualità e manutenibilità superiori.**

**La AFNIC usa Python**

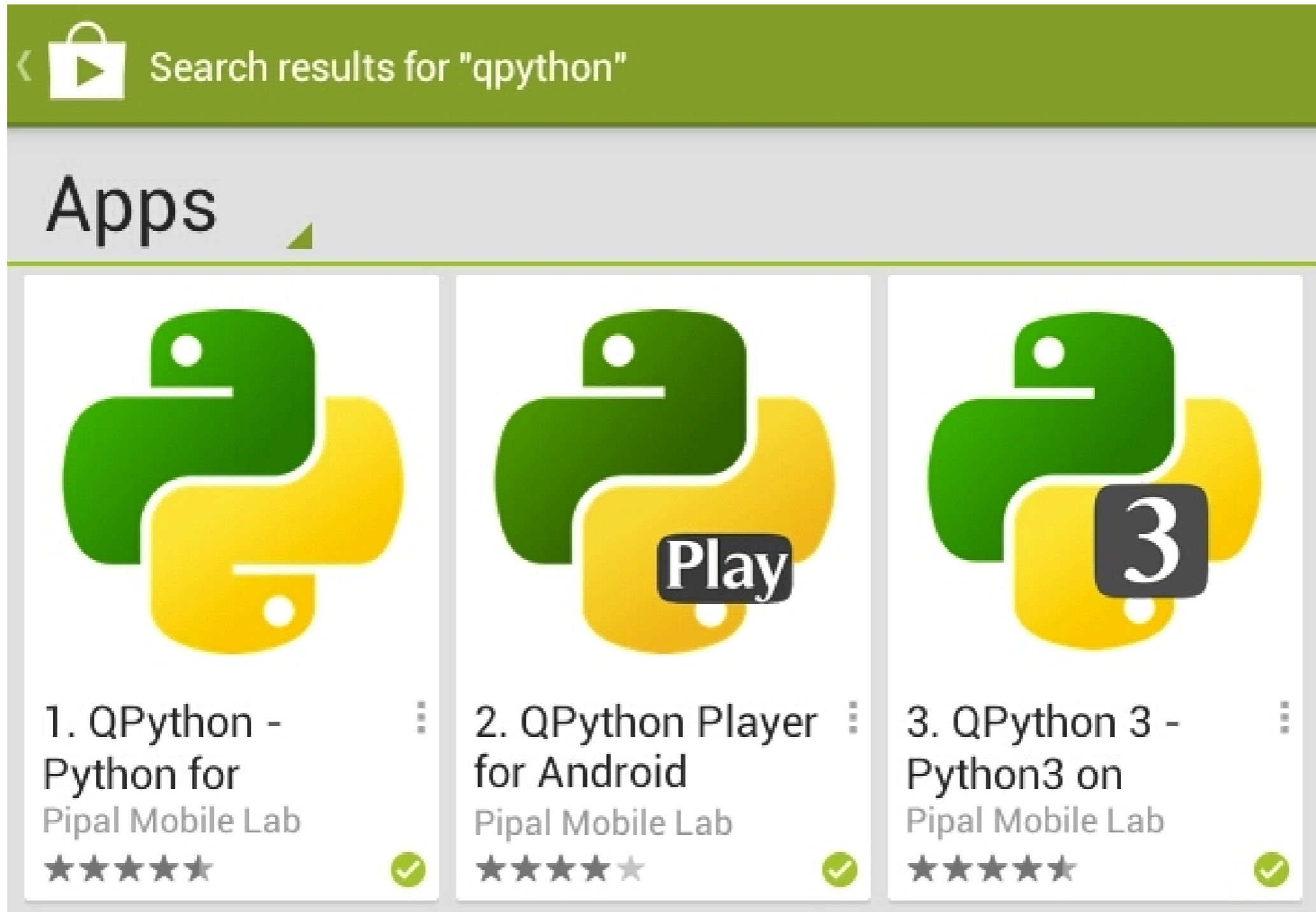
...insieme a molti altri.

