

2009 ♀恋莉の小北♂ Python 教程

第一节 前言&准备



By: 刘志宇 (♀恋莉の小北♂)

给所有的 Python 学习者:

首先让我先感谢一下 Python 给我们提供一个如此简单实用且易于入门的平台。是 Python 使得我们的手机更加智能, 更加多彩。同时也要感谢大家对于 Python 的喜爱与支持, 正是由于大家对于 Python 的喜爱与支持才使得 Python 发展如此迅速。其次, 我借着这个机会感谢一下我所有的学员。如果不是你们提出问题, 也许我也不会对 Symbian 系统了解了如此多, 虽然那仅仅是九牛一毛。有一句话说的很对, 帮助别人就是在提高自己。最后我要感谢八神智能天下 (wap.8zntx.com), 不是虚伪的恭维, 而是因为我从最开始什么都不懂到现在会一些皮毛都是从八神智能天下学习的, 我做过 8 个月的 95 区版主, 然后跳到三版等等。如果没有这些我不会接触到智能机系统, 不会接触到编程, 是智能机成就了今天的我。而我现在在这里给大家讲解并不是说我比你们强, 而是说我们大家都是 Python 的学习者, 我们在互相探讨。手机 Python 虽然是比较容易入门的语言, 但是依然需要一定的高度。如果您现在还不了解手机基本的一些使用例如: 汉化。这是属于最基本的东西了。如果您还没有了解这些, 或许您应该从基础学起。编程属于一种长时间, 长耐性的工作, 不可一蹴而就。如果您抱着想让自己成为一颗耀眼的星星的话, 您还是别来了。因为当您学会一些东西以后就会停滞不前了。如果在编程过程中出现了任何错误, 先不要问别人, 尽力自己去改正! 不停地尝试不停地改, 当您改好的那个时候您就会深刻地理解了这段代码。好了, 说了这么多了, 如果您已经理解了全部, 并决定坚持下去。相信我, 您一定会成功。

准备工作:

由于本教程是针对于手机 Python。并不涉及到电脑 Python。我们仅仅需要一些手机软件。

- ① Python 平台&模块
- ② X-plore (X 管理器) By: Lonely Cat Games
- ③ Ped - Python IDE By: Arkadiusz Wahlig
- ④ Ded By: Juriy Bakunin aka JBAK
- ⑤ py2sis By: Maxim6630&Santehni

很抱歉很多软件我不知道中文名字, 但是英文名字是没错的。如果有必要的话还需要汉化风暴 (By: 雨中彩虹), 我觉得这个是非常非常不错的软件!

2009 ♀恋莉の小北♂ Python 教程

第二节 了解 Python 的世界



By: 刘志宇 (♀恋莉の小北♂)

Python 简介:

了解 Python 对于我们 Python 学习者是相当重要的,

Python(发音:['paiθ(ə)n; (US) 'paiθo n]n.蟒蛇, 巨蛇), 是一种面向对象的解释性的计算机程序设计语言, 也是一种功能强大而完善的通用型语言, 已经具有十多年的发展历史, 成熟且稳定。Python 具有脚本语言中最丰富和强大的类库, 足以支持绝大多数日常应用。(From: <http://baike.baidu.com/view/21087.htm>)

也许你会认为这是废话, 但是对于 Python 学习者, 连 Python 是什么都不知道, 这是一件多么荒谬的事情。

手机 Python 程序结构简单介绍:

一个手机 Python 程序虽然是 sis 格式的, 但是却是 Totally different。当我们解包一个 Python 软件, 这里拿我的软件举例 (因为解包他人的软件然后发布出来是属于侵犯版权的行为)

现在我们解包软件 定时管家 v1.0 By: 刘志宇 (♀恋莉の小北♂)

在 X 管理器中我们看到了 unpacked 文件夹中看到了! 文件 进入发现有三个文件夹: private resource 和 sys 文件夹。我们平时写的代码其实只是在 private 文件夹内一个后缀为 py 的文件。在本软件中, 这个文件在 private 文件夹内的 e35e00df 中。打开之后就是我们实用 Ped 进行开发的时候写的代码了。也就是真正需要我們进行操作的地方。

*手机 Python 程序内容一些简单介绍:

首先在 Python 中需要引入你所需要的模块。就像 C++ 中的头文件, 呵呵 我还是会一些 C++ 的。通常是以 import xxxxx 来开头的或者是 from xxxx import xxxx 这些就是需要引入所有你需要的模块。

