Welcome to CS61A CSM!

Introductions

Name: Aditya

Preferred Name: Adi

Pronouns: (He/Him)

Year and Major: 2nd Year CS and Physics Major

What class are you most excited about this semester: I’m taking 3 math classes so none really rip

Something you enjoy doing: Playing soccer (football tho), and marital arts

What animal do you resonate with most and why: Dog/Tiger

Section Philosophy

The goal of section is to create a safe and inclusive learning environment.

What does this mean practically?

- There is no stupid question. Even “what does 1+1 equal to” is a valid question in section. If something isn’t clear, please make sure to ask. This will only benefit you (in the short and long run).
- Some of you have more programing experience than others, so if a question seems silly to you please don’t be condescending about it and instead try to guide your peers to the right answer. Remember you were once a beginner too.
- If you’re not comfortable asking questions in person, that’s totally fine. I’ll have an anonymous google form set up so you can ask your question there and I’ll get to it via email to everyone or in next weeks section.

What you can expect from me as your mentor?/ general section structure

- During section I’ll give you a mini lecture on the relevant topics for that weeks worksheet. Rationale: Incase you’re behind on lecture (or need a refresher), the section won’t be a complete waste of time for you. If you are caught up with lecture and don’t want a refresher, feel free to get started on the worksheet!
- After mini lecture, you guys can work on the worksheet alone for a bit
- Next, discuss/ideate the solution with other students. We’ll be forming groups of 2-3 based on class size. This step can be particularly awkward, especially in the early days but this is intentional. There is tons of research that says working with others fosters better learning.
- I’ll go over the solutions to the problems
- I’m also available by email or text to answer any questions. For content related questions I’d prefer email (sometimes I open texts and never get back to them, so for questions that need answers, please email- and I’ll typically respond back in 24 hours. If I don’t feel free to prompt me again), I’m open to text messages to talk about anything else!

What I can’t do

Unfortunately, I can’t help with course assignments (projects/labs/hw) etc so make sure to go to office hours or post on 61A Ed!

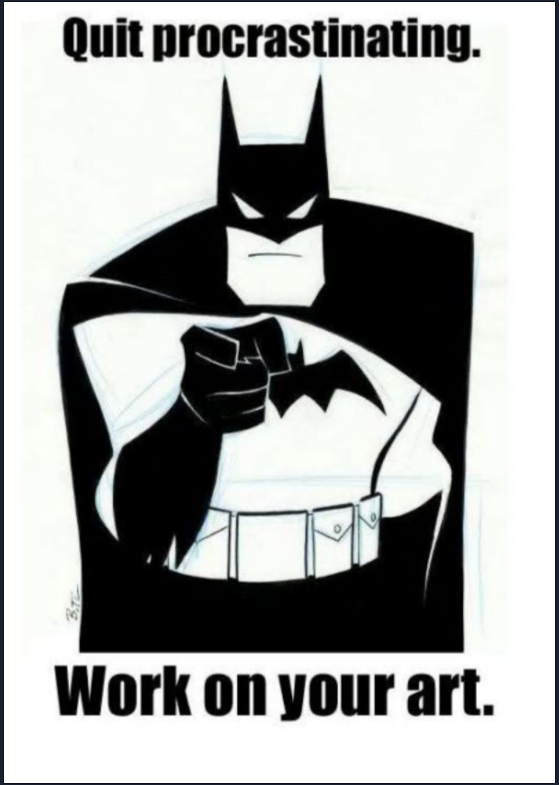
How to get the most out of CSM?

1. Understand worksheet questions thoroughly, Some of the questions are tricky and you might not get them the first time and that's completely okay. I've finished 61A and this is my second time teaching it and sometimes they trip me up too. Stick with it and be patient!
2. Ask questions! Super important
3. We won't be able to go over all questions on the worksheet in section and they're designed that way, so make sure to go home and practice these questions! And of course feel free to ask me if you have any questions.

Logistics

- If you have any unexcused absences in the first 3 weeks, you will automatically be dropped from CSM as there is high demand. If you’re unwell or have another legitimate reason for not being able to attend (MT for example), then email me and I’ll give you an excused absence.
- If you’re taking CSM for a unit you need 90% attendance to pass the course (and you must fill out the course feedback forms sent later on.

