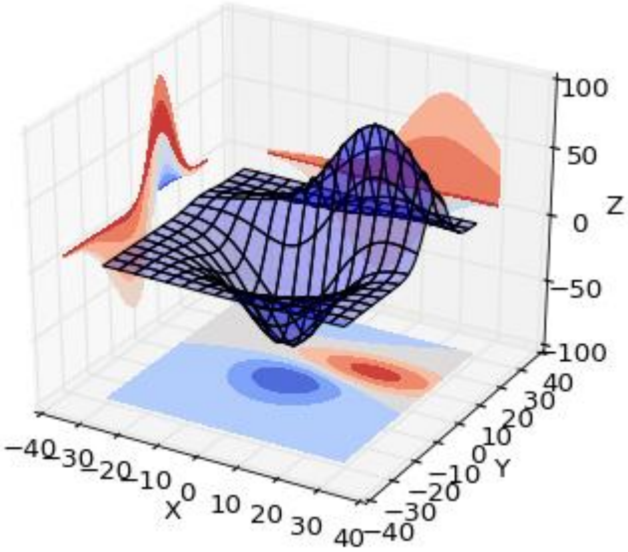
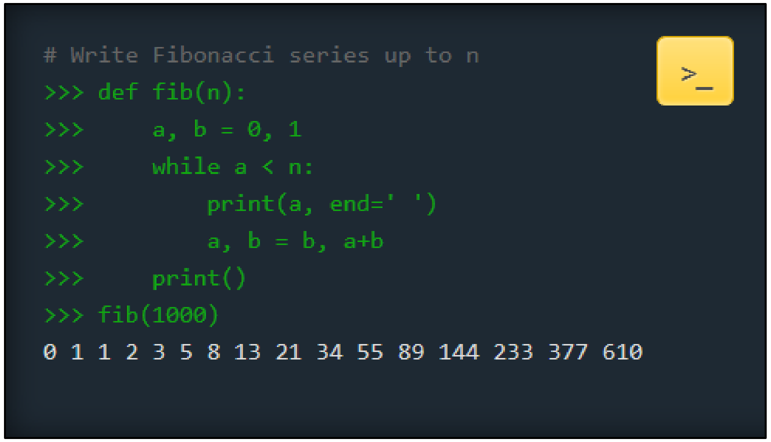
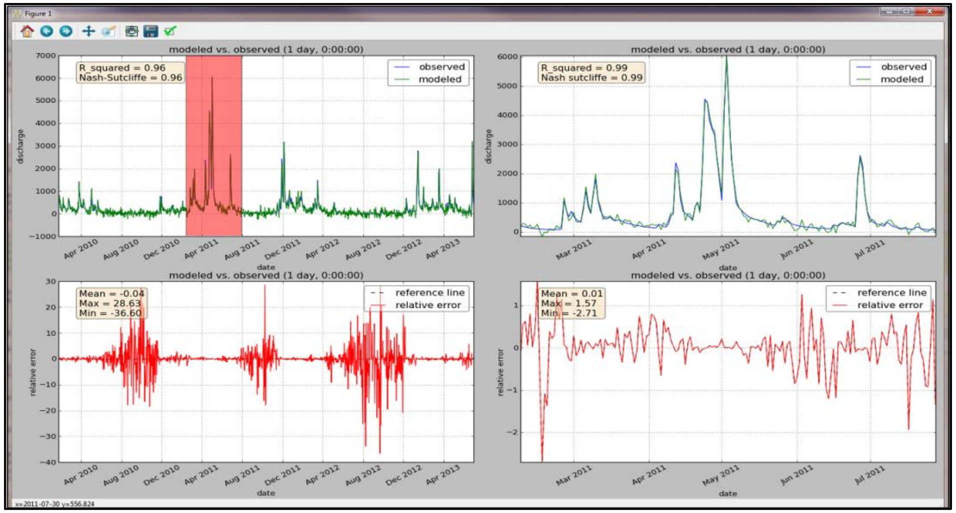
Introduction to Scientific Computing
Meeting 17
Programming with Python





```
# Write Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```



Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Last Meeting

- Learn about main built-in container called a **list**
- Learn how to loop through lists using a **for loop**

Last Meeting

- **List** is a **container** or a **collection of items**.

list = [*item0, item1, item2, …***]**

