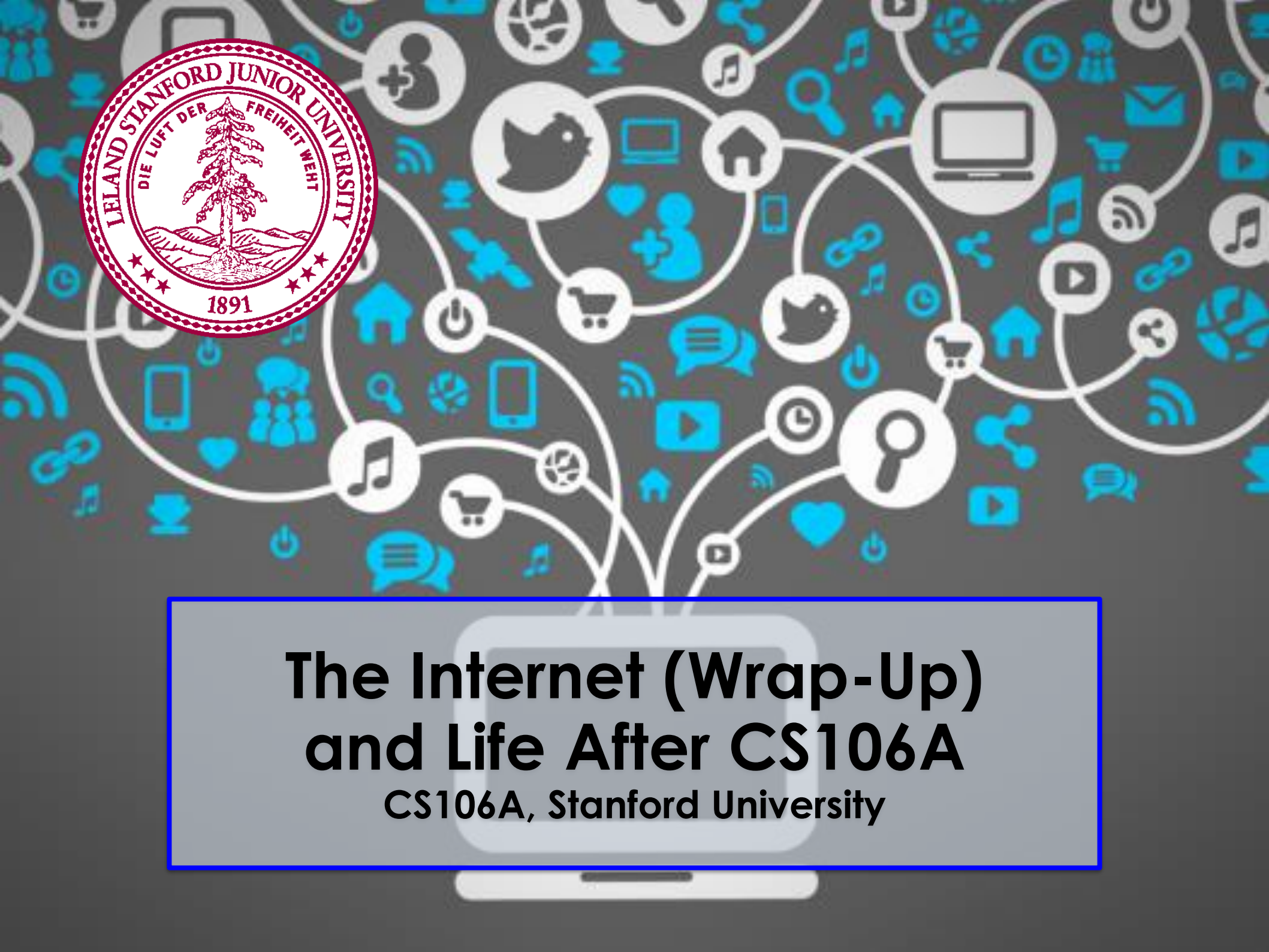
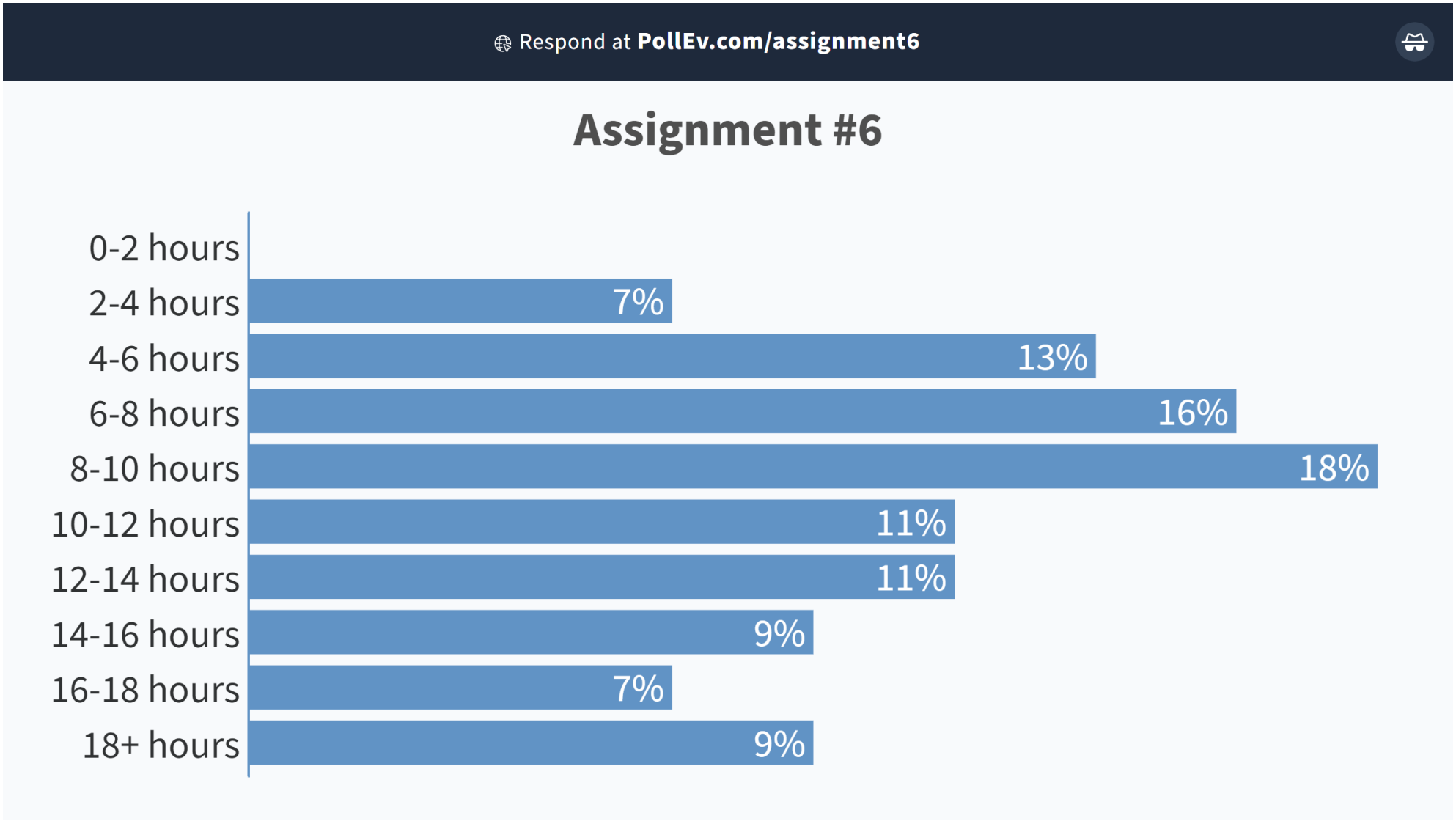


The Internet (Wrap-Up) and Life After CS106A

CS106A, Stanford University

Housekeeping



Learning Goals

1. Write a chat program that can respond to internet requests

Recall, Creating Servers

1

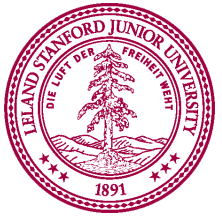
```
# handle server requests (must be in a class)
def handle_request(self, request):
    # return a string response!
```

2

```
# turn on the server
def main():
    # make an instance of your server class
    handler = MyServer()
    # start the server!
    SimpleServer.run_server(handler, 8000)
```

3

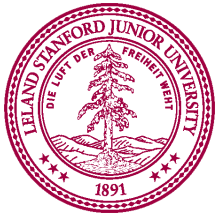
```
# enjoy
```



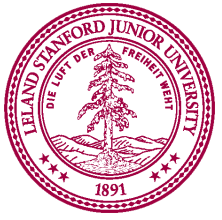
Recall, Requests

```
/* Request has a command */
command (string)

/* Request has parameters */
params (dict)
```

```
// methods that the server calls on requests
request.command
request.params
```



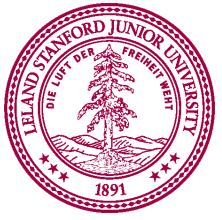
Requests responses are strings, often encoded using JSON



Reading a dictionary from a file

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

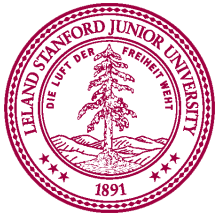


JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

JSON: JavaScript Object Notation
It's a format for storing a data structure as human (and machine) readable text



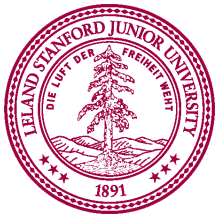
JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



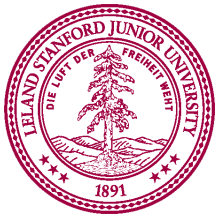
JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



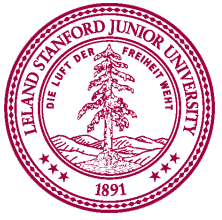
JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



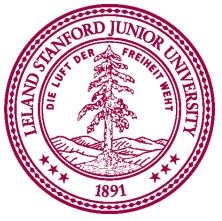
JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: handling data structures

ages.json

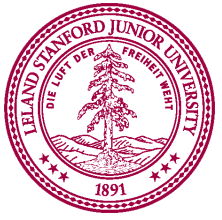
```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```

data

```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```



JSON: handling data structures

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

print_ages.py

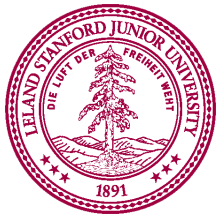
```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```

data

```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```

Console

```
Chris 34
Gary 70
Mehran 52
Brahm 25
Rihanna 34
```

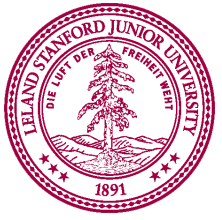


JSON

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

```
import json

# load data from a JSON file
data = json.load(open('ages.json'))

# write a data structure to a JSON file
json.dump(data, open('ages.json', 'w'))
```



JSON

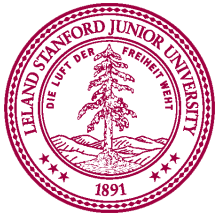
ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

```
import json

# load data from a JSON file
data = json.load(open('ages.json'))

# write a data structure to a JSON file
json.dump(data, open('ages.json', 'w'))
```

data

```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```



JSON

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

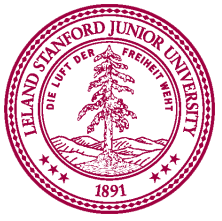
```
import json

# load data from a JSON file
data = json.load(open('ages.json'))

# write a data structure to a JSON file
json.dump(data, open('new.json', 'w'))
```

data

```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```

new.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```



JSON can also create strings!