Tips to succeed in 61A



Higher Order Functions + Intro to Recursion

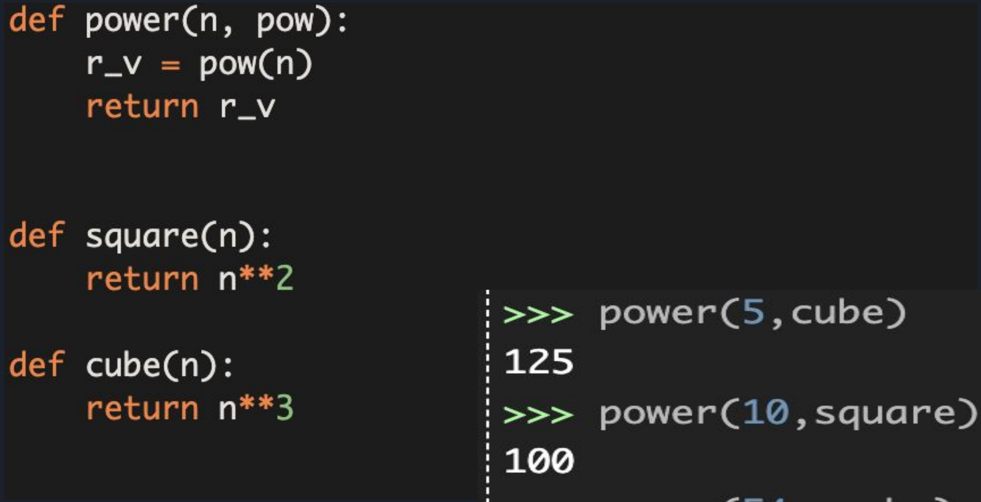
Aditya Iyer
9th Feb 2023

What is HOF?

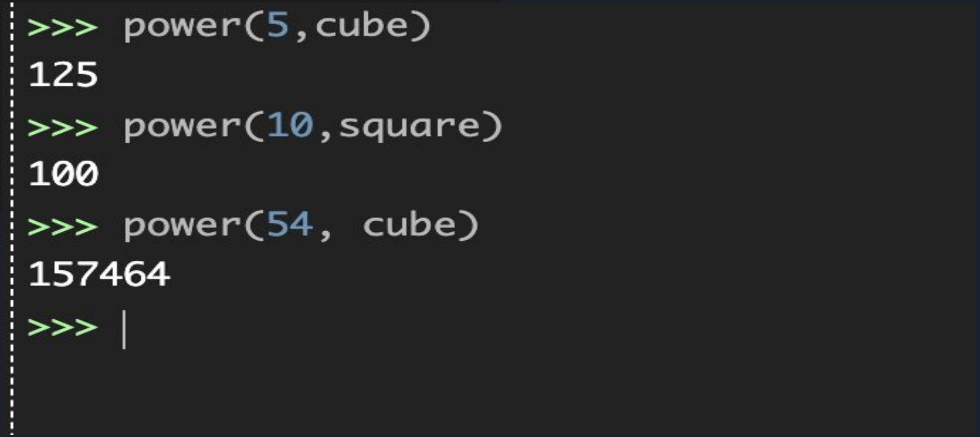
A higher order function is a function that either:

1. Takes in a function as its input
2. Returns a function as its output

HOF Example

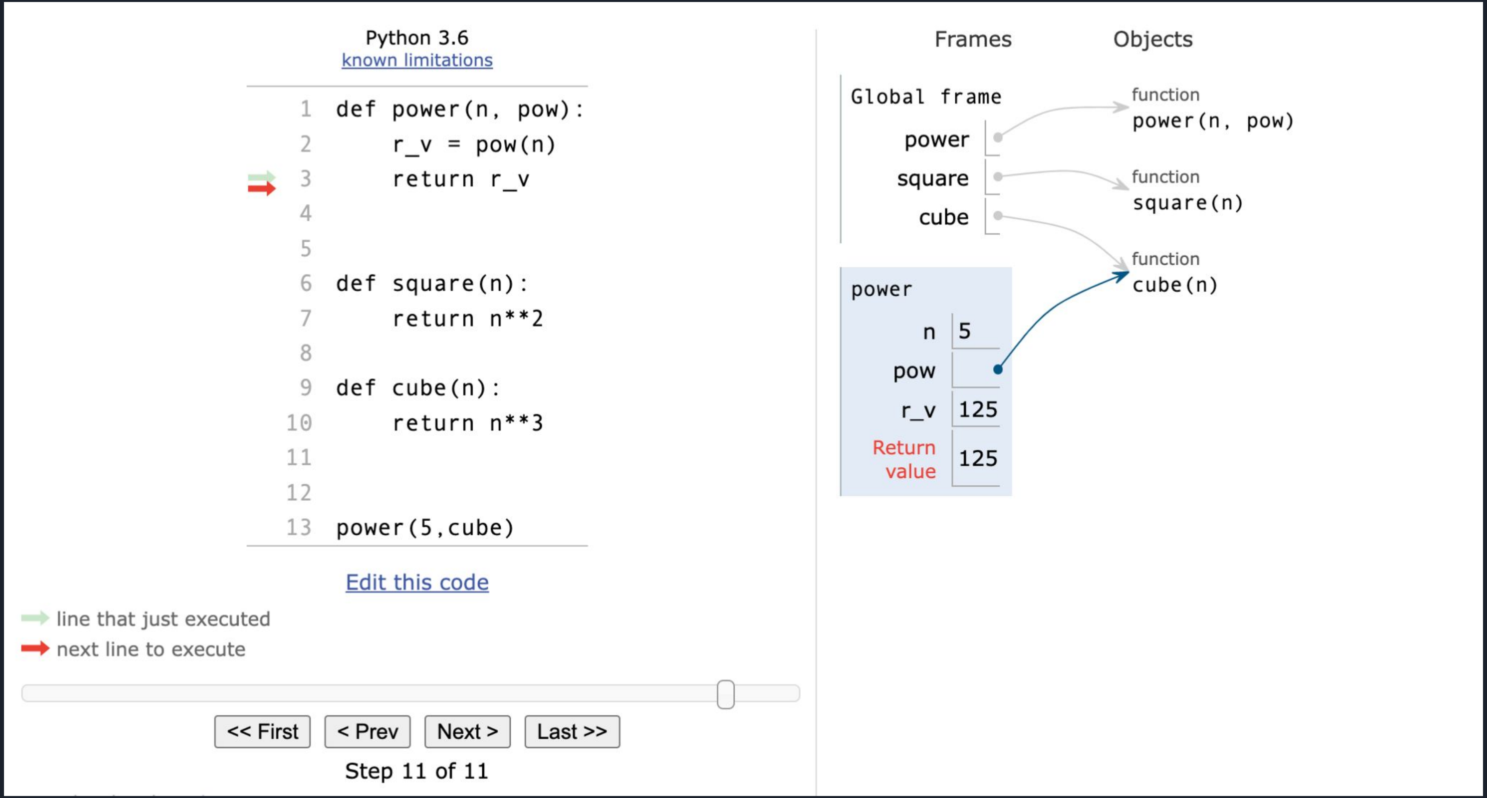


```
def power(n, pow):
    r_v = pow(n)
    return r_v


def square(n):
    return n**2

def cube(n):
    return n**3
```



```
>>> power(5,cube)
125
>>> power(10,square)
100
>>> power(54, cube)
157464
>>> 
```

HOF Environment Diagram



Python 3.6
known limitations

```
def power(n, pow):
    r_v = pow(n)
    return r_v


def square(n):
    return n**2

def cube(n):
    return n**3


power(5,cube)
```

Edit this code

🟩 line that just executed
🟥 next line to execute

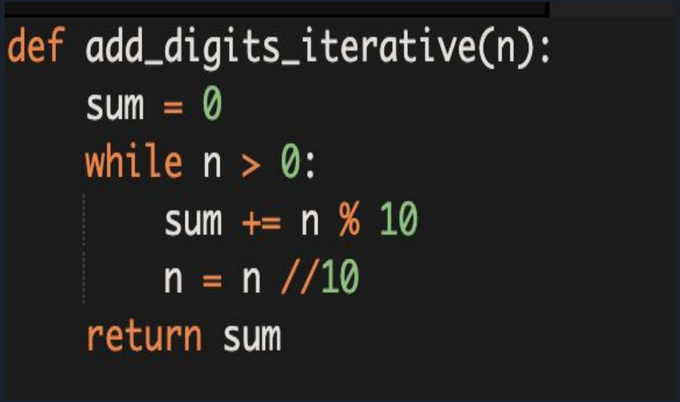
Recursion (also google recursion, its funny)

What is Recursion?

