

# NDKで「Lua」を組み込んでみる

日本Androidの会

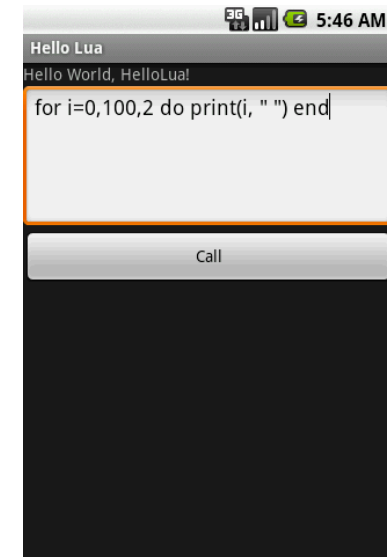
木南英夫

# Luaライブラリの作成

- <http://www.lua.org/> から、lua-5.1.4.tar.gz をダウンロード
- srcディレクトリにAndroidからの呼び出しコードのファイルを追加 (android.c)
- Android.mkを作成
  - LOCAL\_MODULE := lua
  - LOCAL\_SRC\_FILES := 必要なファイルのリスト
    - (locale.hの部分でエラーが出たので、一か所削除)

# Luaアプリの作成

- apps/hellolua/Application.mkの作成
  - APP\_PROJECT\_PATH := \$(call my-dir)/project
  - APP\_MODULES := lua
- Eclipseで通常の手順でアプリを作成
  - \${ndk}/app/hellolua/projectに作成
  - EditText, Button, TextViewを配置



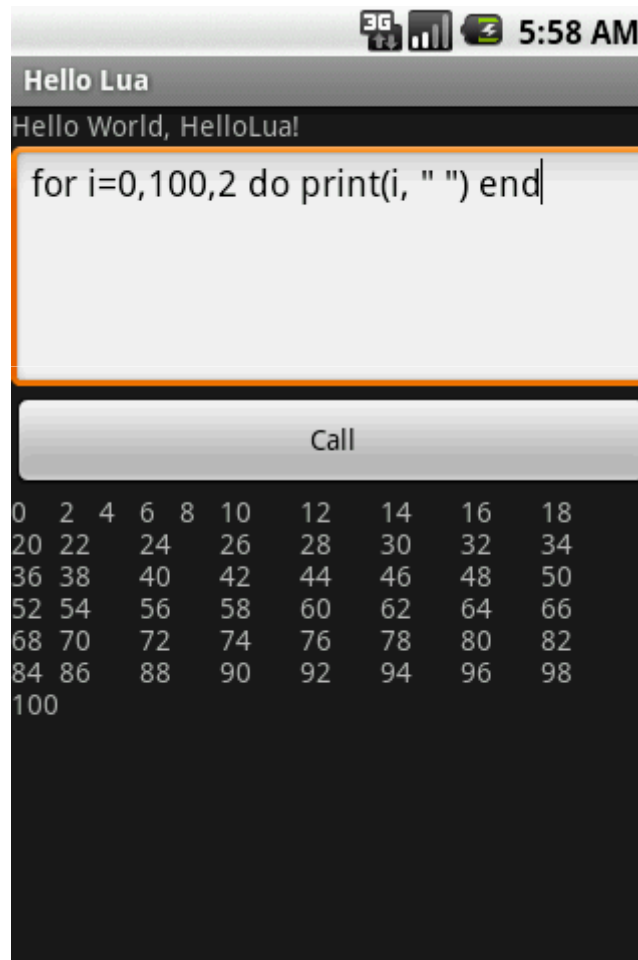
# Javaのnativeメソッドの定義

- Javaのクラスにnativeメソッドを定義する
  - public class HelloLua extends Activity {  
    public native String doCall(String script);
- クラス名を指定して、JNIのヘッダーを生成
  - javah -jni -classpath  
    \${ndk}/apps/hellolua/project/bin  
    -d \${ndk}/sources/samples/lua-5.1.4/src  
    jp.hews.hellolua.HelloLua

# Nativeメソッドの定義

- C言語の呼び出しコードを記述
  - jstring JNICALL  
Java\_jp\_hews\_hellolua>HelloLua\_doCall  
(JNIEnv \* env, jobject jobj, jstring jstr) {}
- JNIの初期化ルーチンで、ライブラリの初期化と、print関数の定義
  - jint JNI\_OnLoad(JavaVM \*vm, void \* reserved)

# 実行



```
3G 5:58 AM
Hello Lua
Hello World, HelloLua!
for i=0,100,2 do print(i, " ") end|
Call
0 2 4 6 8 10 12 14 16 18
20 22 24 26 28 30 32 34
36 38 40 42 44 46 48 50
52 54 56 58 60 62 64 66
68 70 72 74 76 78 80 82
84 86 88 90 92 94 96 98
100
```