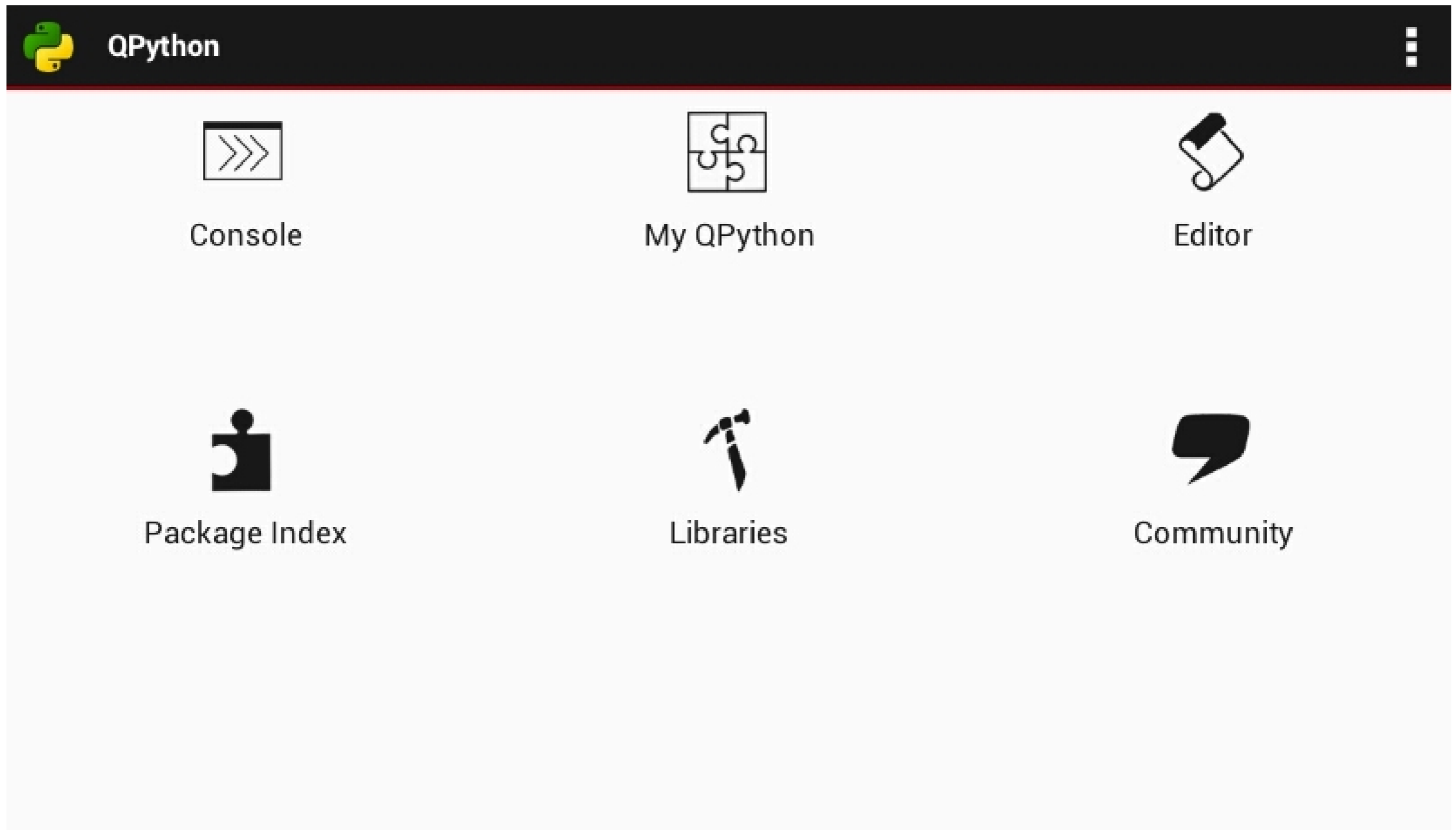
**Notizie su Python**

- » [Creazione sito web dinamico](#)
- » [Sviluppo software scientifico in Python](#)
- » [\[im\]possible living cerca](#)