Recursion is when a function calls itself either directly or indirectly.

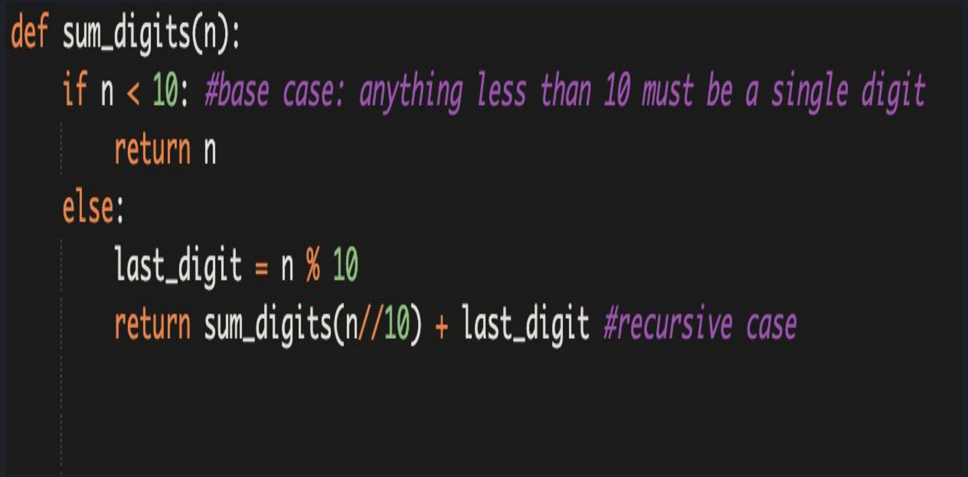
Recursion Basic Example

Iterative Approach



```
def add_digits_iterative(n):
    sum = 0
    while n > 0:
        sum += n % 10
        n = n //10
    return sum
```

Recursive Approach



```
def sum_digits(n):
    if n < 10: #base case: anything less than 10 must be a single digit
        return n
    else:
        last_digit = n % 10
        return sum_digits(n//10) + last_digit #recursive case
```

