

Discussion 7

OOP and String Representation

Antonio Kam

`anto [at] berkeley [dot] edu`

Announcements

- HW 06 Releases on Friday!
 - OOP related
- Ants gets released on Monday
 - my favourite project by far
 - i love this project
 - very good project

Notes from last section

- switch on the lights at the front
- cucumbr
 - 👍
- super comprehensive minilecture at the start of lab; oop tough
 - 🤖
- Sometimes moved on from slides a little too quickly
 - yeah i agree
- hmmm. please discuss one of the lab problems.
 - usually don't have time during discussion to do this
 - ask during lab!

Notes from last section

- what is a regex
 - regular expression - basically lets you do find and replace with more general patterns to find
 - example: `\s*[Pp]ok[eé]mon\s*`
- Resources/exercises for learning about regex? I know it's useful but seems like dark magic to me.
 - tbh i have no idea, i just learned it when i took cs61a
 - fa21.cs61a.org

Notes from last section

- Can you go over how we can study efficiently. I feel like I study long hours but not getting anywhere. What helps what should we focus on how should we manage our time.etc.
 - everyone learns at difference paces and comes into this class with differing levels of experience
 - what i tend to find works well is doing the assignments (hw, optional lab questions) rather than just watching lectures, and afterwards, trying to come up with a real world use case
 - my mind tends to like seeing concepts in action in the real world, which is why this works for me

Notes from last section

- Place you want to travel to/one thing you have on your bucket list?
 - incheon this summer 🧐 world championships is happening then and i'm planning on competing
- secret word: 6 people typed Pokémon out of 19 people 🤔

Temperature Check

- OOP
- Class Methods

All slides can be found on

teaching.roux1.es

Object-Oriented Programming

What is OOP?

- One way I like to think of OOP is as a sort of 'advanced' data abstraction
 - You would use OOP for similar things that you would use data abstractions for
 - Can make a `City` class, for example
- OOP also allows for inheritance (less repetition of code, more on this later/next discussion)
- OOP also allows for mutation
 - Similar to list mutation (`.append`, `.extend`, etc.)
- You may have seen this if you've seen Java before (I didn't have any exposure to OOP when I took CS 61A for the first time)

OOP Terminology

- Class
 - A class is sort of a 'blueprint' for something. You can think of it as a template for creating an object
- Instance
 - An instance of a class is one object of that blueprint, or one physical object that you create based on your template
- Variables
 - Instance Variables: Variables unique to each instance (each actual object)
 - Class Variables: Variables shared between each instance in the same class
- Method
 - Function bound to a class

Functions vs Methods

- Methods need to take in `self` as an argument
 - This is very often *implicitly* passed in when the thing on the left side of the dot is an instance
 - `my_car.drive(100)` is the same as `Car.drive(my_car, 100)`
- `self` refers to the actual instance (rather than the class)
- Two ways of calling methods:
 - `Class.method(self, args)`
 - `instance.method(args)`

Q1 Mini-lecture

Worksheet!

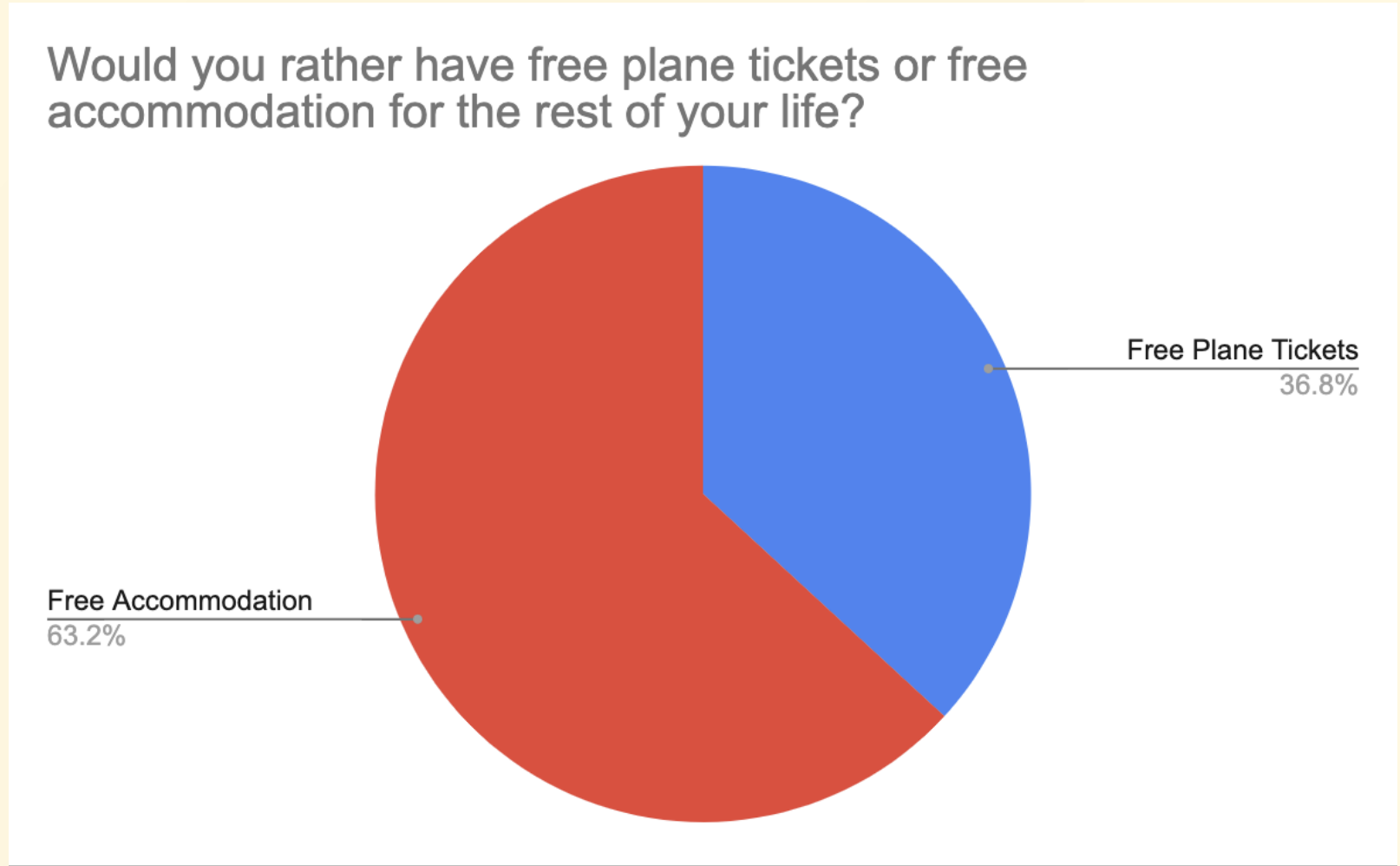
Class Methods

```
class Dog:
    def __init__(self, name, owner):
        self.name = name
        self.owner = owner

    @classmethod
    def robo_factory(cls, owner):
        return cls("RoboDog", owner)
```

- Uses the `@classmethod` decorator
- `Dog.robo_factory('Sally')`
 - Creates new dog with name `RoboDog` and owner `Sally`

Results from last section (links.roux1.es/disc)



Worksheet!

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

Thanks for coming! 🎉

Please give me feedback on what to improve!