CS65: Introduction to Computer Science

Variables and expressions
Comments
Basic input/output

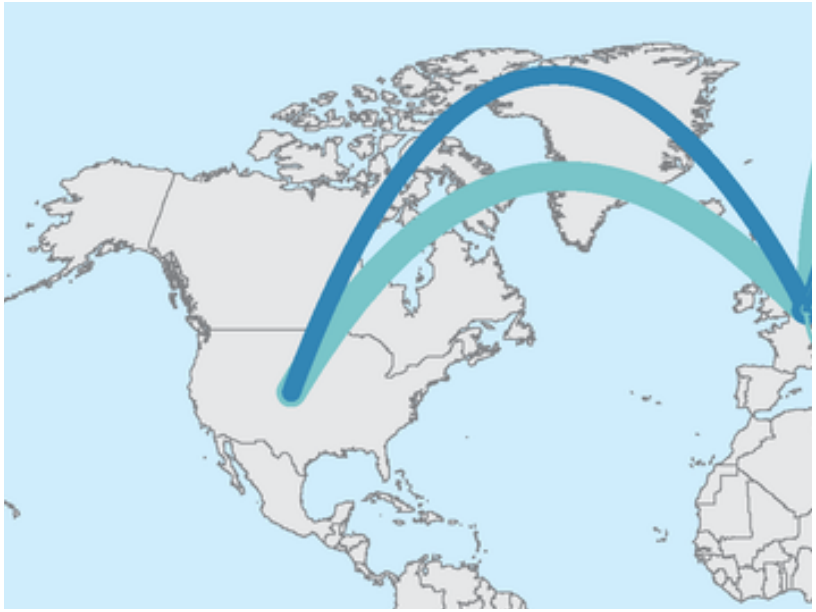
Md Alimoor Reza
Assistant Professor of Computer Science

Recap

- Algorithms
- Programming
- Writing program in a specific language eg Python
- Integrated Development Environment (IDE)

Recap: Algorithm

- step-by-step instructions to be executed by the machine
- Describe the process of making a trip from USA to Europe?
  - Person next to you is your partner
  - Write down the steps



Recap: What are computer programs?

- A program is a sequence of instructions that specifies how to perform a computation
  - can be written by a specific programming language

- Programming languages are formal language to express computations
  - **Python**
  - Java
  - C/C++

- Programming languages have strict rules, known as syntax that must be followed
  - Specific keywords need to be used to perform some action
  - Specific structure to be followed
  - Naming convention

Roadmap

- Setting up Thonny
- Comments in Python
- Variables and expressions
- Receiving input from user
  - Counterpart of showing an user some outputs