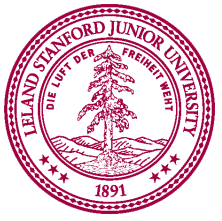
ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

```
import json

# load data from a JSON file
data = json.load(open('ages.json'))

# write a data structure to a JSON file
json.dump(data, open('ages.json', 'w'))

# write a variable to a string
data_str = json.dumps(data)
```

data

```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```



JSON can also create strings!

ages.json

```
{
    "Chris": 34,
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna": 34
}
```

```
import json

# load data from a JSON file
data = json.load(open('ages.json'))

# write a data structure to a JSON file
json.dump(data, open('ages.json', 'w'))

# write a variable to a string
data_str = json.dumps(data)
```

data

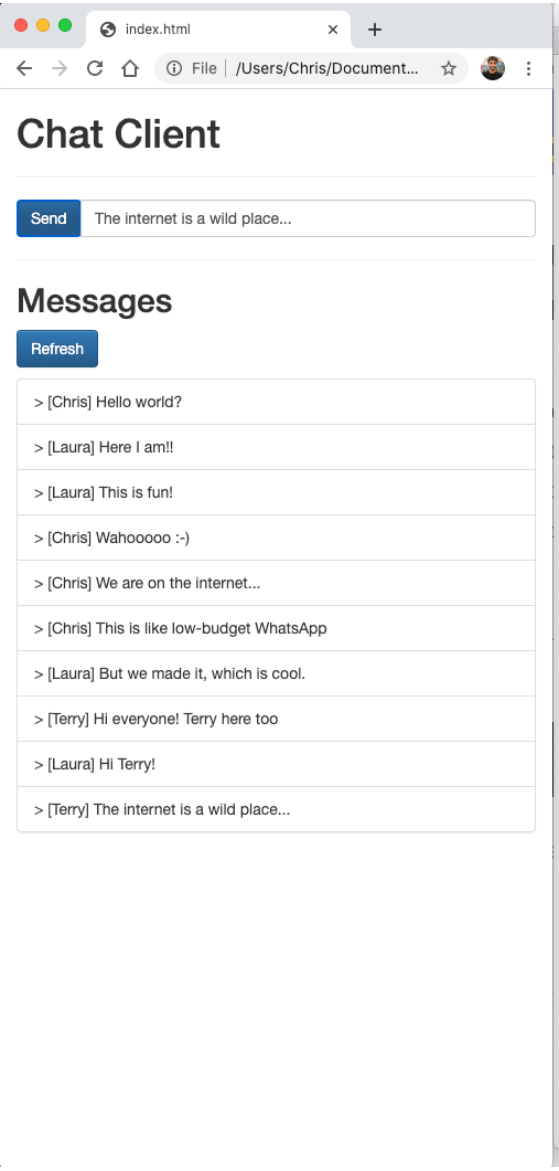
```
{
    "Chris": 34
    "Gary": 70,
    "Mehran": 52,
    "Brahm": 25,
    "Rihanna":34
}
```

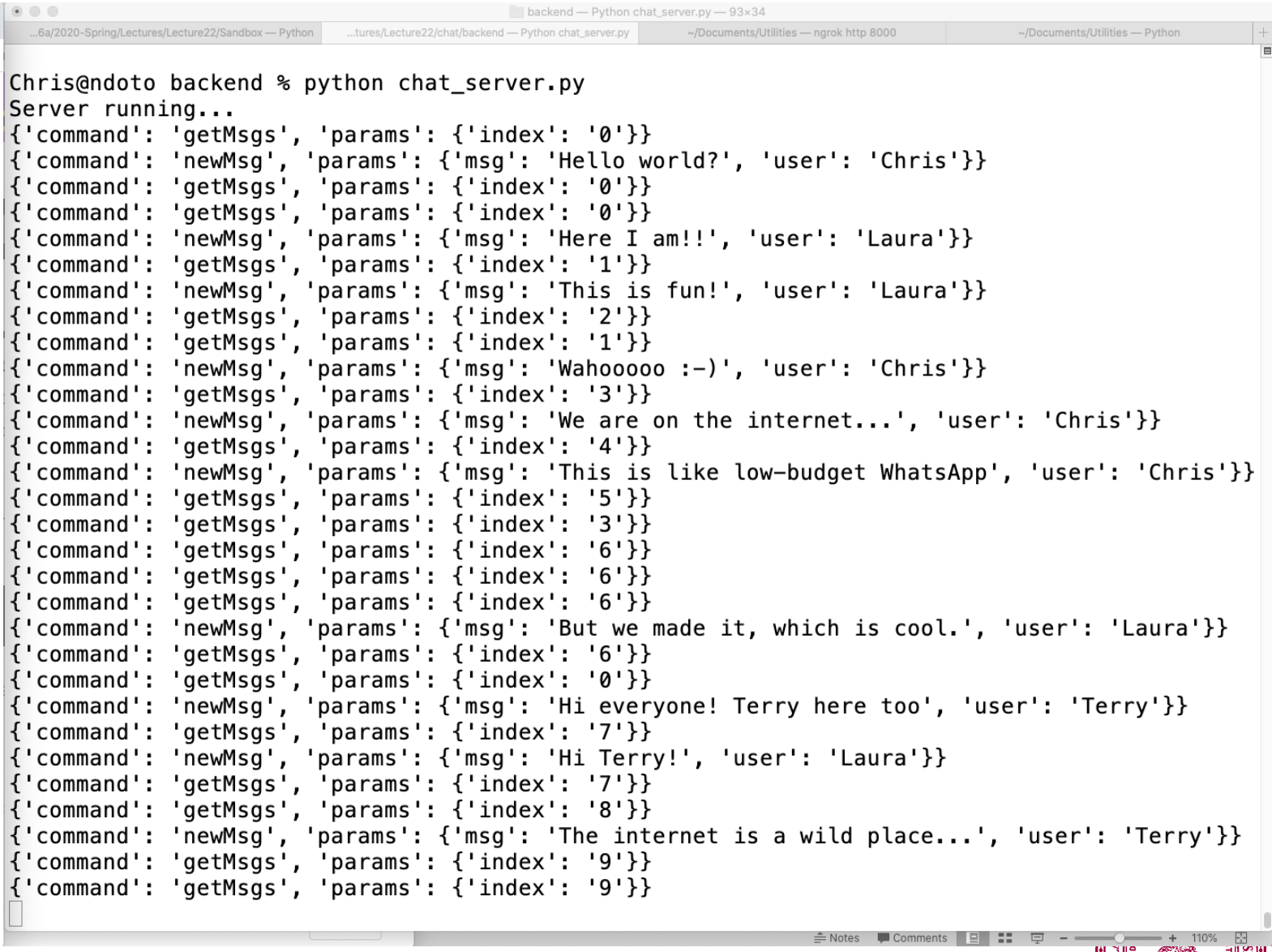
data_str

```
'{"Chris":34, "Gary":70, "Mehran":52, "Brahm":25, "Rihanna":34}'
```

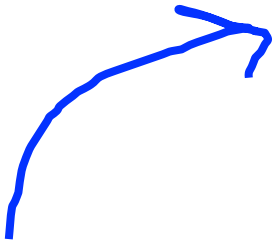
Time for a little chat

Chat Server and Client

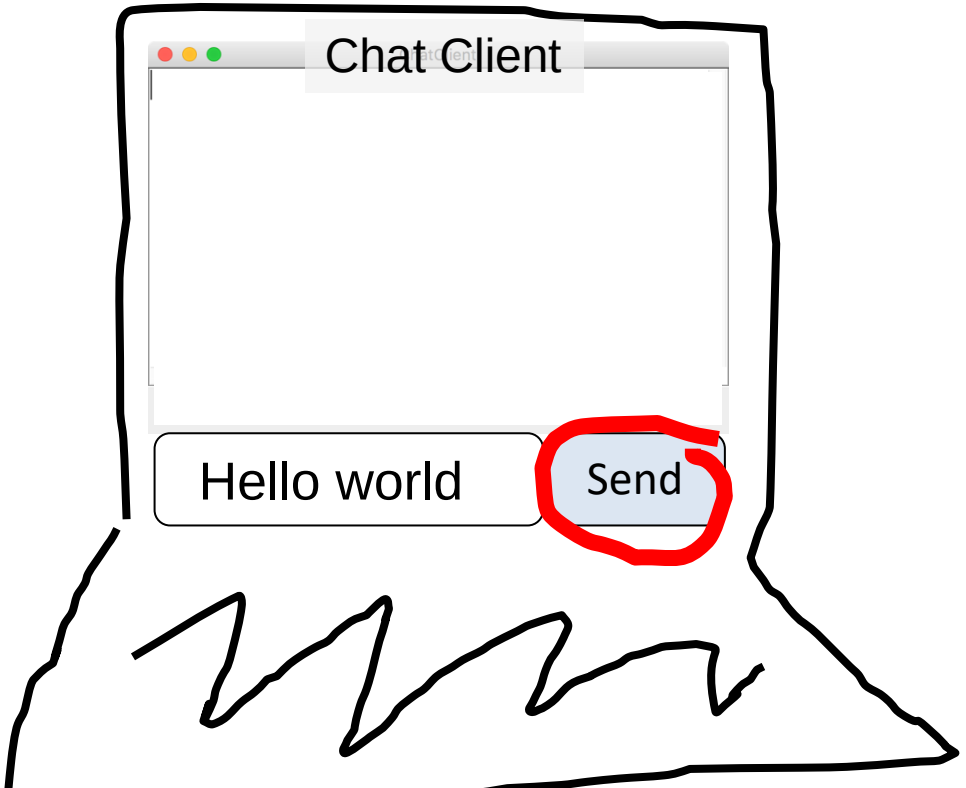


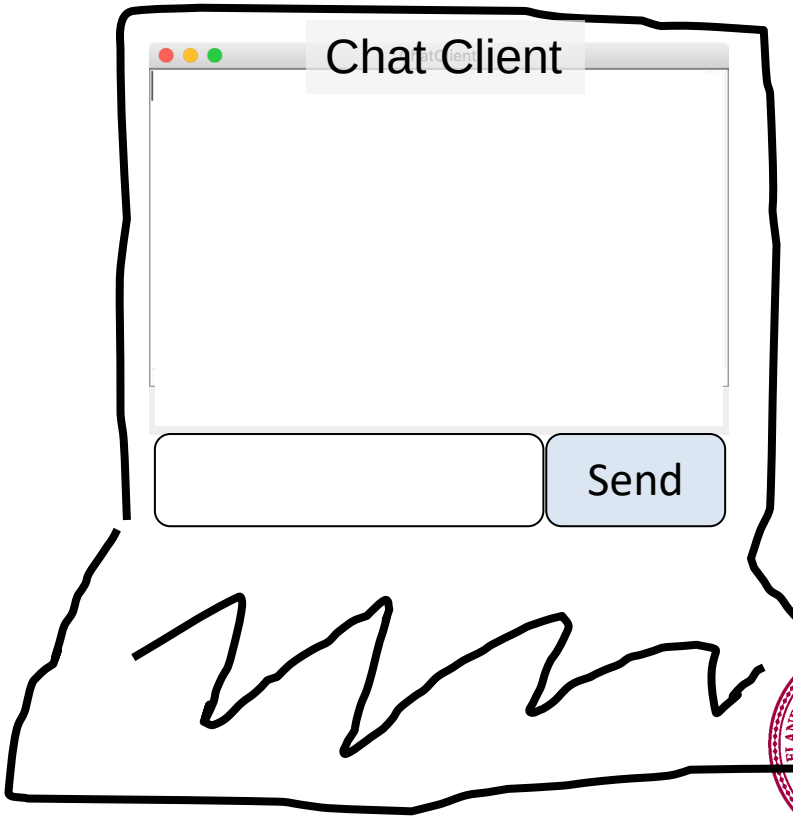