# Qpython: python for Android





```
import web

webapp=web.application('/(.*)', 'hello', globals())

class hello:
    def GET(self, name):
        return "<html><body>Hello world</body></html>"

if __name__ == "__main__":
    webapp.run()
```



# SL4A Script Layer for Android

The screenshot shows the Code.google.com page for the 'android-scripting' project. The browser address bar shows 'https://code.google.com/p/android-scripting/'. The page title is 'android-scripting' with the subtitle 'Scripting Layer for Android brings scripting languages to Android.' There is a search bar and a 'Search projects' button. Navigation tabs include 'Project Home', 'Downloads', 'Wiki', and 'Issues'. Below these are 'Summary' and 'People' tabs. The 'Project Information' section shows '+588 Recommend this on Google', 'Project feeds', 'Code license Apache License 2.0', and 'Labels' including Google, Android, Lua, BeanShell, Scripting, Python, Perl, JRuby, Tcl, JavaScript, Ruby, and Shell. The 'Members' section lists 'damonkoh...@gmail.com', 'rjmatthews62', and '17 contributors'. The 'Featured' section lists several APK downloads: 'PythonForAndroid\_r4.apk', 'beanshell\_for\_android\_r2.apk', 'lua\_for\_android\_r1.apk', 'perl\_for\_android\_r1.apk', 'rhino\_for\_android\_r2.apk', and 'sl4a\_r6.apk', with a 'Show all >>' link. The main content area contains a notice that 'SL4A's source has moved to github', a description of SL4A's capabilities, a list of supported scripting languages (Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl, and shell), and a note that SL4A is alpha quality software. A QR code and an 'Ohloh I USE IT' button are also present.



# SL4A Script Layer for Android

The screenshot shows the Google Code project page for 'android-scripting'. A blue callout box with a black border points to the 'Wiki' link in the navigation menu. The text inside the callout box reads 'Documentazione sulle API'. The page content includes project information, a list of members, a featured downloads section, and a main text area with a QR code and an 'Ohloh I USE IT' button.

**Documentazione sulle API**

Project Home | Downloads | Wiki | Issues

Summary | People

**Project Information**

+588 Recommend this on Google

[Project feeds](#)

**Code license**  
[Apache License 2.0](#)

**Labels**  
Google, Android, Lua, BeanShell, Scripting, Python, Perl, JRuby, Tcl, JavaScript, Ruby, Shell

**Members**  
[damonkoh...@gmail.com](#), [rjmatthews62](#), 17 contributors

**Featured**

**Downloads**  
[PythonForAndroid\\_r4.apk](#)  
[beanshell\\_for\\_android\\_r2.apk](#)  
[lua\\_for\\_android\\_r1.apk](#)  
[perl\\_for\\_android\\_r1.apk](#)  
[rhino\\_for\\_android\\_r2.apk](#)  
[sl4a\\_r6.apk](#)  
[Show all »](#)

SL4A's source has moved to [github](#). The issue tracker, wiki, and downloads will continue to be hosted here.

Scripting Layer for Android (SL4A) brings scripting languages to Android by allowing you to edit and execute scripts and interactive interpreters directly on the Android device. These scripts have access to many of the APIs available to full-fledged Android applications, but with a greatly simplified interface that makes it easy to get things done.

Scripts can be run interactively in a terminal, in the background, or via [Locale](#). Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl, and shell are currently supported, and we're planning to add more. See the [SL4A Video Help](#) playlist on YouTube for various demonstrations of SL4A's features.

