

# Discussion 3

## Recursion

Antonio Kam

`anto [at] berkeley [dot] edu`

# Announcements

- Exam grades have been released
  - I understand that you haven't necessarily gotten the grade you wanted; I'll talk more about this in a bit
- Hog due on Friday
  - Submit by Thursday to get a point of extra credit!
  - OH lately has been less busy than usual semesters
    - Look out for project parties
- We're going to start recursion! (Super fun topic, and will stick with us)

# Temperature Check

- Environment Diagrams
- Higher-order Functions
- Recursion

# Questions and Comments from last section

- Could you turn on the light in the front? If there is such an option. It is a bit dark.
  - i'm putting this here in case i forget
- If there's not enough time to complete an entire problem, could you still show the answers please?
  - good point, will do
- spending too much time on little details, but then again, maybe its just my pace. + give less time to solve questions.
  - people tend to work at different paces, and so far based on discussion responses, people have generally found that im going a little but too *fast*
  - more than happy to answer questions on anything while we go into questions (i'm usually prepared enough to answer questions on the whole worksheet)

# Questions and Comments from last section

- how to recover with a failing midterm grade?
  - clobber policy - it's very good
  - cs61a takes a long time to grasp (might take the whole semester for things to finally click)
- Your favorite restaurants in Berkeley? I wanna try some new places. My favorite is Sichuan Style [...]
  - Taco Sinaloa
  - IBs
  - Marugame

# Questions and Comments from last section

- reviewing environment diagrams and higher order functions a little bit more :)
  - we'll get a bit more time for this with future parts of the course (and a bit today as well)
- Are you excited for Starfield or Hogwarts legacy?
  - surprisingly, i haven't heard of them until now
    - starfield looks very nice 🙄 super excited to listen to the music 🤖
    - hogwarts legacy is interesting, but i've never interacted with the harry potter franchise, so i'm not sure what to expect there
- Your favorite places in Berkeley? Favorite cat breed?
  - victory point cafe (i really like the board game dominion)
  - siberian; british shorthair







# Midterm

- Important to mention that the midterm is worth **40** points out of the total **300**
- Exams are not the only component of your grade in this class - you have discussion/lab attendance, and homeworks/projects where you're given unlimited attempts, and there are also no hidden tests
  - This means that quite a large portion of the points in this course are not based on exam performance.
- Many people struggle on exams; it's completely normal to not feel too good about your own performance (in fact, quite a lot of exams in higher education will have averages lower than what you may have been used to in HS)

**All slides can be found on**

**[teaching.roux1.es](http://teaching.roux1.es)**

# What is recursion?

- A *recursive* function is one where a function is defined in terms of itself.
- Similar to higher-order functions except it returns a *call* to a function rather than the function itself
- Will be hearing me talk about this a lot: **recursive leap of faith**

# Analogy

- Imagine you're in a line waiting for boba, but you don't know how many people there are in front of you (and you want to count how many people there are in front of you)
- In this case, you can ask the person in front of you about how many people they have in front of them, and then they repeat this same process until...
- The person at the front now tells the person behind them that there's nobody in front of them
- Then everyone just needs to add 1 to their answer, and bring it backwards from there.

# 3 Steps of Recursion

## 1. Base Case

- What is the smallest version of the problem we know the answer to?
- I tend to think of this as the simplest input

## 2. Recursive Case (recursive call on a smaller version of the problem)

- What can I do to reduce my input to something simpler?
- Similar to `while` loops

## 3. Connecting it all together

- Assuming your recursive call is correct (**recursive leap of faith!**), how do you solve the real problem

# Example with analogy

## 1. Base Case

- I'm at the front of the line

## 2. Recursive Case (recursive call on a smaller version of the problem)

- I ask the person in front of me to tell me how many people they have in front of them (assume that the answer that they give is correct (**recursive leap of faith**))

## 3. Connecting it all together

- Add 1 to their answer

# Example

```
def factorial(n):  
    if n == 0 or n == 1: # Base Case  
        return 1  
    else: # Recursive Case  
        return n * factorial(n - 1)
```

# Example

- To calculate a factorial of an integer, what you do is multiply the integer itself with the factorial of one less than itself
  - `factorial(5) = 5 * factorial(4)`
- Notice the recursive pattern - `factorial(4)` will call `factorial(3)`, and so on and so forth, until our *base case* is reached.
- We know the result of `factorial(1)`, so calling `factorial(1)` will just return 1 (*base case*)



# Example (Another Perspective)

- What's the smallest input? What's the simplest problem I know the answer to?
  - `0` is the smallest input - `factorial(0)` also returns `1`.
- How can I reduce my problem?
  - If you have `factorial(n)`, you can reduce your problem down by calling `factorial(n - 1)`.
  - In this step, you also assume your reduced problem gives you the correct answer (so `factorial(n - 1)` gives you the correct result - which is the recursive leap of faith)
- How do I use that result to solve my problem?
  - Multiply by `n`
  - `n * factorial(n - 1)`

