CMSC201
Computer Science I for Majors

Lecture 03 - Variables

Last Class We Covered

- Algorithms
- Program Development
- Control Structures
  - Sequential
  - Decision Making
  - Loops
- Types of Errors
  - Syntax
  - Logic

Any Questions from Last Time?

Exercise

- What will each of the following do?

1. `print("Hello')`

   Error – Need to have matching ' and "

2. `Print('Hello')`

   Error – Need to have lowercase `print`

3. `print('Hello World')`

   Hello World

Today's Objectives

- To start learning Python
- To learn more about variables
  - How to use them
  - Different types
- To learn how to use input and output
  - To do interesting things with our program
- To play a party game

Software Development Process

- A quick reminder about the process we follow

1. Analyze the problem
   - Determine specifications (requirements)
2. Create a design
3. Implement the design
4. Test and debug the program
5. Maintain the program

Don’t “Cowboy Code”

- “Cowboy coding” is when you jump right in to writing code without planning beforehand
  - No formal management of project
  - No standard way of coding
  - Not planning things out
    - Forgetting to include important things
    - Having to make big changes later

Image from www.publicdomainpictures.net

Example: Temperature Converter

You have been invited to live in Europe during a semester abroad. You aren’t sure how to dress because the temperature is given in Celsius.

- Problem:
  - Temperature is given in Celsius
- Solution:
  - Write a program to convert Celsius to Fahrenheit

Input/Process/Output

- Input
  - What information do you need for your converter?
- Process
  - What formulas do you need for your converter?
- Output
  - What is the output from your converter?

Introduction to Python (Variables)

Python

- Python is a widely used language
  - General purpose
  - High-level language
- Emphasizes code readability
  - More streamlined than some other languages

“Hello World!”

- In Python:

```
print("Hello World!")
```

- In the C++ programming language:

```
#include <iostream>
int main() {
    std::cout << "Hello World!\n";
}
```

Elements of a Program

- Identifiers
  - Variables
  - Modules (later in the semester)
  - Functions (later in the semester)

- Expressions
  - Code that manipulates or evaluates identifiers

We Start Python Today!

- Two ways to use Python

We will write programs for assignments

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

Use the interpreter to help you test things

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

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

Image from pixabay.com

Rules for Naming Variables

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

Image from https://www.flickr.com/photos/rtgregory

More Rules for Naming Variables

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

- Variables cannot **<u>start</u>** with a digit
  - `2cool4school` is not a valid variable
  - `cool4school` <u>is</u> a valid variable

Variables and Keywords

- Keywords are the reserved words in Python

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

- 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?

`1spam`	**No – Illegal!**
`raise1`	**Yes – legal!**
`Spam_And_Eggs`	**Yes – legal!**

But it doesn't follow our coding standards!

spamAndEggs or spam_and_eggs

Using Variables in Python

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

assignment operator

```
richFiddy  = 50000000
poorFiddy  = 0.50
brokeFiddy = 0
```

Introduction to Python (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

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

value (a "literal")

expression

Image from wikimedia.org

Common Mistake

- Many new programmers mix up the left and right hand sides of the assignment operator
  - Variable being set is 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
    - Integers
    - Floats (decimals)
  - Booleans (**`True`** and **`False`**)
  - Strings (collections of characters)

Variables Types: Examples

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

dogName   = "Ms. Wuffington"
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
  - Python doesn’t have a “short term memory,” so everything needs to be written down for it

Introduction to Python (Input and Output)

Output

- Output is text that is printed to the screen
  - So the user can see it (and respond)

- 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
```

Output Exercise 1

- What will the following code snippet print?

```
a = 10
b = a * 5
c = "Your result is:"
print(c, b)
```

```
Your result is: 50
```

Output Exercise 2

- What will the following code snippet print?

```
a = 10
b = a
a = 3
print(b)
```

There are two possible options for what this could do! Any guesses?

