CLASS 10: INTRODUCTION TO PROGRAMMING IN PYTHON

ENGR 102 – Introduction to Engineering

2

Programming

What is Programming?

- **Programming**
  - The implementation of ***algorithms*** in a particular computer ***programming language*** for execution on a ***computer***

- **Algorithm**
  - A step-by-step procedure for performing a computation, solving a problem, performing some action, etc. – a recipe
  - ***Algorithm design*** is the meat of programming – the rest is just translation into a particular language
- **Programming language**
  - We'll use Python. Others include C, C++, Java, MATLAB, etc.
- **Computer**
  - May be a PC, or may be a microcontroller, FPGA, etc.

Why Programming?

- I don't want to be a *software* engineer. Why do I need to learn to program?
  - ***All*** engineers will need to write computer code throughout their careers
    - Design and simulation
    - Numerical solution of mathematical problems
    - Data analysis – from measurements or simulation
    - Firmware for the control of mechatronic systems
  - More importantly: ***development of algorithmic thinking ability***
    - Learn to think like an engineer – single most important takeaway from your engineering education

Python

- This a course in ***programming fundamentals*** and ***algorithmic thinking***
- The language we'll use to develop these concepts is ***Python*** (in the ***Spyder*** development environment)
  - Could just as well use another language, e.g., C, C++, Java, MATLAB, Fortran, …
  - The important concepts are not language-specific
- ***Two goals*** of this course:
  - Learn to develop basic algorithms and to write structured computer code
  - Learn to use Python

Introduction to Python & Spyder

The remainder of this section of notes is intended to provide a brief introduction to Python and the Spyder development environment.

Python – What is It?

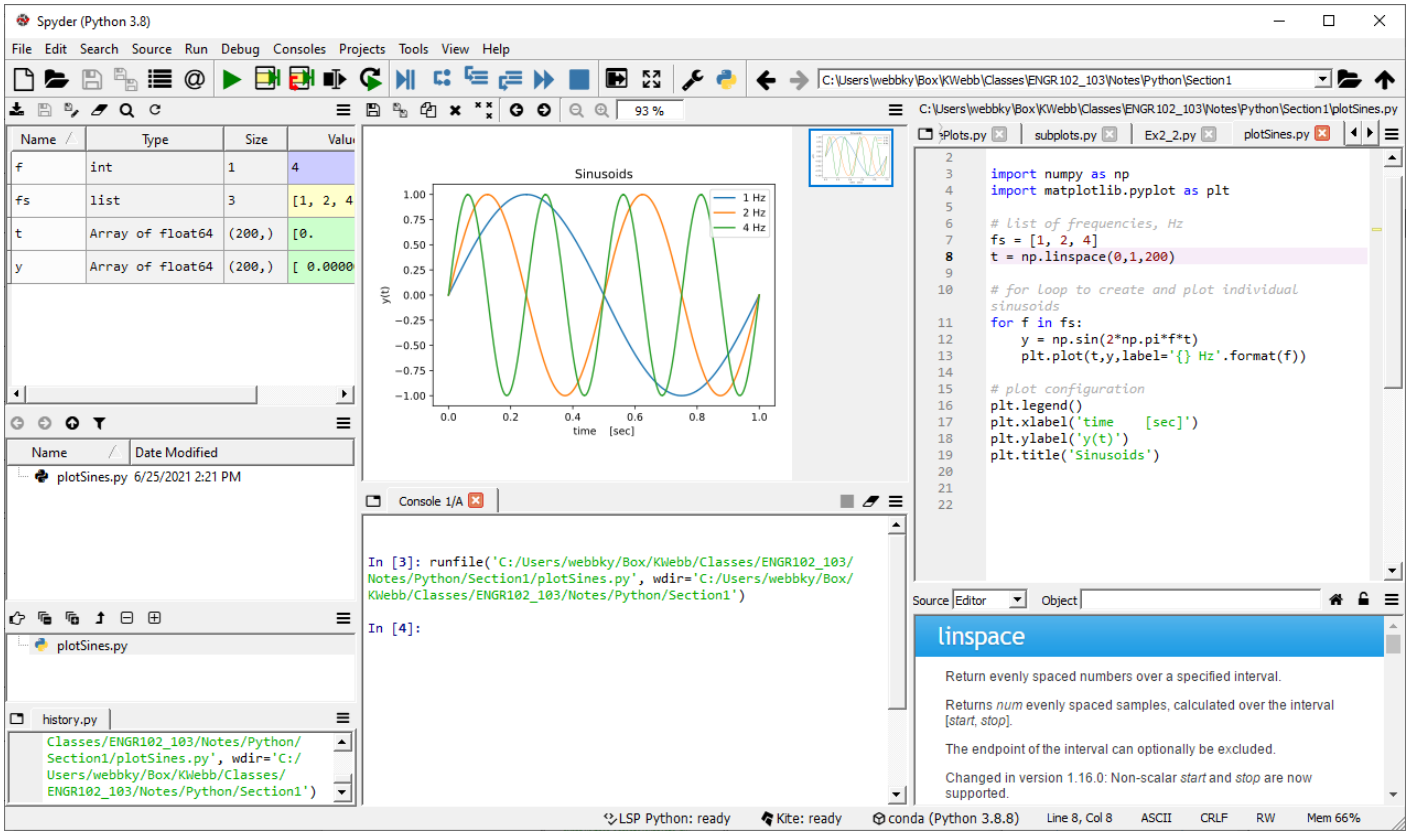
- A ***general-purpose programming language***
  - Used for writing programs to describe procedures to be executed by computers
  - *High-level*
    - Readable code – includes natural-language constructs
    - Makes use of extensive libraries of functions
    - Highly abstracted from the machine-level instructions that will ultimately be passed to the computer
  - *Interpreted*
    - Translation to machine instructions happens at runtime
    - Not *compiled* – translations happens once, creating a separate executable file
  - *Object oriented* – more on this later