SL4A is designed for developers and is *alpha* quality software. Please report bugs and feature requests on the [issues list](#). You can download the current APK by scanning or clicking the following barcode:

 [Ohloh](#) I USE IT

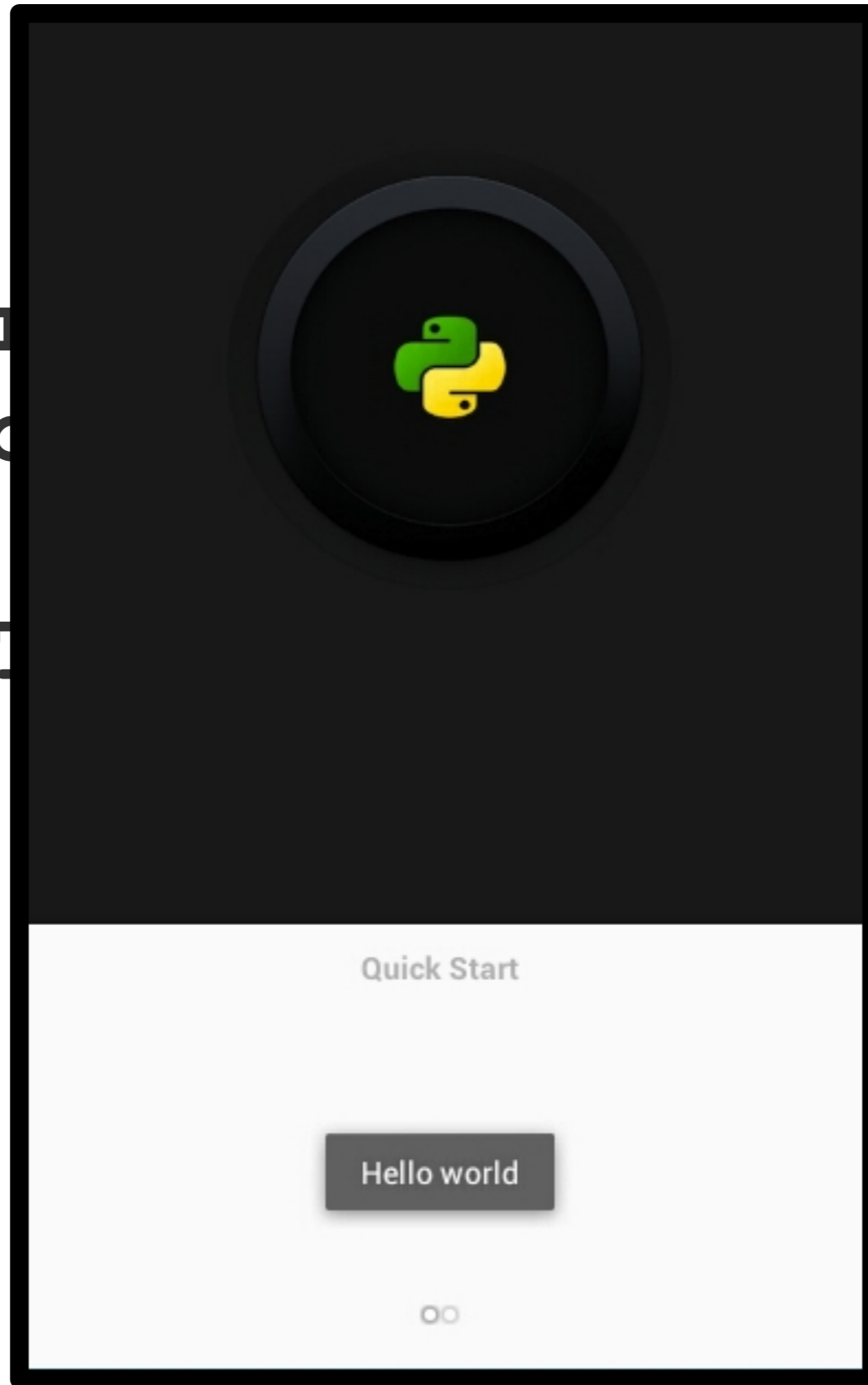




```
import androidhelper  
droid = androidhelper.Android()  
  
droid.makeToast("Hello world")
```



```
import android  
droid = android.R.  
  
droid.makeText(R.  
R.string.hello_world)
```



```
import androidhelper  
droid = androidhelper.Android()  
  
droid.ttsSpeak("Hello world")
```



```
import androidhelper
droid = androidhelper.Android()

droid.smsSend("+39333666999",
              "Hello world")
```



```
import androidhelper
droid = androidhelper.Android()

droid.sendEmail(
    to="you@youraddress.org",
    subject="Ciao",
    body="Hello world")
```



```
import smtplib
from email.mime.text import MIMEText

me = "roberto@bettazzoni.it"
you= "you@youraddress.org"
msg = MIMEText("Hello world")
msg['Subject'] = "Ciao"
msg['From'] = me
msg['To'] = you
s = smtplib.SMTP('127.0.0.1')
s.sendmail(me, [you], msg.as_string())
s.quit()
```



```
import androidhelper, web

urls = ('/(.*)', 'hello')
App = web.application(urls, globals())
Droid = androidhelper.Android()

class hello:
    def GET(self, name):
        location = Droid.getLastKnownLocation().result
        location = location.get('network', location.get('gps'))
        return "<html><body><h3>I am here</h3> </body></html>"

if __name__ == "__main__":
    App.run()
```



```
def gmapUrl(latitude, longitude):  
    return "'http://maps.googleapis.com/maps/api/staticmap?center=%s,  
%s&zoom=12&size=400x400&sensor=false&markers=color:green|label:I|%s,%s'"  
% (  
    latitude, longitude, latitude, longitude)
```





