

Week 04 Studio

Higher-Order Functions, Scope of Names

CS1101S AY21/22 Semester 1
Studio 05E

30 Aug 2021

Yan Xiaozhi (David)
@david_eom
yan_xiaozhi@u.nus.edu

August 27, 2020

gg bros 8:29 PM ✓✓

any hint for quest 8:29 PM ✓✓

stuck at the first qn for 3 hr lol 8:29 PM ✓✓

LOL 8:30 PM

me too 8:30 PM

i was there from last night 10 till like this morning
8:30 PM

Message



CS1101S 2122 All Staff

101 members



Martin Henz

owner

Do you all remember the Quest "Functional Expressionism"? Guess what, it's back!

It's in production and opens at noon tomorrow. Give it a try to make sure you haven't forgotten your lambdas!

1 ↩ 8:22 PM



Ivan Wong

i remember it well, spent 10 hours to earn just under 400 xp
8:23 PM



Admin

- Contact tracing (QR code + class photo)
- Reading Assessment 1 is on THIS FRIDAY!!!
 - Do past year papers (I believe they're on LumiNUS)
 - All MCQs but DO NOT OVERLOOK!
 - Week 3's content is quite important!
 - Post on the group if you have any questions!
- Quest Functional Expressionism (it's okay!)
- Mission Curve Introduction due 1st September 2359

Recap

Recursion

Classic Question



DEPARTMENT OF
**COMPUTER
SCIENCE**



COVID-19 INFORMATION

[HOME](#)

[ADMISSIONS](#)

[RESEARCH](#)

[NEWS & EVENTS](#)

[ABOUT US](#)

[ALUMNI](#)

[OUR STUDENTS](#)

[INNOVATION](#)

[HOME](#) > [ADMISSIONS](#) > [UNDERGRADUATE](#) > [HOW TO APPLY](#) > [INTERVIEWS](#) > [SAMPLE INTERVIEW PROBLEMS](#)

Undergraduate

[Why Computer Science, Why Oxford?](#)

[Undergraduate Degrees](#)

[How to Apply](#)

[Choose a course](#)

[Choose a college](#)

[Submit UCAS Form and Register for MAT](#)

[Maths Admissions Test \(MAT\)](#)

[Interviews](#)

Sample interview problems

Here are some problems that have been used in interviews in the past.

1. *Tidy boxes.* You are given 10 boxes, each large enough to contain exactly 10 wooden building blocks, and a total of 100 blocks in 10 different colours. There may not be the same number in each colour, so you may not be able to pack the blocks into the boxes in such a way that each box contains only one colour of block. Show that it is possible to do it so that each box contains at most two different colours. **A [sample interview dialogue](#) for this question is available.**

Recursion

Classic Question

5. *Lily-pad lunacy*. Eleven lily pads are numbered from 0 to 10. A frog starts on pad 0 and wants to get to pad 10. At each jump, the frog can move forward by one or two pads, so there are many ways it can get to pad 10. For example, it can make 10 jumps of one pad, 1111111111, or five jumps of two pads, 22222, or go 221212 or 221122, and so on. We'll call each of these ways different, even if the frog takes the same jumps in a different order. How many different ways are there of getting from 0 to 10?