Factorial

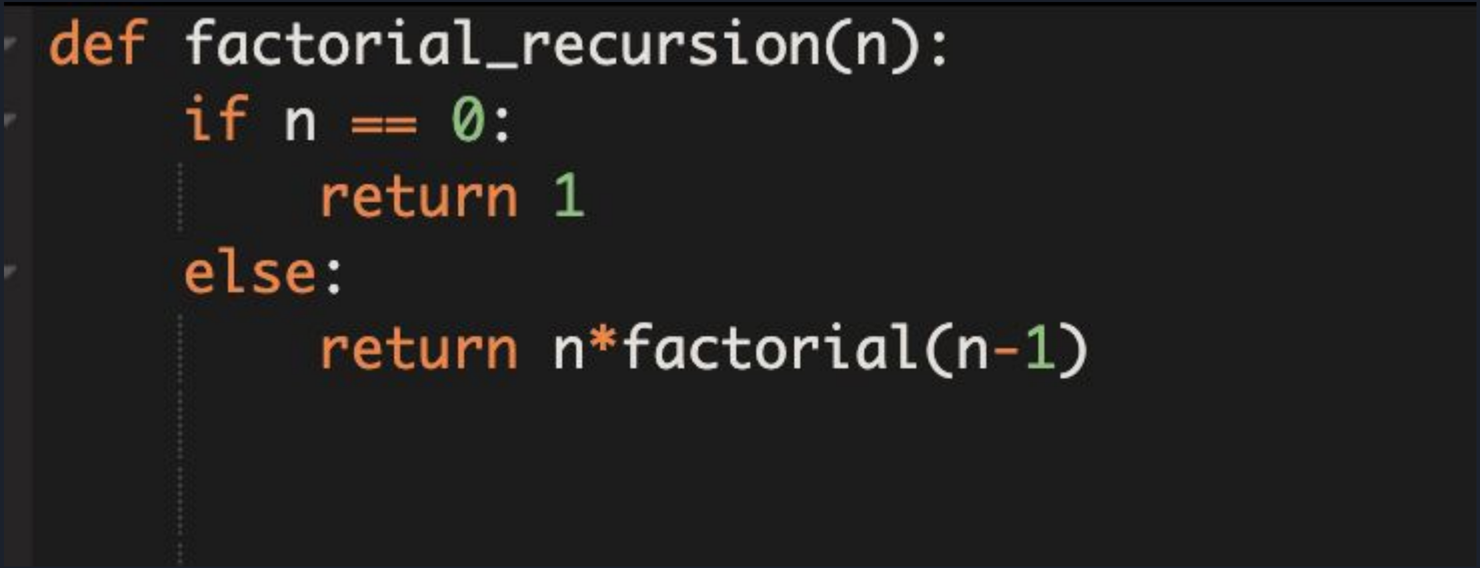
Factorial is denoted by ! symbol

$n! = n(n-1)(n-2)(n-3)....(1)$

$0! = 1$

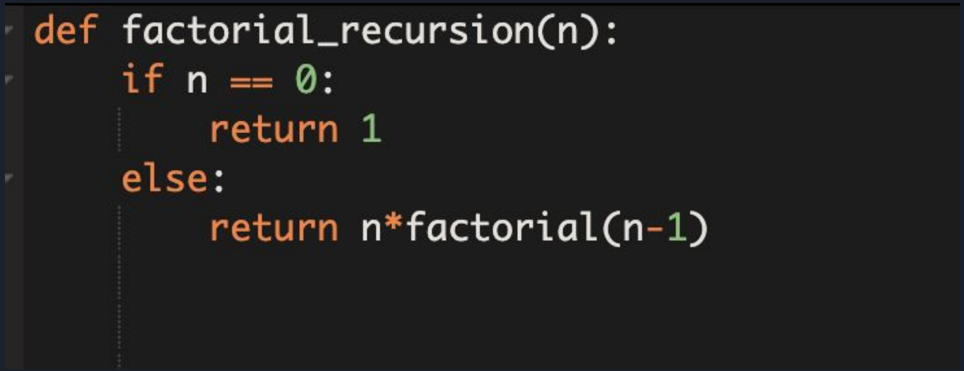
$1! = 1$

Factorial using recursion

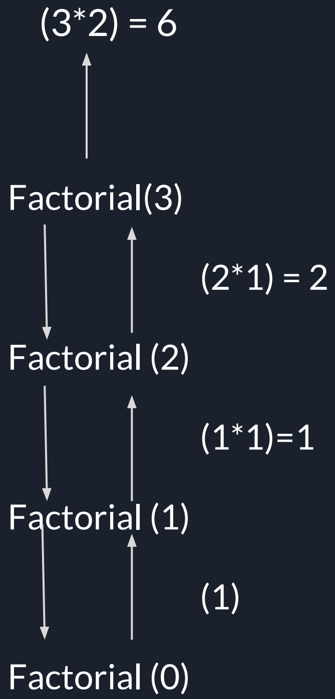


```
def factorial_recursion(n):
    if n == 0:
        return 1
    else:
        return n*factorial(n-1)
```

Understanding Factorial



```
def factorial_recursion(n):
    if n == 0:
        return 1
    else:
        return n*factorial(n-1)
```



$$\begin{tikzcd}
 & (3*2) = 6 \\
\text{Factorial(3)} \arrow[d, shift right] \arrow[ur] & \\
\text{Factorial (2)} \arrow[d, shift right] \arrow[u, shift right, "(2*1) = 2"'] & \\
\text{Factorial (1)} \arrow[d, shift right] \arrow[u, shift right, "(1*1)=1"'] & \\
\text{Factorial (0)} \arrow[u, shift right, "(1)"'] &
\end{tikzcd}$$

Anonymous Feedback Form