Programming Foundations in Python
Adapted From: CMSC 201 Computer Science I for Majors

Lecture 02b – Intro to Python

Written Programs vs Python Interpreter

We Started Python Yesterday!

- Two ways to use Python

We will mostly write programs for assignments

  - You can write a program as a series of instructions in a file and then execute it

Use the interpreter shell to help you test things

  - You can also test simple Python commands in the Python interpreter

Written Programs

- Create, write, and save a Python file (.py)
- File is run via the command line

```
python myProgram.py
```

- File must be complete to run correctly
- Program cannot be edited on the fly
  - Must be exited, file re-opened, changes made, file saved and closed, and then re-run the program

Python Interpreter

- The "interactive" interpreter evaluates each individual line of code as it's typed in
- Type "**`python`**" to launch the interpreter

`>>>` is where the user types their code

```
>>> print("Hello")
Hello
>>> 4 + 7
11
>>>
```

lines without a "`>>>`" are Python's response

Any Questions from Last Time?

Today's Objectives

- To learn about variables
  - How to use them
  - Different types
- To learn how to use input and output
  - To do interesting things with our program

Variables

Elements of a Program

- Identifiers
  - Variables
  - Functions (later in the course)
- Expressions
  - Code that manipulates or evaluates identifiers
- Literals
- Operators

What Is a Variable?

- Something that holds a value
  - Can change (unlimited number of times)
- Similar to variables in math
- In simple terms, a variable is a "box" that you can put stuff in

Rules for Naming Variables

- Variable names can contain:
  - Uppercase letters (**`A-Z`**)
  - Lowercase letters (**`a-z`**)
  - Numbers (**`0-9`**)
  - Underscores (_)
- Variables can't contain:
  - Special characters like **$**, #, **&**, **^**, **)**, **(**, @

More Rules for Naming Variables

- Variables can be any length
  - **`x`**
  - **`IsKanyeRunningForPresidentIn2020`**
  - **`myName`**

- Variables cannot **start** with a digit
  - **`2cool4school`** is not a valid variable
  - **`cool4school`** is a valid variable

Variables and Keywords

- Keywords are "reserved" words in Python

```
False      class      finally    is         return
None       continue   for        lambda     try
True       def        from       nonlocal   while
and        del        global     not        with
as         elif       if         or         yield
assert     else       import     pass
break      except     in         raise
```

- Variables cannot be keywords
  - **`or`** is not a valid variable name
  - **`orange`** is an acceptable variable name

Exercise: Variables

- Are the following legal or illegal in Python?

```
3cart

case2

cats_and_dogs

EXIT_CODE
```

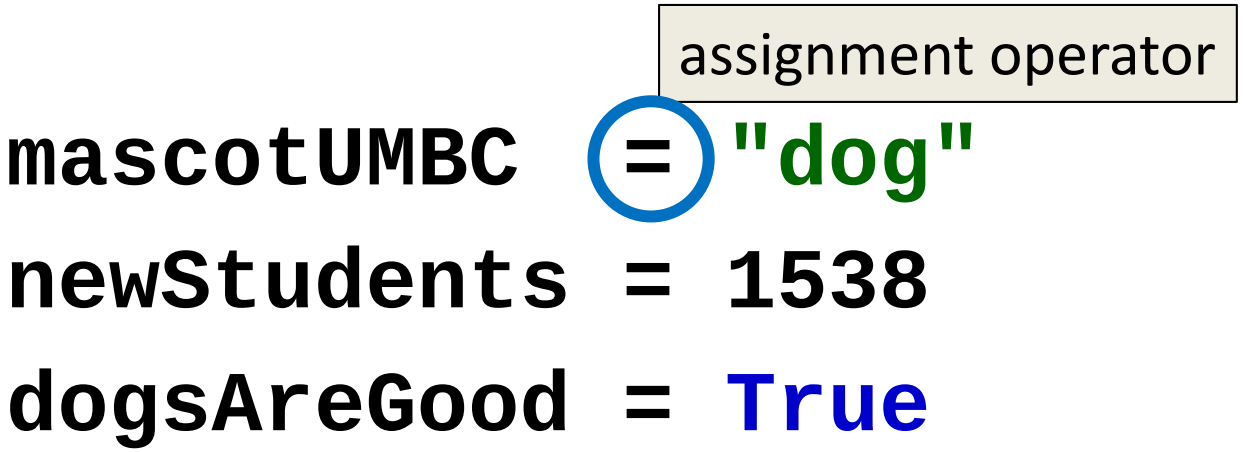
Exercise: Variables

- Are the following legal or illegal in Python?