```
history = [
]
```



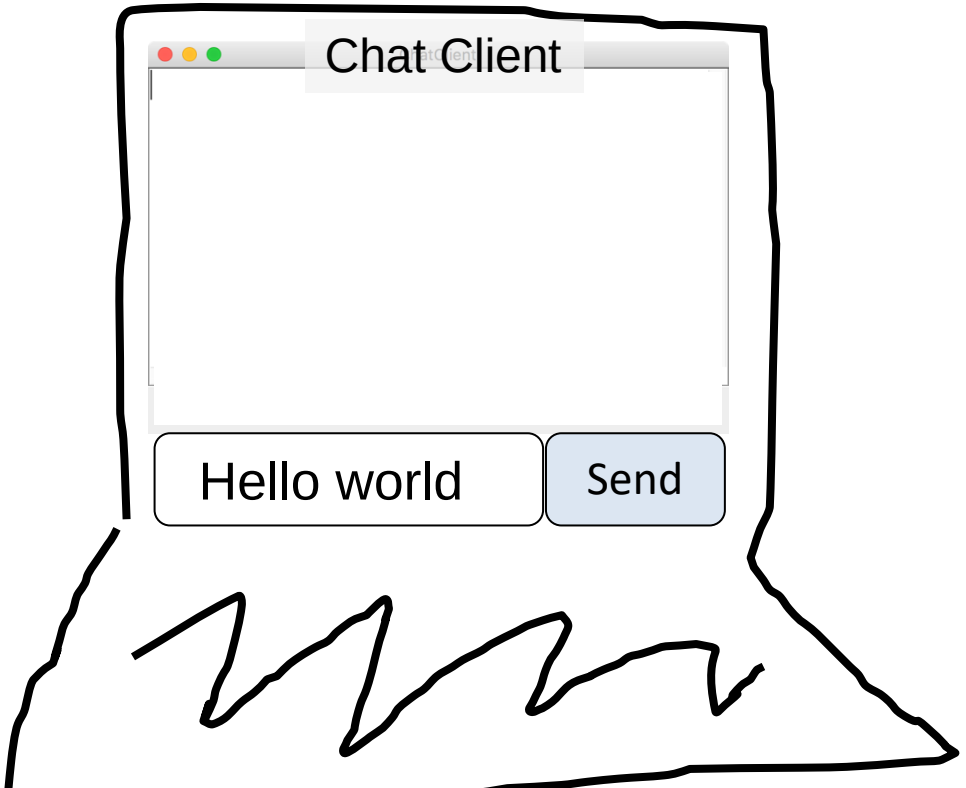
```
newMsg
{
    'msg' : 'Hello world'
    'user' : 'C'
}
```

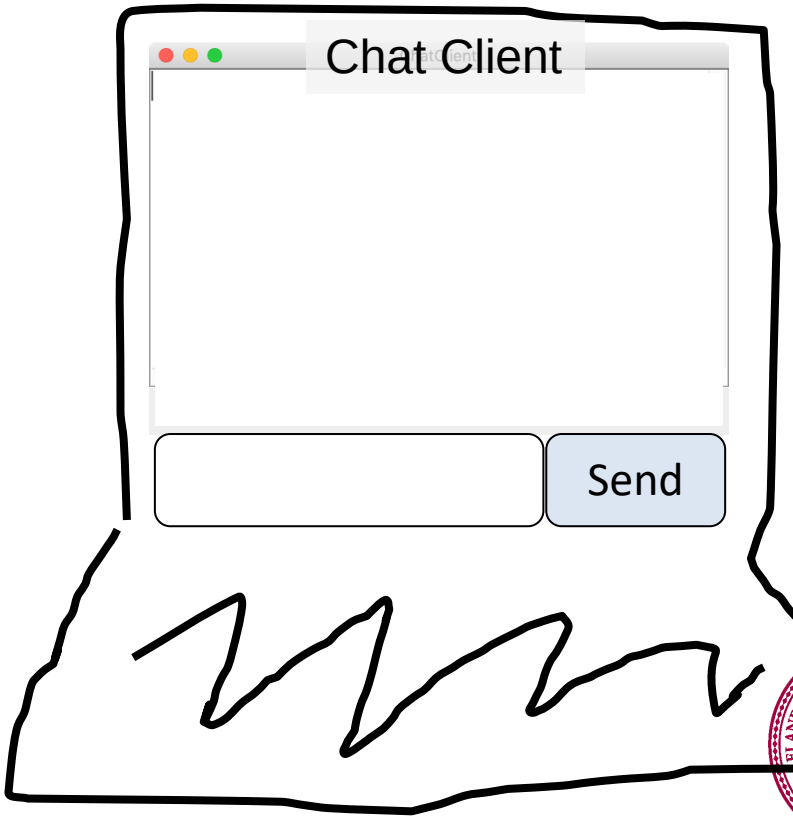


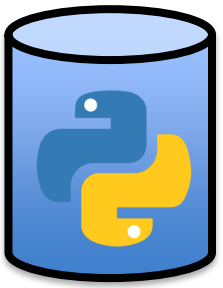


```
history = [
    '[C] Hello world'
]
```

```
newMsg
{
    'msg' : 'Hello world'
    'user' : 'C'
}
```

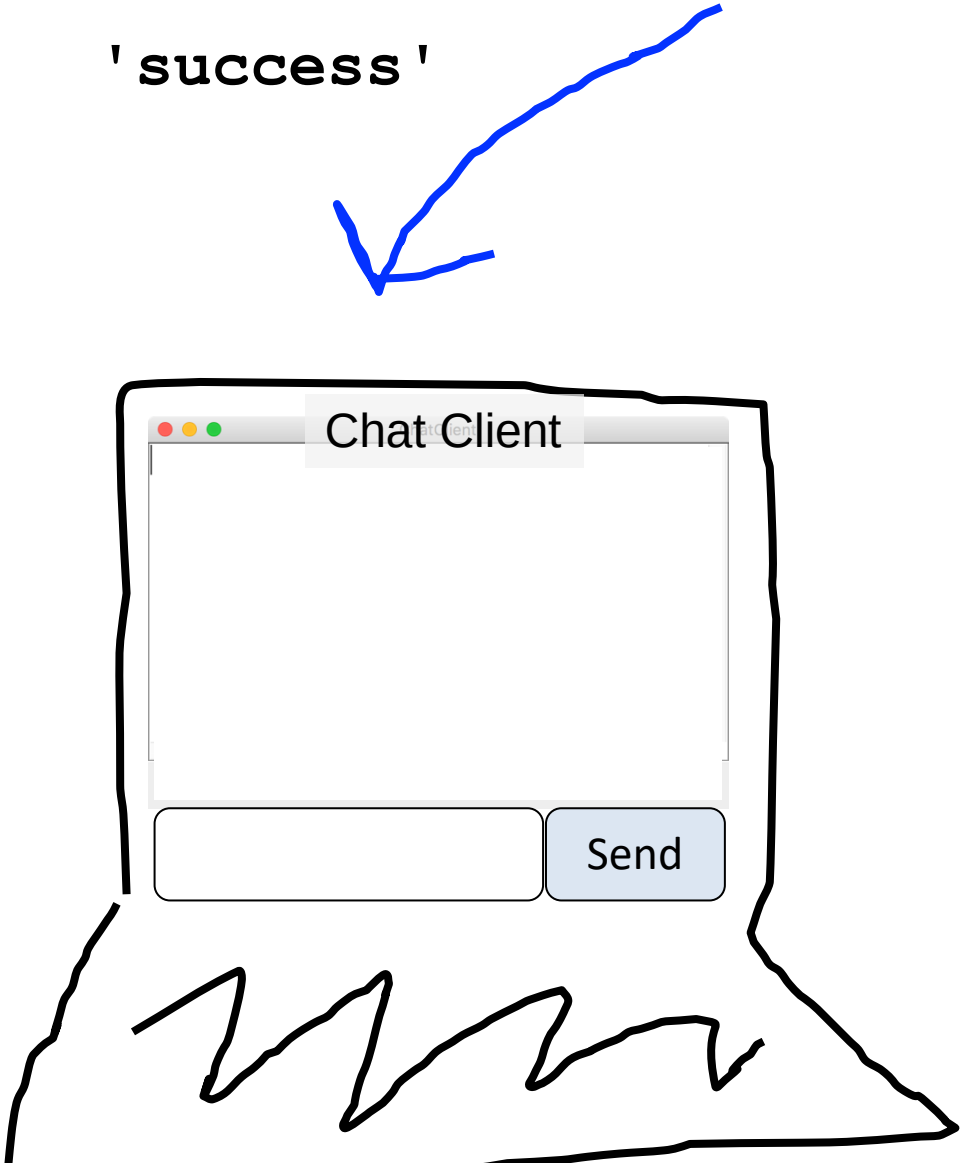


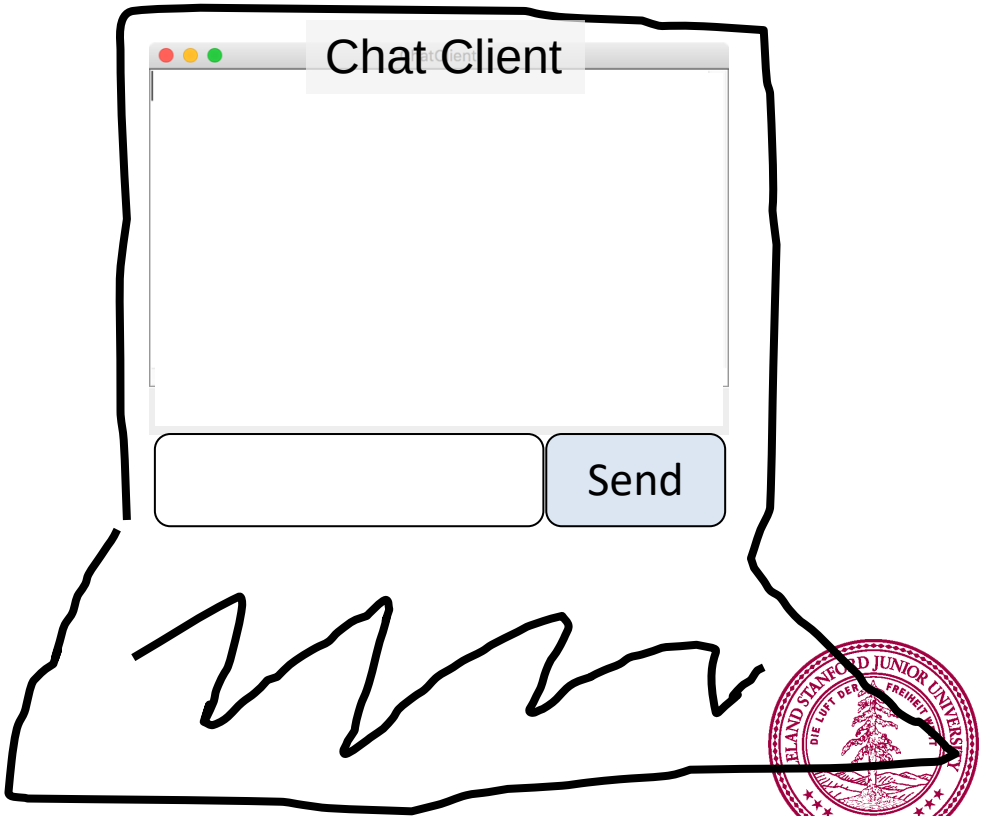




