可以设置断点，单步调试

其中第一个按钮是进行调试按钮，点击或者按Ctrl+F5就进入调式，程序到达你设置的第一个断点，这步是调式所必需的；第二个是单步调式按钮，点击或者按Ctrl+F10就可以在设置的断点之后单步调式；第三个按钮可以进入到光标所在句子中含有的函数体内部，或者按快捷键Ctrl+F11；第四个是从此函数中跳出；第五个是跳转到下一个断点；最后一个按钮是停止调试。

Spyder的功能比较多，这里仅介绍一些常用的功能和技巧：

默认配置下，“Variable explorer”窗格中不显示以大写字母开头的变量，可以单击工具栏中的配置按钮(最后一个按钮)，在菜单中取消“Exclude capitalized references”的选中状态。

在控制台中，可以按Tab按键进行自动补全。在变量名之后输入“?”，可以在“Object inspector”窗格中查看对象的说明文档。此窗格的Options菜单中的“Show source”选项可以开启显示函数的源程序。

可以通过“Working directory”工具栏修改工作路径，用户程序运行时，将以此工作路径作为当前路径。例如我们只需要修改工作路径，就可以用同一个程序处理不同文件夹下的数据文件。

在程序编辑窗口中按住Ctrl键，并单击变量名、函数名、类名或模块名，可以快速跳转到定义位置。如果是在别的程序文件中定义的，将打开此文件。在学习一个新模块的用法时，我们经常需要查看模块中的某个函数或类是如何实现的，使用此功能可以帮助我们快速查看和分析各个模块的源程序。例如下面的程序从不同的扩展库载入了一些模块和类。用Spyder打开此文件，按住Ctrl键，并单击signal、pl、HasTraits、Instance、View、Item、lfilter、plot、title等，将打开定义它们的程序文件，并跳转到相应的行。

常用快捷键：

　　Ctrl + L: 跳转到行号

　　Tab/Shift + Tab: 代码缩进/反缩进

　　Ctrl ＋I：显示帮助

Ctrl + 1: 注释/反注释

块注释/块反注释 Ctrl + 4/5

断点设置 F12

关闭所有 Ctrl + Shift + W

代码完成 Ctrl +空格键

条件断点 SHIFT + F12

配置 F6

向下复制 Ctrl + Alt +向下

剪切 Ctrl + X

调试 Ctrl + F5键

删除 Del

向上复制 Ctrl + Alt +向上

前光标位置 Ctrl + Alt +左

重新运行一个脚本 Ctrl + F6

运行           F5

运行选择 F9

保存所有  Ctrl + Alt + S

保存为     Ctrl + Alt + S

保存文件  Ctrl + S

全选        Ctrl + A

注释/反注释  Ctrl + 1

运行步差    F10

运行分析    F8

清除变量：reset 再输入 y

*#%%* (standard cell separator)

Shortcuts for useful functions

F5 executes the current file

F9 executes the currently highlighted chunk of code: this is very useful to update definitions of functions (say) in the console session without having to run the whole file again. If nothing is selected F9 executes the current line.

Tab auto-completes commands, function names, variable names, methods in the Console (both Python and IPython) and in the Editor. This feature is very useful, and should be used routinely. Do try it now if auto-completion is new to you. Assume you have defined a variable:

mylongvariablename = 42

Suppose we need to write code that computes mylongvariablename + 100, we can simply type my and then press the Tab key. The full variable name will be completed and inserted at the cursor position if the name is unique, and then we can carry on and type + 100. If the name is not uniquely identifiable given the letters my, a list field will be displayed from which the right variable can be chosen. Choosing from the list can be done with the <Arrow up> key and <Arrow down> key and the Enter key to select, or by typing more letters of the name in question (the selection will update automatically) and confirming by pressing Enter when the right name is identified.

Ctrl+Enter executes the current cell (menu entry Run > Run cell). A cell is defined as the code between two lines which start with the agreed tag #%%.

Shift+Enter executes the current cell and advances the cursor to the next cell (menu entry Run > Run cell and advance).

Cells are useful to execute a large file/code segment in smaller units. (It is a little bit like a cell in an IPython notebook, in that chunks of code can be run independently.)

Alt+<Up Arrow> moves the current line up. If multiple lines are highlighted, they are moved up together. Alt+<Down arrow> works correspondingly moving line(s) down.

Ctrl+Left Mouse Click on a function/method in the source, opens a new editor windows showing the definition of that function.

Shift+Ctrl+Alt+M maximizes the current window (or changes the size back to normal if pressed in a maximized window)

Ctrl+Shift+F activates the search pane across all files.

Cmd + + (On MacOS X) or Ctrl + + (otherwise) will increase the font size in the Editor, whereas Cmd + - (Ctrl + -) will decrease it. Also works in the IPython Console.

The font size for the Help, the Python console etc. can be set individually via Preferences > Help etc.

I couldn’t find a way of changing the font size in the variable explorer.

Cmd+S (on MacOS X) and Ctrl+S (otherwise) in the Editor pane saves the file currently being edited. This also forces various warning triangles in the left column of the Editor to be updated (otherwise they update every 2 to 3 seconds by default).

Cmd+S (on MacOS X) and Ctrl+S (otherwise) in the IPython console pane saves the current IPython session as an HTML file, including any figures that may be displayed inline. This is useful as a quick way of recording what has been done in a session.

(It is not possible to load this saved record back into the session - if you need functionality like this, look for the IPython Notebook.)

Cmd+I (on Mac OS X) and Ctrl+I (otherwise) when pressed while the cursor is on an object, opens documentation for that object in the help pane.