

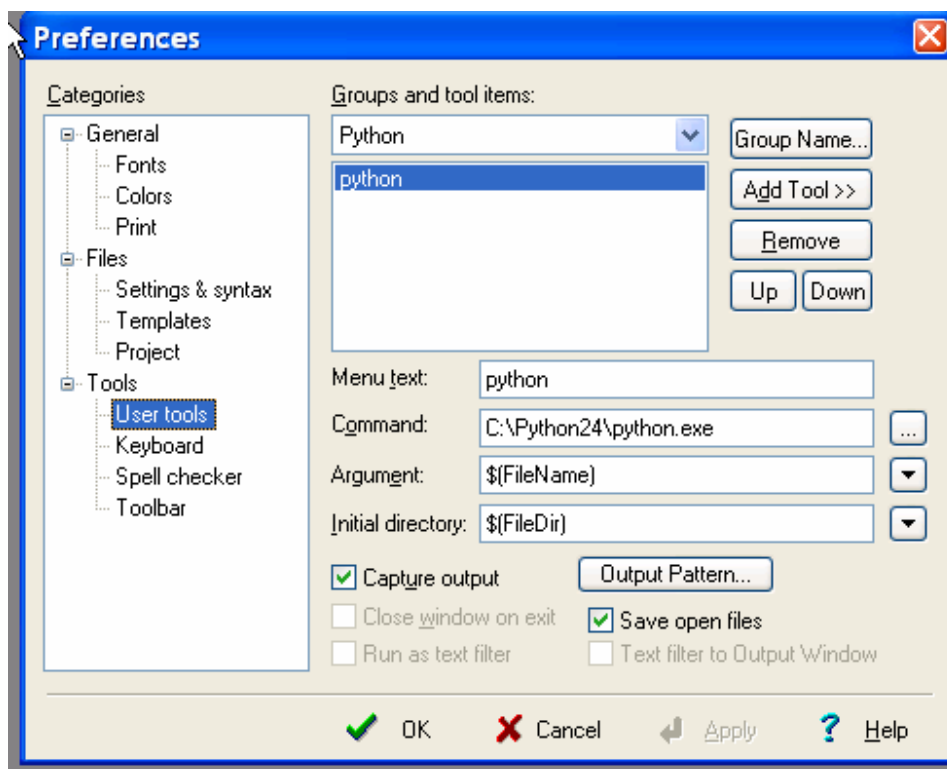
在 Windows 环境下用 Editplus 打造一个 Python 编辑调试环境

