Turtles, Loops, and Painting!

James

U. of West Georgia

2015

Outline

Code Website

All code presented in this talk is posted at this website.

`https://github.com/jcchurch/PythonUCode3/`

Adults: Go to this site for providing assistance.

Our First Python Program

```
name = input('What is your name?   ')
print('Hi, ', name)
```

What are functions?

- A **function** is a set of programming commands with a **name**.
- When we want to perform those commands, we just call that **name**.

Turtles

Drawing with Turtles: drawCircles1.py

```
import turtle

def makeTurtle(color1, color2):
    t = turtle.Turtle()
    t.shape('turtle')
    t.color(color1, color2)
    return t

window = turtle.Screen()
t = makeTurtle('green', 'yellow')

for i in range(4):
    t.circle(40)
    t.forward(100)

window.mainloop()
```

A loop repeats code.

```
for i in range(4):
    t.circle(40)
    t.forward(100)
```

- Loops repeat code.
- **for** means that we wish to loop.
- **i in range(4)** is an index variable must be in the range from 0 to 3.
- Computers always count beginning with 0.
- 40 is the size of each circle.
- 100 is the distance moved between each circle.

Change our code to this:

Change our loop to look like this:

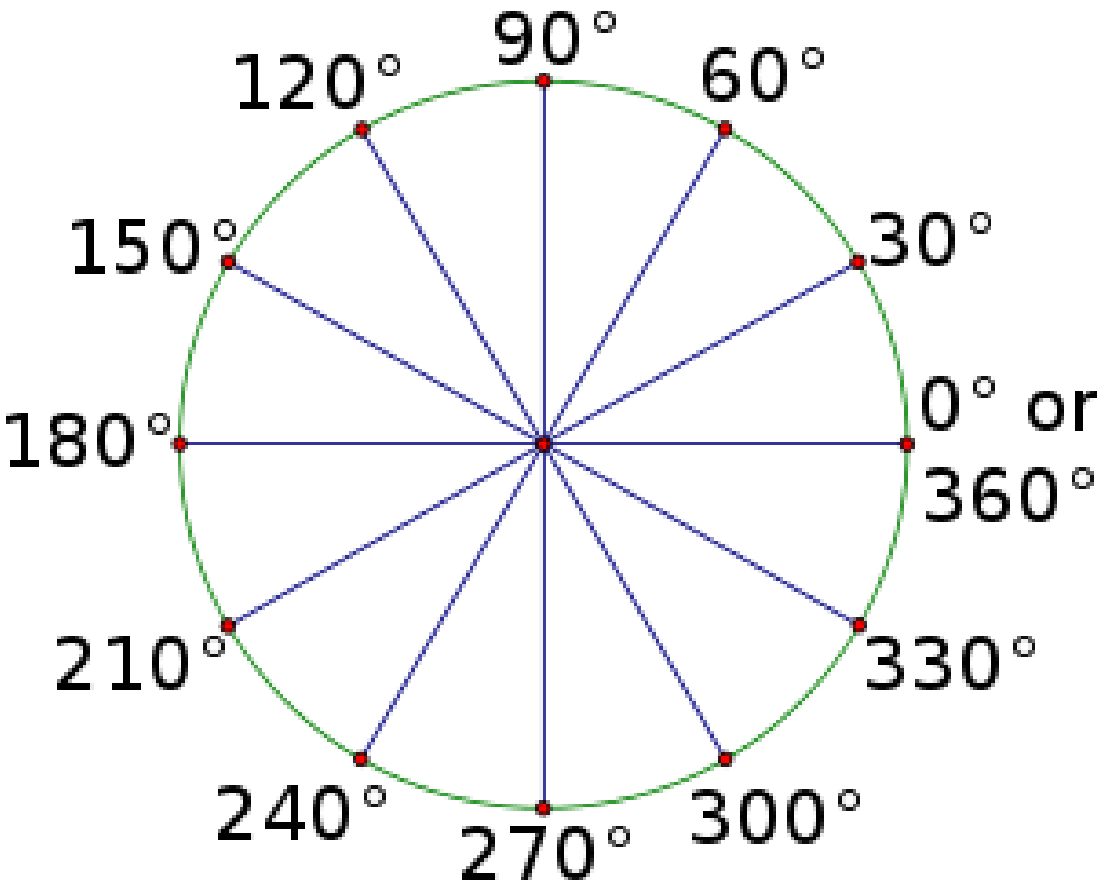
```
numberOfCircles = int(input('How many circles?   '))

for i in range(numberOfCircles):
    t.circle(40)
    t.forward(100)
```

