LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

30 August 2016

Class outline

- Programming project demos
- Week 1 - Recap
- Review of programming concepts
- Practice: Problem Set 1

Programming project demos

Any questions about Week 1 content, course syllabus/policies etc?
.. or about the coursera lectures?

American Association of Corpus Linguistics conference is happening in ISU in a few weeks. Try to register and attend!

Review of programming concepts

Basic building blocks in one slide

- Defining variables, writing different expressions to change the variables in the program.

Basic building blocks in one slide

- Defining variables, writing different expressions to change the variables in the program.
- Writing conditional statements (if condition, do this; else do that)

Basic building blocks in one slide

- Defining variables, writing different expressions to change the variables in the program.
- Writing conditional statements (if condition, do this; else do that)
- Writing loops (for, while) to iteratively repeat some set of expressions until some condition is met.

Basic building blocks in one slide

- Defining variables, writing different expressions to change the variables in the program.
- Writing conditional statements (if condition, do this; else do that)
- Writing loops (for, while) to iteratively repeat some set of expressions until some condition is met.
- Manipulating different data structures (strings, Arrays/lists, Dictionaries/Hashes etc)

Basic building blocks in one slide

- Defining variables, writing different expressions to change the variables in the program.
- Writing conditional statements (if condition, do this; else do that)
- Writing loops (for, while) to iteratively repeat some set of expressions until some condition is met.
- Manipulating different data structures (strings, Arrays/lists, Dictionaries/Hashes etc)
- Reading and Writing data into files on your hard disk.

Some practice with your programming language syntax

Basic Building Blocks: Variables and Expressions

Do the following in a program or in the interactive console.

- Define three variables: a, b, c and assign values: 3, 4, and ”a” respectively.
- print the output of expressions: a + b, a + c
- execute the expression a = b and print the value of a after that.
- execute: a = 5, and then, print the value of (a+b)*(a-b)

Basic Building Blocks: Conditional statements

Do the following in a program.

- Write a program that accepts some input from the user.
- If the user enters "Hello", the program prints "Hello". For any other input, the program prints "Politeness first".

Basic Building Blocks: Loops

- Write a program that prints numbers from 1 to 20.

Basic Building Blocks: Data Structures

- Write a program that takes a sentence, and splits at whitespace.

Basic Building Blocks: Files

- Write a program that reads in a sample .txt file, prints the file contents, and appends one additional line to the file.

Some non-syntax based exercises

What is this below pseudo-code doing?

```
(Let us say str is a string)
length = len(str) #number of characters in the string
i = 0
while i<length:
    if str[i] != str[length-1-i]
          return False
    else:
          continue
    i = i+1
return True
```

What is this below pseudo-code doing?

```
(Let us say str is a string)
length = len(str) #number of characters in the string
i = 0
while i<length:
    if str[i] != str[length-1-i]
          return False
    else:
          continue
    i = i+1
return True
```

What is returned if str has a value ”exception”?

What is this below pseudo-code doing?

```
function/subroutine magic(n):
if n <= 0:
   return 0
else if n == 1:
   return 1
else:
   return magic(n-1) * n
```

What is this below pseudo-code doing?

```
function/subroutine magic(n):
if n <= 0:
   return 0
else if n == 1:
   return 1
else:
   return magic(n-1) * n
```

What is returned if $n = 5$? 6?

What is this below pseudo-code doing?

```
function change(word):
  if word.endswith("ational"):
      replace "ational" with "ate"
  else if word.endswith("tional"):
      replace "tional" with "tion"
  return word
```

What is this below pseudo-code doing?

```
function change(word):
  if word.endswith("ational"):
      replace "ational" with "ate"
  else if word.endswith("tional"):
      replace "tional" with "tion"
  return word
```

What will this return for ”relational” and ”conditional”?

What happens if I change the order of the if statements?

```
function change(word):
  if word.endswith("tional"):
      replace "tional" with "tion"
  else if word.endswith("ational"):
      replace "ational" with "ate"
  return word
```

What happens if I change the order of the if statements?

```
function change(word):
  if word.endswith("tional"):
      replace "tional" with "tion"
  else if word.endswith("ational"):
      replace "ational" with "ate"
  return word
```

What will this return for ”relational” and ”conditional”?
Note: Your rules in Question 1 of Assignment 1 should look like this.

What is this below pseudocode doing?

```
content = read(some_text_file)
freq_data = {} #Empty dictionary.
words_list = split_into_words(content)
for word in words:
   length = len(word)
   if (len > 5):
        if word in freq_data:
            freq_data[word] = freq_data[word] + 1
        else:
            freq_data[word] = 1
display freq_data
```

What will freq_data contain, if we program this logic?

Errors while programming

- Syntax errors: You should know it is a syntax error by seeing the error messages (missing colons, parantheses etc)

Errors while programming

- Syntax errors: You should know it is a syntax error by seeing the error messages (missing colons, parantheses etc)
- Logic errors: Slightly difficult to find out, because you may not get an error message. You may only get a wrong answer. One solution: Debug your code. Put print statements in the middle, wherever you have questions.

Errors while programming

- Syntax errors: You should know it is a syntax error by seeing the error messages (missing colons, parantheses etc)
- Logic errors: Slightly difficult to find out, because you may not get an error message. You may only get a wrong answer. One solution: Debug your code. Put print statements in the middle, wherever you have questions.
- Other errors: such as division by zero, inputting a decimal number when a string is expected, etc. - use try-catch/try-except blocks in your programming to handle such cases.

Start working on Problem Set 1

... we will briefly discuss Problem set 1 on thursday, and move on to other practice exercises.

Something that can be useful: pythontutor.com lets you visualize your python code line by line, showing what variables get what values etc.