

How to get the path of a Python script

Why?

In old fashioned DOS manner I want to have a configuration file in the same folder as my program.

Should it be done this way in Linux?

Unix people would say: No! The better way is to store config files where they should be, namely in the hidden .config folder in the home directory.

So for example for user jcf and program myprog the file location would be /home/.config/myprog.conf

The advantage is clear:

- The path is absolute, the location is clear
- Every user can have his own configuration file

However the Python script must determinate the path to the home folder.

Can it be done?

It is strange how many possibilities to do it are found in diverse forums. However they mostly don't work, if the folder is changed during script run.

To test these, I wrote a simple script that gives the path, changes the working folder to "/" and gives the path again:

```
import os.path, os, __main__
import sys, string

def testprogpah():
    print "1", os.path.dirname(sys.path[0])
    print "2", os.path.dirname(os.path.realpath(sys.argv[0]))
    print "3", os.path.dirname(os.path.abspath(getattr(__main__, '__file__', '__main__.py')))
    print "4", sys.path[0]
    print

testprogpah()
os.chdir("/")
testprogpah()
```

The output looks like this:

```
1 /media/sda2_doc/4_JCProg/Python
2 /media/sda2_doc/4_JCProg/Python/Library
3 /media/sda2_doc/4_JCProg/Python/Library
4 /media/sda2_doc/4_JCProg/Python/Library

1 /media/sda2_doc/4_JCProg/Python
2 /
3 /
4 /media/sda2_doc/4_JCProg/Python/Library
```

when the script is started from the folder /media/sda2_doc/4_JCProg/Python/Library

This means that the methods 2,3,4 work when invoked directly after the start of the script.
But only method 4 works when the current folder is changed by the script.

Conclusion:

`sys.path[0]` gives the folder from which the script was started.

So, to build the name for a config file `config.txt`, you can use

```
cfgfile = os.path.join(sys.path[0], 'config.txt')
```

And if I want to do it the Linux way?

I could use

```
homefolder = os.getenv("HOME")
```

to get the home folder

and

```
cfgfile = os.path.join(os.getenv("HOME"), ".config.myprog.conf")
```

to directly build the name for the config file.