在定义一些动作的时候, 使用 def xxxx0:来进行定义, 注意那两个括号是不能去掉的, 同时下面所定义的动作要进行缩进。缩进就是让手机识别出运行顺序 (不知道怎么说, 缩进可以理解为让程序懂得哪些动作是被定义为什么的, 在后面执行这个命令的时候才会懂得执行你前面定义的动作。不懂得多看几遍吧。或者进行尝试)

缩进一般是四个空格。如果在二级缩进就是8个空格咯。这个以后会用到, 今天只是浅显的提一下缩进。这对于一些逻辑思维不好的人, 这是一个非常难的难点。

定义好了一系列动作之后,在我们后面用到这一系列动作的时候只需要用定义的字符来代替那些长长的动作命令 例如:

```
def exit():
```

```
    appuifw.app.set_exit()
```

定义好了 exit 之后,当我们下面需要程序作出这个动作的时候我们就只需要打出 exit()就可以让程序明白我们要执行 appuifw.app.set_exit() 这个命令

在 Python 中,由于程序作者也许会经常修改软件,也许当几个月之后一看到代码竟然忘了这段代码的作用,于是这时候你就需要注释了。在 py 中,我们需要用“#”来引导注释行,Python 会自动跳过“#”开始的行, In another word, “#”开头的行不会影响程序的运行。例如:

```
def cn(x):return x.decode("utf-8")
```

```
#定义中文编码模式
```

程序不会运行那个带有#号的行。

2009 ♀恋莉の小北♂ Python 教程

第三节 快速入门



By: 刘志宇 (♀恋莉の小北♂)

如何开始手机编写 Python 程序

这里我们将会使用到 Ped - Python IDE By: Arkadiusz Wahlig 我想既然你们想学 Python,英语就一定要好一些。打开 Ped - Python IDE 之后,我们可以建立一个 Python 源码文件然后开始写代码。

编写简单的 Python 程序

每个 python 程序都包含一个或者多个函数,函数有执行函数的功能语句序列组成。操作系统则通过调用函数来执行程序,函数则执行其所被定义的一系列动作。

下面我们建立一个最简单的函数,它仅仅是一个提示框,也就是我们学习任何计算机语言的第一步“Hello world”

```
import appuifw
```

#import 是引入模块的意思, appuifw 是我们需要用到的模块

```
appuifw.note(u"Hello world","info")
```

#appuifw.note 意思使用 appuifw 模块中的 note, u 代表意思是 u 码,注意 u 后面不可以直接跟中文,因为中文的编码模式没有定义。Info 是类型,也就是提示框的类型。

这是一个最简单的 Python 程序,也是每个学习的人必须要经过的一段路。

appuifw 模块用于实现界面设计。它提供了可视化用户界面框

如果想把 Hello world 写成为中文。那么就需要定义为 utf-8 编码模式

```
import appuifw
```

```
def cn(x):return x.decode('utf-8')
```

```
appuifw.note(cn("我是小北,欢迎来到八神智能天下"),"info")
```

讲解一下 def 的意思为定义，我们经常会用到的，用来定义某一系列函数，cn 是名字而已，如果你想也可以定义为 qq，x 在这个里面是你所输入的东西，在：后面定义了执行的函数，即将 x 使用 utf-8 编码模式

我想大家应该理解了吧，试着写出自己独特的 Hello world 程序来看看？

下面我想为大家介绍一个 appuifw 所有可用的函数（作为一种了解而已）

```
['Canvas', 'Content_handler', 'EAColumn', 'EApplicationWindow', 'EBColumn', 'EBatteryPane', 'ECColumn', 'EContextPane', 'EControlPane', 'EControlPaneBottom', 'EControlPaneTop', 'EDCcolumn', 'EEEventKey', 'EEEventKeyDown', 'EEEventKeyUp', 'EFindPane', 'EHCenterVBottom', 'EHCenterVCenter', 'EHCenterVTop', 'EHLeftVBottom', 'EHLeftVCenter', 'EHLeftVTop', 'EHRightVBottom', 'EHRightVCenter', 'EHRightVTop', 'EIndicatorPane', 'EMainPane', 'ENaviPane', 'EScreen', 'ESignalPane', 'EStaconBottom', 'EStaconTop', 'EStatusPane', 'EStatusPaneBottom', 'EStatusPaneTop', 'ETitlePane', 'EUniversalIndicatorPane', 'EWallpaperPane', 'FFormAutoFormEdit', 'FFormAutoLabelEdit', 'FFormDoubleSpaced', 'FFormEditModeOnly', 'FFormViewModeOnly', 'Form', 'HIGHLIGHT_ROUNDED', 'HIGHLIGHT_SHADOW', 'HIGHLIGHT_STANDARD', 'Icon', 'InfoPopup', 'Listbox', 'STYLE_BOLD', 'STYLE_ITALIC', 'STYLE_STRIKETHROUGH', 'STYLE_UNDERLINE', 'Text', '__all__', '__builtins__', '__doc__', '__file__', '__name__', '__appuifw__', 'app', 'available_fonts', 'e32', 'multi_query', 'multi_selection_list', 'note', 'popup_menu', 'query', 'selection_list']
```

