LOOPS (CONT.)

LECTURE 04-1

JIM FIX, REED COLLEGE CSCI 121

UPCOMING COURSE EVENTS

- This coming Wednesday, 2/15, our first ***QUIZ***:
  - On Python scripting, conditional statements, and integer arithmetic.
  - 20 minutes; in-class; closed-note; written code.

LOOPS

- **Reading**: TP Ch 5, CP Ch 1.5
- A **`while`** statement can be used to repeat some code.
- The template below gives the syntax of a while loop statement:

```
lines of "set up" code to execute first
while condition-expression :
    lines of "loop body" code to execute if the condition holds
    ...
lines of "follow up" code to execute once the condition no longer holds
```

SIMPLE EXAMPLE

- This example script counts from 101 down to 1:

```
print("This program will count down by 10.")
count = 51
while count > 1:
    print(str(count) + "...")
    count = count - 10
print("1!!!!")
```

- Output of the script above:

```
51...
41...
31...
21...
11...
1!!!!
```

- **NOTE:** hit [CTRL-c] to terminate the Python script's execution.

SOME LOOP ISSUES TO COVER

- The **`while`** template and what it means.

- Definite versus indefinite loops.
  - **`countdown.py`**, **`guess.py`**, **`guess6.py`**

- Infinite loops happen.
  - Hit [CTRL-c] to terminate a runaway script.

- Using boolean conditions to control loops.

- Using **`break`** and **`continue`**.

- Nested loops.

COUNTING DOWN, GENERALIZED, GIVING PAUSE

- This example script counts from 101 down to 1:

```
print("This program will count down to 1 by an amount.")
start = int(input("Enter a value to start near: "))
decrement = int(input("Enter an amount to step down: "))
#
print("Ready? Counting down to 1:")
input("[Hit RETURN]")
#
count = start - ((start - 1) % decrement)
while count > 1:
    #
    print(str(count) + "...", end='')
    sys.stdout.flush()
    time.sleep(1)
    #
    count = count - decrement
#
print("1!!!!!")
```

GUESSING GAME

- This example script engages the user in a guessing game:

```
number = random.randint(1,100)
print("I have chosen a random number from 1 to 100.")
print("Try and guess what it is.")

guess = int(input("Your guess? "))
while guess != number:
    if guess > number:
        print("That guess was too high!")
    else:
        print("That guess was too low!")
    guess = int(input("What's your next guess? "))

print("You got it right! Great job.")
```

NESTING CONTROL STATEMENTS WITHIN A LOOP

- Of course you can put a conditional statement within a loop's body.

```
count = 0
while count < 6:
    if count % 2 == 0:
        print(str(count) + " is even.")
    else:
        print(str(count) + " is odd.")
    count = count + 1
print("Done.")
```

- Output of the script above:

```
0 is even.
1 is odd.
2 is even.
3 is odd.
4 is even.
5 is odd.
Done.
```

GUESSING GAME WITH 6 GUESSES

- This example script engages the user in a *more challenging* guessing game:

```
number = random.randint(1,100)
print("I have chosen a random number from 1 to 100.")
print("Try and guess what it is.")

guess = int(input("Your guess? "))
guesses = 1
while guesses < 6 and guess != number:
    if guess > number:
        print("That guess was too high!")
    else:
        print("That guess was too low!")
    guess = int(input("What's your next guess? "))
    guesses = guesses + 1

if guess == number:
    print("You got it right! Great job.")
else:
    print("Oh, so sorry. You ran out of guesses.")
    print("The number was "+str(number)+".")
```

PROJECT 1: GAME OF LIFE AND IMAGE PROCESSING

- Posted on the web at nchanath.github.io/csci121/source/_posts/project1.md
- It is a grid simulation.
- It is also an image processing platform.
- You'll write functions that compute a grid cell's value.
  - Based on its current value, from 0 to 100.
  - Based on its neighboring cell's values, also from 0 to 100.
- Applied successively over the entire grid, you obtain interesting behavior.

  (DEMO)
- Start looking at it!!! Play with the existing rule code.
- It's due ***Monday, October 3rd at 1pm***.

PROJECT 1 NEEDS TKINTER

- On some systems running Project 1 causes an error at the code line:

```
from tkinter import *
```

- This is the Python graphics library we use, and apparently isn't installed.

- For a Mac or a Windows machine :
  - Enter the Terminal command:

```
pip3 install tk
```

- For those few using WSL on Windows:

ADVANCED STUFF

  - Enter the terminal command:

```
sudo apt install python3-tk
```

  - Install a (free) tool called **MobaXterm**.
  - Run MobaXterm and create a *"New session..."* of type WSL.
  - Run the Grid program inside that terminal session