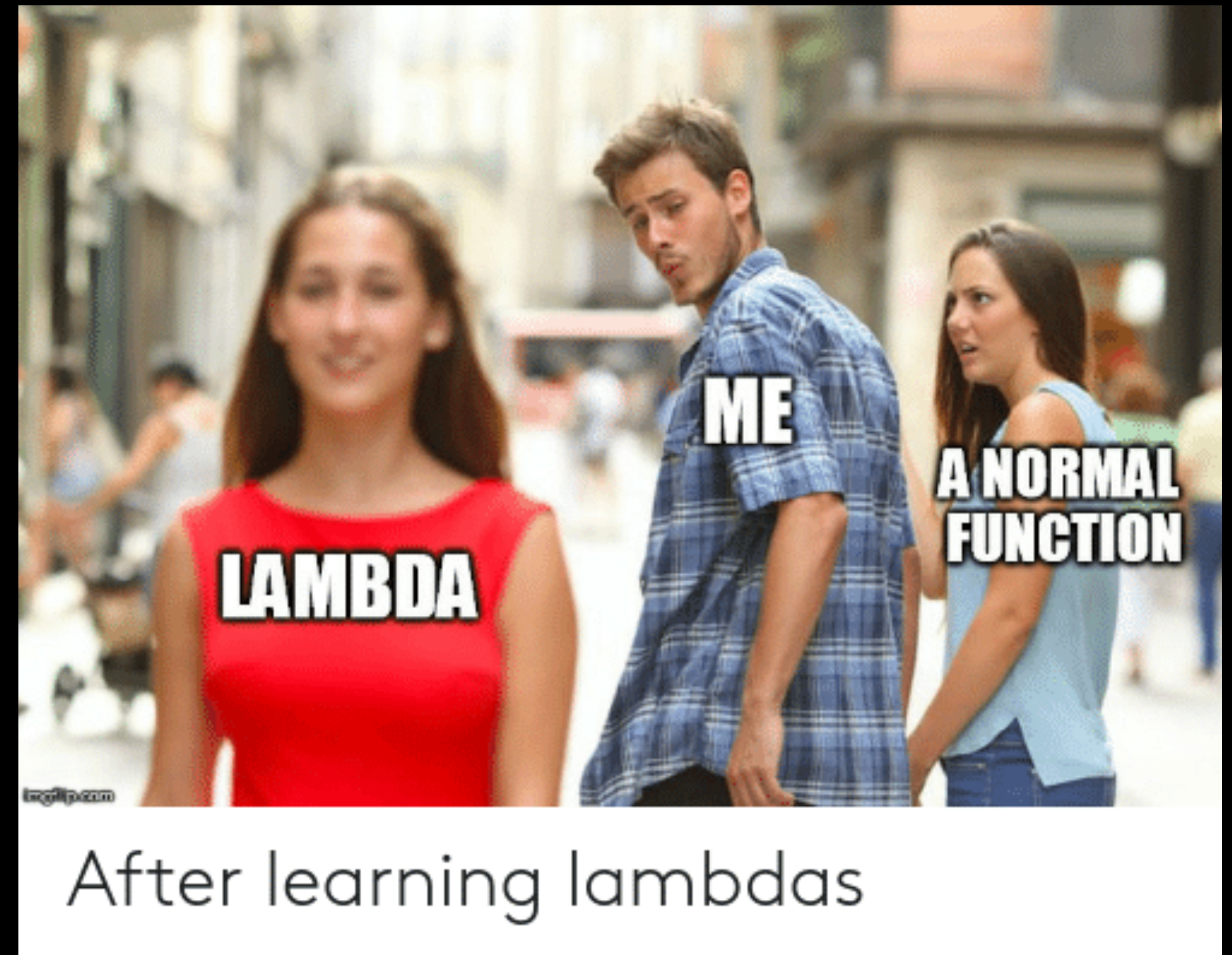
- Try to write out the solution without referring to the lecture slides!
 - ```
function frog(n) {
 return ... ;
}
```
- Time complexity?



# Lambda Expressions

## What is it?

- Functions can be anonymous
- Arrow functions
  - `=>` in JavaScript/Source
  - `->` in Java
  - `lambda ... : ...` in Python)
- `(parameters) => expression`



# Lambda Expressions

## Why Use Lambda?

- Makes the programmes more concise and easier to read
  - Declare outside the function scope?
  - Declare as local functions?
- Makes it possible for a function to return a function
- ```
function sum(term , a, next , b) {  
    return a > b  
        ? 0  
        : term(a) + sum(term , next(a), next , b);  
}
```


Function Signatures

What Is It?

- Each function takes thing(s) in and return something
 - `const sum = (x, y) => x + y;`
 - `sum: (number, number) -> number`
 - sum takes in 2 numbers, does some stuff, returns 1 number
- `show(stackn(heart, sail));`
- `stackn: (number, rune) -> rune`

Function Signatures

Exercises

- Identify the function signatures:
 - `const f1 = () => 1;`
 - `const f2 = number => number ? false : true;`
 - `const f3 = (x, y, z) => x * y + z;`
 - `const f4 = func => x => func(x);`
 - `func = x => x;`
 - `func = x => y => x - y;`

Higher Order Functions

What Is It?

- Functions can take in functions as arguments
 - `sum(term, a, next, b)`
- Functions can return function as result
 - How to make `add(x)(y)` return `x + y`?
- Or even both at the same time LOL
 - `increment_repeater(repeater)`

`(() => ())()`

Higher Order Functions
in JS

Higher Order Functions

Strategy to Interpret

- Find the leftmost arrow
 - Consider two sides
 - Left: parameters
 - Right: returned value
 - Substitute and get rid of the left part once argument is provided!
- `const foo = a => a;`
 - `const bar = abc => def => abc * def;`
- Exercise:
 - `foo(123);`
 - `bar(5)(10);`
 - `bar(foo(1))(foo(7));`
 - `bar(bar(3)(4))(bar(5)(foo(6)));`

Tickle Tickle Your Brain

- Consider the programme below
- What will the output be?
 - (Without running in Source Acad of course...)

```
const cs1101s = path => (quest, mission) =>
    contest => contest(quest, mission);

const cs = ceg => (infosec, bza, infosys) => bza;

cs(cs1101s(false))(NaN, "sike", 12345);
```

Scope of Names

What Is It?

