GEOG 432/832: Programming, Scripting, and Automation for GIS

Week 01.03: Programming exercise

Dr. Bitterman

Today's schedule

- Open discussion
- Exercises

Anything to discuss?

Fridays are typically our work days

The purpose of today's work

- Practice, practice, practice
- Don't worry about completing all the tasks in class
- These are here to stimualte your brain
- ...and to prime you for the rest of the semester

there will be hiccups along the way, don't worry (too much) about them

Python the hard way

- A sort of "classic" resource for learning Python *by doing*, even if you don't know **what** you're doing
- But it's in Python 2.7, while we're using Python 3
- It's unfortunately been monetized: https://learncodethehardway.org/python/
- But some resources (like I'm co-opting here) are available on the web: http://cglab.ca/~morin/teaching/1405/lpthw/book/ex1.html

A good first program

Open Jupyter Notebooks and create a new notebook for today's work

Enter the following - type it in, do **NOT** copy-paste

```
print("Hello World!")
print("Hello Again")
print("I like typing this.")
print("This is fun.")
print('Yay! Printing.')
print("I'd much rather you 'not'.")
print('I "said" do not touch this.')
```

Run the program

- What happens?
- Do you note anything special about the formatting of the strings?

Comments and octothorpes

99 PI: https://99percentinvisible.org/episode/octothorpe/

Comments are very important in your programs. They are used to tell you what something does in English, and they also are used to disable parts of your program if you need to remove them temporarily. Here's how you use comments in Python:

```
# A comment, this is so you can read your program later.
# Anything after the # is ignored by python.

print("I could have code like this.") # and the comment after is ignored

# You can also use a comment to "disable" or comment out a piece of code:
# print "This won't run."

print("This will run.")
```

Comment-out a print command and run again - what happens differently?

Numbers and math

Every programming language has some kind of way of doing numbers and math.

- `+` plus
- `-` minus
- `/` slash
- `*` asterisk
- `%` percent (modulo)
- `<` less-than
- `>` greater-than
- `<=` less-than-or-equal
- `>=` greater-than-or-equal

Numbers and math part 2

Write one at a time and calculate what the output will be prior to running each command:

```
print("I will now count my chickens:")

print("Hens", 25 + 30 / 6)
print("Roosters", 100 - 25 * 3 % 4)

print("Now I will count the eggs:")

print(3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6)

print("Is it true that 3 + 2 < 5 - 7?")

print(3 + 2 < 5 - 7)

print("What is 3 + 2?", 3 + 2)
print("What is 5 - 7?", 5 - 7)

print("Oh, that's why it's False.")

print("How about some more.")

print("Is it greater?", 5 > -2)
print("Is it greater or equal?", 5 >= -2)
print("Is it less or equal?", 5 <= -2)
```

Variables and characters

1. Write a comment above each line explaining to yourself what it does in English

2. Explain to your neighbor what the commands will do

```
cars = 100
space_in_a_car = 4.0
drivers = 30
passengers = 90
cars_not_driven = cars - drivers
cars_driven = drivers
carpool_capacity = cars_driven * space_in_a_car
average_passengers_per_car = passengers / cars_driven

print("There are", cars, "cars available.")
print("There are only", drivers, "drivers available.")
print("There will be", cars_not_driven, "empty cars today.")
print("We can transport", carpool_capacity, "people today.")
print("We have", passengers, "to carpool today.")
print("We need to put about", average_passengers_per_car, "in each car.")
```

Do it yourself (but let's start together)

Say hello

Create a string variable called x and assign it the value "Hello". Display the contents of the x variable in the Console.

Concatenate two strings

Create a string variable called *first* and assign to it your first name. Likewise, create a string variable called *last* and assign to it your last name. Concatenate (merge) the two strings together, making sure to also include a space between them.

Do some easy math and report results

Create a variable score1 and assign it a value of 60. Then create a variable score2 and assign it a value of 40. Compute the sum and average of these values and output to the Console the following messages, filling in the blanks:
The sum of these scores is _______.
Their average is _______.

tips:

In constructing your messages, you're probably going to want concatenate text strings with numbers. This requires converting the numeric values into strings to avoid a syntax error. The str() function can be used to do this conversion. **For example:**

```
print('Score 1 is ' + str(score1))
```

For next week

Topic: Geoprocessing in ArcPro and Python fundamentals

- Read: chapters 3 & 4 in textbook
- Lab 1 starts next week