`3cart`	**No – Illegal!**
`case2`	**Yes – legal!**
`cats_and_dogs`	**Yes – legal!**
`EXIT_CODE`	**Yes – legal!**

Using Variables in Python

- You create a variable as soon as you declare it
- You also need to initialize it before using it
  - Use the assignment operator (equal sign)



Expressions

Expressions

- Programs manipulate data
  - Allows us to do interesting things
- Expressions calculate new data values
- Use assignment operator to set new value

Expressions Example

variable being set

assignment operator

value (a "literal")

```
numCandy    = 4
priceCandy = 0.58
totalCost   = numCandy * priceCandy
```

expression

Common Mistake

- Many new programmers mix up the left and right hand sides of the assignment operator
  - Variable being set must be on the ***left***
  - Expression is on the ***right***
  - Evaluate the expression first, then assign the value

```
numCandy = 4 + 1
```

```
4 + 1 = numCandy
```

Variable Types

- There are many different kinds of variables!
  - Numbers
    - Whole numbers (Integers)
    - Decimals (Floats)
  - Booleans (**`True`** and **`False`**)
  - Strings (collections of characters)

Variables Types: Examples

```
aString   = "Hello class"
float_1   = 1.12
myBool    = True
anInteger = 7

className   = "Python coding"
classCode = 201
```

Variable Usage

- Variables are designed for storing information
- Any piece of information your program uses or records must be stored in a variable

Literals and Operators

Literals

- Literals in Python are values you use "literally"
  - Can be assigned to a variable or not
- For example:
  - 2 is an integer literal
  - "Hello" is a string literal
  - 4.0 is a float literal
  - False is a Boolean literal

Using Literals

- The expression below assigns the string literal “CMSC” to a variable called major

```
major = "CMSC"
```

- The expression below prints the integer literal 50 without assigning it to a variable

```
print(50)
```

Operators

- Operators are special symbols that allow Python to perform different operations

- There are many types of operators
  - Mathematical
  - Comparison
  - Assignment
  - Logical

Operator Types

- We won't cover all the types in detail today, but here are some simple examples

- Mathematical

  + - * / %

- Comparison

  < <= != >= ==

- Assignment

  = += *=

Practice Exercises

- Print the value of the variable **`myDog`**
  - Remember to assign a value to **`myDog`** first!
- Set a value for a variable called **`bill`**, and calculate and print the 15% tip for that **`bill`**
- Create your own expression using at least two variables, and print out the result

Input and Output

Output

- Output is text that is printed to the screen
  - So the user can see it

- The command for this is **`print`**
  - Use the keyword "**`print`**" and put what you want to be displayed in parentheses after it

Output Example

```
print (3 + 4)
print (3, 4, 3 + 4)
print()
print("The answer is", 3 + 4)
```

What does this output to the screen?

```
7
3 4 7

The answer is 7
```

Input

- Input is information we get from the user
  - We must tell them what we want first

```
userNum = input("Please enter a number: ")
print(userNum)
```

- The input and output will look like this:

```
Please enter a number: 22
22
```

How Input Works

```
userNum = input("Please enter a number: ")
```

- Takes the text the user entered and stores it
  - In the variable named **`userNum`**

- You can do this as many times as you like!

```
userNum  = input("Enter another number: ")
userNum2 = input("Enter a new number: ")
userAge  = input("Please enter your age: ")
```

Input as a String

- Everything that is stored via `input()` will come through in the form of a string

- There is a difference between `"10"` and `10`
  - `"10"` is a string containing two characters
  - `10` is understood by Python as a number

Converting from String

- To turn an input string into a number, you can do the following:

```
aNum = input("Enter a number: ")
aNum = int(aNum)
```

- "int" stands for "integer" (a whole number)

- You can also do it in one line:

```
aNum = int(input("Enter a number: "))
```

Converting from String

- We can cast to other data types as well

```
gpa = float(input("Enter GPA: "))
```

- Do you think the string `"1,024"` will work if we try to cast it as an integer? Why?
- It won't work
  - The comma character isn't a number

Image Sources

- Cardboard box:
  - https://pixabay.com/p-220256/
- No cursing sign (adapted from):
  - https://www.flickr.com/photos/rtgregory/1332596877
- Rock candy:
  - https://commons.wikimedia.org/wiki/File:Rock-Candy-Sticks.jpg
- Broken chain:
  - https://pixabay.com/p-297842/