ENGL 516X:
Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

18 Jan 2018

Class outline

- Algorithms and Flowcharts: Discussion
- Python - review of last class
- Logical operations
- Useful python functions
- Writing your first python program
- Practice exercises

Algorithms: A small quiz

Write an Algorithm/flowchart for this description

There are 10 Mangoes. Of which 5 are ripe. 4 are raw. One is spoilt. Ripe mangoes have a yellow or reddish yellow color and are soft to touch. Raw mangoes are green in color and hard to touch. Spoilt mango is too soft to touch and is brownish yellow in color. Now, write an algorithm that looks at a mango, and decides whether it is raw or ripe or spoilt.

Algorithms: A small quiz

Write a flowchart/algorithm to calculate the factorial of a number N. (Factorial of 5 is represented by 5! and is calculated as: 5*4*3*2*1 = 120. Factorial of 6 is 6*5*4*3*2*1 or 6*5!. That is 720.)

From the last class

- Different mathematical operations and symbols in Python
- Different ways of assigning and re-assigning values to variables
- Some errors we may see if we make mistakes

- so far, we worked with the Python console. What we wrote goes away if we close PyCharm. We need to save what we wrote as python program files (.py) to reuse these.

Python Console vs Python program

- Python console: interactive, instant result
- Python program: You should "run" or execute the program to see your results.
- Console: You write x=3, and write x in next line, 3 gets printed in the other line
- Program: You have to write print(x)
- Program: You can save it and reuse it again and again

Your First Python Program

1. Go to PyCharm. In File Menu, choose "New Project" and name it Week2.
   (A project in this context is a collection of programs you write).
2. In this new project Week2, Rightclick and choose New->Python File. Name your file: MyFirstProgram or something.
3. Go to MyFirstProgram.py and type: print('Hello World!')
4. In the next line, type: print("This is my first Python program")
5. Click on the green pennant/triangle to run/execute your program.

A cool quote about programmers

"The ideal programmer would have the vision of Isaac Newton, the intellect of Albert Einstein, the creativity of Miles Davis, the aesthetic sense of Maya Lin, the wisdom of Benjamin Franklin, the literary talent of William Shakespeare, the oratorical skills of Martin Luther King, the audacity of John Roebling, and the self-confidence of Grace Hopper" (Source: Introduction to

Computing: Explorations in Language, Logic and Machines, By David Evans, Page 35.)

Python Basics

Write a program basics.py with the following lines.

```
print(4)
print(4*7)
print("hello " * 4)
print("hello" * 4)
x = 3
print(x)
print(x*4)
y = x
print(y)
name = input('Enter your name: ')
print ('Hello', name)
```

-Does it run? If it does not run, make it run. What do you see? It ran completely if you saw the message: "Process finished with exit code 0"

Python Basics - Continued

To err is human. To not forgive is computerish.

Go back to interactive/console mode now, and do these:

1. What happens when you type:
   - print("name)
   - print("name')
   - print("John's pen")
   - print('John's pen')
   - print('name')

Python Basics - Continued

To err is human. To not forgive is computerish.

On console, what happens when you type:

- x = "hello"
- print(x+5)

Python Basics - Continued

To err is human. To not forgive is computerish.

On console, what happens when you type:

- x = "hello"
- print(x+5)
- Why?

Python Basics - Continued

To err is human. To not forgive is computerish.

On console, what happens when you type:

- x = "hello"
- print(x+5)
- Why?

What happens when you type:

- print(3)
- print(1,000,000)

Python Basics - Continued

To err is human. To not forgive is computerish.

On console, what happens when you type:

- x = "hello"
- print(x+5)
- Why?

What happens when you type:

- print(3)
- print(1,000,000)
- Did python give you what you expected in both cases? Why? Why not?

Type of variable: type() function

- type() function is useful when we don't know what is the type of data stored in a variable.
- type(3) returns a integer.
- type("3") returns a string
- type(3.9) returns a float
- type([1,2,3]) returns a list
- type(print) returns "builtin function or method"

... and so on.

print(1,000,000)

How do you think Python sees this?

Asking the user for input

- input() is a built-in function in Python to seek input from user. (we will soon write our own functions)
- type: input() on your console. Press enter. What happened?