这些是 appuifw 模块的所有可用函数，我们的 hello world 就是使用了其中的 note，在代码中写成为 appuifw.note(u"hello world","info")，呵呵 尝试一下使用别的代码试试，仅仅作为一个了解而已 呵呵 当然我们以后会详细讲解的

下一节:模块的三种加载方式

2009 ♀恋莉の小北♂ Python 教程

第四节 三种加载模块的方式



By: 刘志宇 (♀恋莉の小北♂)

其实在 Python 的世界里，不仅仅有那么一种加载方式，在 Python 的世界里其实是有多种的变通方式，我相信在你们打开别人的源码时候会发现有些软件在引入模块的时候使用的是 import xxxx 有些是 from xxx import *。那么到底有多少种加载方式呢？

在 Python 编程中，我们要应用 xxx 模块的函数 ooo，我们最常用的方式就是

```
Import xxx
```

```
Xxx.ooo()
```

这种的意思其实就是说引入 xxx 模块，然后执行 xxx 模块中的 ooo 函数

其实我们还有其他两种方式来引入这个模块同时应用这个命令

```
Import xxx as qq
```

```
qq.ooo ()
```

这种方式只是把 xxx 模块定义为 qq，也就是 shortcut（快捷？好像是这么叫）更容易方便简洁的引入。

现在我们介绍最后一种 这种方式我不太喜欢 因为容易造成错误

```
From xxx import *
```

```
Ooo()
```

或者

```
From xxx import ooo
```

```
Ooo()
```

这种加载方式的意思就是在 xxx 模块中加载 ooo 函数，这种方式在下面不需要写出模块名，所以相对方便但是又极为容易重名，使用这种方式一定要注意重名的问题，在第一段中的*号的意思就是全部，from xxx import * 意思就是在 xxx 模块中引入全部函数。

讲的这么简单不知道大家明白没有，希望大家不要着急，基础知识才是最重要的东西，写出程序不代表你基础知识很扎实。

下一节：构建简单的程序界面

2009 ♀恋莉の小北♂ Python 教程

第五节 构建简单的程序界面



By: 刘志宇（♀恋莉の小北♂）

上节我们讲到了三种模块的加载方式，这一节，让我们继续领略 py 的魅力。相信大家已经可以写出了简单的 py 程序“hello world”。

但是那个简单到不能再简单的 py 程序怎么能让我们满足呢。所以这一节我们来讲一讲简单的程序界面的构建。

今天，我们就来讲一讲文本格式的程序界面和图片格式的程序界面。

首先，对于文本格式的程序界面 我们需要 appuifw 模块来进行操作。

```
import appuifw
```

```
def cn(x):return x.decode('utf-8')
```

```
appuifw.app.body=appuifw.Text()
```



```
appuifw.app.body.set(cn"我是小北，欢迎访问八神智能")
```

当你在 pyshell 中输入了上面的代码之后，会发现程序一闪而过，这是为什么呢？

对了，这是因为我们需要让程序等待我们操作而不是仅仅运行一遍程序，那么我们就在程序的最后面加上让程序等待我们操作的代码 `e32.Ao_lock().wait()`，所以还要在前面引入模块的地方加上 `e32` 模块 于是程序变形为

```
import appuifw, e32

def cn(x):return x.decode('utf-8')

appuifw.app.body=appuifw.Text()

appuifw.app.body.set(cn"我是小北，欢迎访问八神智能")

e32.Ao_lock().wait()
```

到这里，我想你应该学会了如何添加文本程序界面了吧？

如果你想要在这个程序界面上再添加一些文字的话，那么你只需要在中间加上一个代码

```
appuifw.app.body.add(cn("小北 2009"))
```

