

Crash Course in Python

Devansh Agarwal

da0017@mix.wvu.edu

devanshkv.github.io

In today's class

1. Using the Python: Interactive Shell and scripts
2. Hello world in python
3. Lists and loops
4. Numpy: Dealing with numbers in python
5. Matplotlib: Plotting with Python

Python: interactive Shell

- To enter the python interactive shell:
 - Open a Terminal
 - Type `python`
- Fairly straightforward, Try the following:
 - Printing Strings (NOTE: Strings are always in quotes)
 - `print("hello world")`
 - Defining and Printing a variable
 - `a=10`
 - `print(a)`
 - `print(a+100)`
- To exit
 - Type `exit()`

Python: Scripts

- Use your favorite text editor (gedit, vim, emacs, nano).
 - Hello World
 - `emacs hello_world.py`
 - `print("hello world")`
 - Save the file
 - `python hello_world.py`
 - Printing a Number
 - `emacs a_is_10.py`
 - `a=10`
 - `print(a)`
 - Save the file
 - `python a_is_10.py`

Python: Lists

- Lists are collections of objects
 - `emacs python_lists.py`
 - `my_list = ["apple", "mango", "orange"]`
 - `print(my_list)`
 - Save the file
 - `python python_lists.py`
- Lists can contain anything!
 - `emacs python_lists.py`
 - `my_number_list = [1,2,3,4,5,6,7]`
 - `print(my_number_list)`
 - Save the file
 - `python python_lists.py`

The for loop

- If you want to do something recursively, for loop is for you!
 - emacs python_lists.py
 - my_number_list = [1,2,3,4,5,6,7]
 - for number in my_number_list:
 - print(number)
 - Save the file
 - python python_lists.py
- Exercise: Use the `my_number_list` (without changing the list) and the for loop to print `[2,4,6,8,10,12,14]`

Numerical Computation in python: numpy

- If you want to do something recursively, for loop is for you!
 - `emacs python_numpy.py`
 - `import numpy as np`
 - `my_numpy_array = np.array([1,2,3,4,5,6,7])`
 - `print(my_numpy_array)`
 - Save the file
 - `python python_lists.py`
- Exercise: Use the `my_number_list` (without changing the list) and the for loop to print `[2,4,6,8,10,12,14]`
 - `print(my_numpy_array*2)`
- Numpy has tonnes of functions: `np.sin()`, `np.sqrt()`, `np.power()`, `np.fft`, `np.random ...`

Numerical Computation in python: numpy

- Useful numpy functions:
 - `np.linspace(start=0, stop=1, num=5)`
 - `>>> array([0. , 0.25, 0.5 , 0.75, 1.])`
 - `np.arange(3)`
 - `>>> array([0, 1, 2])`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`

Plotting in python: Matplotlib

- Simple Plotter:
 - `import numpy as np`
 - `import matplotlib.pyplot as plt`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`
 - `plt.plot(x,y)`
 - `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:

- `import numpy as np`
- `import matplotlib.pyplot as plt`
- `x = np.linspace(start=-1, stop=1, num=50)`
- `y = x**2`
- `z = np.sqrt(x)`
- `plt.plot(x,y)`
- `plt.plot(x,z)`
- `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:

- `import numpy as np`
- `import matplotlib.pyplot as plt`
- `x = np.linspace(start=-1, stop=1, num=50)`
- `y = x**2`
- `z = np.sqrt(x)`
- `plt.plot(x,y, 'r--')`
- `plt.plot(x,z, 'k.-')`
- `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:
 - `import numpy as np`
 - `import matplotlib.pyplot as plt`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`
 - `plt.plot(x,y, 'r--', label='y=x^2')`
 - `plt.plot(x,z, 'k.-', label='y=x^0.5')`
 - `plt.legend()`
 - `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:
 - `import numpy as np`
 - `import matplotlib.pyplot as plt`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`
 - `plt.plot(x,y, 'r--', label='y=x^2')`
 - `plt.plot(x,z, 'k.-', label='y=x^0.5')`
 - `plt.xlabel('x axis label goes here')`
 - `plt.ylabel('x axis label goes here')`
 - `plt.legend()`
 - `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:
 - `import numpy as np`
 - `import matplotlib.pyplot as plt`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`
 - `plt.plot(x,y, 'r--', label='y=x^2')`
 - `plt.plot(x,z, 'k.-', label='y=x^0.5')`
 - `plt.xlabel('x axis label goes here')`
 - `plt.ylabel('y axis label goes here')`
 - `plt.title('Title goes here')`
 - `plt.grid()`
 - `plt.legend()`
 - `plt.show()`

Plotting in python: Matplotlib

- Simple Plotter:
 - `import numpy as np`
 - `import matplotlib.pyplot as plt`
 - `x = np.linspace(start=-1, stop=1, num=50)`
 - `y = x**2`
 - `z = np.sqrt(x)`
 - `plt.plot(x,y,'r--',label='y=x^2')`
 - `plt.plot(x,z,'k.-',label='y=x^0.5')`
 - `plt.xlabel('x axis label goes here')`
 - `plt.ylabel('y axis label goes here')`
 - `plt.title('Title goes here')`
 - `plt.grid()`
 - `plt.legend()`
 - `plt.show()`