Thonny: Integrated Development Environment

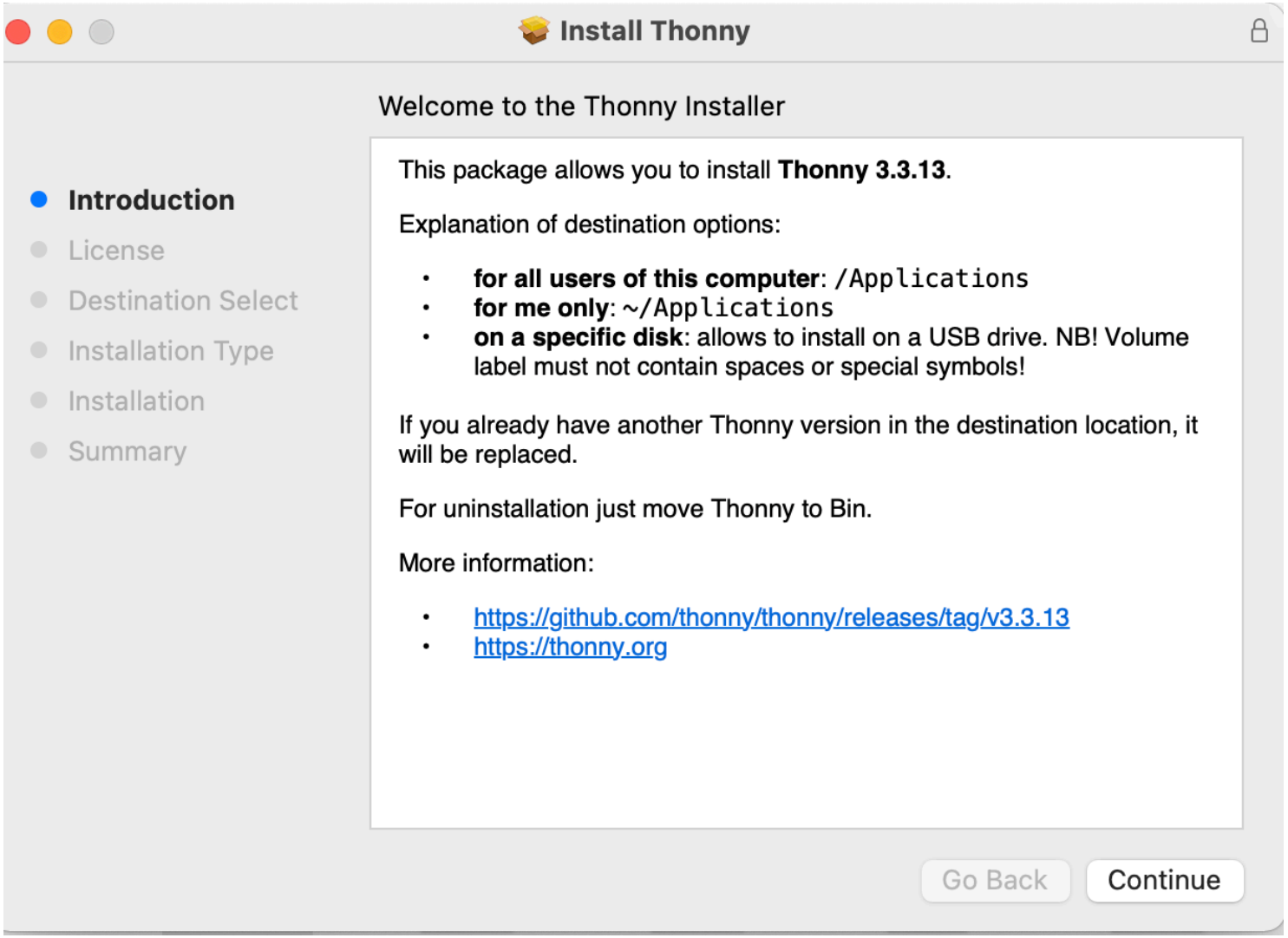
Step 1: Download and installation: Open your favorite web browser, eg, *Mozilla Firefox, Google Chrome, etc.* and go to the website https://thonny.org. On the upper-right corner you will notice the download link as shown below. Click on the *Mac* button and save the **.pkg* file.

