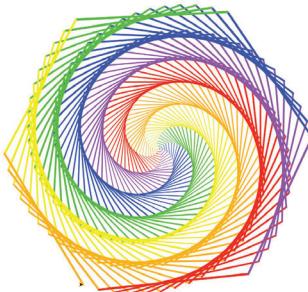


# CompSci 101

## Turtle, Bagels, Loop Tracing, Files

### Part 1 of 4



Susan Rodger  
Nicki Washington  
February 23, 2021

2/23/21

CompSci 101, Spring 2021

1

## I is for ...



- **Identity**
  - Who are you? Computer Science Student
- **Invariant**
  - Reasoning formally and informally about loops
- **Internet**
  - Network of networks
  - Far more than that!

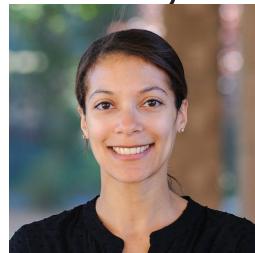
2/23/21

CompSci 101, Spring 2021

2

## Prof. Beth Trushkowsky

- Computer Science BS, Duke '07
- Computer Science MS '10, PhD '14, UC Berkeley
- Associate Professor, CS, Harvey Mudd College
- Her research involves leveraging human intelligence via crowdsourcing to create hybrid human/machine query processing systems.



2/23/21

CompSci 101, Spring 2021

3

## PFTD

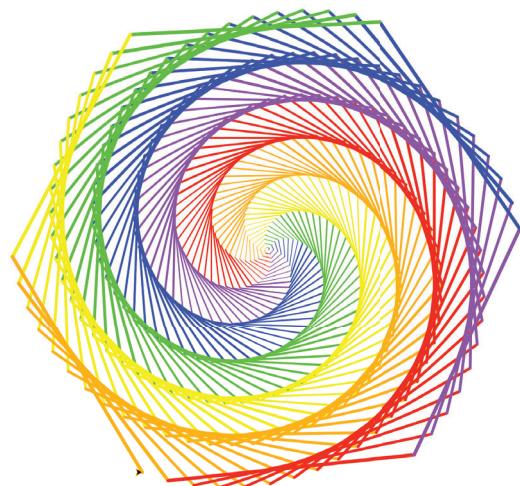
- Turtle
- Bagels APT
- Trace through loops
- Files

2/23/21

CompSci 101, Spring 2021

4

# Run Turtle, Run



2/23/21

CompSci 101, Spring 2021

5



# Turtle Programming

- Must:
  - Import turtle module
  - Create window/Screen
  - Last thing - exit on click
  - Create turtles to use, name/type/value
- Review **Turtle commands and concepts**
  - [http://bit.ly/turtle\\_tutorial](http://bit.ly/turtle_tutorial) for more, and book
- See **Snowpeople.py, ColorMyWorld.py, and Spiro.py** for some ideas
  - Color, Position, Leaving Turtle where started
  - Many more commands than this

2/23/21

CompSci 101, Spring 2021

6

## Put yourself in the turtle t...

```
t.forward(50)          # turtle moves forward  
                      # drawing a line  
  
t.left(90)            # turtle turns to its left  
t.pencolor("blue")    # change pen color  
t.forward(100)         # turtle moves forward  
                      # drawing line, new color
```

2/23/21

CompSci 101, Spring 2021

7

```
import turtle  
  
def drawPicture(turt):  
    t.forward(50)  
    t.left(90)  
    t.forward(80)  
    t.pencolor('red')  
    t.right(60)  
    t.forward(100)  
    t.pencolor('green')  
    t.left(60)  
    t.forward(50)  
    t.left(90)  
    t.forward(200)
```

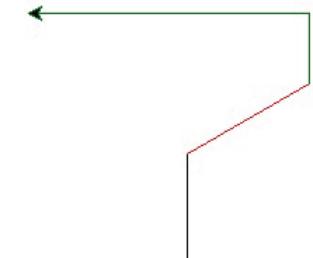
```
if __name__ == '__main__':  
    win = turtle.Screen()  
    t = turtle.Turtle()  
    drawPicture(t)  
    win.exitonclick()
```

2/23/21

CompSci 101, Spring 2021

8

## Example: simple.py



## Example: Simple.py parts

```
import turtle ← • Import at the top
```

```
)if __name__ == '__main__':
    win = turtle.Screen()
    t = turtle.Turtle()
    drawPicture(t)
)    win.exitonclick()
```