- Recap: forms of name declaration
 - Pre-declared names e.g. `math_PI`
 - Constant declarations e.g. `const a = 1;`
 - Parameters & function names
 - e.g. `function f(x) { ... }` OR `const f = x => ...`
- Lexical scoping
 - A name occurrence refers to the closest surrounding declaration
 - The interpreter only looks upwards and outwards

Scope of Names

“Scopes”

- What entails a scope?
 - Constant: closest surrounding pair of `{ ... }`, or the whole program (global scope)
 - Parameters: body of the function
 - Function name: as if it was declared with `const`

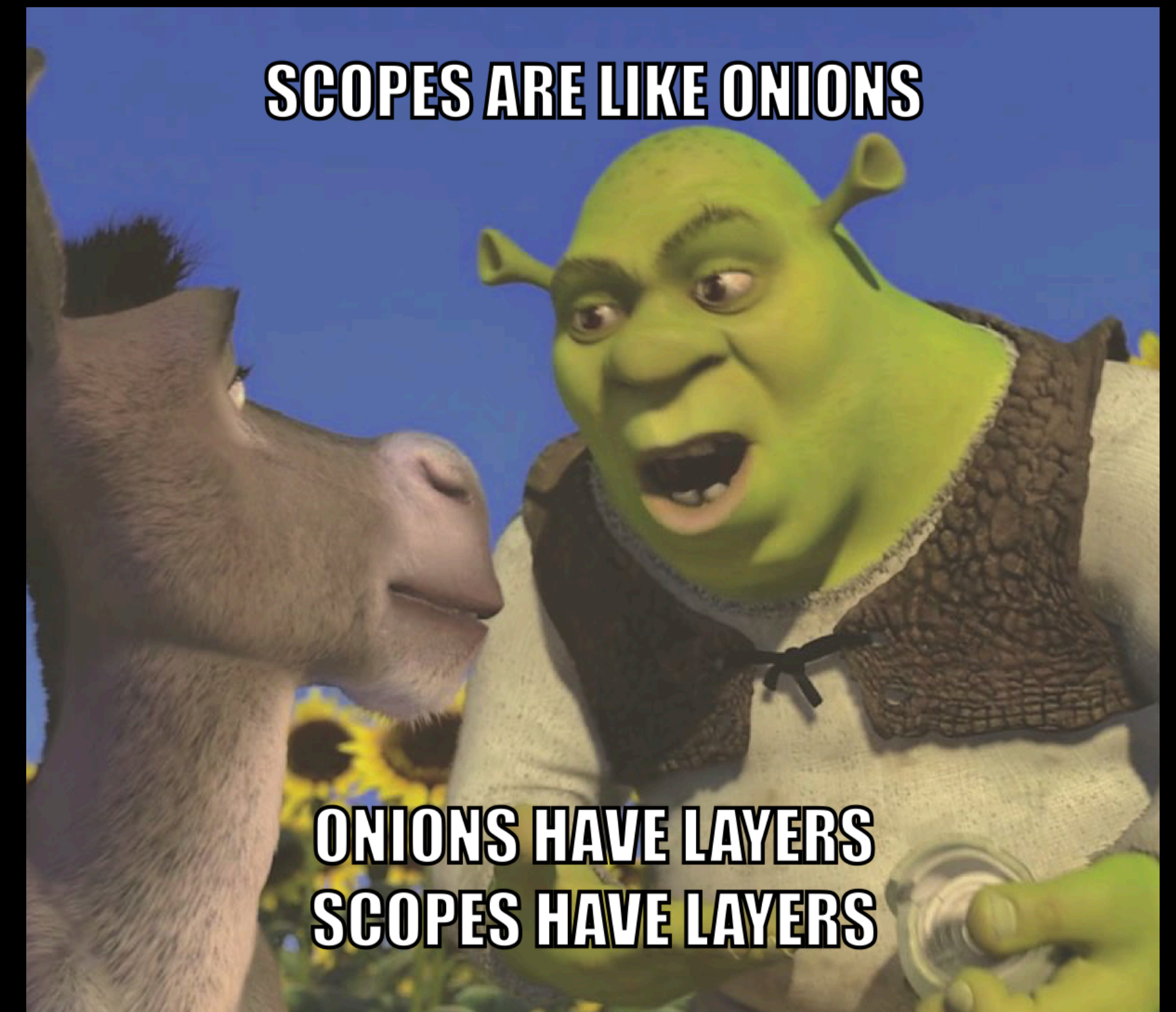
- ```
function add_one(x) {
 return x + 1;
}
```

- $(x, y) \Rightarrow x + y;$

# Scope of Names

## “Scopes”

- Interpreter looks for the current scope
- If it does not exist, it looks outwards in the layers until it finds it, or the global scope is reached
- If it is not found, interpreter will throw error:  
“Name <xxx> not declared.”





**Any Questions?**