Click on the *Download* button for Mac

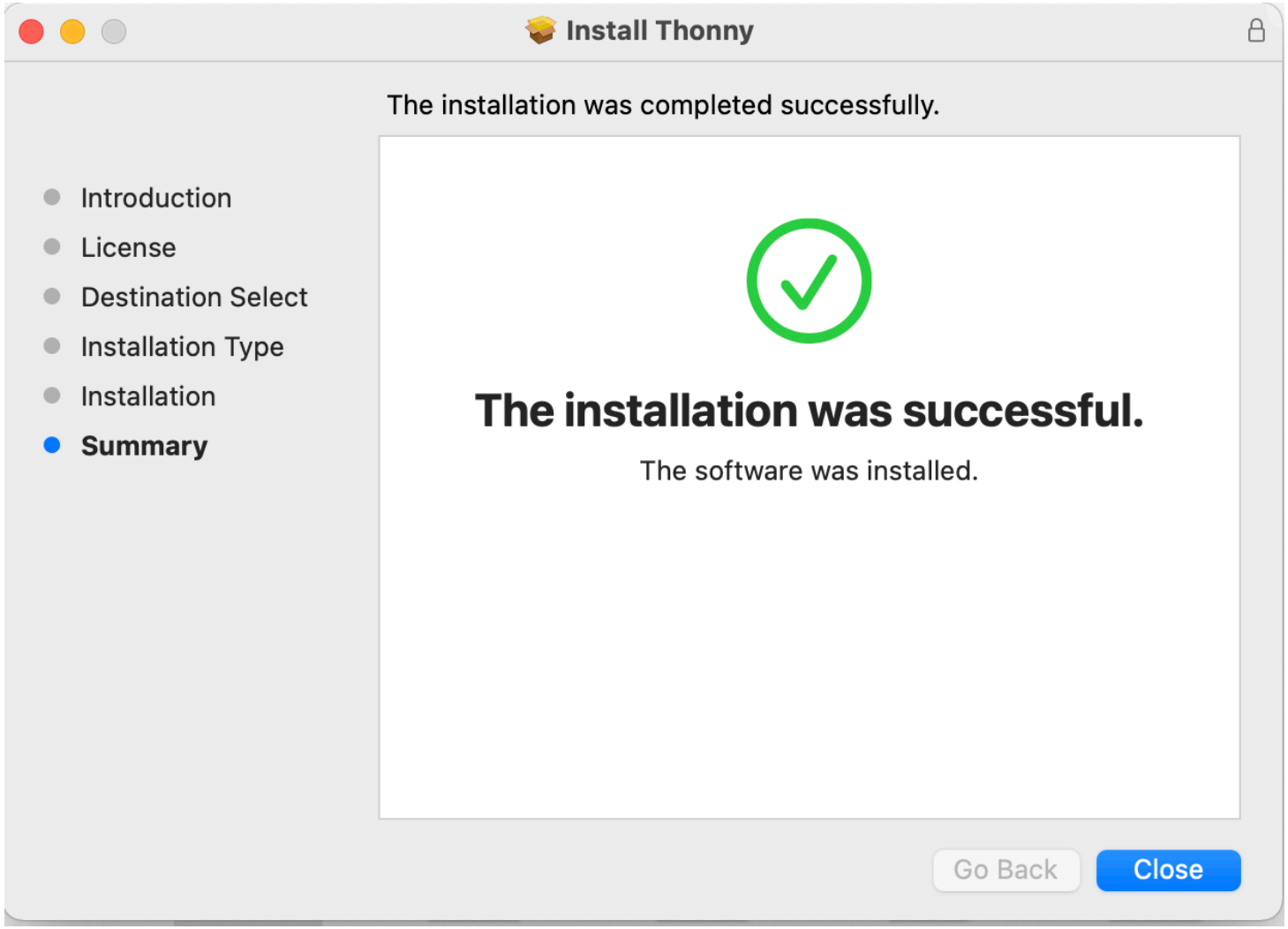
Double-click on the *'thonny-3.3.13.pkg'* and it will pop up a window for installation. Follow the steps as instructed. To help you guide the installation process, we are attaching the relevant snapshots during the installations.

Double-click on the *'thonny-3.3.13.pkg'*.

