Session Title: Input, Loops and Functions with Turtle
Session No: 2

Learning Objectives:

By the end of this session students will be able to:

- Construct programs that use user input
- Describe the purpose of Python modules
- Solve problems to create drawings using the Turtle module
- Modify duplicated code to use for Loops
- Construct reusable functions

Session Outline

Intro & Framing
5 mins

This session focuses on repetition and reusability in code. Students will be introduced to the following Python programming concepts:

- Use input
- Modules
- For loops
- Functions

Several of the exercises require the students to create basic shapes and drawings using the Python turtle module in order to develop their problem solving skills.

Block #1: User Input
5 minutes

The first block introduces students to the input() function. Time should be taken to show students how to input values with PyCharm.

5 minutes

The second half of the block introduces the int() function which can be used to convert the string value entered in input() to an integer. When demonstrating the example, show what happens when you enter text instead of a number and explain the error message.

Relevant Exercise(s):

Exercise 2.1: 5 minutes

- In this exercise students will practice using the input() function
- They need to come up with their own questions and use string formatting from the first session to format the output

- Students should be encouraged to adapt the example shown in the exercise description

Exercise 2.2: 5 minutes

- This exercise introduces the use of int() when used in combination with input()
- Students should use the code in the exercise as a starting point for their solutions

Block #2: Python Modules
3 minutes

This block explains the purpose of Python modules i.e. reusable code that someone else has written that you can use. This lesson uses the turtle module in the standard library. The Python Package Index (PyPI) is introduced in a later session.

Block #3: Problem Solving (with Turtle)
5 minutes

This block introduces drawing with the turtle module. From this block students should start to understand how parts of programs can be repeated and reordered to solve problems.

The students should follow along as the instructor demonstrates how to draw a square with the turtle module. By rewriting the square program to use variables students are reminded that data values and variables are interchangeable.

Tell the students not to name any of their files “turtle.py” as this will cause an import error when they try to use the turtle module.

Relevant Exercise(s):

Exercise 2.3: 10 minutes

- This exercise helps develop the students' ability to adapt an existing piece of code to solve a different problem
- The square program can be used as a starting point for students
- Extension 1: Draw an octagon instead of square
- Extension 2: Draw a circle (it is unlikely that the students will complete this one, it is instead meant to setup them up to understand why repetition with for loops is so useful later in the session)

Block #4: For Loops
10 minutes

This block introduces for loops. The repeated code in the square example is replaced with a for loop in order to help the students understand how for loops help with repetition

Relevant Exercise(s):

Exercise 2.4: 10 minutes

- In this exercise students will finish the code included on the slide in order to draw a shape with any number of sides
- Extension: Draw a spiral. Students will need to work out that the values provided by range() to the for loop can be used to set the length of each line in the spiral
- Extension: Write code that draws a circle using a for loop

Block #5: Functions
5 minutes

This block introduces students to writing their own functions. So far students have used pre-written functions like `print()` and `str()`. Emphasis for this block is that functions can be reused and that functions also make the purpose of blocks of code easy to understand.

Relevant Exercise(s):

Exercise 2.5: 5 minutes

- Students will rewrite their triangle code as a function
- Extension: Write a function that draws a circle

Block #6:
5 minutes

This block builds on the previous one to introduce function arguments.

Relevant Exercise(s):

Exercise 2.6: 5-10 minutes

- Students will adapt the triangle function that they created in the previous exercise to include an argument that sets the length of each side of the triangle
- They should call the function multiple times with different arguments
- Extension: Add a second argument to set the colour of each triangle

Block #7: Returning Values from Functions

5 minutes

In this block students will learn how to return values from functions. Instructors should point out that like variables, functions that return values are interchangeable with any data value or variable.

Relevant Exercise(s):

Exercise 2.7: 5-10 minutes

- In this exercise students will need to add the return operator to the function to return the value calculated by the function

Recap & Closing
Time: 5 mins

Recap questions:
Question 1:
What is a Python module?

Answer:
A Python module is a collection of reusable code that has been written by someone else that you can use in your program

Question 2:
What is more suitable name for this function?

```
def x(days):
   minutes = days * 24 * 60
   return minutes

print(x(10))
```

Answer:
Anything along the lines of calculate_minutes(), hours_to_minutes() etc.

Question 3:
Why won't this program run?

```
for number in range(100)
print(number)
```

Answer:
Missing a colon and second line is not indented

Homework Tasks

Learning Task:

Session 2 homework questions in the student guide

Guide for Instructors

General comments

Make sure you can run the example code before the start of the session. You may have to install Tkinter beforehand.

Students may get an import error when using turtle. The most likely cause of this is that they’ve named one of their python files “turtle.py”. To fix this, rename the file named “turtle.py” to something else.