# Interfaccia Utente



kivy.org



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



**Cross platform**

Kivy is running on Linux, Windows, MacOSX, Android and IOS. You can run the same code on all supported platforms.

It can use natively most inputs protocols and devices like Wacom, Touch, Wacom, Pen, Mac OS X



**Business Friendly**

Kivy is 100% free to use, under MIT licence (for 1.8 and 1.7.2), and LGPL 3 for the previous versions. The toolkit is professionally developed, backed and used. You can use it in a product and sell your

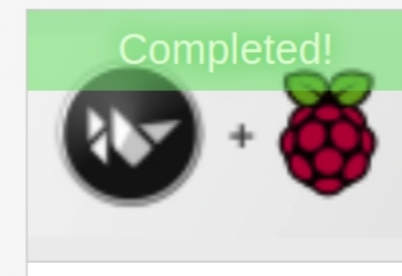


**GPU Accelerated**

The graphics engine is built over OpenGL ES 2, using modern and fast way of doing graphics.

The toolkit is coming with more than 20 widgets designed to be extensible. Many parts are written in C using Python, tested

**Funded by the community**



```
from kivy.app import App
from kivy.uix.label import Label

class HelloWorldApp(App):
    def build(self):
        return Label(text='Hello World')

if __name__ == '__main__':
    HelloWorldApp().run()
```



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

hw='Hello World'

class HelloWorldApp(App):
    def build(self):
        b = Button(text=hw, font_size=70)
        def callback(instance):
            b.text=hw if b.text[0]!="H" else "Ciao mondo"
        b.bind(on_press=callback)
        return b

if __name__ == '__main__':
    HelloWorldApp().run()
```



```
from kivy.app import App
from kivy.uix.widget import Widget
from kivy.uix.label import Label

class HelloWorld(Widget):
    def on_touch_down(self, touch):
        with self.canvas:
            Label(font_size=50, pos=(touch.x, touch.y),
                text = "Hello World")

class HelloWorldApp(App):
    def build(self):
        return HelloWorld()

if __name__ == '__main__':
    HelloWorldApp().run()
```



# GRAZIE PER L'ATTENZIONE

**Le slides e le riprese audio/video  
dell'intervento saranno disponibili su:**

**<http://erlug.linux.it/linuxday/2013/>**