<http://lwkj.blogchina.com>

2005-9-25

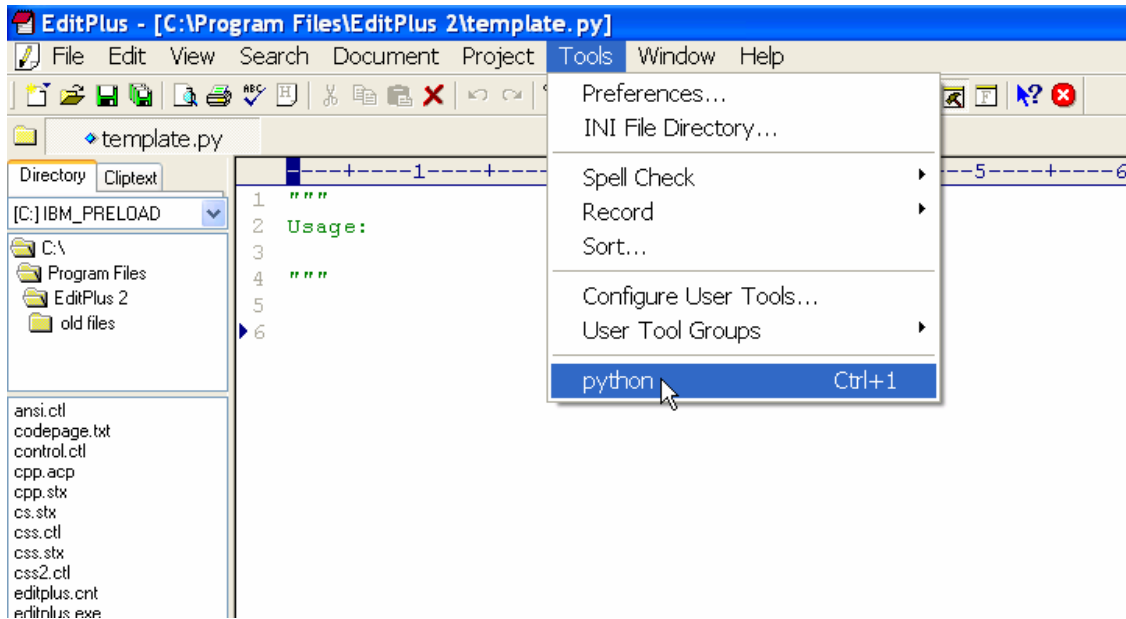
Python 有很多集成开发工具，商业的有 komodo, Wingide, Boa 等，还有 Python 自带的集成环境 IDLE, windows 下还有 Pythonwin 等。但是，这些工具有的是过于复杂，有的是对 Unicode 支持的不太好。因此，今天我们利用一个很流行的，功能也非常强大的文本编辑软件 Editplus，改造成一个能够很好的支持 Python 编辑，浏览，运行及调试的开发环境，并且具备语法加亮，自动完成，自动缩进，新建 python 文件模板等功能支持。我们用的 Editplus 是 v2.20, Python 是 2.4 版。

1. 要使 Editplus 能够运行 python 程序，首先打开 tools->Preferences 对话框，在 User tools 里新建一个 group，名字我们叫 python，在 python group 里新建一个工具也起名为 python，分别在 Menu text: 填入 python，在 Command: 填入 python 的安装路径，Argument: 填入 \$(FileName)，Initial Directory: 填入 \$(FileDir)。完成这一步的结果如下图（1）：



(图 1)

这样我们在 Editplus 里打开一个 python 文件，在 tools 菜单下就会有一个 python 选项，直接单击就可以运行 python 程序了，也可以利用快捷键（Ctrl+1），根据你设置用户工具的个数，这个快捷键可能有所不同。如下图（2）：



(图 2)

在此处我们想实现下面的一个功能：就是当程序出错的时候，python 解释器会给出类似如下的提示信息：

```
----- python -----
```

```
File "test.py", line 6
```

```
print 'a
```

```
^
```

```
SyntaxError: EOL while scanning single-quoted string
```

```
Output completed (1 sec consumed) - Normal Termination
```

此时，我们一般会根据提示的行号到文本中去找错误的地方，而不能象其他的编译器一样直接双击错误的提示，将光标直接移动到错误行。此时如果我们直接双击某行，则会出现找不到某文件的错误。此处，我们就利用 Editplus 的 Output Pattern 的实现这个功能。具体做法如下，在图（1）中的 Capture Output 选项前的复选框要选上，然后打开 Output Pattern 按钮，在 Output Pattern 对话框中，首先取消 Use default output pattern 选项前的复选框，在 Regular expression 栏里填入正则表达式如下：

