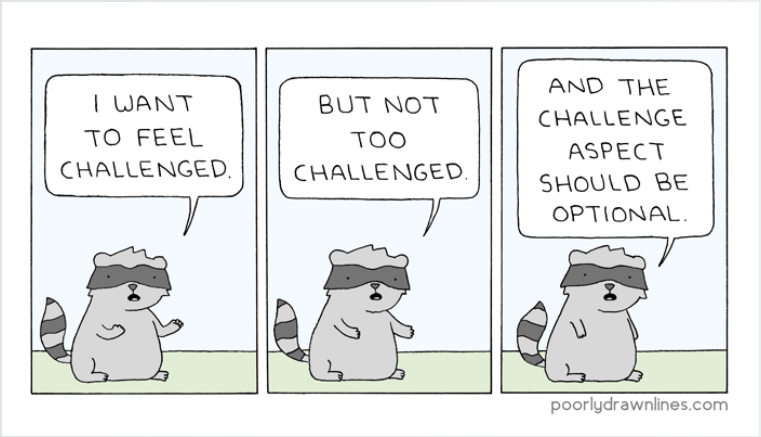


Loops

CS 8: Introduction to Computer Science, Winter 2018
Lecture #5

Ziad Matni
Dept. of Computer Science, UCSB

Administrative

- No more adds to this class: list is closed

Lecture Outline

- Reviewing basic functions
- Flow Control: If-Else Statements
- Loops: For Statements

Yellow Band = Class Demonstration! ☺

Class Exercise

Get together with 2 or 3 other people around you and answer this question. You can use your notes from last time:

a) Write a short Python code that asks a user their age. Once you do that, decide whether to print out "**Your age is an even number!**" or "**Your age is an odd number!**" depending on their answer.

b) Now modify your code so that it can detect if someone entered a number less than 1 as their age. If so, print out a rejection message and quit. Challenge: do this twice: once by using the **and** operator and once without using **and**!

Class Exercise

```
age = int(input("How old are you? ”))

if (age % 2 == 0):
    print("Your age is an even number!")
else:
    print("Your age is an odd number!")
```

Class Exercise

```
age = int(input("How old are you? ”))

if (age % 2 == 0) and (age > 0):
    print("Your age is an even number!")
elif (age % 2 != 0) and (age > 0):
    print("Your age is an odd number!")
else:
    print("You have entered an illegal age!")
```

Class Exercise

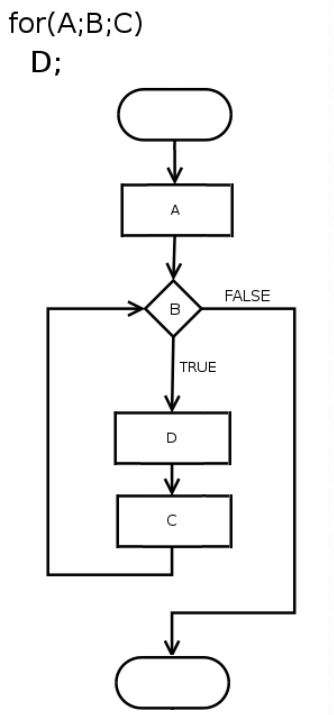
```
age = int(input("How old are you? ”))

if (age > 0):
    if (age % 2 == 0):
        print("Your age is an even number!")
    else:
        print("Your age is an odd number!")
else:
    print("You have entered an illegal age!")
```

Loops

- Sometimes we want to be able to **repeat** a part of the program a certain number of times
  - Called a “loop”

- A popular way to do this is with the `for` command.



Repetition with a `for` loop

- `for` *ref* `in` *a list*`:`
  `#` *block – ref refers to current object in list*
  `#` *note that the block is all indented*
  - `for`, `in`, `:` – mandatory parts
  - *ref* – a name for referring to objects in the list

- Example:

```
for numbers in (0, 1, 2, 3, 4, 5):
    print (numbers)
```

This will print out the numbers 1 thru 5 in sequence

Using `range` with `for` loops

- The `range()` built-in function provides a handy list
- Simplest use: `range(n)`
  - Creates a list with `n` items `[0, 1, 2, …n-1]`

- Example:

```
for numbers in range(6):
    print (numbers)
```

This will print out the numbers 1 thru 5 in sequence
(like the last example)

More `range` with `for` loops!

- You can also do a range with **start** & **stop** parameters.
- Example:

```
for numbers in range(5, 8):
    print (numbers)
```

This will print out the numbers 5 thru 7 (excludes 8) in sequence

- Or you can have **start**, **stop** *and* **step** parameters.
- Example:

```
for i in range(1, 11, 4):
     print(i)
```

This will print out the numbers 1, then 5, then 9

Let's try these out!

Live Examples of For Loops

Let's try a bunch of these out!

REMEMBER:

Code we use in class is provided to you on the class website

Simpler Drawing By Repetition

- Listing 1.3 from the text (p. 34)

```
def drawSquare2(myTurtle, sideLength):
    for i in range(4):
        myTurtle.forward(sideLength)
        myTurtle.right(90)
```

- Small variation draws a spiral (Listing 1.4)

```
def drawSpiral(myTurtle, maxSide):
    for sideLength in range(1, maxSide+1, 5):
        myTurtle.forward(sideLength)
        myTurtle.right(90)
```

Let's try these out!

More Drawing Abstraction

- Contrast – a triangle vs. a square (Listing 1.5)

```
def drawTriangle(myTurtle, sideLength):
    for i in range(3):        # draw 3 sides, not 4
        myTurtle.forward(sideLength)
        myTurtle.right(120)  # 120°× 3
```

- Hmm…any regular polygon? (Listing 1.6, p. 38)

```
def drawPolygon(myTurtle,sideLength,numSides):
    turnAngle = 360 / numSides
    for i in range(numSides):
        myTurtle.forward(sideLength)
        myTurtle.right(turnAngle)
```

Let's try these out!

Problem Solving:
Draw A Circle With a Given Radius as a Polygon!

- *Notice: a polygon with many sides looks like a circle*
  - But how many sides to draw?
  - And how long should each side be?

- *Start simple*: decide to draw 360 sides every time
  - Think: *length of 1 side = circumference / 360*

- And remember from math that circumference equals $2\pi r$

Draw A Circle With a Given Radius as a Polygon!

- Put it all together: Listing 1.7 from the text (p. 40)

```
def drawCircle(myTurtle, radius):
    circumference = 2 * 3.1415 * radius
    sideLength = circumference / 360
    drawPolygon(myTurtle, sideLength, 360)
```

- The easy way to draw a circle in Turtle:

```
myTurtle = turtle.Turtle()
myTurtle.circle(100)  # draws a circle r = 100
# now he tells us…
```

YOUR TO-DOs

- ❑ Read **Chapter 3**
- ❑ Start **Homework2** (due next **Monday!**)
- ❑ Prepare for **Lab1** this week

- ❑ Be nice to others

</LECTURE>