Thonny: Integrated Development Environment



Thonny: Integrated Development Environment



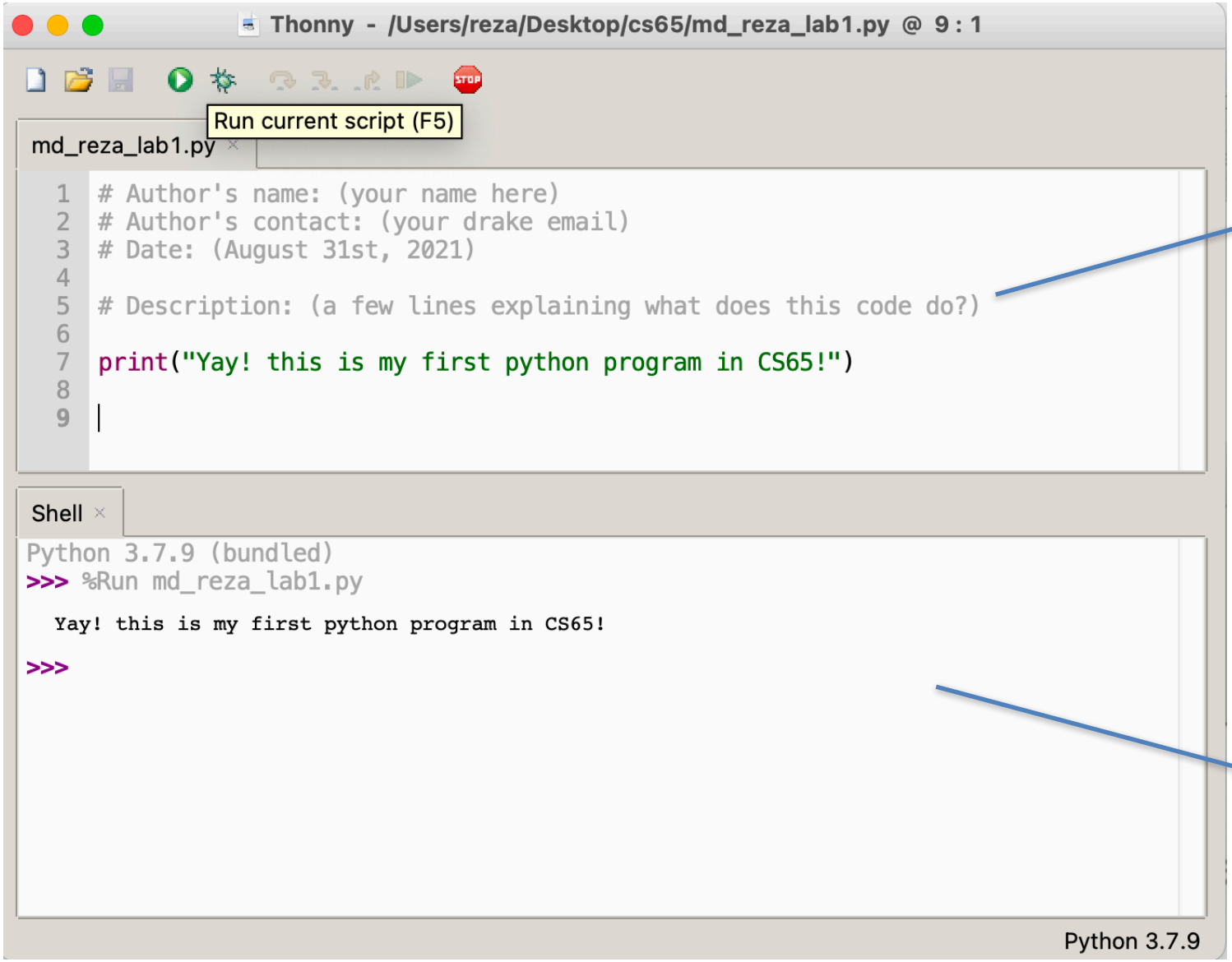
Thonny: Integrated Development Environment

Step 2: Openning up Thonny If you correctly follow the installation process, Thonny will be installed in your *Applications* folder as shown below.

Click on the *Thonny* icon to launch it.

Finally, we will be able to see the Thonny interface as shown below. We will be writing all our assignments, labs, and other practice programming in Python here. *Yay!!! We installed it successfully and ready to write program in Python!.*

Demo



Editor

Executable environment

Roadmap

- Setting up Thonny
- Comments in Python
- Variables and expressions
- Receiving input from user
  - Counterpart of showing an user some outputs

Comments

- Comments are notes explaining the functionality of your computer program (source code)

- Python comments are denoted with
  - # for a single line
  - triple quotes (either ‘ or “) for multiple lines

- Other languages eg, C++ has different syntax for comments

```
1  # Author's name: Md Alimoor Reza
2  # Author's contact: md.reza@drake.edu
3  # Date: (September 1st, 2021)
4  # Collaborator:
5  #   Your partner's name
6
7  #print("Yay! this is my first python program in CS65!")
8
9
10 #----------------------------   Task 1 ------------------------------------
11 # add, multiplication, divide, etc arithmetic operation
```

Comments

- Ignored by the Python interpreter
  - Won't see any output in the shell environment or any error message

- Helpful for people who may be reading the source code
  - Peer/partner
  - Grader
  - Professor

```
10 """
11  Author's name: Md Alimoor Reza
12  Author's contact: md.reza@drake.edu
13  Date: (September 1st, 2021)
14  Collaborator:
15    Your partner's name
16
17 #print("Yay! this is my first python program in CS65!")"""
18
```