Asking the user for input

- input() is a built-in function in Python to seek input from user. (we will soon write our own functions)
- type: input() on your console. Press enter. What happened?
- If you now type something on the console, and press enter, what happens?

Asking the user for input

- input() is a built-in function in Python to seek input from user. (we will soon write our own functions)
- type: input() on your console. Press enter. What happened?
- If you now type something on the console, and press enter, what happens?
- i = input("Enter a number: ") - type this on the console and see what happens.

Asking the user for input

- input() is a built-in function in Python to seek input from user. (we will soon write our own functions)
- type: input() on your console. Press enter. What happened?
- If you now type something on the console, and press enter, what happens?
- i = input("Enter a number: ") - type this on the console and see what happens.
- if you now type i and press enter, what does it print?
- Is that what you would expected to see?

Conversion between different types

- If we ask the user to enter a number, and it is read by default as string, what should we do?
- Python has some built in functions to convert between data types.
- int() converts whatever you give it into a integer if possible, otherwise, it throws an error.
- Type int("3"), int("3.4"), int("a") on your console -which of these runs without error?

Conversion between different types

- If we ask the user to enter a number, and it is read by default as string, what should we do?
- Python has some built in functions to convert between data types.
- int() converts whatever you give it into a integer if possible, otherwise, it throws an error.
- Type int("3"), int("3.4"), int("a") on your console -which of these runs without error?
- similar functions: float(), str()

Yesterday, you said syntax errors are easy to fix

But if it just says: "invalid syntax" without details, may be that is difficult.

- 76trombones = 'big parade'
- more@ = 1000000
- class = 'Advanced Theoretical Zymurgy'

-Why are these errors?

Yesterday, you said syntax errors are easy to fix

But if it just says: "invalid syntax" without details, may be that is difficult.

- 76trombones = 'big parade'
- more@ = 1000000
- class = 'Advanced Theoretical Zymurgy'

-Why are these errors?
Type 17/0 on console - what error do you see?

Operators in Python

- Operators are those symbols between variables that allow you to perform some "operations".
- mathematical operators are: +, -, /, *, % etc.
- Python also has logical operators: and, or, not
- comparisons: ==, <, ,>, <=, >= (there are few more - more on those later)
- When a line of code has more than one operator, rules of precedence exist to decide what operation should be performed first.
- e.g., 5-3*2 - what is the output of this?

Operators in Python

- Operators are those symbols between variables that allow you to perform some "operations".
- mathematical operators are: +, -, /, *, % etc.
- Python also has logical operators: and, or, not
- comparisons: ==, <, ,>, <=, >= (there are few more - more on those later)
- When a line of code has more than one operator, rules of precedence exist to decide what operation should be performed first.
- e.g., 5-3*2 - what is the output of this?
- e.g., (5-3)*2 - what is the output?

Rules of precedence

- parenthesis come first - anything within parentheses gets executed first.
- Exponents come next 2**1+1 is 3, not 4, and 3*1**3 is 3, not 27.
- Multiplication and division have the same level of precedence
- Finally, addition and subtraction have same precedence.
- When you have two operators of same precedence, execution is left to write. 5-3+2 is 4.

general advice: use parentheses, to avoid confusion.

Python Basics: Practice

Switch back to Python Project mode and create a new code file called SecondProgram and write a code that does these things:

- Prompt the user to enter their name.
- Prompt the user to enter their country.
- Write a print statement which says: Hello [Name] from [Country]. Welcome to Ames.
- Run your program, and test with the following values:
  1. Name = Bond, Country = UK
  2. Name = 3, Country = Poland
- Add a line in your program that accepts a phone number (10 digits, without space, any signs) from the user, and prints out the number.

Python Basics: More Practice

source: https://goo.gl/yPVqDn

The formula for compound interest calculation is given as:
$A = P * (1 + \frac{r}{n})^(n * t)$ Where:
P = principal (initial investment) r = annual nominal interest rate n = number of times interest is compounded per year t = number of years. - Write a program which has initial values as: P=12000, n=12, r=8%. Then, the program should ask the user for t i.e., number of years, and print the final amount after t years.

Next Class

1. Topics: Conditional statements
2. ToDo before the class
   - 2.1 Readings: Chapter 3 in the textbook
   - 2.2 Submit Assignment 1
3. Post your program on the discussion forum for today if you want to discuss differences between your programs.
4. Try to do exercises at the end of Chapter 2