2/23/21

CompSci 101, Spring 2021 9

## Example: Simple.py parts

```
import turtle
```

```
|if __name__ == '__main__':
|    → win = turtle.Screen() • Create window
|        t = turtle.Turtle()
|        drawPicture(t)
|    win.exitonclick()
```

2/23/21

CompSci 101, Spring 2021 10

## Example: Simple.py parts

```
import turtle
```

```
)if __name__ == '__main__':
    win = turtle.Screen()
    → t = turtle.Turtle() • Create turtle
    drawPicture(t)
)    win.exitonclick()
```

2/23/21

CompSci 101, Spring 2021 11

## Example: Simple.py parts

```
import turtle
```

```
|if __name__ == '__main__':
|    win = turtle.Screen()
|        t = turtle.Turtle()
|        → drawPicture(t) • Call function to draw
|    win.exitonclick()
```

2/23/21

CompSci 101, Spring 2021 12

## Example: Simple.py parts

```
import turtle

if __name__ == '__main__':
    win = turtle.Screen()
    t = turtle.Turtle()
    drawPicture(t)
→ win.exitonclick()
```

- Close canvas when click on it

2/23/21

CompSci 101, Spring 2021 13

## Example: Simple.py DrawPicture

```
def drawPicture(turt):
    → turt.forward(50)
    turt.left(90)
    turt.forward(80)
    turt.pencolor('red')
    turt.right(60)
    turt.forward(100)
    turt.pencolor('green')
    turt.left(60)
    turt.forward(50)
    turt.left(90)
    turt.forward(200)
```



2/23/21

CompSci 101, Spring 2021 14

## Example: Simple.py DrawPicture

```
def drawPicture(turt):
    turt.forward(50)
    turt.left(90)
    turt.forward(80)
    turt.pencolor('red')
    turt.right(60)
    turt.forward(100)
    turt.pencolor('green')
    turt.left(60)
    turt.forward(50)
    turt.left(90)
    turt.forward(200)
```

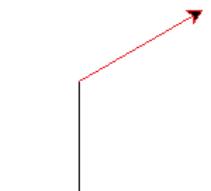


2/23/21

CompSci 101, Spring 2021 15

## Example: Simple.py DrawPicture

```
def drawPicture(turt):
    turt.forward(50)
    turt.left(90)
    turt.forward(80)
    turt.pencolor('red')
    turt.right(60)
    turt.forward(100)
    turt.pencolor('green')
    turt.left(60)
    turt.forward(50)
    turt.left(90)
    turt.forward(200)
```

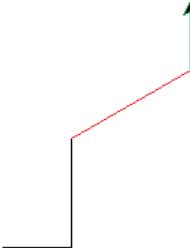


2/23/21

CompSci 101, Spring 2021 16

## Example: Simple.py DrawPicture

```
def drawPicture(turt):
    turt.forward(50)
    turt.left(90)
    turt.forward(80)
    turt.pencolor('red')
    turt.right(60)
    turt.forward(100)
    turt.pencolor('green')
    turt.left(60)
    turt.forward(50)
    turt.left(90)
    turt.forward(200)
```

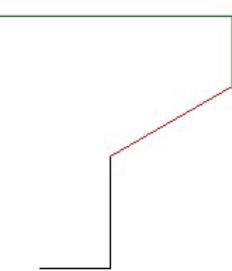


2/23/21

CompSci 101, Spring 2021 17

## Example: Simple.py DrawPicture

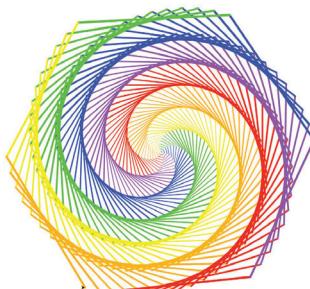
```
def drawPicture(turt):
    turt.forward(50)
    turt.left(90)
    turt.forward(80)
    turt.pencolor('red')
    turt.right(60)
    turt.forward(100)
    turt.pencolor('green')
    turt.left(60)
    turt.forward(50)
    turt.left(90)
    turt.forward(200)
```