Python – How Do We Use it?

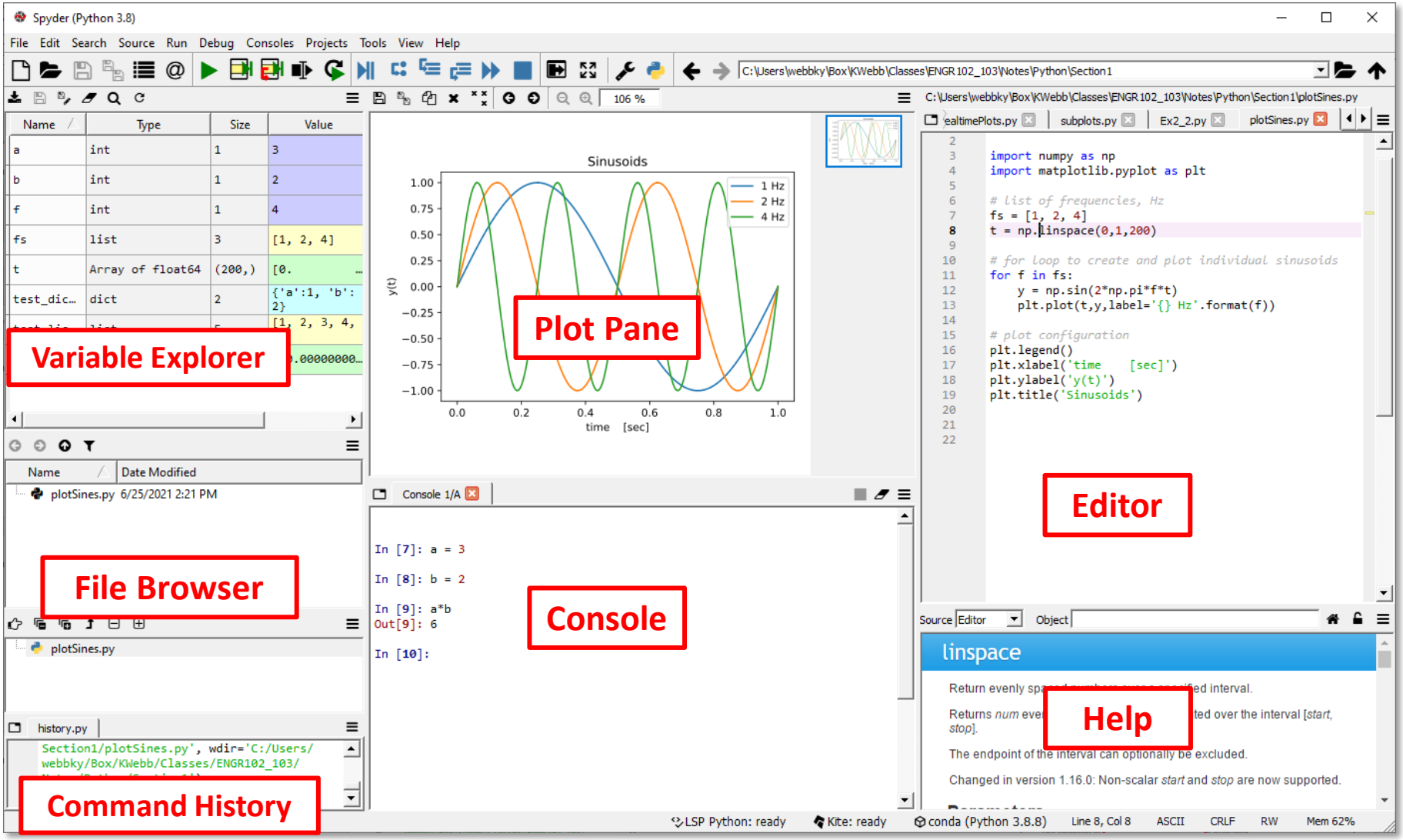
- Different ways to write and execute Python code
  - ***Text editor***
    - Simple editor for writing code
    - May include language specific formatting/coloring, etc.
    - E.g. Vi/Vim, Sublime Text, etc.
  - ***Integrated development environment (IDE)***
    - Software interface to facilitate code development
      - Code editor
      - Debugger
      - Console
      - Variable explorer
      - File browser,
      - Plotting support, etc.
    - E.g. Spyder, Pycharm, IDLE, Visual Studio, etc.