```
history = [
    '[C] Hello world'
]
```

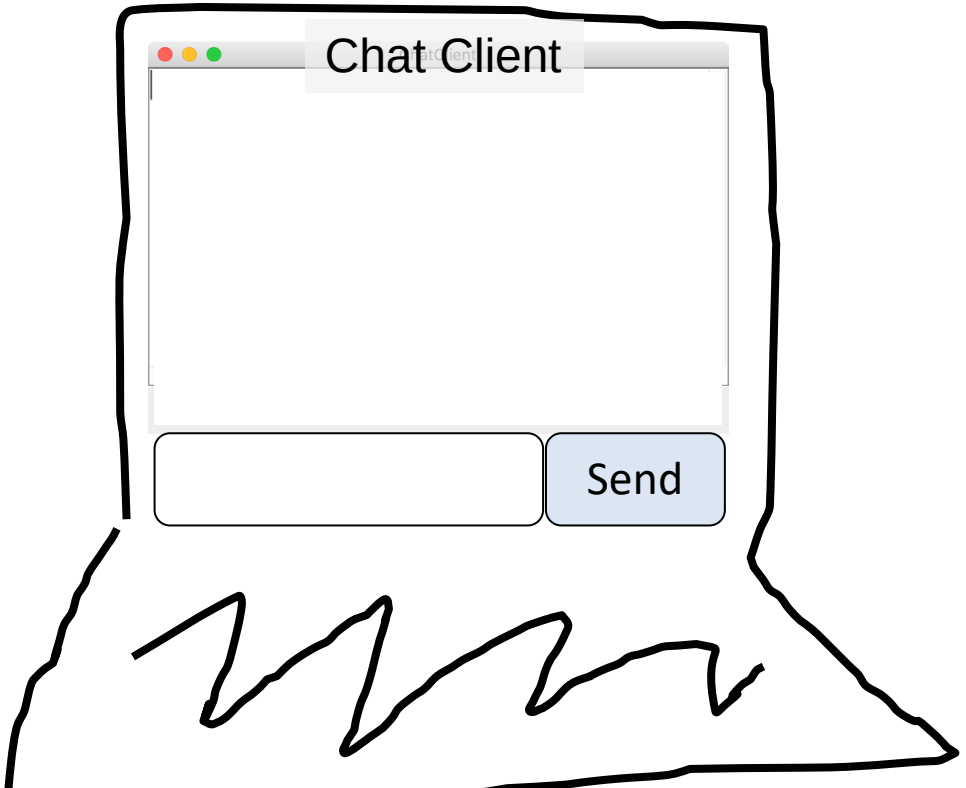
'success'

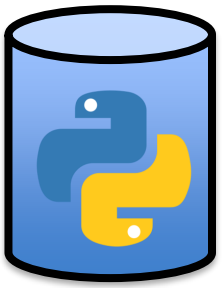




```
history = [
    '[C] Hello world'
]
```

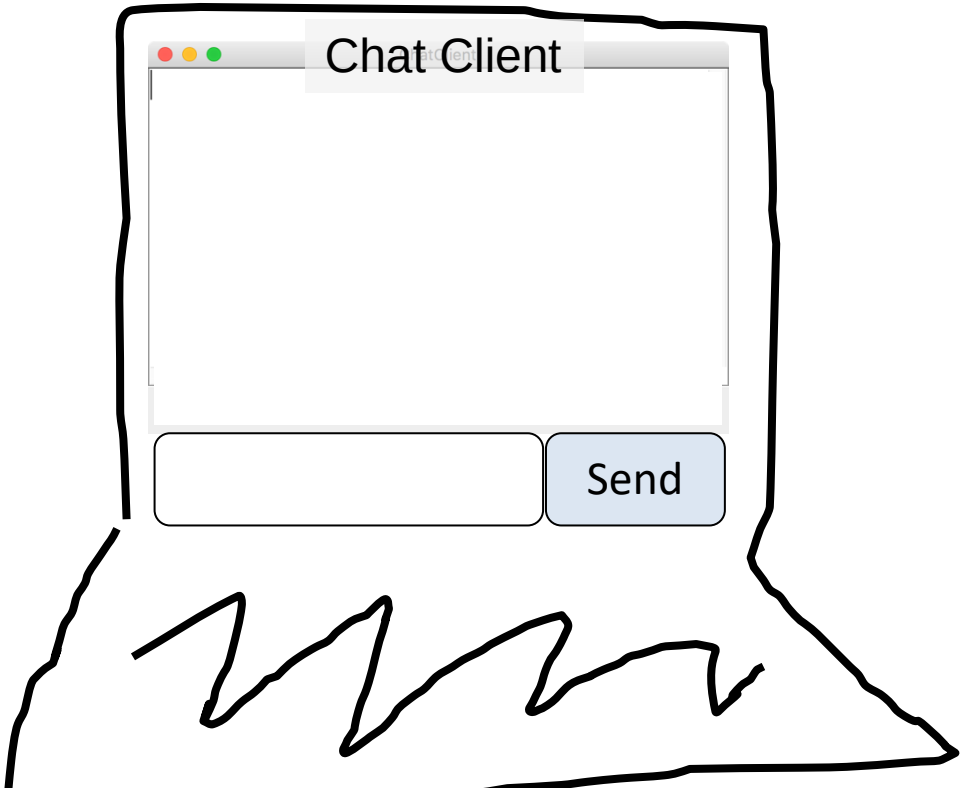
```
getMsgs
{
    'index' : 0
}
```

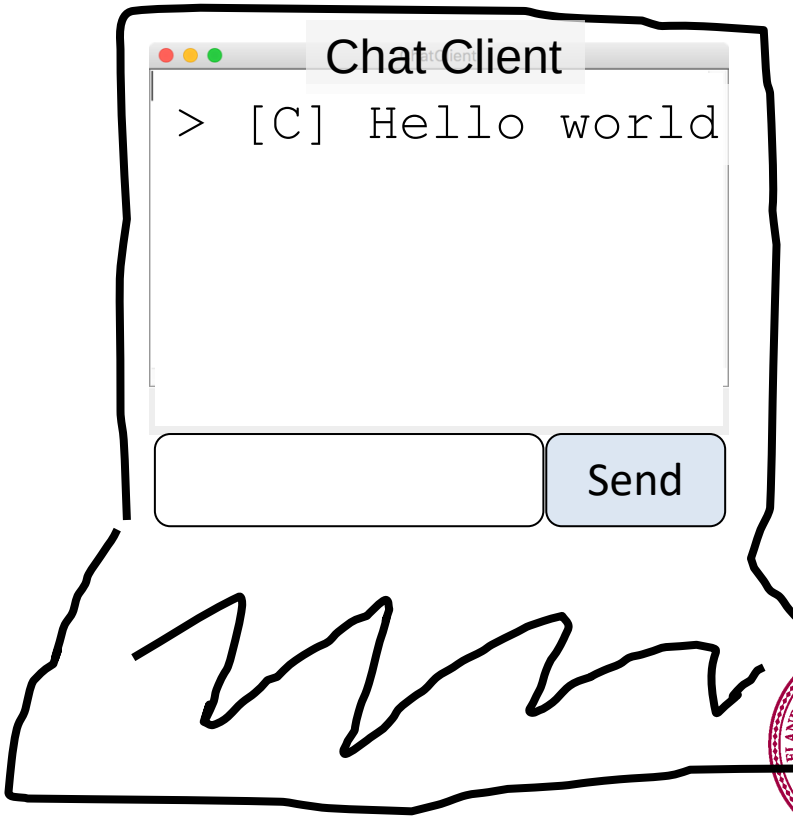


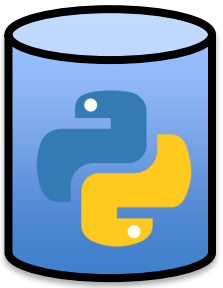


