Introduction to Python

In this course, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to the process. We'll take a first look at variables, assignment, and input/output.

Content Learning Objectives

After completing this activity, students should be able to:

- Explain differences between a program's code and output.
- Use a Python shell to execute input and output functions.
- Write assignment statements and use assigned variables.

Process Skill Goals

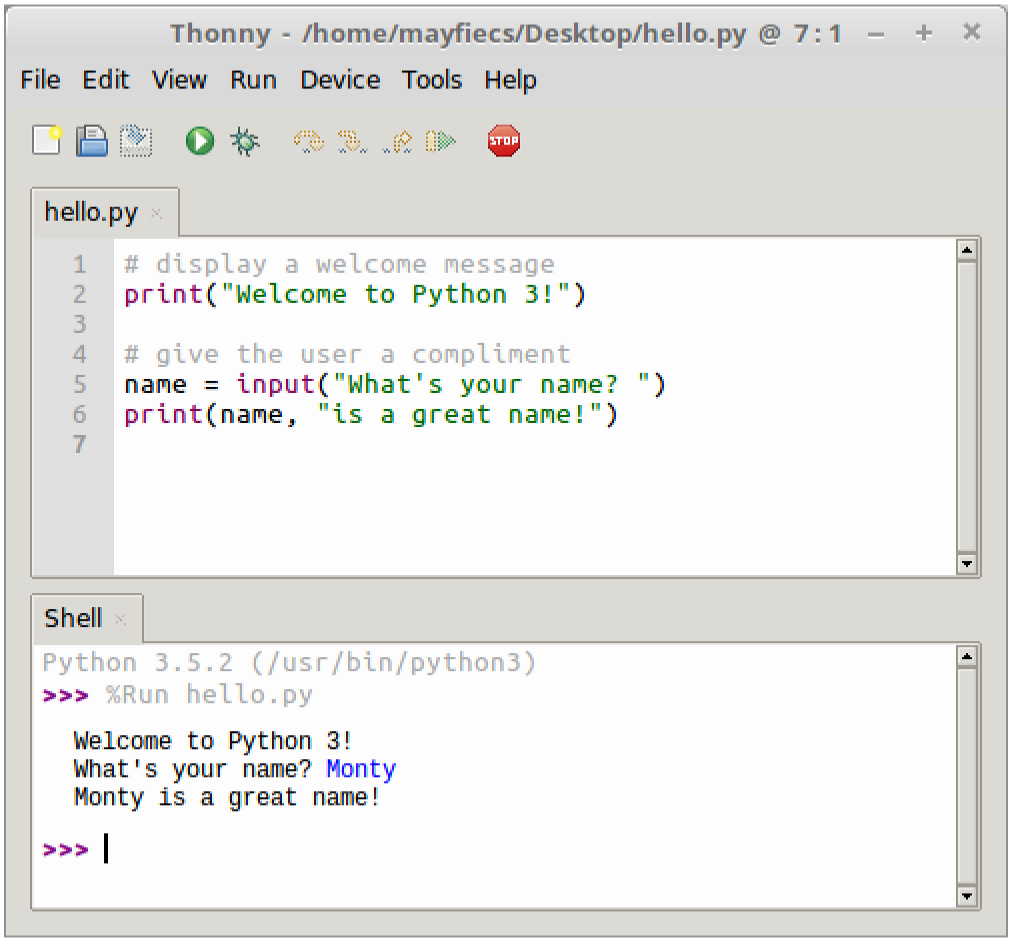
During the activity, students should make progress toward:

- Leveraging prior knowledge and experience of other students. (Teamwork)

Model 1 Editor and Shell

Thonny is an ***integrated development environment*** (IDE) for Python designed for learning and teaching programming. Thonny is freely available at https://thonny.org/.

Do not run Thonny yet! Answer the questions first!


Thonny - /home/mayfiecs/Desktop/hello.py @ 7:1
File Edit View Run Device Tools Help
hello.py
Shell

```
# display a welcome message
print("Welcome to Python 3!")

# give the user a compliment
name = input("What's your name? ")
print(name, "is a great name!")

```

```
Python 3.5.2 (/usr/bin/python3)
>>> %Run hello.py

  Welcome to Python 3!
  What's your name? Monty
  Monty is a great name!

>>> 
```

Questions (15 min) **Start time:**

1. Based on the screenshot in Model 1:

a) where is the ***Shell*** window?

b) where is the ***Editor*** window?

c) what is the name of the file in the Editor?

d) what is the directory where this file is located?

2. Identify the number of lines corresponding to:

a) the program (in the Editor)

b) the output of the program

3. What is the symbol at the start of a line of program text not displayed as output?

4. Consider the three program lines (in the Editor) that do not include text displayed as output. Describe what might be the purpose of:

a) a ***comment*** line (starts with a pound sign: #)

b) a blank line

Now open Thonny on your computer, type the code shown in Model 1, save the file as hello.py, and run the program. Ask for help if you get stuck!

