

LEC 09

CSE 123

Linked Nodes

Questions during Class?

Raise hand or send here


sli.do #cse123



BEFORE WE START

*Talk to your neighbors:**Any fun Summer plans once the quarter is over?*Music: [123 24su Lecture Tunes](#) **Instructor:** Joe Spaniac**TAs:** Andras Eric Sahej Zach
Daniel Nicole Trien


Lecture Outline

- **Announcements** 
- Review: Recursive Tracing
- Recursive Programming
 - Files
 - Public / Private pairs


Announcements

- Resubmission Period 3 due tonight (7/19) at 11:59pm
- Programming Assignment 2 is due Wednesday (7/24) at 11:59pm
- Resubmission Period 4 opening tonight, due next Friday (7/26)
 - Assignments available: P1, C2
- Check-in 3 next Thursday (7/25)
 - Taken in quiz section, should help in preparation for the quiz.
 - Reminder: you only *need* to attend 2 of these for the grade, but they should be beneficial regardless.

Lecture Outline

- Announcements
- **Review: Recursive Tracing** 
- Recursive Programming
 - Files
 - Public / Private pairs

Lecture Outline

- Announcements
- Review: Recursive Tracing
- **Recursive Programming** 
 - Files
 - Public / Private pairs

Recursive Methods [Review]

- 2 components of every recursive method:
- Recursive case
 - Decompose problem into subproblem
 - Make the actual recursive call
 - Combine results meaningfully
- Base case
 - Simplest version of the problem
 - No subproblems to break into
 - Return known answer



If decomposing moves you closer to the base, no infinite recursion!

Why Recursion?

- Generally, anything you can write iteratively you can write recursively
 - So why write anything recursively?

Recursion is particularly useful when dealing with something that's recursively defined

- Math examples:
 - Factorial: $n! = n * (n - 1)!$
 - Exponent: $x^n = x * x^{n-1}$
 - Fibonacci: $fib(n) = fib(n - 1) + fib(n - 2)$
- Non-math examples?
 - ListNodes (int data, ListNode next)
 - Other ideas?

Language

As Gregor Samsa awoke one morning from uneasy dreams
he found himself transformed in his bed into a monstrous vermin.

Kafka, *Metamorphosis*



Language

- Below are formal definitions for English parts of speech:

sentence = noun_phrase + verb_phrase

noun_phrase = determiner_phrase + adjective_phrase + noun OR
proper_noun

adjective_phrase = adjective OR
adjective + adjective_phrase

verb_phrase = transitive_verb + noun_phrase OR
intransitive_verb

- Early work in NLP leveraged this recursive nature of language

Files

- We'll say that computer files fall into one of the following categories:



Standard file (.txt, .csv, .java)

```
f.isDirectory() -> false
```




Directory w/ subfiles

```
f.isDirectory() -> true  
File[] subFiles = f.listFiles()
```

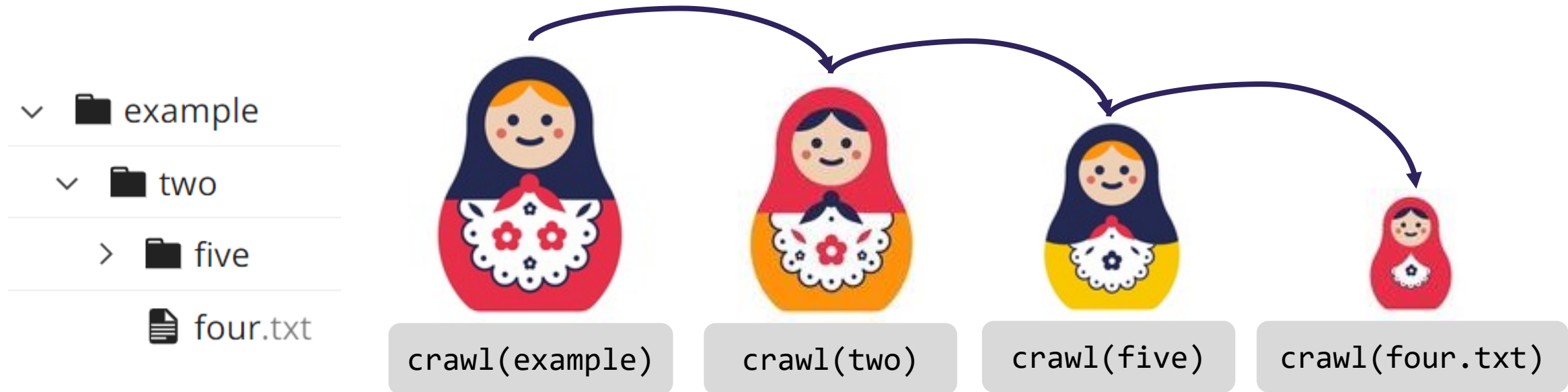
This is a recursive definition! A File is either normal, or a directory with a File[] of subFiles

Lecture Outline

- Announcements
- Review: Recursive Tracing
- Recursive Programming
 - **Files** 
 - Public / Private pairs

Crawl w/ Indentation

- How can one of our files know what level it's on?



- What if a bigger doll told the next smaller doll the level?
 - So long as the first doll is told the right value, this will work!
- Remember, recursive method calls are still method calls
 - How can we pass information from a bigger doll to a smaller doll?

Public / Private Pairs

- Used when we need additional information between recursive calls
- Private helper method hides additional info
 - Clients shouldn't have to worry about it
- All public method does is kick-start the private one
 - What's the correct starting value(s) for additional param(s)?

Question to ask: "Do I need to keep track of any additional information?"