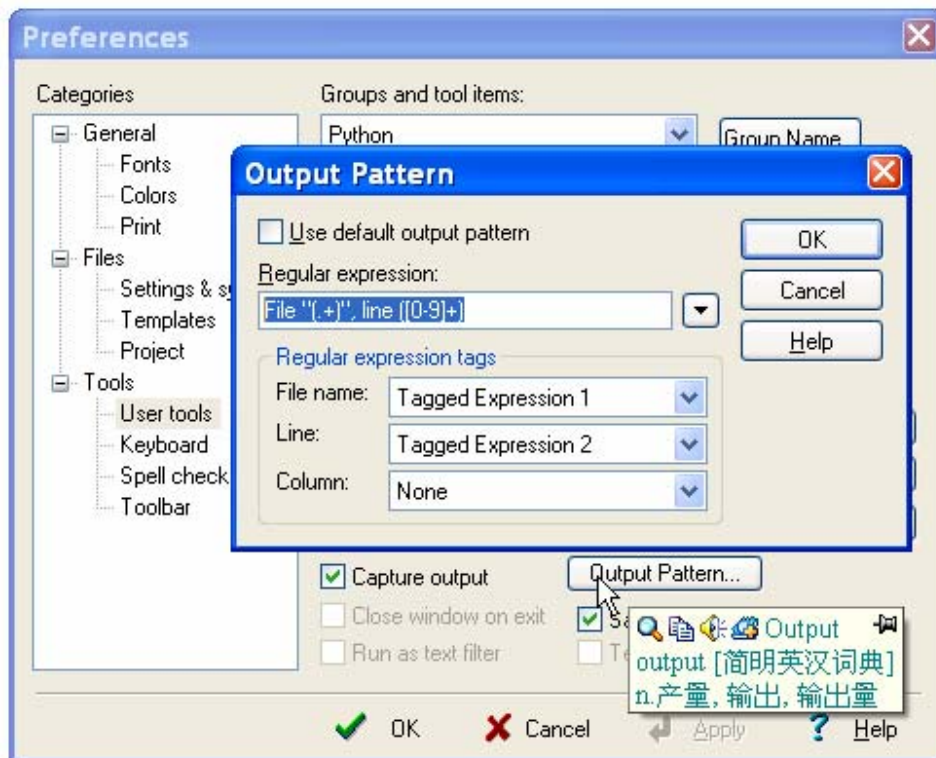
```
File "(.+)", line ([0-9]+)
```

```
File Name 栏里选择 Tagged expression 1，
```

```
Line 栏里选择 Tagged expression 2，
```

```
Column 栏里为 None。
```

结果如图（3）：



(图 3)

然后我们直接双击错误提示信息行

File "test.py", line 6

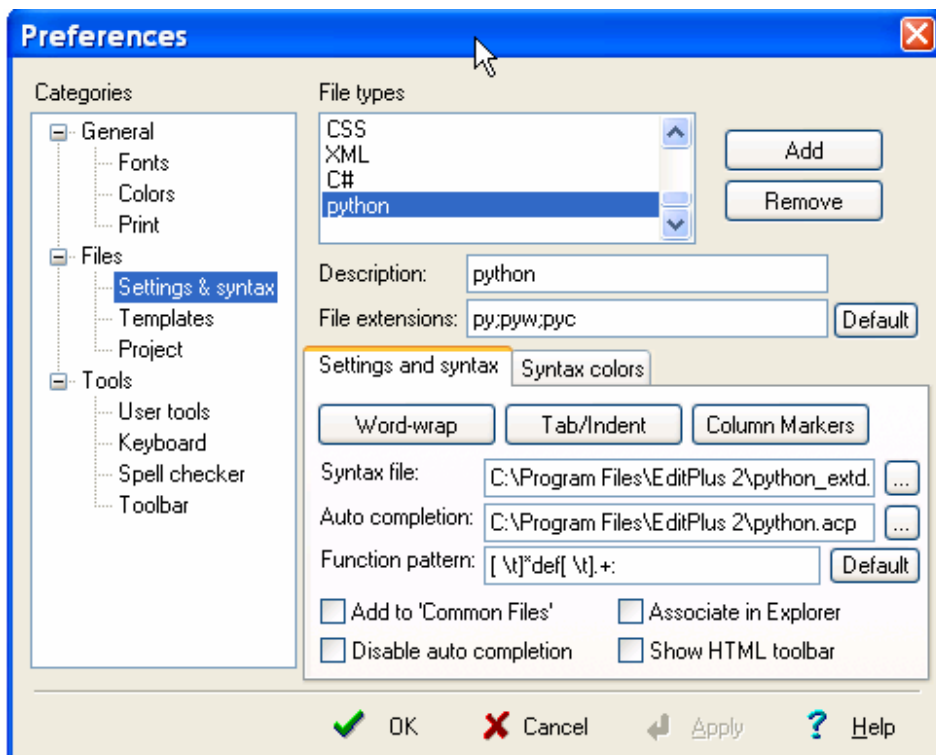
光标就会直接跳到第六行，非常的方便，尤其是当多个文件互相调用的时候，这种方法可以方便打开文件找到错误的位置。