5. What was required before the third line of the program output was displayed?

6. In the Shell window, what is the color of:

a) the program's output?

b) the user's input?

7. Based on your experience so far, what is the difference between the text in the Editor window and the text in the Shell window?

8. Describe what appears to be the purpose of each line of Python code in the Editor window.

   a) line 1:

   b) line 2:

   c) line 3:

   d) line 4:

   e) line 5:

   f) line 6:

Model 2 Python Built-In Functions

You can use ***functions*** to perform specific operations. Some functions require values, known as ***arguments***, to perform their operation. Functions may also ***return*** a value. For example:

```
name = input("What's your name? ")
```

`input` is a function, `"What's your name? "` is an argument, and the return value (typed by the user) is stored in `name`.

The following table shows additional examples of functions. They were written by a scientist to set up an experiment.

Do not type anything yet! Read the questions first!

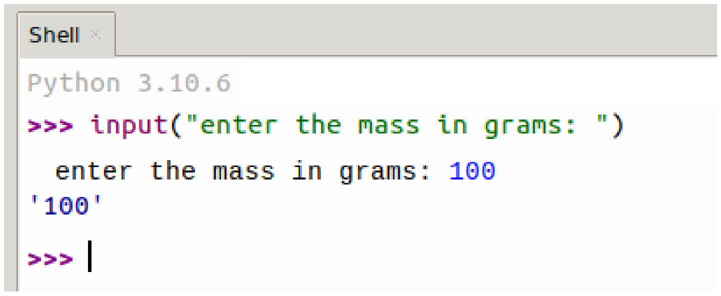
Python code	Shell output
`input("enter the mass in grams:  ")`	
`mass = input("enter another mass in grams:  ")`	
`mass`	
`unit = input("enter the units for mass:  ")`	
`print(mass, unit)`	
`print(mass / 2)`	
`ten = 10`	
`print(ten / 2)`	
`abs(-1)`	
`abs(-1 * ten)`	

Questions (15 min) Start time:

9. List the names of the three functions used in Model 2.

10. What are the arguments of the first use of the `print` function?

11. Type the first three lines of code in a Python **Shell**, one line at a time, and write the resulting output in the right column of the table. When prompted, enter a number of your choice:



```
Python 3.10.6
>>> input("enter the mass in grams: ")
  enter the mass in grams: 100
'100'
>>> |
```

12. Type the remaining lines of code in a Python **Shell**, one line at a time, and write the resulting output (if observed) in the right column of the table. If an error occurs, write what type of error it was (i.e., the first word of the last line of the error message).

13. Which function delayed execution until additional input was entered?

14. Which term, ***user*** or ***programmer***, best defines the role of the person who entered the additional input? Explain.

15. Based on the Shell output, what does the word `mass` represent, and how did it get its value?

16. What does the word `ten` represent, and how did it get its value?

17. Do the values of `mass` and `ten` both represent a number? Explain why or why not.

Model 3 Variables and Assignment

In programming, an ***assignment statement*** sets or updates the value of a ***variable***. The variable is set to the value after the ***assignment operator*** (=). Choosing short yet descriptive variable names is considered good programming style and will make your programs easier to read.

Do not type anything yet! Read the questions first!

Python code	**Shell output**
`data = 12`	
`data`	
`Data`	
`Data = 34`	
`data`	
`Data`	
`3data = "hello"`	
`data3 = "world"`	
`data3 = hello`	
`hot = 273 + 100`	
`273 + 100 = hot`	
`hot`	
`hot - 100`	

Questions (15 min) Start time:

18. Pick one assignment statement from the table above, and identify the following:

a) the variable being assigned

b) the assignment operator

c) the value of the variable immediately after the assignment

19. Similar to Model 2, type each line of code in a Python Shell and write the resulting output in the space provided. If an error occurs, write what type of error. Place an asterisk (*) next to any output for which you were surprised.

20. Circle each *successful* assignment statement in Model 3. How many are there?

21. What is the observed output of a successful assignment statement?

22. After the successful execution of an assignment statement, how can you confirm the value of this variable?

23. For each assignment statement that executed without an error, write the corresponding variable name.

24. Based on the Model 3 output, indicate whether each statement below is true or false.

a) Variable names in Python can start with a number.

b) Variable names in Python must start with a lower-case letter.

c) Variable names in Python are case-sensitive.

25. Each of the following assignment statements has an error. Write a valid line of Python code that corrects the assignment statement. Test your code on a computer in a Python Shell.

a) `3 + 4 = answer`

b) `oh well = 3 + 4`

c) `2x = 7`

26. Predict the value of the variable `hot` after executing all lines of code in Model 3. Then test your prediction on a computer, and explain the result.

27. Write a line of Python code to assign the current value of `hot` to the variable `temp`. Show output that confirms that you have done this correctly, and explain the code.