Spyder – What is It?

- We will use the ***Spyder*** IDE
  - **S**cientific **PY**thon **D**evelopment **E**nvi**R**onment
  - Designed for scientific, engineering, and data science applications



The Spyder Interface

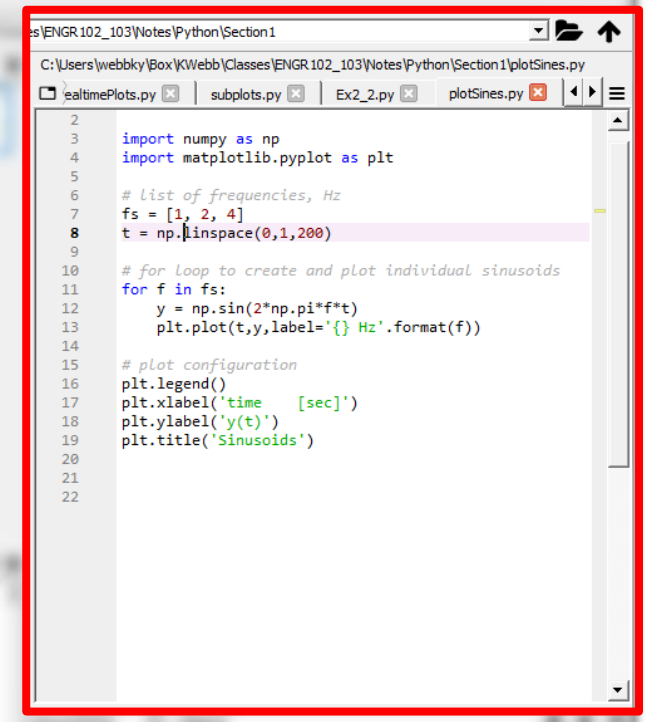


The Spyder Interface - Console

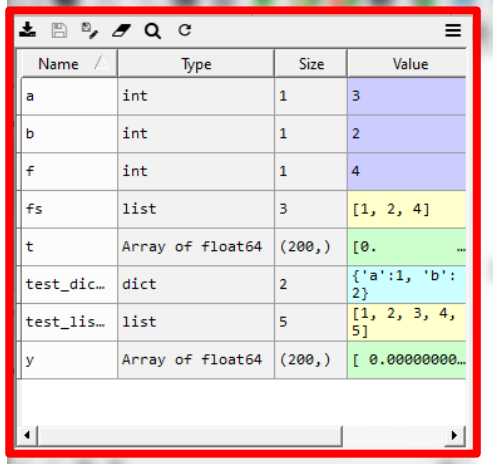
- Run Python commands interactively
- Behaves like a ***calculator***
- Useful for:
  - Quick calculations
  - Simple debugging tasks

