

ME400_Lecture_1_Slides

August 17, 2017

1 ME 400 Lecture 1 - Getting Started

First, a brief overview of the [Syllabus](#)!

```
In [1]: from IPython.core.interactiveshell import InteractiveShell
        InteractiveShell.ast_node_interactivity = "all"
```

1.1 Contact Information

- **Lecture** MW 1:30--2:20pm, DUE 1073
- **Instructor:** Jeremy Roberts
- *Phone:* 785-532-7182
- *Email:* jaroberts@ksu.edu
- *Office:* Ward 137D
- *Office Hours:* M/W, 2:30-4:00pm (or by appointment; refer to my [calendar](#) before suggesting a time)
- **GTAs:** Will Asher and John Boyington
- **UTAs:** Cole Hoing, Keith Huddleston, and Sam Oxandale

1.2 Textbook

No book required

Online notes: robertsj.github.io/me400_notes

Other good resources: - [Object-Oriented Programming in Python](#) - Downey, Allen, Jeffrey Elkner, and Chris Meyers. *How to Think Like a Computer Scientist: Learning with Python*. Green Tea Press, 2002. ISBN: 9780971677500.

(This book is available for [free as a PDF](#), or you may purchase a paper copy.) - Shapiro, Bruce. *Scientific Computation: Python Hacking for Math Junkies*. Sherwood Forest Books, 2015. ISBN: 9780692366936. - Stewert, John. *Python for Scientists*. Cambridge University Press. ISBN: 9781107686427. - www.python.org - www.scipy.org - www.scipy.org

1.3 Amount of Study Required

6 to 9 hours per week of active study outside class for the average student to earn the average grade.

1.4 Lectures

Your job is to **come to lecture prepared** to learn and to engage in any in-class activities.

Every lecture has a **pre-lecture activity** (almost always a Canvas quiz) due one hour before lecture.

No phones, but laptops okay if used for examples.

1.5 Assessment

- Pre-Lecture Activities (10%)
- Laboratory Exercises (10%)
- Homework Exercises (20%)
- Examinations (60%)

1.6 Course Schedule

Lecture	Date	Topics
1	Monday, August 21	Administrivia. Overview of Python. Spyder and Jupyter. Basic types.
2	Wednesday, August 23	Basic operators and common, built-in functions.
	Friday, August 25	LAB 1
3	Monday, August 28	Basic data processing with NumPy and Matplotlib.
4	Wednesday, August 30	More on NumPy arrays. Slicing operations. np.linalg
	Friday, September 1	LAB 2

Thereafter: - lectures 5-11 on elementary programming concepts (e.g., algorithms, if statements, for loops, and container types) - lectures 12-17 on modular programming (e.g., function definitions and modules) - lectures 18-29 on applications (e.g., symbolic computing, optimization, curve fitting)

1.7 Overview, Objectives, and Key Terms

In this lesson, we'll walk through the installation of, access to, and use of Python.

1.7.1 Objectives

By the end of this lesson*, you should be able to

- *access Python on departmental machines (or your own machine)*
- *define variables using basic Python types*
- *explain what a variable is*

(*Lecture slides and other content used in class may (1) include only parts of the required, online reading and (2) provide materials beyond the reading. In this class, "lesson" refers to the coupled reading/lecture pair.)

1.7.2 Key Terms

- interpreted programming language
- compiled programming language
- integrated development environment
- Spyder
- Jupyter notebook
- variable
- int
- float
- str
- bool
- assignment

1.8 Overview of Python

Python is a scripting language, which means it is [interpreted](#) and not [compiled](#).

Interpreted languages are converted from human-readable text files to binary (i.e., the 1's and 0's a computer understands) on-the-fly.

A compiled language is converted to binary once-and-for all by an explicit compilation step.

Several versions of Python are in current use, but the most popular versions are 2.7 and 3.6. Despite the numbering, both versions are current (but [not forever](#)).

In this class, **we will work exclusively with Python 3.6**.

1.9 Accessing Python

For folks who want to install Python on their own machines, the Conda folks have provided an [installation guide for all operating systems](#).

Conda is installed on MNE machines---see the online notes to get an early start, or wait until Lab 1.

1.9.1 Interactive Python via IDLE

In the CMD window (or, on OS X, the terminal), type "python" and press enter to open Python.

1.9.2 Spyder

The recommended way to use Python in ME 400 is Spyder, an [integrated development environment](#).

1.9.3 Jupyter Notebook

Another way to interact with Python is through use of [Jupyter notebooks](#). Jupyter allows one to produce documents within a web browser that contain text, images, links, and, of course, Python code. In fact, these online materials were all produced using the notebook format and converted to the form you are now seeing!

1.10 Basic Types

Python uses **variables**, which consist of a *name* to which a *value* can be *assigned*.

```
In [2]: x = 1
```

```
In [3]: y = 1.0
```

```
In [4]: z = '1.0'
```

Are these all the same?

```
In [5]: x
        y
        z
```

```
Out[5]: 1
```

```
Out[5]: 1.0
```

```
Out[5]: '1.0'
```

```
In [6]: type(x)
        type(y)
        type(z)
```

```
Out[6]: int
```

```
Out[6]: float
```

```
Out[6]: str
```

Note: str values can be defined with ' ' or " " but not ' " or " '.

One last type:

```
In [7]: b = True
```

Warning: The value of a bool is either True or False and the *capitalization* is *required*

We can change a value of one type to another.

```
In [8]: z = float(z)
        z
        type(z)
```

```
Out[8]: 1.0
```

```
Out[8]: float
```

```
In [9]: bool(1)
        bool('a')
        bool(0.0)
        bool('')
        bool(None)
```

```
Out[9]: True
```

```
Out[9]: True
```

```
Out[9]: False
```

```
Out[9]: False
```

```
Out[9]: False
```

For conversion to `bool`, 0 and its cousins like 0.0, the empty string `''`, and the special `None` value all evaluate to `False`.

Throughout, use of `=` is as the *assignment operator*, which gives *values* to *names* (i.e., defines **variables**).

1.11 Recap

By this time, you should be able to

- *access Python on departmental machines* (or your own machine)
- *define variables using basic Python types*
- *explain what a variable is*

Today's pre-lecture quiz is special and can be completed by 11:59 pm tonight. Be sure to do all activities by 12:30 pm each lecture day.

Questions?