Notice that we add 1 new line, replace "4" with "numberOfCircles".
This is in our folder as "drawCircles2.py".

Drawing with Turtles: drawCircles3.py

```
import turtle

def makeTurtle(color1, color2):
    t = turtle.Turtle()
    t.shape('turtle')
    t.color(color1, color2)
    return t

window = turtle.Screen()
t = makeTurtle('green', 'yellow')

for i in range(4):
    t.circle(40)
    t.left(90)

window.mainloop()
```

Degrees in a Circle



Every circle has 360 degrees.

Understanding our code

```
for i in range(4):
    t.circle(100)
    t.left(90)
```

- 4 is the number of circles.
- 100 is the size of each circle.
- 90 is the degrees between each circle.
- $4 \times 90 = 360$.
- Every circle has 360 degrees.

Change our code to this:

Change our loop to look like this:

```
numberOfCircles = int(input('How many circles?    '))

for i in range(numberOfCircles):
    t.circle(100)
    t.left(360 / numberOfCircles)
```

Notice that we add 1 new line, replace "4" with "numberOfCircles", and "90" with "360 / numberOfCircles".
This is in our folder as "drawCircles4.py".

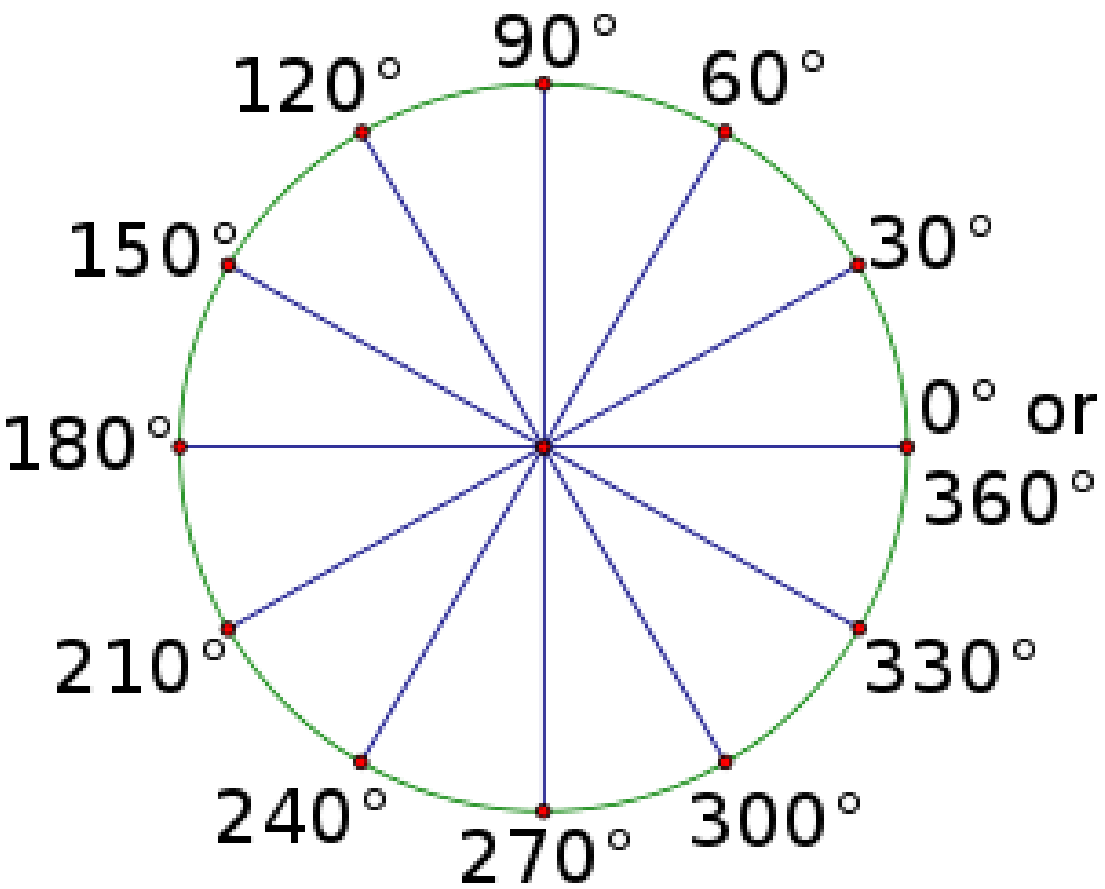
Making an Interactive Program

- We are going to make an art creation software application.
- You can show this to your friends.
- You can add new functionality.
- The only thing this program won't do is save.

onkey

- Every key on your keyboard has a name.
- We can associate each key with an action using the function "onkey".
- The method is "turtle.onkey(function, keyname)".
- 1 rule: the function cannot have parameters.
- Step 1: Make the function. Step 2: Make the onkey.
- Remember: we can't associate the same key with two functions.

Degrees in a Circle



Notice the headings on this circle: right=0, up=90, left=180, down=270.

Part 1

```
import turtle
import random

t = turtle.Turtle()
t.shape('turtle')
t.color('green', 'green')

def up(): head(90)
def right(): head(0)
def left(): head(180)
def down(): head(270)

def head(x):
    t.seth(x)
    t.forward(2)
```

Part 2

```
turtle.onkey(up, 'Up')
turtle.onkey(right, 'Right')
turtle.onkey(left, 'Left')
turtle.onkey(down, 'Down')

turtle.listen()