The Spyder Interface - Editor

- Editor for Python scripts
- Write and execute our Python code here
- Auto formatting
  - Highlighting
  - Indenting
  - Code completion
- Built-in debugger
  - Set breakpoints
  - Step through code line-by-line or by section

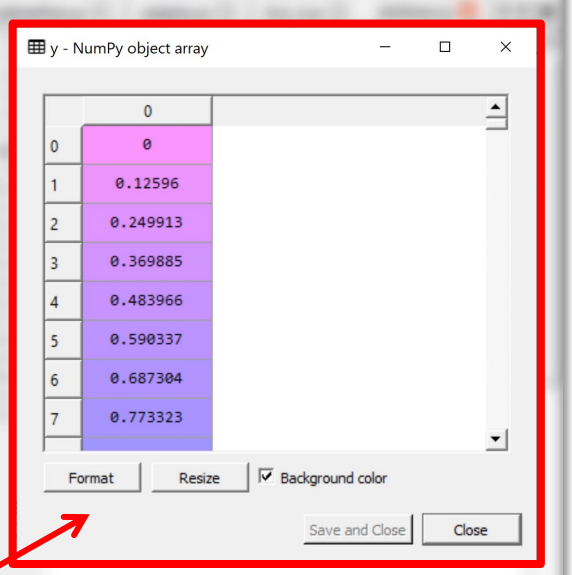


The Spyder Interface – Variable Explorer

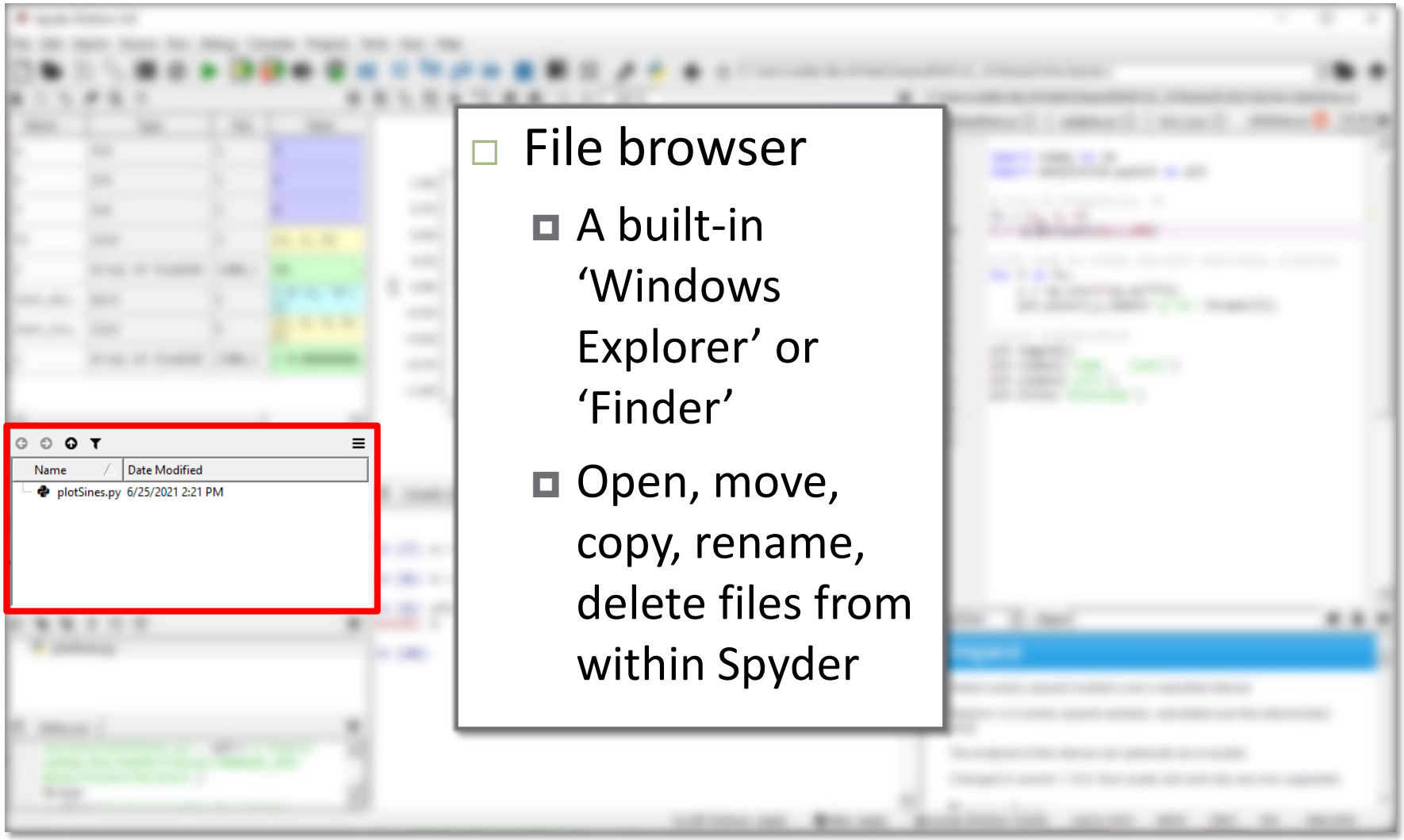


Name	Type	Size	Value
a	int	1	3
b	int	1	2
f	int	1	4
fs	list	3	[1, 2, 4]
t	Array of float64	(200,)	[0. …
test_dic…	dict	2	{'a':1, 'b': 2}
test_lis…	list	5	[1, 2, 3, 4, 5]
y	Array of float64	(200,)	[0.00000000…

- Lists all variables currently stored in memory
  - Values for scalars and small arrays
  - Size and data types for larger arrays
- Double-click a variable to open in a separate window

