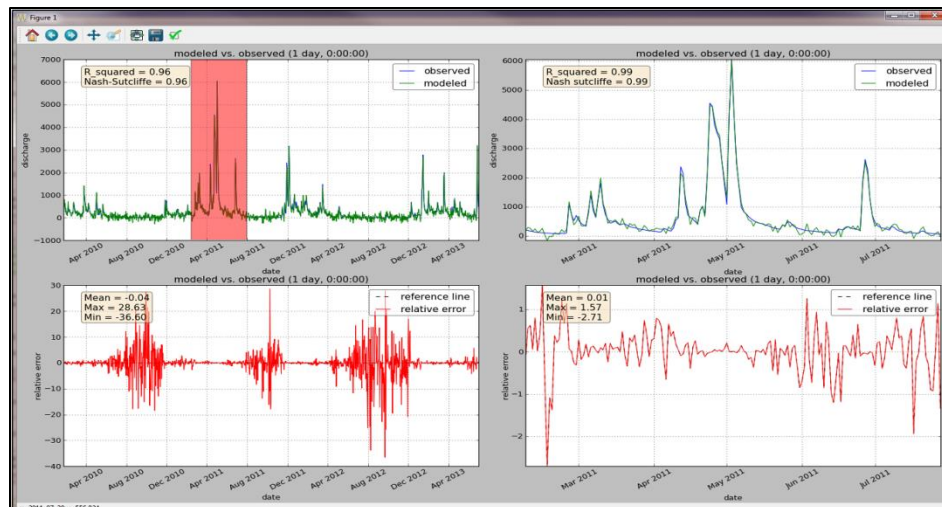
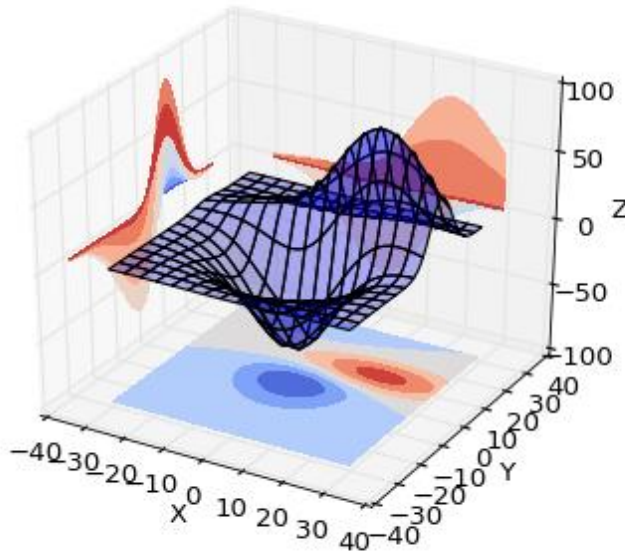


# Introduction to Scientific Computing

## Meeting 14

### 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 what Python is and why it is used
  - Python is a general purpose high level programming language.
  - Python design philosophy is on code readability.
  - Python can be used for any programming task and has libraries for just about anything
    - Scientific computing
    - GIS programming
    - GUI programming
    - Web programming
    - Game programming
    - ...
  - Python has great and extensive documentation and tutorials

# Last Meeting

- Learn some basics of Python
  - How to use Python interactively vs. writing a script

>>> print("Hello World!")

\$ python hello\_world.py

hello\_world.py

print("Hello World!")



# Last Meeting

- Creating variables through assignment; =
  - **A variable is a name for a value**
  - Python is **dynamically typed**
    - >>> x = 10
    - >>> x = "hello"
    - No need to declare a type (e.g. integer) for the variable x
    - Instead, you have variable names, and you bind them to entities whose type stays with the entity itself.
  - Python is **strongly typed**; Every type conversion must be done explicitly
    - >>> x = 10
    - >>> y = "5"
    - >>> x + y                      # is this integer 15, string "15", or string "105"?
    - TypeError

# Last Meeting

- Basic data types; **integers, floats, booleans, strings**
  - Can check data type using:

- **`type(variable)`**

```
>>> x = 2
```

```
>>> type(x)
```

```
<type 'int'>
```

```
>>> is_fun = True
```

```
>>> type(is_fun)
```

```
<type 'bool'>
```

# Last Meeting

- Learn some basics of Python

- Arithmetic operations; +, -, %, \*, \*\*, %**

```
>>> 3/2      # integer division
```

```
1
```

```
>>> 3.0/2.0  # float division
```

```
1.5
```

+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation
%	Modular division

- Augmented assignment operators; a = a + 1 is the same as a += 1**

```
>>> a = 0
```

```
>>> a = a + 1
```

```
>>> print(a)
```

```
1
```

```
>>> a = 0
```

```
>>> a += 1
```

```
>>> print(a)
```

```
1
```

a += b	a = a + b
a -= b	a = a - b
a *= b	a = a*b
a /= b	a = a/b
a **= b	a = a**b
a %= b	a = a%b