turtle.mainloop()
```

Save this program as "etch.py". Play with it! Use the up, down, left, and right keys to move the turtle. (This program is "etch1.py" in our folder.)

Flipping the Turtle's Tail

Let's add a new function to flip the turtle's tail if we hit the space bar. In other words, if the turtle is drawing, stop drawing. But if the turtle is not drawing, start drawing again. This program is "etch2.py" in our folder.

```
def flipTail():
    if t.isdown():
        t.up()
    else:
        t.down()
```

In our "onkey" section:

```
turtle.onkey(flipTail, 'space')
```

Understanding our code

```
if t.isdown():
    t.up()
else:
    t.down()
```

- This is an **if statement**. **if statements** do something if the answer to a question is right or something else if the answer to the question isn't right.
- Here, our question is "Is turtle t's tail down?". If it is right, we put the tail up. If it is not right, we put the tail down.
- Again, go play!

Changing colors!

Can we change the pen color of our turtle while we draw? Yes! Add these new functions to your program. You can also make more colors if you remember the color names.

```
def yellow(): t.pencolor('yellow')
def green(): t.pencolor('green')
def red(): t.pencolor('red')
def blue(): t.pencolor('blue')
```

Also, we can't forget the key presses. Now, "r" is red and "b" is blue!

```
turtle.onkey(yellow, 'y')
turtle.onkey(red, 'r')
turtle.onkey(blue, 'b')
turtle.onkey(green, 'g')
```

This is program "etch3.py" in our folder.

Adding Shapes!

Shapes? Did someone say shapes? (If you didn't, I still had this slide ready.) Make "s" draw a square:

```
def square():
    t.down()
    for i in range(4):
        t.forward(100)
        t.left(90)
```

Also:

```
turtle.onkey(square, 's')
```

This is found in "etchFinal.py" in our folder.

Draw a triangle!

Let's make "t" draw a triangle!

```
def triangle():
    t.down()
    for i in range(3):
        t.forward(100)
        t.left(120)
```

Also:

```
turtle.onkey(triangle, 't')
```

See if you can figure out how to make "c" draw a circle.

Play Time!

- Let's continue to play and add onto our art program!