# Recursion vs Iteration

Recursion	Iteration
Base case is needed for a recursive problem	A condition for a <code>while</code> loop is needed
Need to reduce down to the base case	Need to reduce down to the <code>while</code> condition
Can't use variables to keep track of values because they reset (need a helper function for that)	Can have variables to keep track of values.
Needs lots of frames - takes up memory	Loops happen in 1 frame

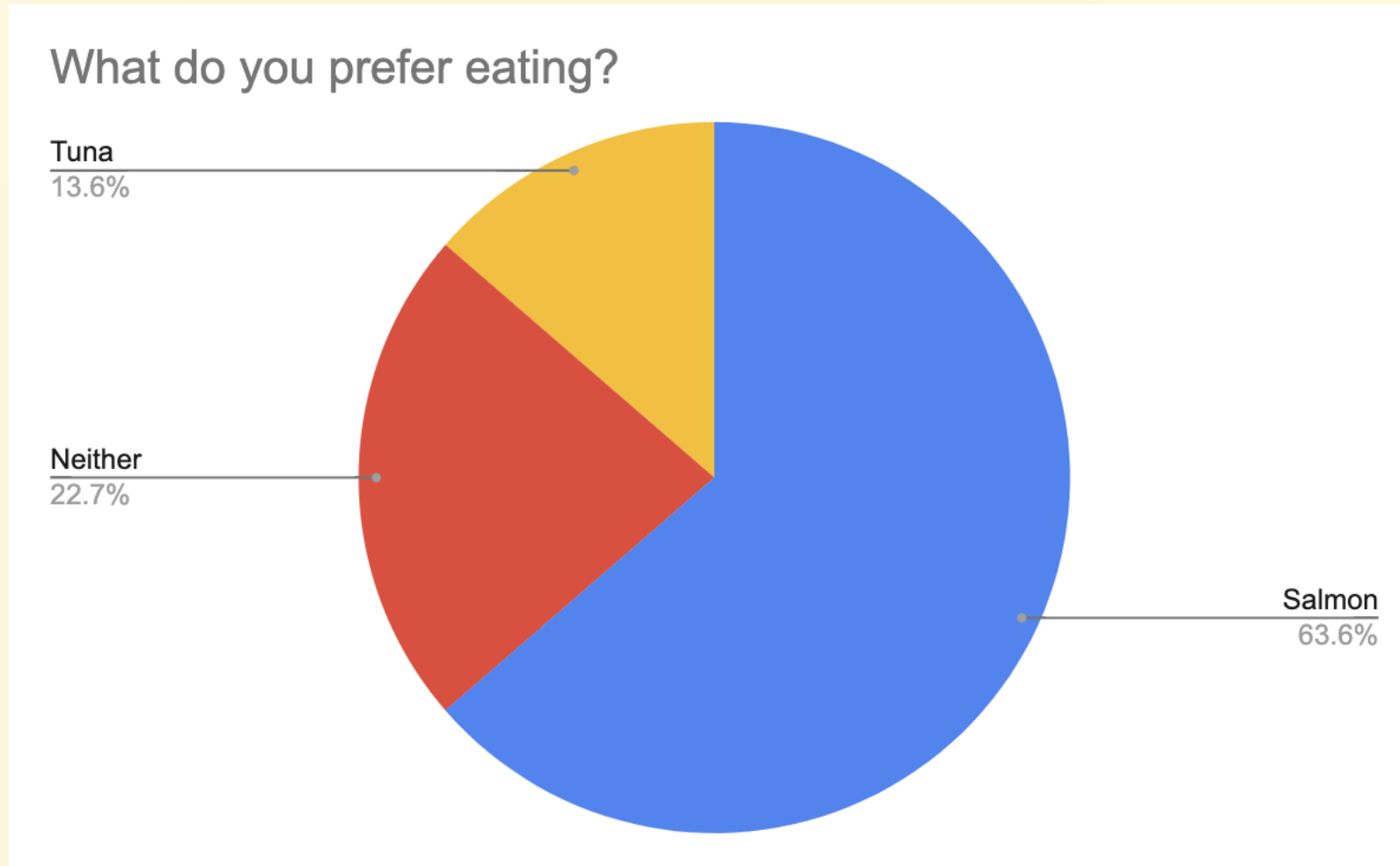
# Recursion vs Iteration

```
# Recursion  
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)
```

```
# Iteration  
def factorial(n):  
    result = 1  
    while n > 0:  
        result = result * n  
        n -= 1  
    return result
```

# Worksheet!

# Results from last section ( [links.roux1.es/disc](https://links.roux1.es/disc) )



# Mental Health Resources

- CAPS:
  - If you need to talk to a professional, please call CAPS at 510-642-9494.
- After Hours Assistance
  - For any assistance after hours, details on what to do can be found at [this link](#)

# Anonymous Feedback Form

[links.roux1.es/feedback](https://links.roux1.es/feedback)

Thanks for coming! 🎉

*Please give me feedback on what to improve!*