# Last Meeting

- Learn some basics of Python
  - **Comparison operators**; returns **True or False**

```
>>> 5 < 10
```

```
True
```

```
>>> 2 != 2
```

```
False
```

<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

# Today's Objectives

- Learn multiple ways to print; **print**
- Learn how to control the flow of a program using **while loop** and **if, elif, and else**



# Demo - Print

```
>>> print "hello"    # Python 2.* print is a statement
```

```
>>> print("hello")   # Python 3.* print is a function
```

```
>>> x = 2
```

```
>>> y = 3
```

```
>>> print "The answer is", x*y
```

The answer is 6

```
>>> print ("The answer is"), x*y
```

The answer is 6

# Demo - Print

- A taste of string formatting; more to come when we cover strings in depth.

```
>>> print "The answer is %s" % (x*y)
```

```
The answer is 6
```

```
>>> print("The answer is %s") % (x*y)
```

```
The answer is 6
```

```
>>> print("The answer is %f") % (x*y)
```

```
The answer is 6.000000
```

```
>>> print("The answer is %.2f") % (x*y)
```

```
The answer is 6.00
```

```
>>> print("The answer is {}".format(x*y))
```

```
The answer is 6
```

```
>>> print("The answer is {:.f}".format(x*y))
```

```
The answer is 6.000000
```

```
>>> print("The answer is {:.2f}".format(x*y))
```

```
The answer is 6.00
```

# Demo – While loop

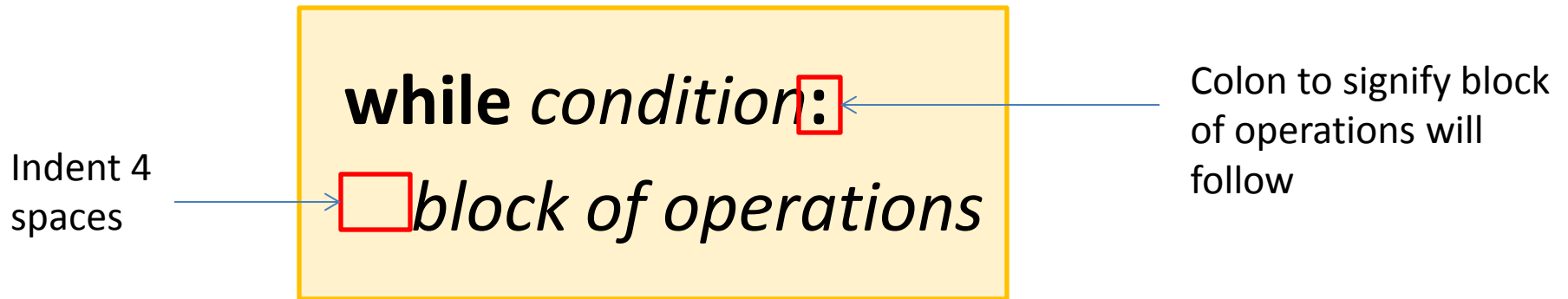
- While loops are used to repeat an operation or set of operations until a certain condition is met.

**while** *condition:*  
*block of operations*

```
>>> number = 0
>>> while number <= 5:
...     print(number)
...     number += 1
```

# Demo – While loop

- While loops are used to repeat an operation or set of operations until a certain condition is met.



```
>>> number = 0
```

```
>>> while number <= 5:
```

```
...     print(number)
```

```
...     number += 1
```

