Multiple Modules

Python comes with an extensive library of built-in modules for accomplishing everyday tasks. With just a few lines of code, you can generate random numbers, draw graphics, send emails, access websites, and more!

Source files:
move.py
stop.py
draw.py

Also for Unit 2 (https://docs.python.org/3/library/turtle.html).

Unit 1 Importing a Module

A. Create a new file `move.py`, and enter the code:

```
1  import random
2
3  def angle():
4      number = random.randint(-90, 90)
5      return number
6
7  print("in move: __name__ ==", __name__)
8  print("will always execute: angle ==", angle())
9
10 if __name__ == "__main__":
11     print("only if True: angle ==", angle())
```

Run `move.py`, and record the output below.

Output Line 1	
Output Line 2	
Output Line 3	

B. Create a new file `stop.py` (in the same folder), and enter the code:

```
1  import move
2
3  print("in stop: __name__ ==", __name__)
4  print("from module: angle ==", move.angle())
```

Run `stop.py`, and record the output below. Draw an arrow from each line of output to its corresponding print statement in the code.

`Output Line 1`	
`Output Line 2`	
`Output Line 3`	
`Output Line 4`	

Questions

1. Upon execution of `move.py`:

a) what is the value of the variable `__name__`?

b) does the output correspond solely to the print statements contained in this file?

2. Upon execution of `stop.py`:

a) what is the value of the variable `__name__` from the print statement in `move`

b) what is the value of the variable `__name__` from the print statement in `stop`

c) does the output correspond solely to the print statements contained in this file?

3. What was the reason to include the `import move` statement in `stop.py`?

4. Based on the output of `stop.py`, describe what happens (as a side effect) when another module is imported.

5. What line in `move.py` did not print when `stop.py` was executed? Why?

6. In order for the output of `stop.py` to correspond solely to the print statements contained in `stop.py`, what modifications need to be made to `move.py`?

7. Describe what code in general to include inside `if __name__ == "__main__"`, and why.

Unit 2 Turtle Graphics

The `turtle` module can be used to create graphics. Create a new file `draw.py` (in the same folder), and enter the following code. Run the program and see what happens.

```
1  import move
2  import turtle
3
4  def randomwalk(steps):
5      turtle.shape("turtle")
6      turtle.color("green")
7      for i in range(steps):
8          turtle.left(move.angle())
9          turtle.forward(10)
10     turtle.bye()
11
12 if __name__ == "__main__":
13     randomwalk(100)
```

Questions

8. For each outcome, describe the type of edit necessary to `draw.py` and `move.py`:

a) a blue turtle

b) a longer simulation

c) a smaller range of angles (e.g., -45 to 45) that define the direction of the turtle

d) a random range of integers (e.g., 10 to 20) that define the length of a turtle move

9. Describe the type of edit necessary to produce the same outcome in Question #8d if the argument of `forward` is `move.length()` instead of 10:

10. Go to https://docs.python.org and click the modules link in the upper right corner. Find at least two built-in modules that interest you, and summarize what functions they provide.