Demo

Roadmap

- Setting up Thonny
- Comments in Python
- Variables and expressions
- Receiving input from user
  - Counterpart of showing an user some outputs

Variables

- Variable is a named storage space in computer memory for one Python value
  - Either we can write a value into a variable
  - Or we can read the value stored in that variable

```
33  time_sec = 60
34  temp_degree = 27
35
36  mile_to_kilometer = 1.609
37  price_in_dollars = 1500.89
```

- Above mentioned named-entities are variables
  - time_sec
  - temp_degree
  - miles_to_kilometer

Variable and assignment operator

- Need to use assignment operator (**=**) to store a value
- Location of assignment on the left
- Single value or some calculated value on the right
- **variable_name** = value

```
33  time_sec = 60
34  temp_degree = 27
35
36  mile_to_kilometer = 1.609
37  price_in_dollars = 1500.89
```

Numbers

```
first_name = "Md Alimoor"
last_name = "Reza"
```

Textual data

Demo

Rules for Variable Naming

- Give meaningful variable name to make it easily readable

```
x = 1.609
```

Vs

```
mile_to_kilometer = 1.609
```

- Name should begin with a lowercase letter

- Use underscore to connect multiple words

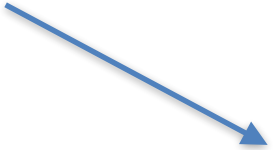
```
milestokilometer = 1.609
MilesToKilometer = 1.609
milesToKilometer = 1.609
```

Vs

```
mile_to_kilometer = 1.609
```

Rules for Variable Naming

- Names can only contain letter, numbers, and underscores
- First character must be a letter or an underscore
  - Then use letter/numbers/underscore
- Cannot be a Python keyword



```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

- Cannot contain spaces
- Variable names are case sensitive
  - Uppercase and lowercase name will signify different variable

Exercise

- Which of following are bad variable names and why?

```
import

First name

?my_variable

555_battery

X

FirstName

global

while
```

Rules for Variable Naming

- Names can only contain letter, numbers, and underscores
- First character must be a letter or an underscore
  - Then use letter/numbers/underscore
- Cannot be a Python keyword

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

- Cannot contain spaces
- Variable names are case sensitive
  - Uppercase and lowercase name will signify different variable

Expression

- A fragment of Python code that calculates a new value called an expression

- For example, you can convert miles into meters using the following expression:

```
num_of_miles = 10
miles_to_kilometer = 1.609
```

```
num_of_meter = num_of_miles*miles_to_kilometer*1000
```

Demo

Exercise

- Can you compute the area of a rectangle?
  - Length of the two sides will be given in variables

- Can you compute the area of a circle?
  - Radius of the circle is given
  - Value of Pi is 3.14159

Roadmap

- Setting up Thonny
- Comments in Python
- Variables and expressions
- Receiving input from user
  - Counterpart of showing an user some outputs

Getting Input from Users

- Built-in function in Python ***input("….")***
  - Step 1: displays the prompt to the user
  - Step 2: waits for user to type in something
  - Step 3: returns the typed content when user hits enter
  - Step 4: this value is stored if assigned to a variable

```
rect_a = input("enter the length of rectangle side a: ")
print(rect_a)
```

Demo

Errors (will be discussed more later)

- Syntax error
  - violating a programming language's rules on how symbols can be combined to create a program

- Runtime error
  - wherein a program's syntax is correct but the program attempts an impossible operation
    - dividing by zero
    - entering a string instead of an integer

Summary

- Takeaway from this lecture:
  - variable syntax, naming convention
  - expression for complex calculation in a Python line
  - comments are helpful notes
  - input() built-in function is useful for getting user input

- **To do: follow the instructions in Lab1**
  - Part 1 (we just did it today) & part 2 (finish by yourself)
  - Finish before the next class on Thursday, 03, 2022

Date	Topic	Reading
week 1 (Tue: 01/25)	Introduction to Computer Science Lecture 1 slide Lab 1 (release)	
week 1 (Thu: 01/27)	Variables, expression, and statements Lecture 2 slide	Reading: Chapter 1, Chapter 2