Programming Fundamentals

Program Flow with Turtles

João Correia Lopes

INESC TEC, FEUP

02 October 2018

GOALS

By the end of this class, the student should be able to:

- Describe how to import and do simple graphics with the module “turtle”
- Describe an instance of Turtle, its own attributes and methods
- Describe the flow of execution of the for loop
- Describe the range function

BIBLIOGRAPHY

- Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers, How to Think Like a Computer Scientist — Learning with Python 3, 2018 (Section 3.1) [PDF]
- Brad Miller and David Ranum, Learning with Python: Interactive Edition. Based on material by Jeffrey Elkner, Allen B. Downey, and Chris Meyers (Chapter 4) [HTML]

TIPS

- There's no slides: we use a script and some illustrations in the class. That is NOT a replacement for **reading the bibliography** listed in the *class sheet*
- "Students are responsible for anything that transpires during a class—therefore **if you're not in a class**, you should get notes from someone else (not the instructor)"—David Mayer
- The best thing to do is to **read carefully** and **understand** the documentation published in the Content wiki (or else **ask** in the class)
- We will be using **Moodle** as the primary means of communication

CONTENTS

Python Modules

- There are many modules in Python that provide very powerful features that we can use in our own programs:
  - to do maths
  - to send email
  - to fetch web pages
  - . . . and many others
- With `turtle` one creates turtles and get them to draw shapes and patterns
- . . . but the aim is to develop the theme: “computational thinking”

Simple graphics

Every window contains a *canvas*, which is the area inside the window on which we can draw

```python
import turtle                # Allows us to use turtles

window = turtle.Screen()     # Creates a playground for turtles
alex = turtle.Turtle()       # Create a turtle, assign to alex

alex.forward(50)             # Tell alex to move forward by 50 units
alex.left(90)                # Tell alex to turn by 90 degrees
alex.forward(30)             # Complete the second side of a rectangle

window.mainloop()            # Wait for user to close window
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/turtles.py

Simple graphics (2)

```python
import turtle

window = turtle.Screen()

window.bgcolor("lightgreen")   # Set the window background color
window.title("Hello, Tess!")   # Set the window title

tess = turtle.Turtle()
tess.color("blue")          # Tell tess to change her color
tess.pensize(3)             # Tell tess to set her pen width

tess.forward(50)
tess.left(120)
tess.forward(50)

window.mainloop()
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/turtles.py

Instances

From a *class* (Turtle) one may have many *objects* (**instances** of Turtle);
Each instance has its own **state** and **behaviour**

```
import turtle

window = turtle.Screen()    # Set up the window and its attributes
window.bgcolor("lightgreen")
window.title("Tess & Alex")

tess = turtle.Turtle()      # Create tess and set some attributes
tess.color("hotpink");
tess.pensize(5)

alex = turtle.Turtle()      # Create alex
...
```

Instances (2)

```
...
tess.forward(80)            # Make tess draw equilateral triangle
tess.left(120);
tess.forward(80);
tess.left(120);
tess.forward(80)
tess.left(120)              # Complete the triangle

tess.right(180)             # Turn tess around
tess.forward(80)            # Move her away from the origin

alex.forward(50)            # Make alex draw a square
alex.left(90)
alex.forward(50)
alex.left(90)
alex.forward(50)
alex.left(90)
alex.forward(50)
alex.left(90)

window.mainloop()
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/herd.py

FOR LOOP

- It is a basic building block of all programs to be able to *repeat* some code, over and over again.
- In computer science, we refer to this repetitive idea as *iteration*
- it has a *loop variable*, an indented *loop body*, and a terminating condition

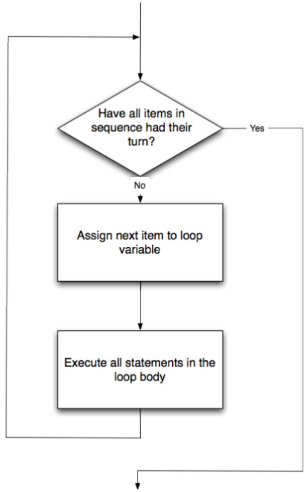
```
for friend in ["Joe", "Zoe", "Zuki", "Thandi", "Paris"]:
    invite = "Hi " + friend + ". Please come to my party!"
    print(invite)
# more code to follow
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/for.py

FLOW OF EXECUTION

- As a program executes, the interpreter always keeps track of which statement is about to be executed.
- We call this the **control flow**, of the **flow of execution** of the program.
- Control flow until now has been strictly top to bottom, one statement at a time. The `for` loop changes this.
- See it in `pythontutor.com`

Flow of Execution of the for loop


Have all items in sequence had their turn?
Yes
No
Assign next item to loop variable
Execute all statements in the loop body

FOR LOOP EXAMPLE 1

```
import turtle                # set up alex
wn = turtle.Screen()
alex = turtle.Turtle()

for i in [0, 1, 2, 3]:       # repeat four times
    alex.forward(50)
    alex.left(90)

wn.exitonclick()
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/for.py

FOR LOOP EXAMPLE 2

```python
# 1
for aColor in ["yellow", "red", "purple", "blue"]:  # repeat four times
    alex.color(aColor)
    alex.forward(50)
    alex.left(90)


# 2
colors = ["yellow", "red", "purple", "blue"]
for color in colors:                                # for each color
    alex.color(color)
    alex.forward(50)
    alex.left(90)
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/for.py

The range Function

- Python gives us special built-in range objects
- Computer scientists like to count from 0!
- The most general form of the range is `range(start, beyondLast, step)`

```
for i in range(4):
    # Executes the body with i = 0, then 1, then 2, then 3
    print(i)

for _ in range(10):
    # Sets x to each of ... [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    print(_)

for i in range(0, 20, 2):
    print(i)
```

Some more turtle methods

- `tess.left(-30)` / `tess.right(330)` ?
- `alex.backward(-100)` / `alex.forward(100)` ?
- `alex.penup()` and `alex.pendown()`
- `alex.shape("turtle")`
- `alex.speed(10)`

A final example

```python
import turtle

wn = turtle.Screen()
wn.bgcolor("lightgreen")

tess = turtle.Turtle()
tess.color("blue")
tess.shape("turtle")

print(list(range(5, 60, 2)))
tess.up()                           # this is new

for size in range(5, 60, 2):        # start with size = 5 and grow by 2
    tess.stamp()                    # leave an impression on the canvas
    tess.forward(size)              # move tess along
    tess.right(24)                  # and turn her

wn.exitonclick()
```

⇒ https://github.com/fpro-admin/lectures/blob/master/03/stamp.py

EXERCISES

- Moodle activity at: LE03: Program Flow with Turtles