

## Review

- What are the two ways to import a module into a program?
  - How does using the imported functions change with each type of import statement?
- How can we tell a Python program to repeat something a set number of times?

Jan 25, 2011

Sprenkle - CSCI1111

## Review: import statement

- Two ways to use `import` statement:
  1. `import <modulename>`
    - Then, need to prepend modulename to each constant or function
    - Ex: `math.ceil`, `math.pi`
  2. `from <modulename> import <defn>`
    - Can then just use function or constant
    - Ex: `ceil`, `pi`

Jan 25, 2011

Sprenkle - CSCI1111

## Review: for loop

```
for i in xrange(5):  
    # like assigning i values(0,1,2,3,4)  
    # loop body ...
```

- When have `xrange(5)`, `i` gets values (0, 1, 2, 3, 4), for each time through the loop
  - Which means that loop executes 5 times
- Loop variable name doesn't matter

Jan 25, 2011

Sprenkle - CSCI1111

## Programming Practice

- Add 5 numbers, inputted by the user
- Refining the program
  - More intuitive output
  - More general program (get different numbers of numbers to add up).

Jan 25, 2011

Sprenkle - CSCI1111

[sum5.py](#)

## Accumulator Design Pattern

1. Initialize accumulator variable
2. Loop until done
  - Update the value of the accumulator
3. Display result

Jan 25, 2011

Sprenkle - CSCI1111

## Programming Practice

- Average 5 numbers inputted by the user

Jan 25, 2011

Sprenkle - CSCI1111

[average5.py](#)

## Summing 5 Numbers from User

```
# keep track of running total
total = 0

# repeat: get user input for 5 numbers,
# keep running total
for i in xrange(1, 6):
    userNum = input("Enter number " + str(i) + ": ")
    # update running total
    total = total + userNum
    # Alternative: total += userNum
# display total
print "The total is", total
```

Note: slight  
change to our  
solution

Jan 25, 2011

Sprenkle - CSCI111

## Review: Accumulator Design Pattern

1. Initialize accumulator variable
2. Loop until done
  - Update the value of the accumulator
3. Display result

Jan 25, 2011

Sprenkle - CSCI111

## Lab 1 Feedback

- Good test cases
  - Ex: Use well-known values for F→C conversion
- Good variable naming
- Good high-level descriptions
  - I use to make sure you understand the purpose of the program

Jan 25, 2011

Sprenkle - CSCI111

## Lab 1 Feedback: Common Mistakes

- Not executing program **more than one** time if have input from user
  - Test with negative values and floats, as appropriate
- Unlabeled output
  - Tell user what is being output
- Area of triangle: not a **float**
  - Test with two odd numbers
  - Common correct solutions: /2.0 or \*.5

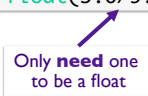
Jan 25, 2011

Sprenkle - CSCI111

## Lab 1 Feedback: Common Issue

- "Over **floating**"
  - Only need to ensure floats when doing division
  - Example:  

```
float(5.0/9.0)*(float(fahrTemp)-32.0)
```



Only need one to be a float

Correct but makes code difficult to read/understand
  - Alternative: 

```
5.0/9*(fahrTemp-32)
```

Jan 25, 2011

Sprenkle - CSCI111

## Lab 1 Feedback: Common Issue

- "**floating**" too late
  - Example:  

```
float(num1 * num2 / num3)
```
  - How evaluated:
    1. Computes num1\*num2/num3
    2. Float-ifies the result
  - Correct:  

```
num1 * num2 / float(num3)
```

Jan 25, 2011

Sprenkle - CSCI111

## Printing %

- Use %% in the format specifier template

```
"%8.3f%%" % 25.4316
```

```
→ " 25.432%"  
12345678
```

Jan 25, 2011

Sprenkle - CSCI111

## Lab 2 Overview

- Practice Python programming
  - String operations
  - Using Functions, Modules
  - for loops

Jan 25, 2011

Sprenkle - CSCI111

## Network Addresses

- A computer on a network has an **address**.
  - Address is used to uniquely identify the computer (also known as a host) on the network
- The most common address system in use today is the **Internet Protocol** (IPv4) addressing system
  - a 32-bit address, typically written as a "dotted-quad": four numbers, 0 through 255, separated by dots, e.g.,  
137.113.48.2

Jan 25, 2011

Sprenkle - CSCI111

## DNS: Domain Name System

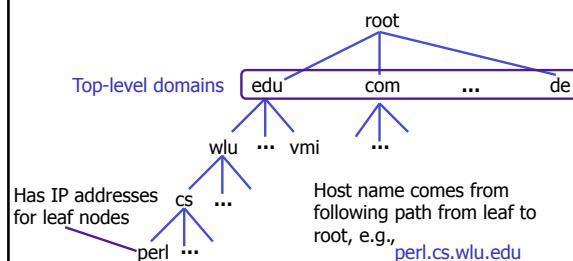
- Translate IP addresses to human-understandable host names and vice versa
  - Example: going from **www.cnn.com** to IP address **64.236.16.20**

Jan 25, 2011

Sprenkle - CSCI111

## DNS: Domain Name System

- Unique names for computers
- Hierarchical system (tree structure)



Jan 25, 2011

Sprenkle - CSCI111

## Using UNIX network utilities

- **host <ipaddress or name>**
  - Examples:
    - host 64.236.16.20
      - For host name **www2.cnn.com**
    - host **www.espn.com**
      - For IP address **199.181.132.250**
- **nslookup <ipaddress or name>**
  - Gets similar information
- **whois <domainname>**
  - Get information about registered name

Jan 25, 2011

Sprenkle - CSCI111