# Demo – if, elif, else

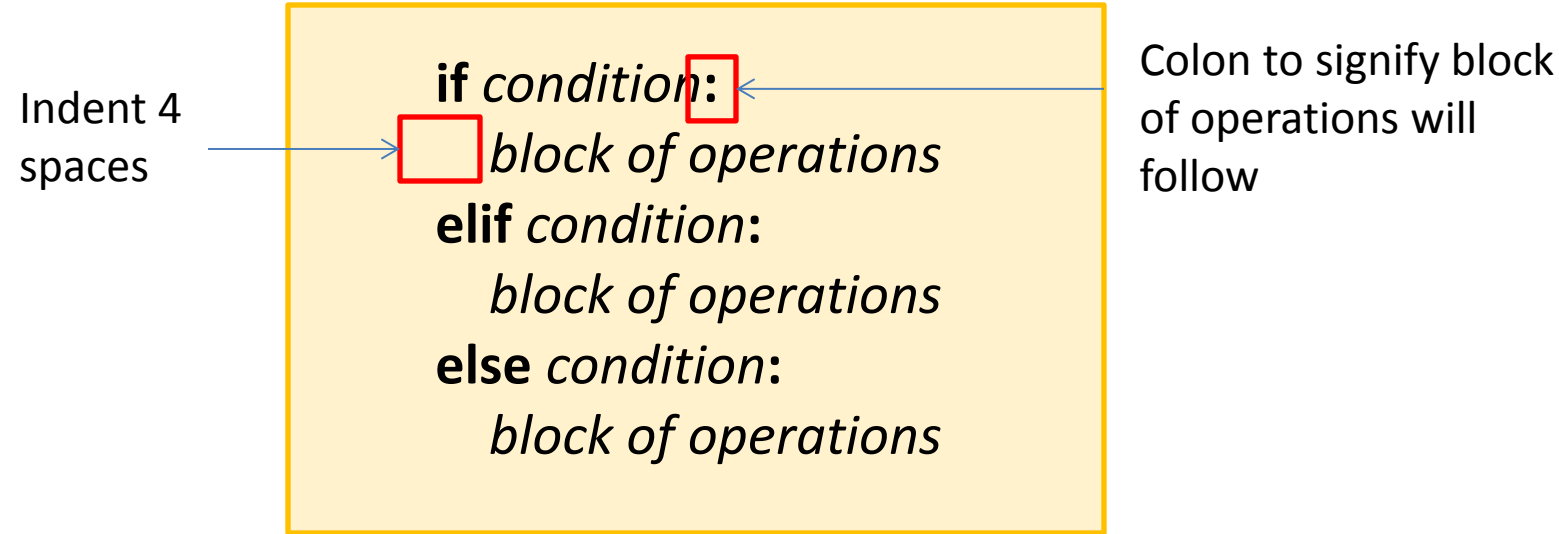
- if, elif, else are used to make decisions (choices)

```
if condition:  
    block of operations  
elif condition:  
    block of operations  
else condition:  
    block of operations
```

```
>>> number = 10  
>>> if number == 0:  
...     print("equals zero")  
... elif number < 0:  
...     print("negative")  
... else number < 0:  
...     print("positive")  
positive
```

# Demo – if, elif, else

- if, elif, else are used to make decisions (choices)



```
>>> number = 10  
>>> if number == 0:  
...     print("equals zero")  
... elif number < 0:  
...     print("negative")  
... else number > 0:  
...     print("positive")  
positive
```

# Video – Python Basics



- Software Carpentry, Greg Wilson
  - Python: Control flow

<http://software-carpentry.org/v4/python/flow.html>

# Practice Objectives - Print

- Write a print statement to print the following using variables *name* and *city*

My name is <fill with your name> and I live in <fill with city>.



- Write a print statement to print the following using a variable  $x = 2$ .

2 + 2 is 4





# Practice Objectives – While loop

- Write a while loop to print out numbers starting from 0 to 10.



- Write a while loop to print out numbers starting from 10 to -10



# Practice Objectives – While loop

- Write a while loop to print out even numbers 2 to 10.



OR



# Practice Objectives – if, elif, else

- Write the following an if, elif, else condition to make a decision about today's temperature

temperature = *<look up current temperature in your area>*

if temperature <= 32:

    print("It is freezing")

elif 33 <= temperature <= 59:

    print("It is chilly")

elif 60 <= temperature <= 80:

    print("It is comfortable")

elif 81 <= temperature <= 100:

    print("It is hot")

else

    print("It is boiling")

# Review – page 1

1. Which of the following prints the following:

The temperature in Louisville is 90 degrees

a) `city = "Louisville"`

`temp = 90`

`print("The temperature in %s is %s degrees") % (city, temp)`

b) `city = "Louisville"`

`temp = 90`

`print("The temperature in {} is {} degrees").format(city, temp)`

c) `city = "Louisville"`

`temp = 90`

`print("The temperature in %s is %.0f degrees") % (city, temp)`

d) All of the above

# Review – page 2

1. How many times does the following program print the variable count?

```
count = 0
while count <= 5:
    print(count)
    count +=1
```

2. What is the output of the following program?

```
number = 0
while number <= 20:
    if number % 2 != 0:
        print(number)
    number +=1
```

# Review – page 3

1. What is the output of the following program?

```
x = 2.5
```

```
y = 3.0
```

```
if y <= x:
```

```
    print("less than")
```

```
elif y >= x:
```

```
    print("greater than")
```

```
elif y == x:
```

```
    print("equal to")
```

```
else:
```

```
    print("unable to compare")
```

2. What is the output of the following program?

```
hungry = True
```

```
if hungry:
```

```
    print("Yes, lunch time!")
```

```
else:
```

```
    print("Not lunch time yet.")
```

# Next meeting

- Python – Learn about built-in containers (collections)
  - **lists**
  - **dictionaries**