Lesson 11:

1. Introduction

Unit 3 – Structured data for web crawler.

String is a kind of structured data.

Lists are more powerful than strings, more than just characters.

1. Quiz: Stooges

stooges = ['Moe', 'Larry', 'Curly']

1. Quiz: Days in a Month

def how\_many\_days(month\_number):

return days\_in\_month[month\_number - 1]

1. Nested Lists

mixed\_up = [‘apple’, 3, ‘oranges’, 27, [1, 2, [‘alpha, ‘beta’]]]

1. Quiz: Countries

print(countries[1][1])

1. Quiz: Relative Size

print(countries[0][2] / countries[2][2])

1. Mutation

You may modify the existing object.

1. A List of Strings

This can happen.

1. Quiz: Different Stooges

stooges[2] = 'Shemp'

1. Yello Mutation

Aliasing p, q = object, changing p’s object also changes q’s

1. Aliasing

James Bond, 007

1. Quiz: Secret Agent Man

008

1. Quiz: Replace Spy

def replace\_spy(spy\_list):

spy\_list[2] = spy\_list[2] + 1

1. List Operations

<list>.append(<element>)

stooges.append(‘Shemp’) adds Shemp to the list, increasing the element size

1. List Addition and Length

<list> + <list>

len(<list>)

1. Quiz: Len Quiz

5

1. Quiz: Append Quiz

len(p) == 3 because it is appending the list q as a single element

1. How Computers Store Data
2. DRAM
3. Quiz: Memory Hierarchy

300000

.12

3.6

1. Quiz: Hard Drives

n$0.01, 2098km

1. Quiz: Loops on Lists

i < len(p)

1. For Loops

for each in list:

do something

1. Quiz: Sum List

# probably wanted a for loop but this is better

def sum\_list(list\_of\_numbers):

return sum(list\_of\_numbers)

1. Quiz: Measure Udacity

def measure\_udacity(list\_of\_strings):

count = 0

for string in list\_of\_strings:

if string[0] == 'U':

count += 1

return count

1. Quiz: Find Element

def find\_element(elements\_list, value):

index = 0

for element in elements\_list:

if element == value:

return index

index += 1

return -1

1. Quiz: Index

def find\_element(input\_list, value):

if value in input\_list:

return input\_list.index(value)

return -1

1. Guest Speaker

Sergey Brin, Co-founder of Google

Go through the whole class. (I’m trying!)

1. Quiz: Union

def union(list\_a, list\_b):

for element in list\_b:

if element not in list\_a:

list\_a.append(element)

1. Pop

Remove last element in a list.

1. Quiz: Pop Quiz

All except x pop, y pop, x append, y append.

1. Collecting Links

Put them in a list.

1. Get All Links

Rather than print all links. Output as list.

1. Links
2. Quiz: Starting Get All Links

links = []

1. Quiz: Updating Links

links.append(url)

1. Quiz: Finishing Get All Links

return links

1. Finishing the Web Crawler

Seed page, find the links, get them into a list, crawl, add more links, etc.

Must keep track of list of pages crawled.

1. Quiz: Crawling Process

Never return, the seed page was not working though.

1. Quiz: Crawl Web

tocrawl = [seed]

crawled = []

1. Quiz: Crawl Web Loop

page = tocrawl.pop() # depth first search

1. Quiz: Crawl If

page not in crawled

1. Quiz: Finishing Crawl Web

def crawl\_web(seed):

tocrawl = [seed]

crawled = []

while tocrawl:

page = tocrawl.pop()

if page not in crawled:

union(tocrawl, get\_all\_links(get\_page(page)))

crawled.append(page)

return crawled

1. Conclusion

We’ve learned to crawl. Lists!

Unit 4 gets content from web pages. Build an index to respond to queries.

Anna Patterson from Google: Politeness, multiple machines, bandwidth.