```
history = [
    '[C] Hello world'
]
```

```
'["[C] Hello world"]'
```

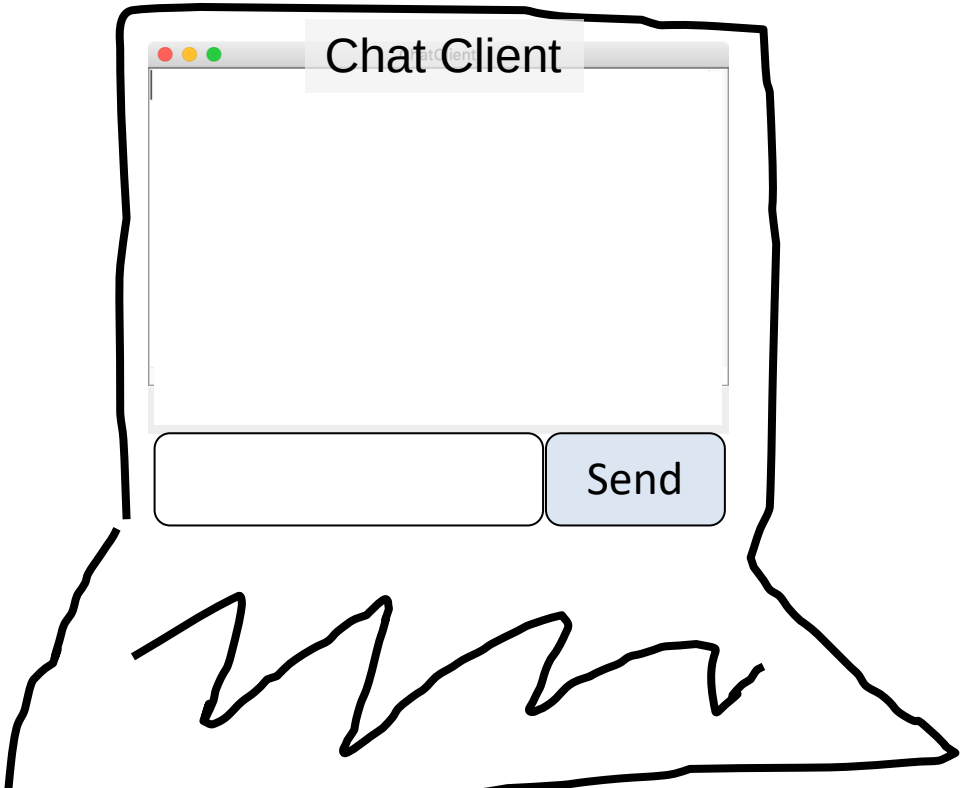


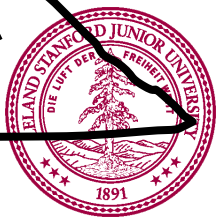


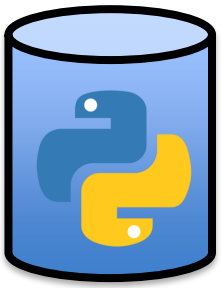


```
history = [
    '[C] Hello world'
]
```

```
newMsg
{
    'msg' : 'Im here too'
    'user' : 'B'
}
```

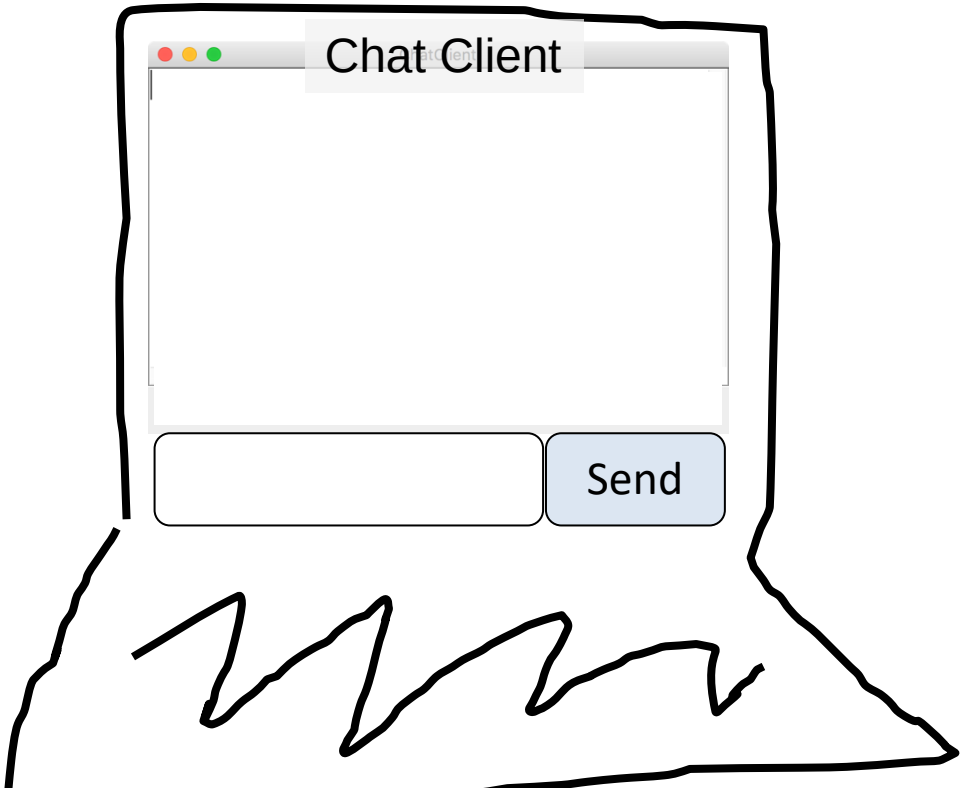


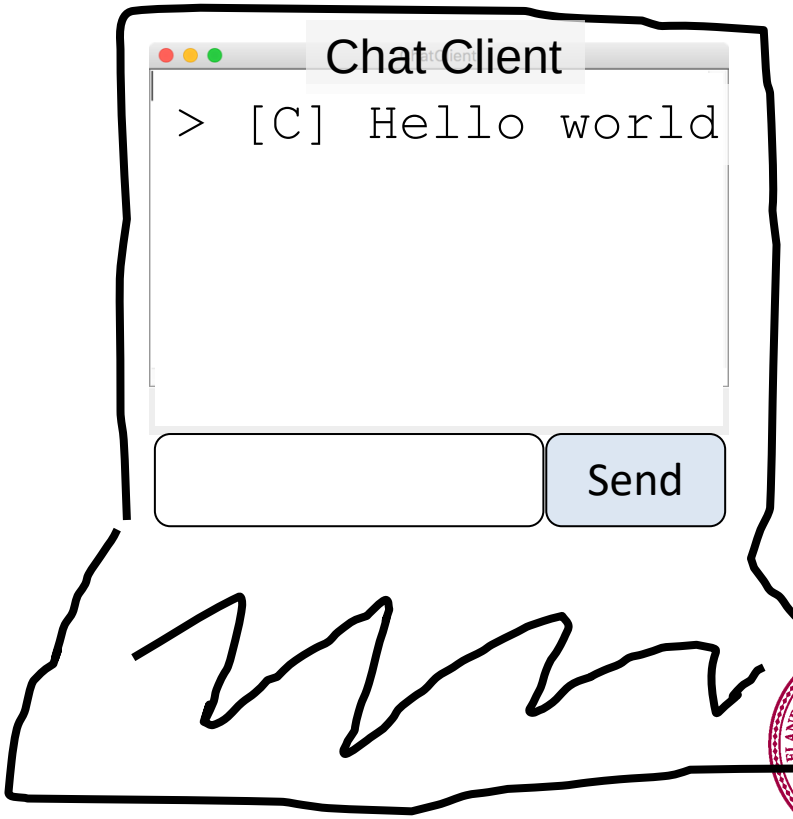


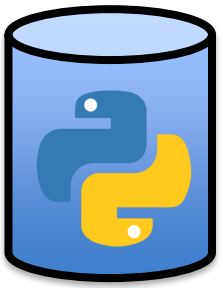


```
history = [
    '[C] Hello world',
    '[B] Im here too'
]
```

'success'

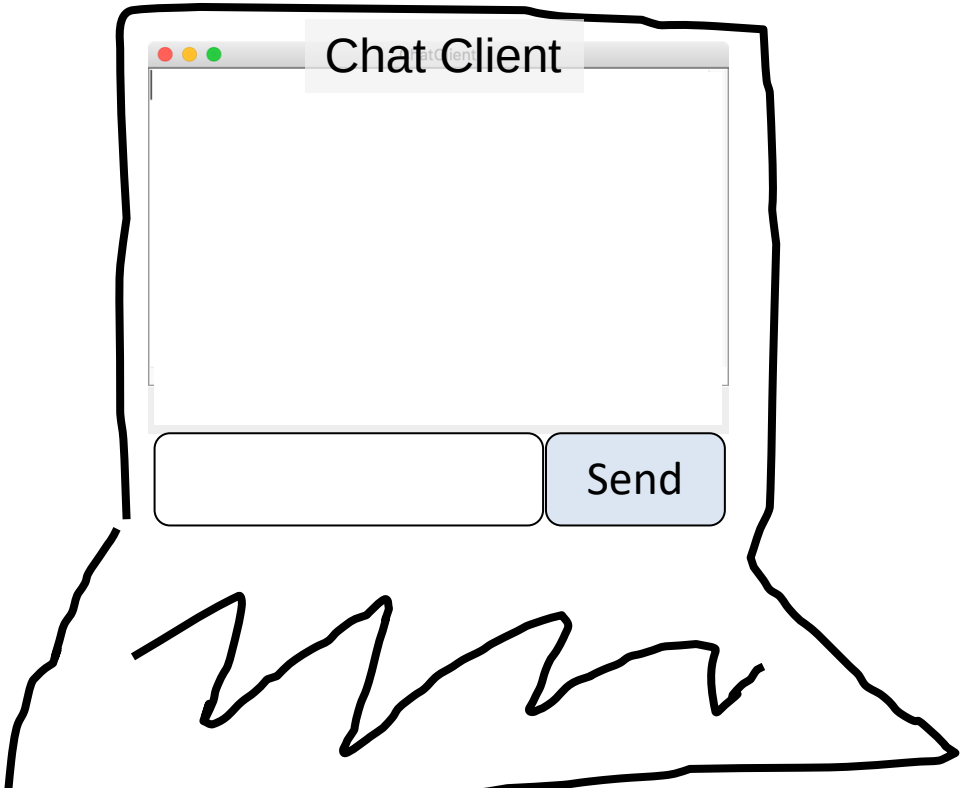


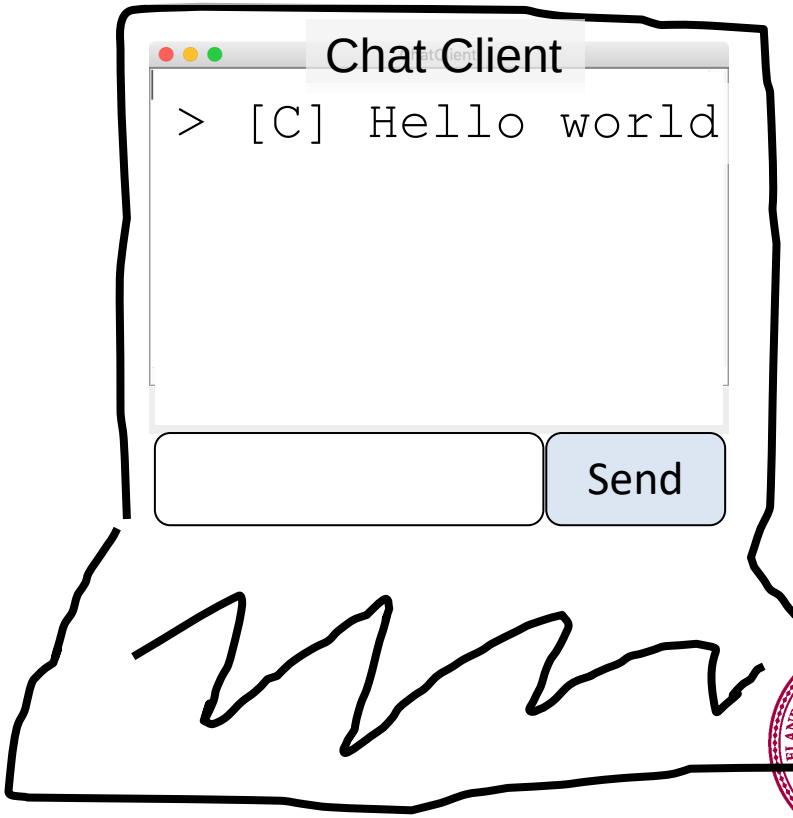


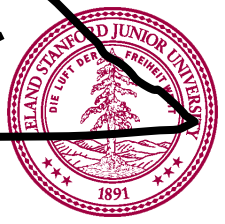


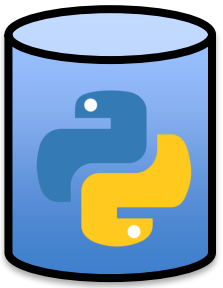
```
history = [
    '[C] Hello world',
    '[B] Im here too'
]
```

