

INTRODUCTION TO PYTHON WITH PANDAS

Chapter 1: Getting Started

Python Programming

1. Getting Started

2. Variables

3. Program Flow Control

4. Lists, Tuples, and Dictionaries

5. Functions and Lambdas

6. Iterators

7. Exception Handling

8. File Handling

9. Python Classes

10. Numpy

11. Pandas

12. Matplotlib

Chapter Concepts

Using ipython

Very Simple I/O

Creating and Executing Programs

Debugging

Chapter Summary

Class Exercise 1.1: Introduction to `ipython`

- Please turn to Exercise 1.1 in the Exercise Manual
- For this exercise, you and your instructor will work through a series of steps which will give you a working knowledge of `ipython`

Chapter Concepts

Using ipython

Very Simple I/O

Creating and Executing Programs

Debugging

Chapter Summary

Writing to Standard Out

- Syntax:

```
print( [[expression] [, expression]...])
```

- Note all `expressions` are converted to strings (a sequence of characters)

- Example:

```
print("hello!")
```

- The expression is a string
- A string is a sequence of characters included in single or double quotes

Writing to Standard Out (continued)

- Separator between objects
 - Default separator is a single space
 - Change the separator with `sep` argument

```
print( [[expression][, expression]...], sep='string')
```

- Example:

```
print("Cost", "Title", "Author", sep=', ')
```

- End of line terminator
 - Default end of line terminator is a newline character (`'\n'`)
 - Change the end of line terminator with an `end` argument

```
print( [[expression][, expression]...], sep='string',  
end='string')
```

- Example:

```
print("Cost", "Title", "Author", sep=', ', end='!')
```

Writing to Standard Error

- Syntax:

```
print( [[expression][, expression]...], file=sys.stderr)
```

- Note all `expressions` are converted to strings (a sequence of characters)
- *Note:* `sys` must be imported (`import sys`)

- Later will see how to write to any file

Reading from `stdin`

- Syntax:

```
input(<prompt_string>)
```

- Notes:

- Returns only string
- Returns all characters up to but not including new line (`'\n'`) which is discarded

- Example:

```
[In [18]: name = input("what is your name? ")  
what is your name? haha  
  
[In [19]: name  
Out[19]: 'haha'
```

Exercise 1.2: Simple I/O

- Please turn to Exercise 1.2 in the Exercise Manual or open the Notebook

Chapter Concepts

Using ipython

Very Simple I/O

Creating and Executing Programs

Debugging

Chapter Summary

Exercise 1.3: Editing with `ipython`

Please turn to Exercise 1.3 in Exercise Manual and work through this with the instructor

Chapter Concepts

Using ipython

Very Simple I/O

Creating and Executing Programs

Debugging

Chapter Summary

Exercise 1.4:

Debugging with `ipython`

Please turn to Exercise 1.4 in Exercise Manual and work through this with the instructor

This is a very, very gentle introduction to debugging

- As the programs become more complex debugging will be revisited

Exercise 1.5: Chapter Exercises

Please turn to Exercise 1.5 in Exercise Manual or open the Notebook

Please let the instructor know when you have finished

Chapter Concepts

Using ipython

Very Simple I/O

Creating and Executing Programs

Debugging

Chapter Summary

Chapter Summary

- o `input`
 - `prompt string`
- o `print`
 - `sep, end`
- o `ipython`
- o `%run -d`
- o `%lsmagic`
- o `help(<object>), ?/??, dir`
- o `#!`
- o `math, sys`
- o `string, docstring`