kids code
jeunesse

Python on Trinket – Lesson 3

Review

- turtle library
- comments
- error handling

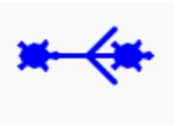
Lesson Outline

- Lead the class through program creation, starting with a basic snowflake, then introducing for loops, variables, and random integers
    1. Snowflake arm
    2. Full snowflake
        - Demonstrate changes to improve the program: add for loops, random colours, rotation, variables
    3. Have students utilize loops and variables to create custom shapes.

Lesson

Initialize program
`import turtle as t`
`import random`
`t.shape(“turtle”)`
`t.speed(speed)`

1. Snowflake Arm	

`t.forward(50)`
`t.stamp()`
`t.backward(20)`
`t.left(45)`
`t.forward(20)`
`t.backward(20)`
`t.right(90)`
`t.forward(20)`
`t.backward(20)`
`t.left(45)`
`t.backward(30)`
	

Demonstrate the following changes we can make to improve our program functionality:

- Introduce for loop structure and variables (so students can create a snowflake with multiple arms)
    - Loops allow us to repeatedly run lines of code a certain number of times (For a snowflake, we would not want to rewrite all the lines of code it takes to create an arm, for every arm. Using a for loop, we can specify the number of times we want the arm code to run).

    - For any value we use more than once, saving the value in a variable allows us to use this value as needed by referencing the variable name. If we want to change the value, we must only modify the one line of code where we initialize the variable (as opposed to changing the value every time it is used).

```
for count in range(10):
    lines of code...
    lines of code...
    t.left(360/10)
```

If we decide we want 15 arms instead of 10, we'd have to find every time we use the number 10 in our program to change it. Or, we can create a variable to store the number of arms…

```
num_arms = 10
for count in range(num_arms):
    lines of code...
    lines of code...
    t.left(360/num_arms)
```

```
num_arms = 15    # only one line changes
for count in range(num_arms):
    lines of code...
    lines of code...
    t.left(360/num_arms)
```

- for count in range (10) **vs.** for count in range (num_arms)

  - t.left(360/10) **vs.** t.left(360/num_arms)

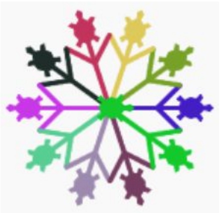
- random integers (for random colors on each arm)

```
‘‘‘ Get a random value for red, green, and blue within the
numeric range of colours (0-255) ’’’
# t.color(r, g, b)
t.color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
```

- Angles – after creating an arm, turn (360˚ / #arms) for an equal distance between all snowflake arms.

2a. Snowflake (+ loops, random values, rotation)

```
for count in range(10):
    t.color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    t.forward(50)
    t.stamp()
    t.backward(20)
    t.left(45)
    t.forward(20)
    t.backward(20)
    t.right(90)
    t.forward(20)
    t.backward(20)
    t.left(45)
    t.backward(30)
    t.left(36)
```



2b. Snowflake (+ variables)

```
size = 10
for count in range(size):
    t.color(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255))
    t.forward(50)
    t.stamp()
    t.backward(20)
    t.left(45)
    t.forward(20)
    t.backward(20)
    t.right(90)
    t.forward(20)
    t.backward(20)
    t.left(45)
    t.backward(30)
    t.left(360/size)
```