```
getMsgs
{
    'index' : 1
}
```



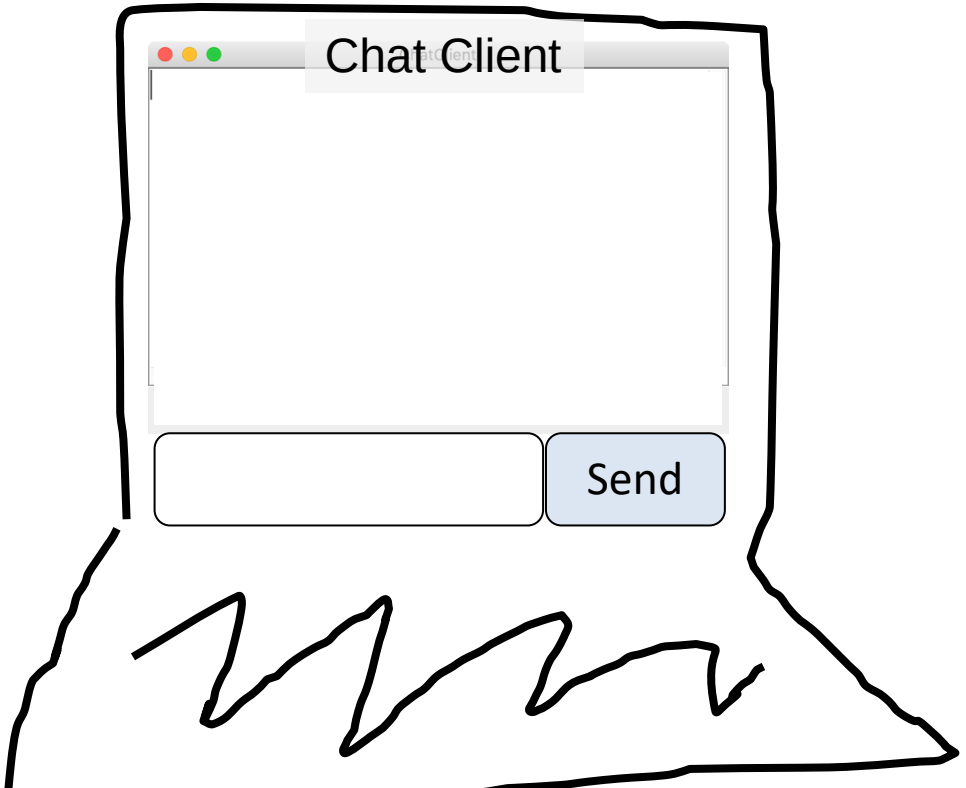


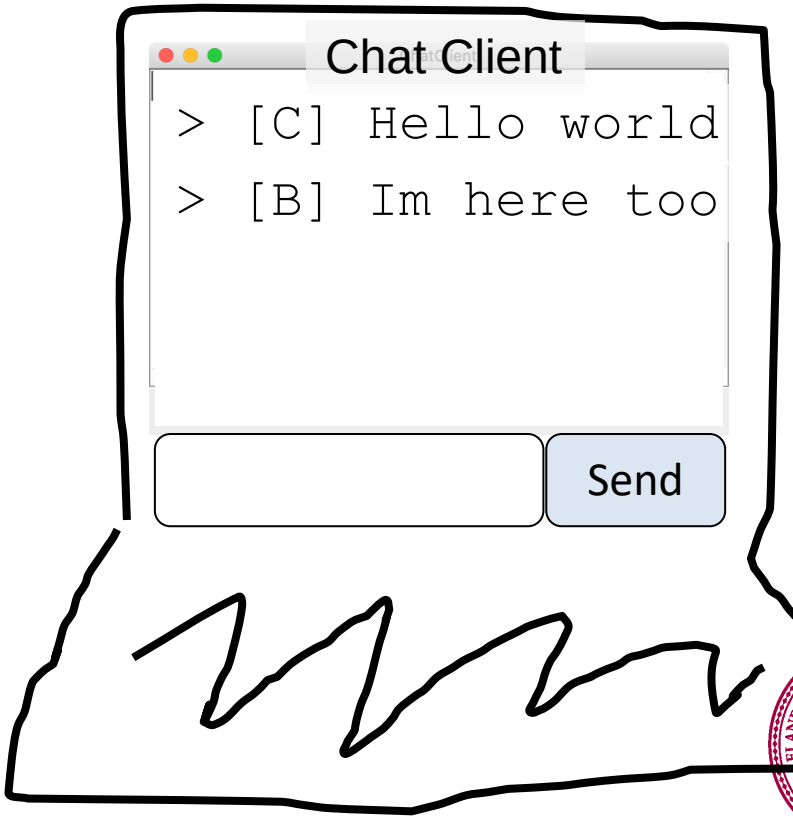


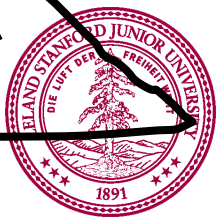


```
history = [
    '[C] Hello world',
    '[B] Im here too'
]
```

```
'["[B] Im here too"]'
```

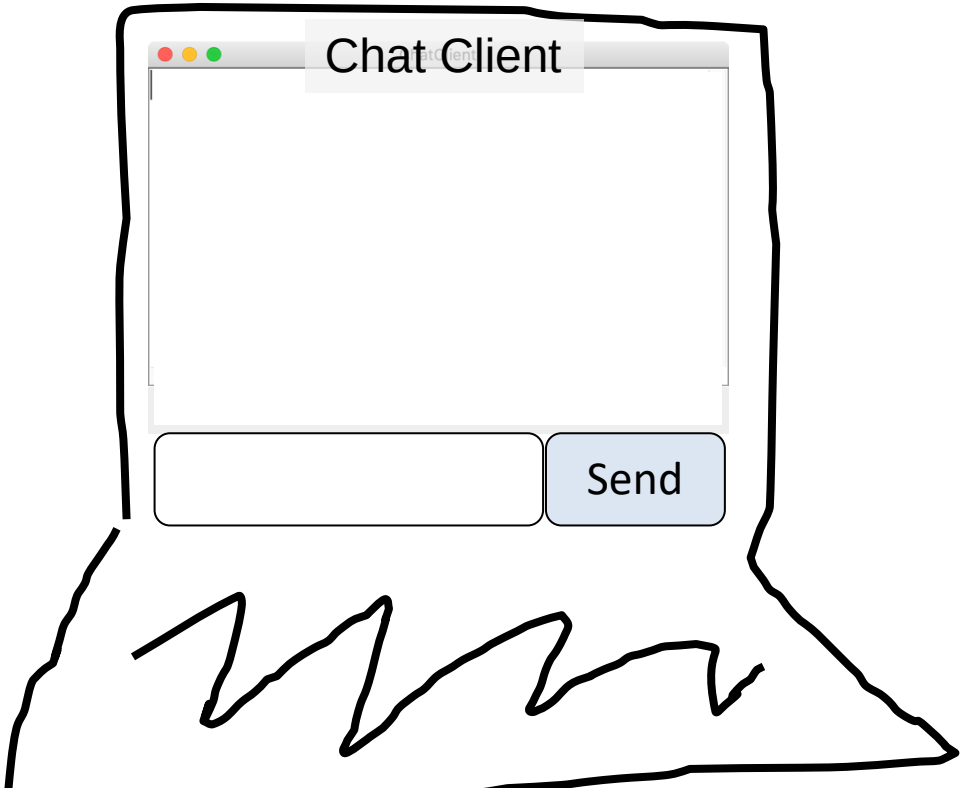


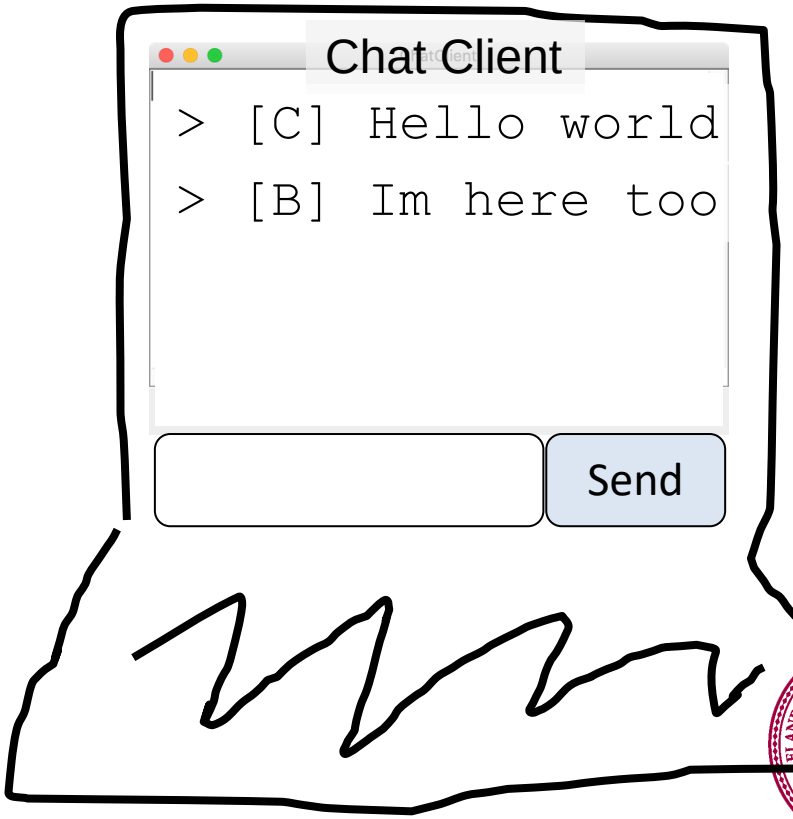