2/23/21

CompSci 101, Spring 2021 18

## Compsci 101 Turtle, Bagels, Loop Tracing, Files Part 2 of 4



Susan Rodger  
Nicki Washington  
February 23, 2021

2/23/21

CompSci 101, Spring 2021 19

## What are key concepts in Spiro.py?

```
8 import turtle
9
10 def draw(turt):
11     colors = ['red', 'purple', 'blue', 'green', 'yellow', 'orange']
12     turt.speed(0)
13     for x in range(360):
14         turt.pencolor(colors[x % 6])
15         turt.width(x/100 + 1)
16         turt.forward(x)
17         turt.left(59)
18
19 if __name__ == '__main__':
20     win = turtle.Screen()
21     t = turtle.Turtle()
22     draw(t)
23     win.exitonclick()
```

Import turtle

1 – slowest  
10 – fastest  
0 – No animation

Create screen/window

Create turtle

Close on click

2/23/21

CompSci 101, Spring 2021 20

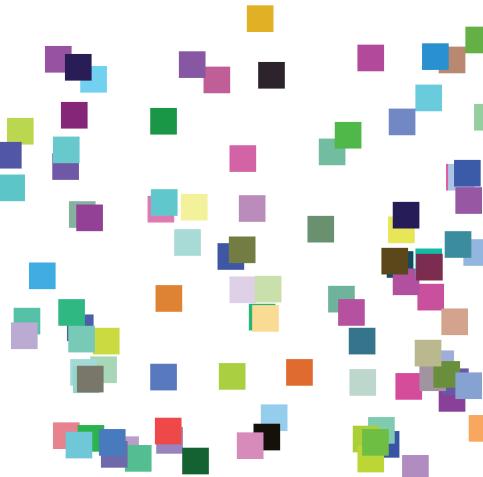
# Useful turtle functions

- **forward(n)/backward(n)** – move turtle n pixels
- **left(n)/right(n)** – turn turtle n degrees
- **pendown()/pendup()** – whether actually drawing
- **setposition(x, y)** – puts turtle in this (x,y) coordinate (a.k.a. **goto**, **setpos**)
- **sethead(n)** – points turtle in this direction (n=0 is east)
- Many more in documentation!
  - <https://docs.python.org/3/library/turtle.html>

2/23/21

CompSci 101, Spring 2021 21

# ColorMyWorld.py



2/23/21

CompSci 101, Spring 2021 22

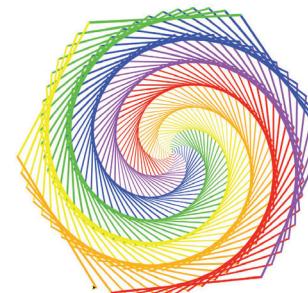
# Turtle Concepts

- Create a screen so you can ..
  - Exit On Click
  - Some other Screen Functions
- Create a turtle so you can ...
  - Move and draw using the turtle
- Drawing Concepts
  - Pen [up and down]
  - Fill
  - Color
  - Position

2/23/21

CompSci 101, Spring 2021 23

# CompSci 101 Turtle, Bagels, Loop Tracing, Files Part 3 of 4



Susan Rodger  
Nicki Washington  
February 23, 2021

2/23/21

CompSci 101, Spring 2021 24

# Code-Tracing a Loop

1. Find the changing variables/expressions
2. Create table, columns are variables/expressions
  1. First column is loop variable
  2. Add columns to help track everything else
3. Each row is an iteration of the loop
  1. Before execute code block, copy down each variable's value
  2. Execute code block, update a value in the row as it changes

2/23/21

CompSci 101, Spring 2021 25

# Code-Tracing a Loop

1. Find the changing variables
2. Create table, columns are the variables
  1. First column is loop variable
  2. Add columns to help track everything else

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i
    return idxMax
```

Loop variable

Other variable

Useful expression to track

2/23/21

CompSci 101, Spring 2021 27

# Code-Tracing a Loop

1. Find the changing variables/expressions
2. Create table, columns are variables/expressions
  1. First column is loop variable
  2. Add columns to help track everything else

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

What should be the table's columns?

2/23/21

CompSci 101, Spring 2021 26

# Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

mystery([2, 12, 4, 15, 15])

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
|   |        |             |        |                      |

2/23/21

CompSci 101, Spring 2021 28

## Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

#1

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

mystery([2, 12, 4, 15, 15])

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      |             |        |                      |
|   |        |             |        |                      |

2/23/21

CompSci 101, Spring 2021 29

## Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

mystery([2, 12, 4, 15, 15])

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
|   |        |             |        |                      |

2/23/21

CompSci 101, Spring 2021 30

## Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

#1

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

mystery([2, 12, 4, 15, 15])

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      |             |        |                      |

2/23/21

CompSci 101, Spring 2021 31

## Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax
```

mystery([2, 12, 4, 15, 15])