```
10
```

Output Exercise 2 Explanation

- Why does it print out 10?

- When you set one variable equal to another, they don’t become linked!
  - They are separate copies of a value

- After **`b`** is set to 10, it no longer has anything else to do with **`a`**

Output Exercise 2 Explanation

```
a = 10
b = a
a = 3
print(b)
```

Image from pixabay.com

Output Exercise 2 Explanation

```
a = 10
b = a
a = 3
print(b)
```

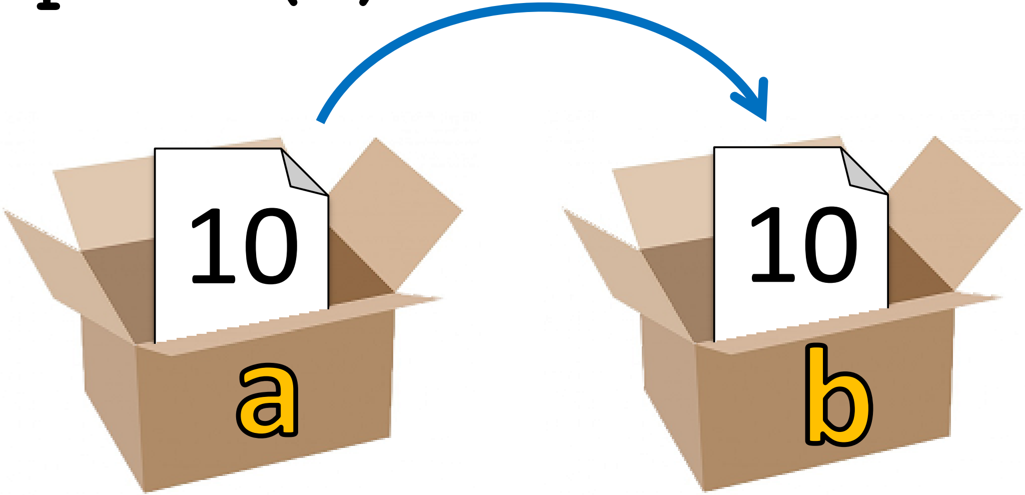


Image from pixabay.com

Output Exercise 2 Explanation