```
history = [
    '[C] Hello world',
    '[B] Im here too'
]
```

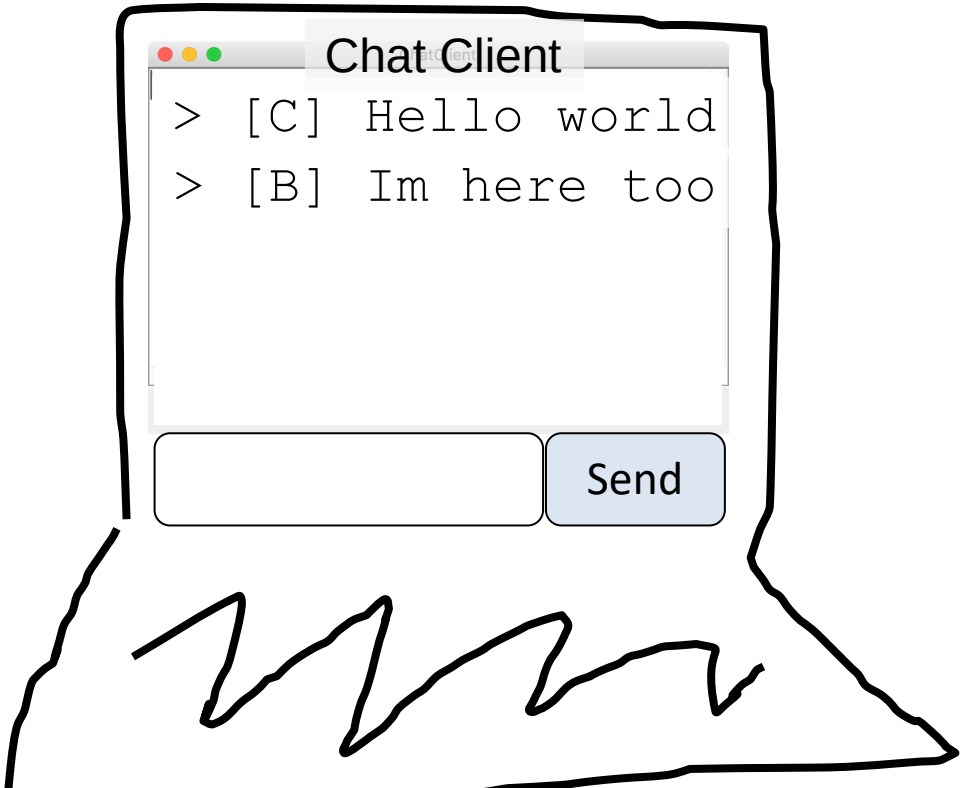
```
getMsgs
{
    'index' : 0
}
```

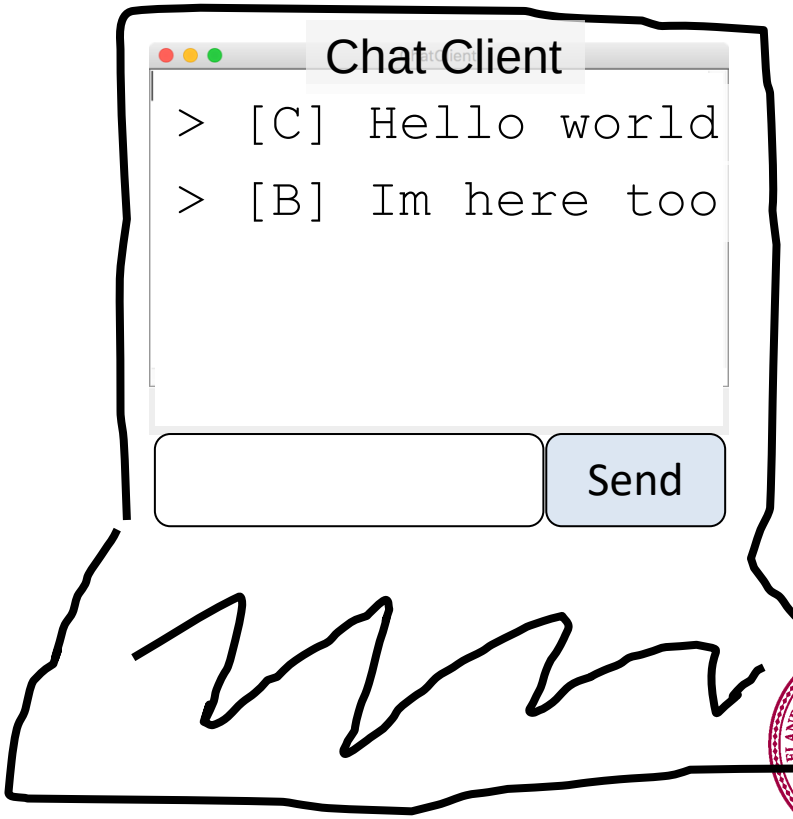