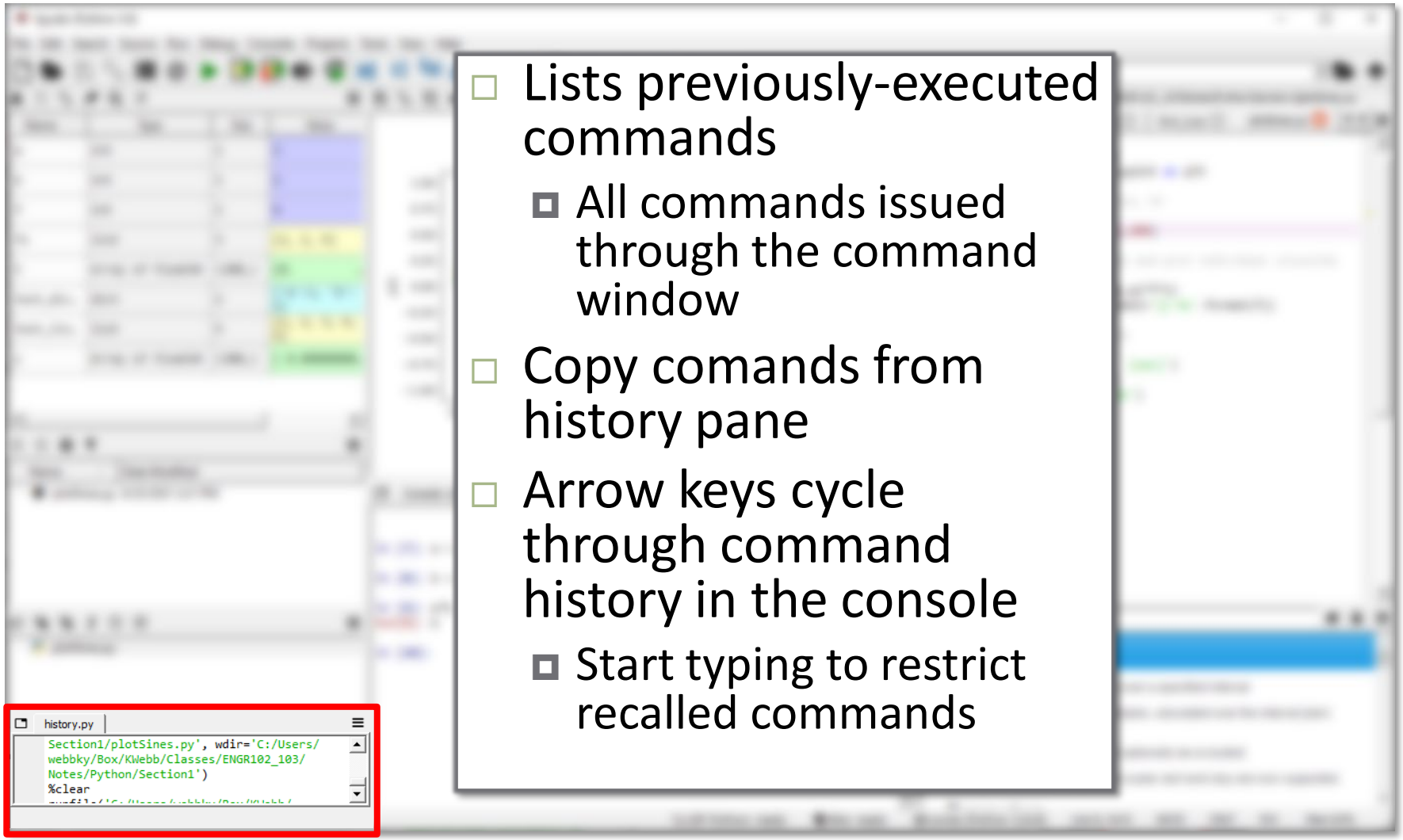


The Spyder Interface – File Browser



- File browser
  - A built-in 'Windows Explorer' or 'Finder'
  - Open, move, copy, rename, delete files from within Spyder

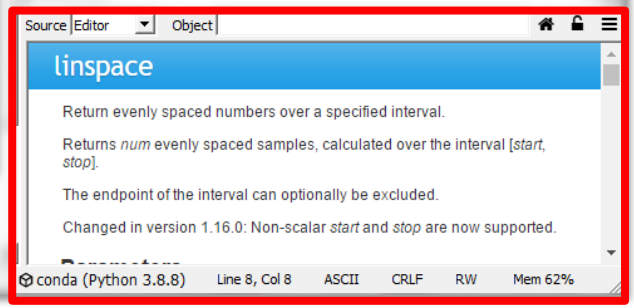
The Spyder Interface – Command History



- Lists previously-executed commands
  - All commands issued through the command window
- Copy comands from history pane
- Arrow keys cycle through command history in the console
  - Start typing to restrict recalled commands

The Spyder Interface – Help Pane

- Display help documentation for modules classes functions and methods