2. 为了实现语法加亮和自动完成功能，我们必须下载两个文件，下载地址为 <http://www.editplus.com/files/pythonfiles.zip>，解压到 editplus 的安装目录下。压缩包有三个文件，此处我们用到两个，python.acp 文件和 python_extd.stx 文件。Acp 文件是自动完成文件，stx 文件是语法加亮文件。怎样加载呢？方法如下：

tools->preferences->Setting & syntax 下，首先新建一个文件类型，我们命名为 python，扩展名（file extensions）栏里我们填入 py。在 syntax file 栏里我们找到刚才下载的 python_extd.stx 文件，在 Auto completion 栏里找到 python.acp 文件，在 Function Pattern 栏里填入：

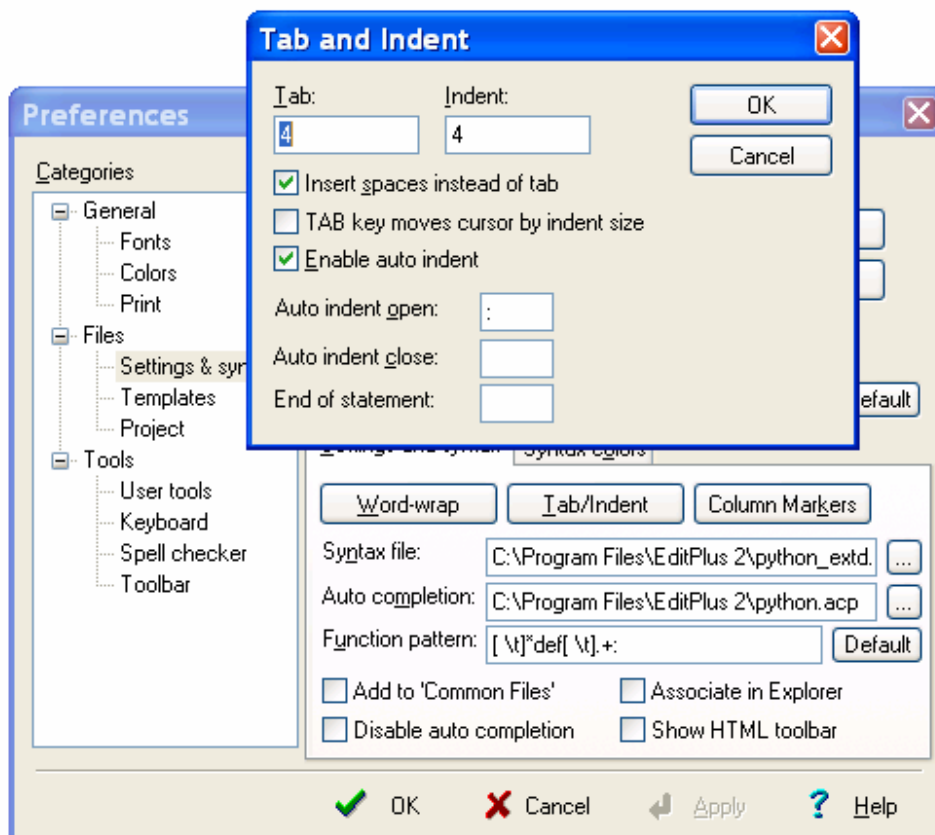
```
[ \t]*def[ \t].+:
```

这样我们就可以通过 Ctrl+F11 快捷键直接显示出 python 文件中的函数列表。设置完成如下图（4）：



(图 4)