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |

2/23/21

CompSci 101, Spring 2021 32

## Fill in table

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

#2

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0 1    | 2           | 12     | True                 |

2/23/21

CompSci 101, Spring 2021 33

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0 1    | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |

2/23/21

CompSci 101, Spring 2021 35

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0 1    | 2           | 12     | True                 |
| 2 | 1      |             |        |                      |

2/23/21

CompSci 101, Spring 2021 34

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0 1    | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 1      |             |        |                      |

2/23/21

CompSci 101, Spring 2021 36

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 1      | 12          | 15     | True                 |

2/23/21

CompSci 101, Spring 2021

37

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

#2

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 1      | 12          | 15     | True                 |

2/23/21

CompSci 101, Spring 2021

38

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 1      | 12          | 15     | True                 |
| 4 | 3      |             |        |                      |

2/23/21

CompSci 101, Spring 2021

39

1. Before execute code block, copy down each variable's value
2. Execute code block, update a value in the row as it changes

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 1      | 12          | 15     | True                 |
| 4 | 3      | 15          | 15     | False                |

2/23/21

CompSci 101, Spring 2021

40

What is always true about the loop?

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 0      | 12          | 15     | True                 |
| 4 | 3      | 15          | 15     | False                |

2/23/21

CompSci 101, Spring 2021

41

What is always true about the loop?

1.  $\text{lst}[\text{idxMax}] \geq \text{lst}[k]$  for all  $k \leq i$

2.  $i < \text{len}(\text{lst})$

3.  $\text{idxMax} < \text{len}(\text{lst})$

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 0      | 12          | 15     | True                 |
| 4 | 3      | 15          | 15     | False                |

2/23/21

CompSci 101, Spring 2021

43

What is always true about the loop?

1.  $\text{lst}[\text{idxMax}]$  is always the largest value seen so far, up through value of  $i$

```
def mystery(lst):
    idxMax = 0
    for i in range(len(lst)):
        if lst[idxMax] < lst[i]:
            idxMax = i

    return idxMax

mystery([2, 12, 4, 15, 15])
```

| i | idxMax | lst[idxMax] | lst[i] | lst[idxMax] < lst[i] |
|---|--------|-------------|--------|----------------------|
| 0 | 0      | 2           | 2      | False                |
| 1 | 0      | 2           | 12     | True                 |
| 2 | 1      | 12          | 4      | False                |
| 3 | 0      | 12          | 15     | True                 |
| 4 | 3      | 15          | 15     | False                |

2/23/21

CompSci 101, Spring 2021

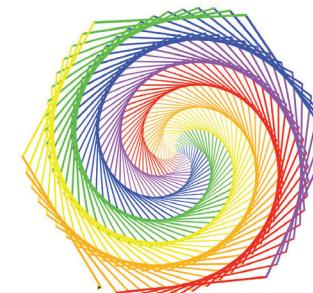
42

# Compsci 101

## Turtle, Bagels, Loop Tracing, Files

### Part 4 of 4

Susan Rodger  
February 23, 2021



2/23/21

CompSci 101, Spring 2021

44

# Examples of Processing Data

- Lecture 1: count letters in Bible
- Another example: Google Ngram viewer
  - <https://books.google.com/ngrams>

2/23/21

Compsci 101, Spring 2021

45

# Studying Language Evolution

- Ngram informs how words evolve
- From dove to dived
- <https://www.youtube.com/watch?v=tFW7orQsBuo>