```
>>> numbers = [0, 1, 2, 3, 4, 5]
>>> names = [“Jeremiah”, “Justin”, “Dave”, “Loren”]
>>> names[2]
 “Dave”
>>> names[1:3]
[“Justin”, “Dave”]
```

Last Meeting – For Loop

- For loops are used to repeat an operation or set of operations a certain number of times.

for *target* **in** sequence:

block of operations

```
>>> numbers = [0, 1, 2, 3, 4, 5]
>>> for num in numbers:
. . .     print(num)
>>> for i in range(len(numbers)):
. . .     print(numbers[i])
```

Video – Python Basics

software carpentry

- Software Carpentry, Greg Wilson
  - Python: Lists and for loop

    http://software-carpentry.org/v4/python/lists.html

Last Meeting

- **Lists** are **mutable**
  - Example:
- **Lists** can contain **arbitrary types**
  - Example?
- **Lists** have **length**
  - Name of function?
- **Lists** have **methods** that operate on a list
  - Examples ?
- **Function** that creates a list of numbers
  - ?
- Can **test** for **item membership** in **list**
  - Example?

Practice Objectives: lists

- Create a list called *cities* that contains 5 strings of the following city names in order:
  - Louisville, London, Paris, New York, Barcelona
- Select the city “Paris” out of the list
- Select the last city, “Barcelona”, out of the list
- Slice out the cities “Louisville”, “London”, “Paris”
- Append a new city to the end of the list
- Sort the list in alphabetical order
- What is the length of the list

Practice Objectives: lists and for loop

- Write a for loop to print out each city name in the list called *cities*.

- Write a program that tests if the cities London, Detroit, Miami, Cincinnati, Paris are in the list called *cities*. If item is in *cities*, print the following:

  City <*name of city*> is already in list cities

  else add city to list *cities* and print the following:

  Added the city <*name of city*> to the list cities

Review – page 1

1. What are some features of Python's lists?

   a) Ordered sequence, arbitrary type, mutability
   b) Arbitrary sequence, fixed type, mutability
   c) Arbitrary sequence, arbitrary type, mutability
   d) Ordered sequence, arbitrary type, immutability
   e) Arbitrary sequence, fixed type, immutability

2. What function could you use to construct a list of odd numbers from 0 to 10?

Today’s Objectives

- Learn about main built-in container called a **dictionary**

Demo - dictionary

- **Dictionary** is a **container** or a **collection of items**.
  - Unordered collection of key/value pairs
  - Mapping/association between keys and values

dictionary = {*“key”: value, “key”: value***}**

```
>>> info = {“name”: “Bob”, “age”: 30, “height”: 6}
>>> info[“name”]
Bob
>>> info[“age”]
30
```

Demo - dictionary

- Unlike a **list** where you must **use an integer index** to **access items**, in a **dictionary**, you **use a key** to **access items**.

```
>>> info_list = [“Bob”, 30,  6,]
>>> info_list[0]
Bob
>>> info_list[1]
30

>>> info_dict = {“name”: “Bob”, “age”: 30, “height”:  6}
>>> info_dict[“name”]
Bob
>>> info_dict[“age”]
30
```

Demo - dictionary

- Adding items to a dictionary after it is created

```
>>> info_dict[“weight”] = 160
>>> info_dict[“hobbies”] = [“golf”, “tennis”]
```

- Test whether key is present using **in**

```
>>> “weight” in info_dict
True
>>> “birthday” in info_dict
False
```

Demo - dictionary

- Print all key/value pairs

```
>>> for key in info_dict:
. . .     print(“Key {} maps to value {}”.format(key, value))

>>> for key, value in info_dict.items():
. . .     print(“Key {} maps to value {}”.format(key, value))

>>> for key, value in info_dict.iteritems():
. . .     print(“Key {} maps to value {}”.format(key, value))
```

Demo - dictionary

- Dictionaries have **methods**
  - https://docs.python.org/2/library/stdtypes.html

Video – Python Basics

- Software Carpentry, Greg Wilson
  - Python: Dictionaries

    http://software-carpentry.org/v4/setdict/dict.html

Practice Objectives: dictionary

- Create a dictionary called *states* that has a mapping of state name to state abbreviation.

  ```
  states = {“Kentucky”: “KY”, “Indiana”: “IN”}
  ```

- Create a dictionary called *cities* that has a mapping of state abbreviation to city name.

  ```
  cities = {“KY”: “Louisville”, “IN”: “Indianapolis”}
  ```

- Add another state mapping to states and city mapping to cities.
- Print out all the key/value pairs

Next meeting

- Python – Learn about
  - **Little more with dictionaries**
  - **Strings**
  - **Input and output**