CS 100: Python Drawings with Conditionals, Iteration, and Functions

Chris Kauffman

Week 5-1

Logistics

Homework 3

- Due Thursday
- Can work with partner
- Submit both Word Doc/PDF AND Python code
- Questions?

Reading

- Pattern Ch 4
- Think: Python Range Function (Ch 4)
- Think: If/Else for Even/Odd Iterations (Ch 7)

Goals Today

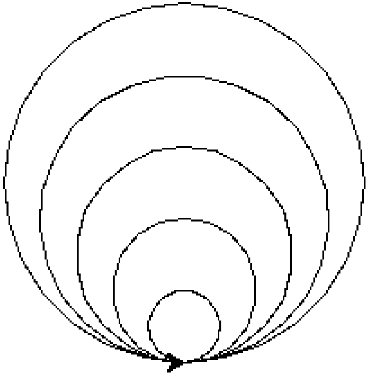
- Alternating in Loops for Drawing
- Drawing Exercises

Exercise: Draw Circles

`def draw_circles(layers):`

- Draws concentric circles
- Each differs in radius by 20 pixels
- Parameter `layers` is how many circles to draw
- Use the `circle(size)` turtle function
- Use a `for` loop

`draw_circles(5)`



Solution: Draw Circles

Solution 1

```
def draw_circles(layers):
  size = 20
  for i in range(layers):
    circle(size)
    size = size + 20

draw_circles(5)
```

Solution 2

```
def draw_circles(layers):
  for i in range(layers):
    size = (i+1) * 20
    circle(size)

draw_circles(5)
```

Python Conditionals

```
myVar = 7                               # Assign a variable
if(myVar == 5):                         # Check something
    print("It’s five");
else:
    print("It’s not five");

for i in range(10):
    if i == 7:
        print("Lucky!")
    else:
        print("Boring")
```

- Using == allows one to check whether a variable is equal to a number
- An `if/else` statement allows conditional execution

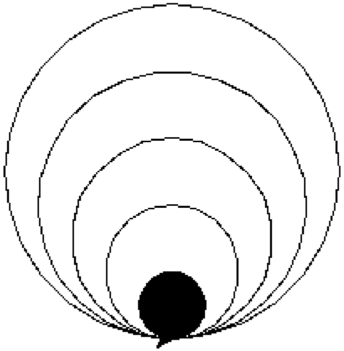
The Eye

```
def the_eye(layers):
```

- Similar to `draw_circles(layers)`
- Only on the first iteration, draw a filled circle
- Use an `if/else` for this
- Answer for `draw_circles(layers)`, a good place to start:

```
def draw_circles(layers):
  for i in range(layers):
    size = (i+1) * 20
    circle(size)
```

`the_eye(5)`



Solution: The Eye

```
def the_eye(layers):
  size = 20
  for i in range(layers):
    if i==0:
      color("black")
      begin_fill()
      circle(size)
      end_fill()
    else:
      circle(size)
    size = size+20

the_eye(5)
```

Alternating with Conditionals in Loops

```
# Print whether the numbers are odd or even
for i in range(10):
    if(i % 2 == 0):             # % is remainder op
        print(str(i) + " is Even")
    else:
        print(str(i) + " is Odd")
```

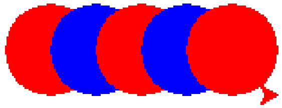
- Useful when you want to alternate drawing different colors
- Nesting and combining things is what makes programming interesting

Alternating Circles

```
def alt_circles(count,col1,col2):
  # your code here
```

- Draws a sequence of circles
- Each circle has size 25
- Move to the right by 25 pixels
- Notice the overlap: later circles go on top of earlier circles

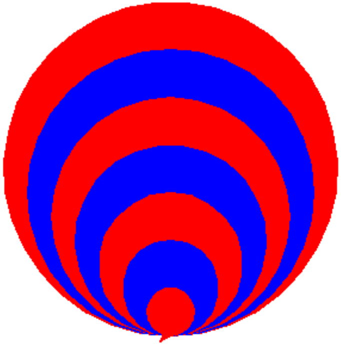
```
alt_circles(5,"red","blue")
```



Goal: Magic Eye

```
def magic_eye(layers,col1,col2):
  # your code here
```

```
magic_eye(7,"blue","red")
```



- Concentric circles
- Alternating colors
- Expect Problems: big circles are later, overwrite little circles

Printing sequences with loops

From HW3 Knowledge, how would you print the following sequences of numbers easily with a loop?

Seq 1: 0, 1, 2, 3, 4
Seq 2: 3, 4, 5, 6, 7
Seq 3: 0, 2, 4, 6, 8
Seq 4: 4, 3, 2, 1, 0
Seq 5: 8, 6, 4, 2, 0

Range Variants

range() can generate many kinds of sequences aside from from 0, 1, 2, . . .

range(start,stop)

```
for i in range(0,5):
  print(i)
# 0, 1, 2, 3, 4

for i in range(3,8):
  print(i)
# 3, 4, 5, 6, 7
```

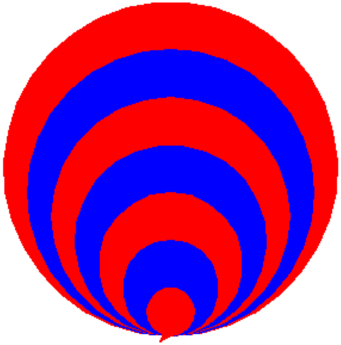
range(start,stop,change)

```
for i in range(0,10,2):
  print(i)
# 0, 2, 4, 6, 8

for i in range(4,-1,-1):
  print(i)
# 4, 3, 2, 1, 0 -- stop before -1

for i in range(4,0,-1):
  print(i)
# 4, 3, 2, 1    -- stop before 0

for i in range(8,-1,-2):
  print(i)
# 8, 6, 4, 2, 0 -- stop before -1
```

The Big One: Magic Eye

```
def magic_eye(layers,col1,col2):
  # your code here
```

`magic_eye(7,"blue","red")`



- Concentric circles
- Alternating colors
- Use `range(layers,0,-1)` for loop

Solution

```
def magic_eye(layers,col1,col2):
  for i in range(layers,0,-1):
    size = i * 20
    if i % 2 == 0:
      color(col1)
    else:
      color(col2)
    begin_fill()
    circle(size)
    end_fill()

magic_eye(7,"blue","red")
```