看一看代码就知道了，这段代码的意思就是在程序界面上添加一段文字

除了简单的文本程序界面，我还想大家能够了解一下将一个图片设置为程序界面的代码

```
Import appuifw, e32, graphics

def cn(x):return x.decode('utf-8')

#首先引入模块和定义中文模式

img=graphics.Image.open("C:\\System\\Apps\\xxxxxx")

#定义图片的位置

def handle_redraw(rect):

****canvas.blit(img)

Appuifw.app.body=appuifw.Canvas(event_callback=None,redraw_callback=handle_redraw)

#定义背景为图片

e32.Ao_lock().wait()
```

这个图片程序界面还是比较有难度的，希望大家能够认真的理解这部分代码

好了，今天就到这里了，明天继续。

下一节：构建简单的程序菜单

2009 ♀恋莉の 小北♂ Python 教程

第六节 构建菜单选项



By: 刘志宇 (♀恋莉の小北♂)

对于简单的程序界面，相信大家都已经了解的很清楚了，这一节，我们来讲解构建简单的菜单选项。

Appuifw 模块是我们构建菜单选项的重要模块。

```
appuifw.app.menu = [(cn("菜单选项"),执行命令)]
```

其中，菜单选项就是在程序选项中所显示的文字，例如：退出 输入等等。

而执行命令就是在前面源代码中所定义的一系列指令。

例如：

我们在前面的源代码中定义了 **exit** 这个指令，然后想在菜单中显示出“退出”。

那么：

```
import appuifw  
  
def cn(x):return x.decode('utf-8')  
  
def exit():  
  
****appuifw.app.set_exit()  
  
appuifw.app.menu = [(cn("退出"),exit)]
```

这里的意思就是，首先引入了 **appuifw** 模块，然后定义 **exit** 执行命令。

Appuifw.app.set_exit()这行命令是退出软件的意思

然后定义菜单选项“退出”为执行 **exit** 命令。

那么，如果想再加一个选项怎么办呢？

这时你需要在两个选项的中间加上", "即可

即

```
import appuifw  
  
def cn(x):return x.decode('utf-8')  
  
def exit():  
  
****appuifw.app.set_exit()  
  
def body()  
  
****appuifw.app.body.set(u"thanks for visiting")  
  
appuifw.app.menu = [(cn("退出"),exit),(cn("屏幕增加"),body)]
```

这就是两个选项的菜单选项。

还在等什么 赶快尝试一下 5 选项的菜单吧。

下一节，构建多级选项菜单

2009 ♀恋莉の小北♂ Python 教程

第七节 构建二级菜单



By: 刘志宇 (♀恋莉の小北♂)

对于构建简单的菜单选项，相信大家已经清楚了，这一节，我将会介绍二级菜单的构建方法。

二级菜单我们依然需要 **appuifw** 模块

```
import appuifw
```

```
#引入 appuifw 模块
```

```
def cn(x):return x.decode("utf-8")
```

```
#定义中文
```

```
appuifw.app.body = m = appuifw.Text(cn("欢迎访问八神智能天下"))
```

```
#定义软件界面
```

```
def information():
```

```
appuifw.note(cn("欢迎访问"),"info")
```

```
#定义弹出信息
```

```
def exit():
```

```
****appuifw.app.set_exit()
```

```
#定义退出
```

```
appuifw.app.menu = [(cn("主菜单"),((cn("弹出信息"),infomation),(cn("退出"),exit,)))]
```

```
#定义菜单
```

大家赶快去试试吧。

其实是一个很简单的代码。

今天我主要想介绍一下多级菜单的构建，其实很简单，可以使用多级方法来构建比如，在二级菜单的执行命令中再定义一个选项列表。

```
import appuifw
```

```
#引入 appuifw 模块
```

```
def cn(x):return x.decode("utf-8")
```

```
#定义中文
```

```
appuifw.app.body = m = appuifw.Text(cn("欢迎访问八神智能天下"))
```

```
#定义软件界面
```

```
def information():
```

```
appuifw.note(cn("欢迎访问"),"info")
```

```
#定义弹出信息
```

```
def exit():
```

```
****appuifw.app.set_exit()
```



```

#定义退出

def choice():
    ***choices = [cn("弹出信息"),cn("退出")]
#定义列表选项
    ***index= appuifw.popup_menu(choices,cn("选择"))
    ****if index == 0:
#如果选择第一个选项
        *****information()
#执行命令 information
    ****elif index == 1:
#如果选择第二项
        *****exit()
#执行退出命令
    appuifw.app.menu = [(cn("主菜单'),((cn("选择"),choice))]
#定义菜单和二级菜单

```

今天就到这里了，相信大家知道了如何建立多级菜单，大家多多温习，还是蛮有用的东西。

编程班的同学请把作业直接传到本贴下面就可以了