  - Enter object name directly in the 'Object' field
  - Place cursor on object in the editor window and type Ctrl-i

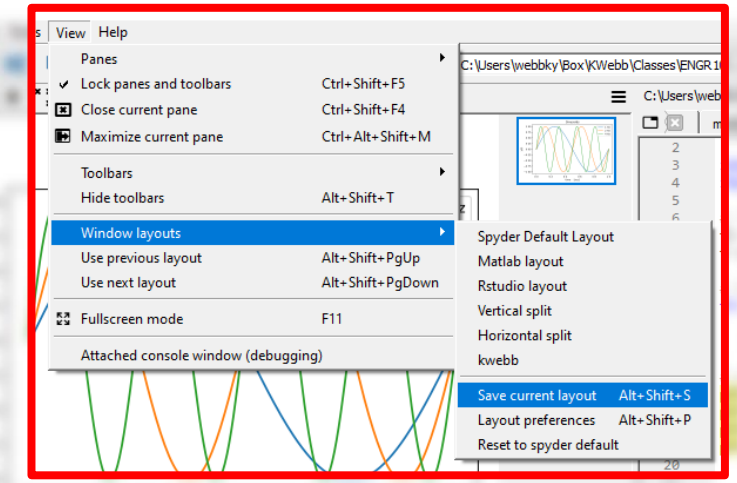


The Spyder Interface – Plots Pane

- Plots displayed here
- Docked to the Spyder desktop
- Plot history thumbnails displayed to right

- Two ways to display plots:
  - Plot pane
    - no cursors or zooming
  - Interactive window
    - Cursors and zooming, but can't dock

The Spyder Interface – Saving Layouts



- Configure the panes in your Spyder desktop to suit your workflow
- Save one or more layouts to suit your preferences