这时我们打开一个 `python` 文件，就可以看到有语法加亮了。但是我们在编辑的时候会发现一个问题就是，在输入冒号后，没有自动缩进，当然，如果我们手工缩进后会发现后面的输入也进行了同样的缩进，怎样让 `editplus` 自动识别冒号然后直接缩进，并且如何修改缩进的长度呢？在图（4）中，我们可以看到一个 `Tab/Indent` 标签，打开他可以看到 `Tab`，`Indent` 的选项，分别填入自己习惯的距离。我两个都填入 4，并且选中 `Insert spaces instead of tab` 和 `Enable auto indent`。然后在 `Auto Indent open` 里填入一个冒号“`:`”。完成设置以后如下图（5）：



(图 5)

其他的设置也可以根据类似的设置进行。

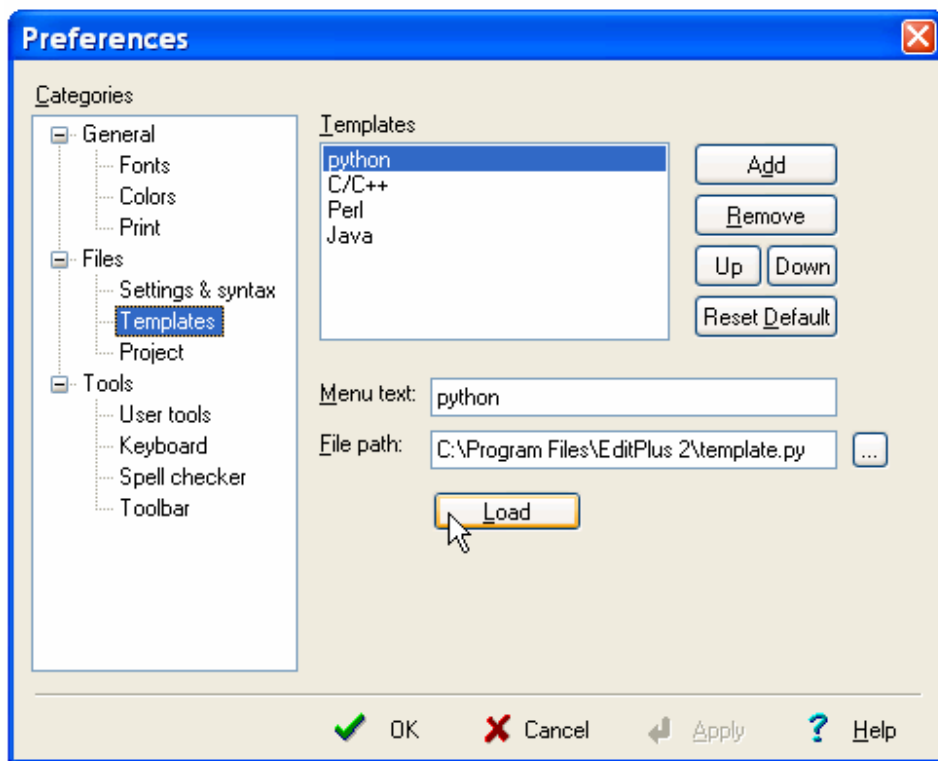
3. 还有一点很方便的是，可以通过建立一个 python 文件的模板，来快速新建一个 python 文件。首先我们建立一个名为 template.py 的文件，内容自定，我们这里假设如下：

```
-----
"""
Usage:
"""
-----
```

保存，将 template.py 文件放到 editplus 安装目录下。

Tools->preferences->templates

打开模板设置对话框，首先添加一个名为 python 的模板，把 file path 设置到我们刚才建立的 template.py 文件上。配置结果如图（6）：



图（6）

点击 load 按钮。完成。这样在 file->New->下就会有一个 python 模板，直接可以方便的建立一个 python 文件。

至此，Editplus 已经可以方便的编辑，浏览，调试和运行 python 程序了。最后还要提醒一点的是，个人使用习惯不同，可以根据类似的方法进行修改和调整。通过以上的介绍，我们可以看出，Editplus 可以很方便的配置，能够完成 python 常用的大部分功能，有效的提高编程效率。