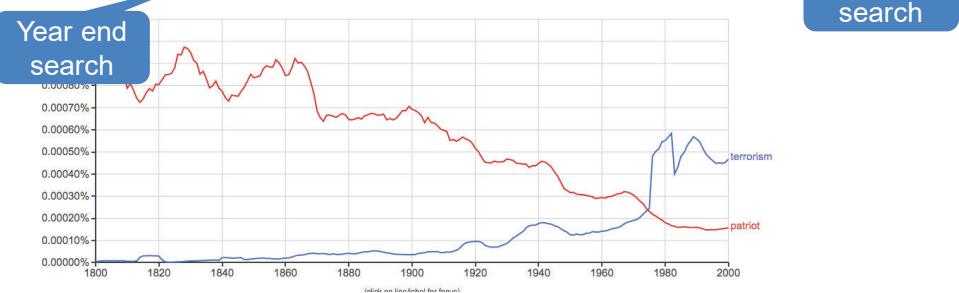
# Sequences, Repetition

- Parameters? What are they to this query?
  - [https://books.google.com/ngrams/graph?content=terrorism%2Cpatriot&year\\_start=1800&year\\_end=2000&corpus=15&smoothing=3](https://books.google.com/ngrams/graph?content=terrorism%2Cpatriot&year_start=1800&year_end=2000&corpus=15&smoothing=3)



# Sequences, Repetition

- Parameters? What are they to this query?
  - [https://books.google.com/ngrams/graph?content=terrorism%2Cpatriot&year\\_start=1800&year\\_end=2000&corpus=15&smoothing=3](https://books.google.com/ngrams/graph?content=terrorism%2Cpatriot&year_start=1800&year_end=2000&corpus=15&smoothing=3)



# Processing Data

- How do we find the longest word in .. Any text?
- How do we find the word that occurs the most?
- How is this related to how Google Search works?
- Text files can be viewed as sequences
  - Sequences of lines
  - Each line is a string
  - Some clean-up because of '\n'



2/23/21

CompSci 101, Spring 2021

49

## lineCount function

```
7  def lineCount(fname):  
8      """  
9          return # lines in file fname  
10         """  
11         f = open(fname)  
12         lc = 0  
13         for line in f:  
14             lc = lc + 1  
15  
16         f.close()  
17         return lc
```

2/23/21

CompSci 101, Spring 2021

51

## File Pattern: One line at a time

- Simplest and reasonably efficient Python pattern
  - Open, loop, close, return/process
  - LineCounter.py
- File as sequence
  - One line at-a-time
- Asymmetry in Open vs Close steps

```
7  def lineCount(fname):  
8      """  
9          return # lines in file fname  
10         """  
11         f = open(fname)  
12         lc = 0  
13         for line in f:  
14             lc = lc + 1  
15  
16         f.close()  
17         return lc
```

A diagram showing the lineCount function code. Two blue arrows point from the opening line 'f = open(fname)' to the closing line 'f.close()', highlighting the asymmetry between the two steps.

2/23/21

CompSci 101, Spring 2021

50

## altCount function

```
19  def altCount(fname):  
20      """  
21          return # lines in file fname  
22          """  
23          f = open(fname)  
24          lc = len(f.readlines())  
25          f.close()  
26          return lc
```

2/23/21

CompSci 101, Spring 2021

52

## main

```
28 ► if __name__ == "__main__":
29     name = "data/poe.txt"
30     pc = lineCount(name)
31     print("# lines:", pc)
32     pc2 = altCount(name)
33     print("# lines:", pc2)
```

2/23/21

CompSci 101, Spring 2021 53

## File Objects

- A file is an object, like a string
  - Functions applied to object: `len("word")`
  - To get file object use `open("data.txt")`
  - What is returned? Integer value, file object
- Often methods (aka function) applied to object
  - `f.readlines()`, `f.read()`, `f.close()`
  - Just like: `st.lower()`, `st.count("e")`

2/23/21

CompSci 101, Spring 2021 54

## Text File Processing Pattern

- See module `FileStuff.py`
  - If newline '\n' is read, call `.strip()`
  - If want to break line into "words", call `.split()`
- Process the list returned by `.split()`
  - May need to convert strings to int or float or ...
- The `for line in f:` pattern is efficient
  - Contrast list returned by `f.readlines()`

2/23/21

CompSci 101, Spring 2021 55

## FileStuff.py: avgWord

```
def avgWord(fname):
    f = open(fname, encoding="utf-8")
    totalWords = 0
    totalLen = 0
    for line in f:
        line = line.strip() #remove newline
        data = line.split()
        for word in data:
            totalWords = totalWords + 1
            totalLen = totalLen + len(word)

    f.close()
    return totalLen/totalWords
```

2/23/21

CompSci 101, Spring 2021 56