```
a = 10
b = a
a = 3
print(b)
```

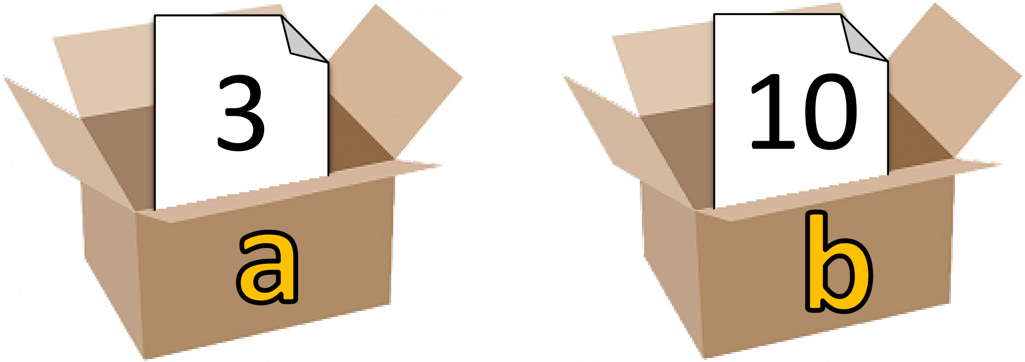
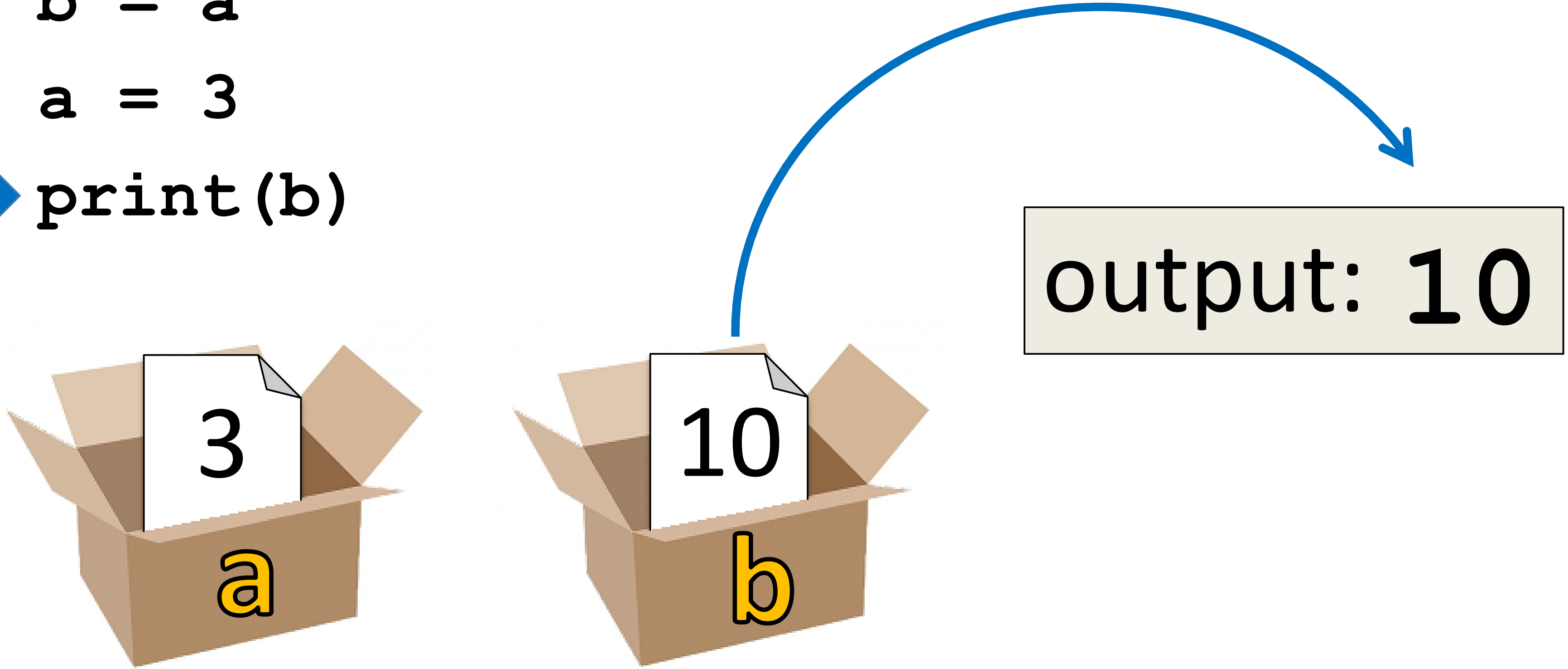


Image from pixabay.com

Output Exercise 2 Explanation

```
a = 10
b = a
a = 3
print(b)
```



output: 10

Image from pixabay.com

Input

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

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

- The output and input 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 comes through **`input`**`()` will come 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

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

- We can cast to other data types as well

```
flt = float(input("Enter float: "))
```

Exercises

Exercise: Calculating Averages

- Write, on paper or on your computer, a program that asks the user for two numbers and prints out the average.

- Make sure to use variables, and to get the input from the user!
- Does the order of operations come into play for this exercise?

Exercise: Assignment Weighting

- Pretend you're writing a program to compute someone's weight grade. You have so far:

```
hwWeight         = 0.4
examWeight       = 0.5
discussionWeight = 0.1
```

- Write a program that then asks the user for their homework grade, exam grade, and discussion grade and prints out their total grade in the class.

Class Exercise: Mad Libs

- Mad Libs is a word game where one player prompts the others for different types of words, using them to fill the blank in a story.
- The result is often hilarious, and almost always nonsensical.

Announcements

- Your discussions (Labs) start in person this week!
  - Go to your scheduled location and time

- Homework 1 is out (on Blackboard)
  - Due by this Wednesday (Sep 14) at 8:59:59 PM
  - Complete the Syllabus/Course Website Quiz to see it

- Academic Integrity Quiz on Blackboard soon
  - Must complete to see Homework 2