

Object Oriented Programming for Collaborative Development in Python

An introduction to writing well-formed modular code to facilitate collaborative programming

What makes collaborating with others easy?

- good communication / shared vocabulary
- clarity in expectations
- access to resources
- work style
- feedback
- goals
- synchronous / asynchronous work

- coding styles diff.
- different goals
- lack of understanding

Object Oriented Programming (OOP)

Modularises code into chunks or "Objects" which is a data field that has unique attributes & behaviours

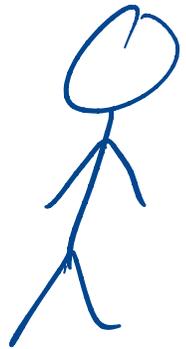
An Example: Humans

class Human :

attributes

Has :

- Name
- energy level
- friends



~~behaviours~~ → Methods

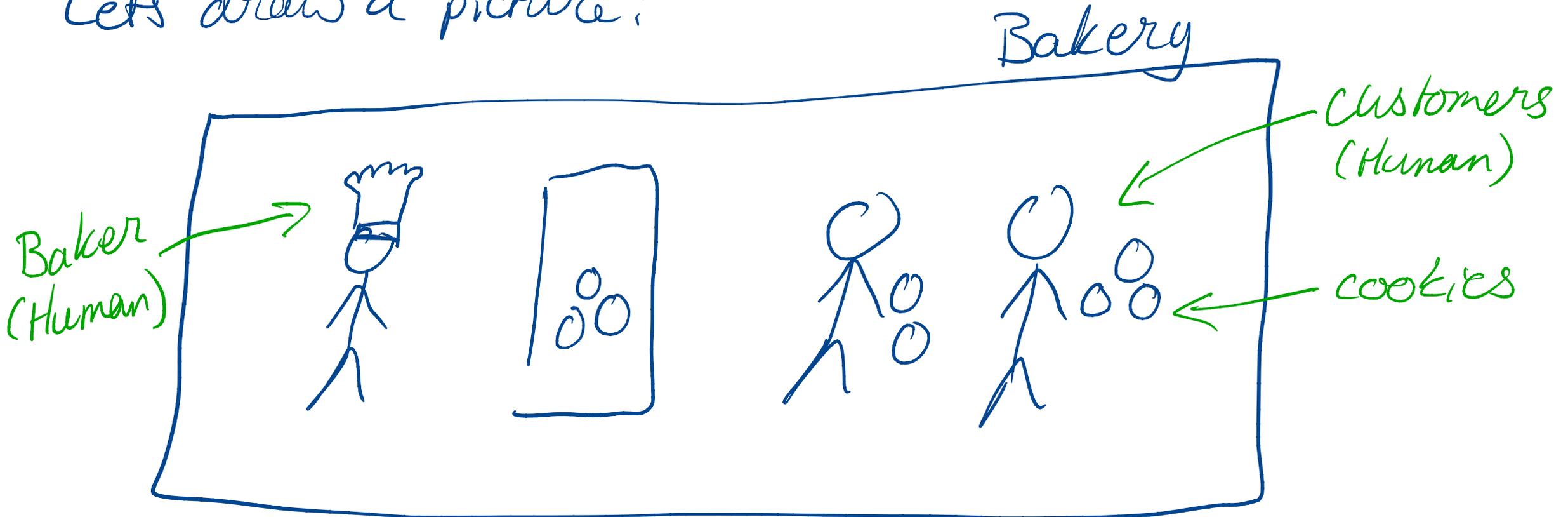
does :

- move
- sleep
- make friends

An Example: Bakery

There are different kinds of *Humans* with different Roles. Think in the context of a Bakery for example. We have groups of *bakers* and *customers* which are both types of *Humans*.

Let's draw a picture!



Bakery Has
- Baker (Human)
- cookies (int) does

Baker Has
- commission does
- make cookies
- sell cookies

customer Has
- money
- cookies does
~~- buy cookies~~



cookies Has
→ price

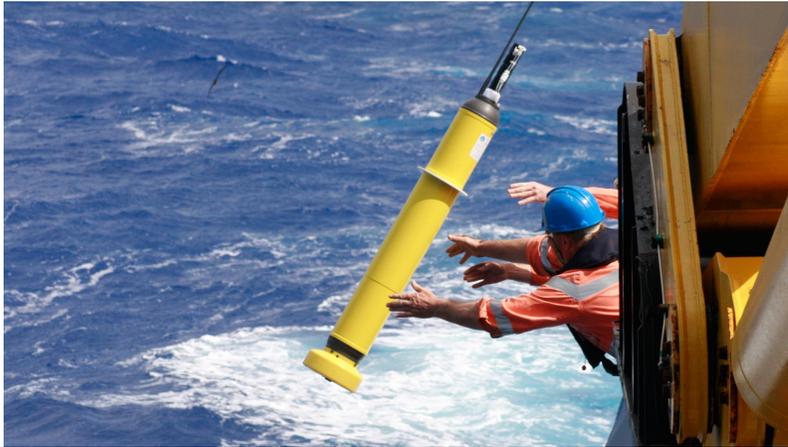
An application: icepyx

QUEST: Querying Unifying Exploring SpatioTemporal data

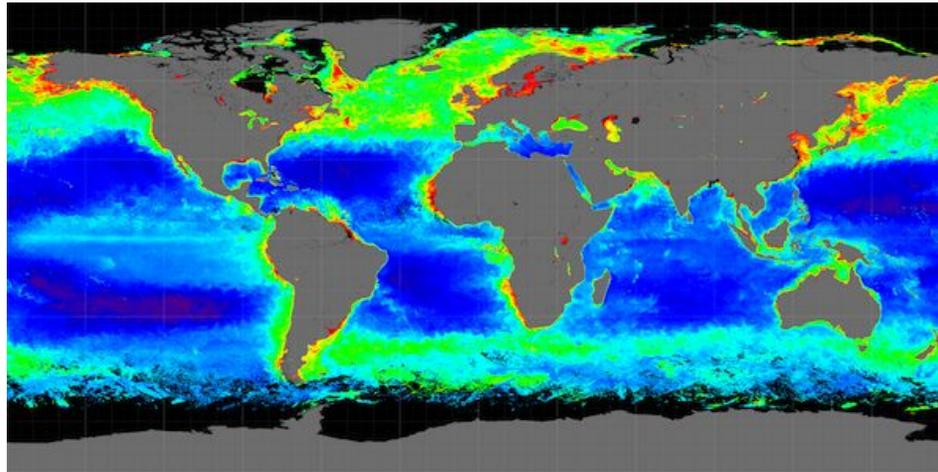
QUEST is a subclass of the super query, stores object information for querying datasets.

We need community input to add datasets and extend the capabilities of ICESat-2 data!

Argo



MODIS Aqua



Your favorite
dataset here



What's achievable when we combine the power of ICESat-2 data with other datasets?

Now you try!

Let's take Aliens for example. Below are some things that all Aliens have in common. Create your own kind of Alien using the *alien.ipynb* notebook as a template.