```
history = [
    '[C] Hello world',
    '[B] Im here too'
]
```

```
'["[C] Hello world",
  "[B] Im here too"]'
```



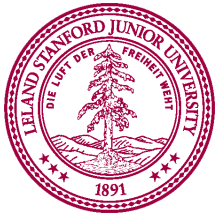


Chat Server

Chat Server

newMsg
msg = *text*
user* = *user

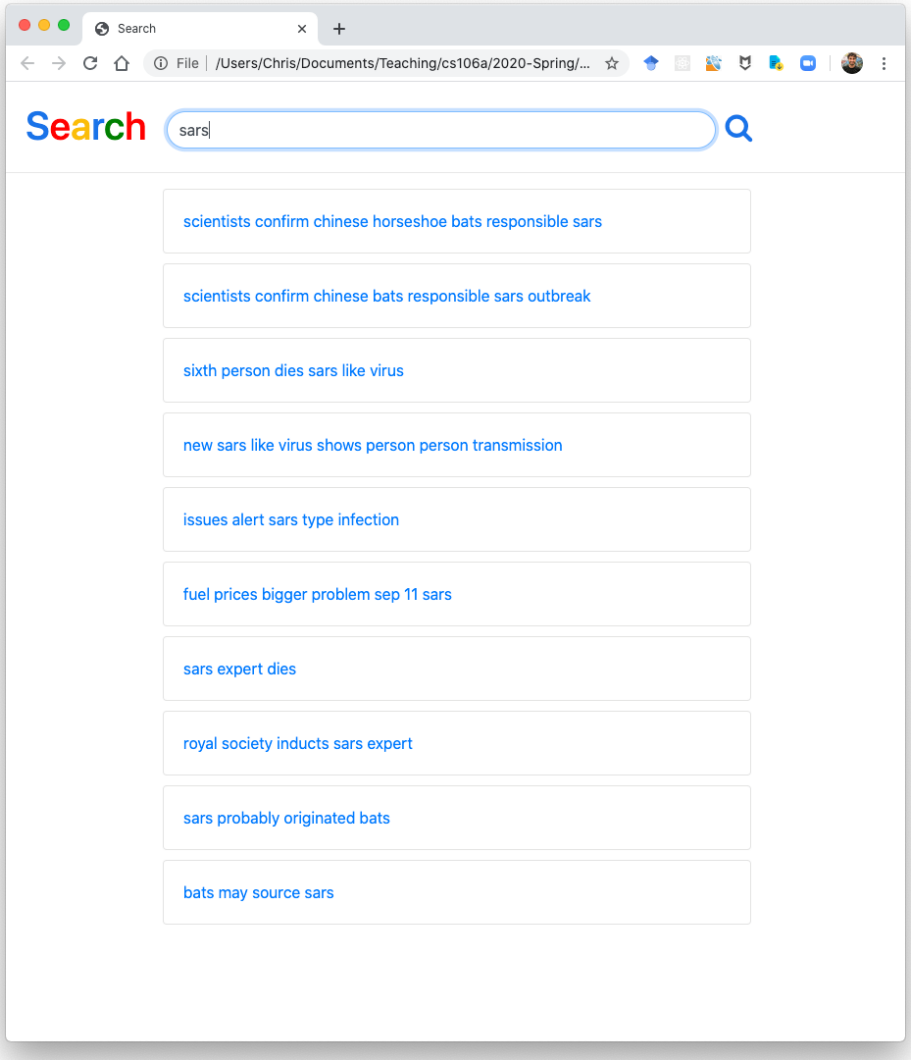
getMsgs
index = start_*index*



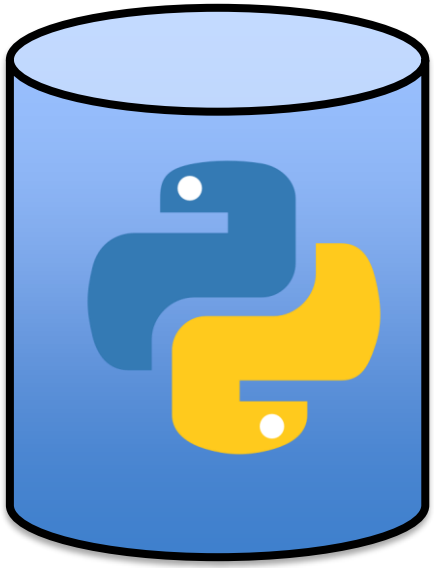
Let's take it out for a spin:

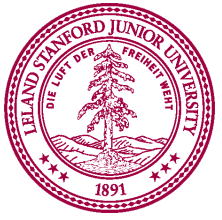
`chat_server.py`

Optional web server for Assign #7



Search Engine





Learning Goals

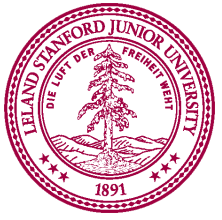
1. Write a chat program that can respond to internet requests

Life After CS106A!

Is there any?

“Any sufficiently advanced technology is indistinguishable from magic.”

—Arthur C. Clarke



Test Run Video

"It's a no-brainer that 50 to 60 years from now, cars will drive themselves"

—Sebastian Thrun

Faculty director, "Junior" autonomous car project quoted in Forbes, May 11, 2011

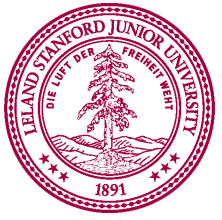
“Nevada has become the first state to issue an ‘autonomous’ license for a driverless car”

—USA Today, May 8, 2012

Google Self-Driving Car on El Camino Real
August 2015

Autonomous Waymo minivan in Los Altos
November 2017

Computing as a Growth Accelerant



“Google Data Center” Circa 1997

Image courtesy of Google

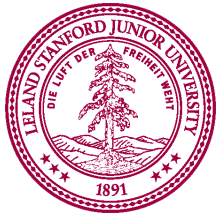
http://google.stanford.edu

Image courtesy of Google

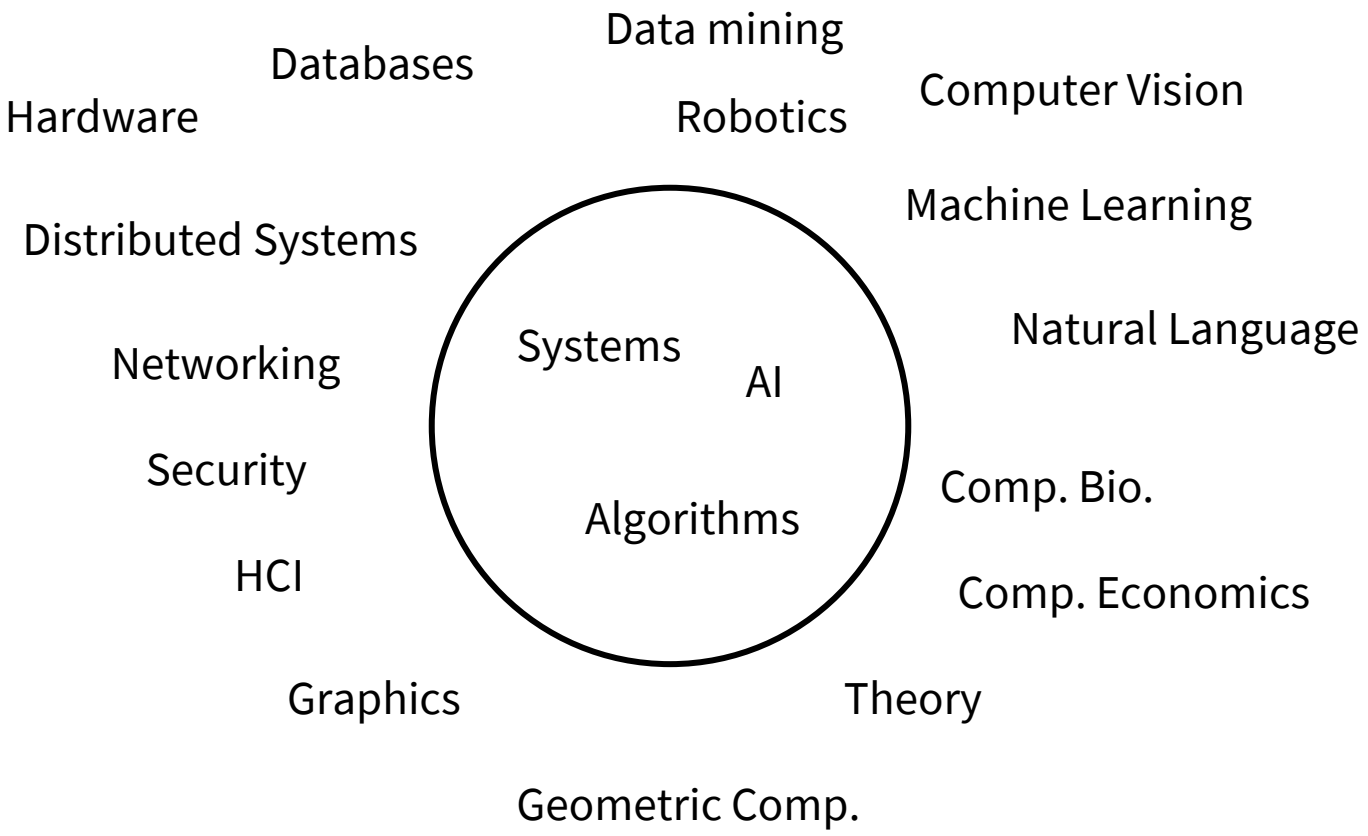
The Power of Computing

- Creating awareness of “CS in the large”
- Computing is increasingly needed for work in other fields
- Providing context for computing
  - Programming is a *means*, not an *end*

It’s about empowerment!

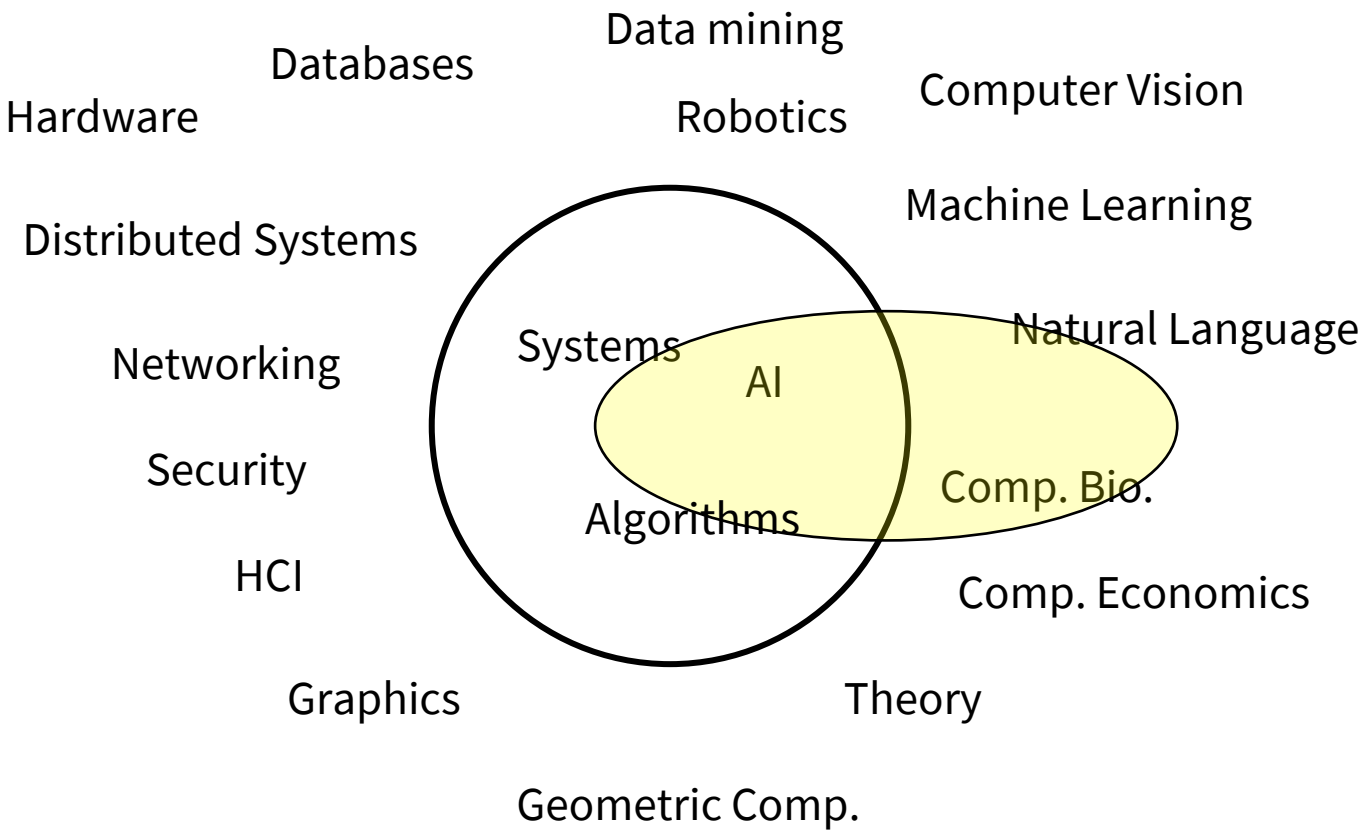


A Broad View of Computer Science

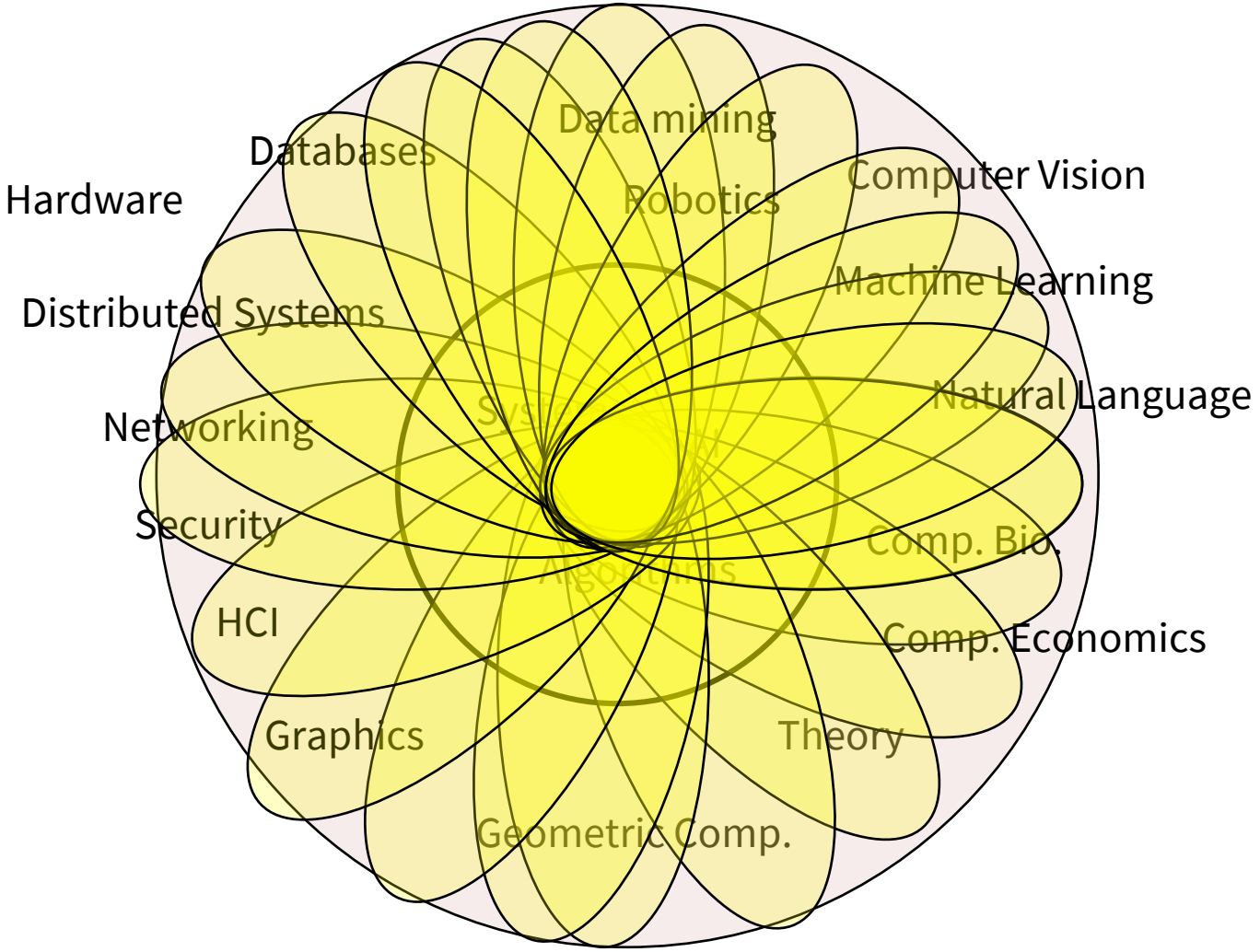


Editor's Note: Two-dimensional projection clearly does not capture the relative importance or organizational nuances of the field.
Some topics may be closer to you than they appear on this slide.

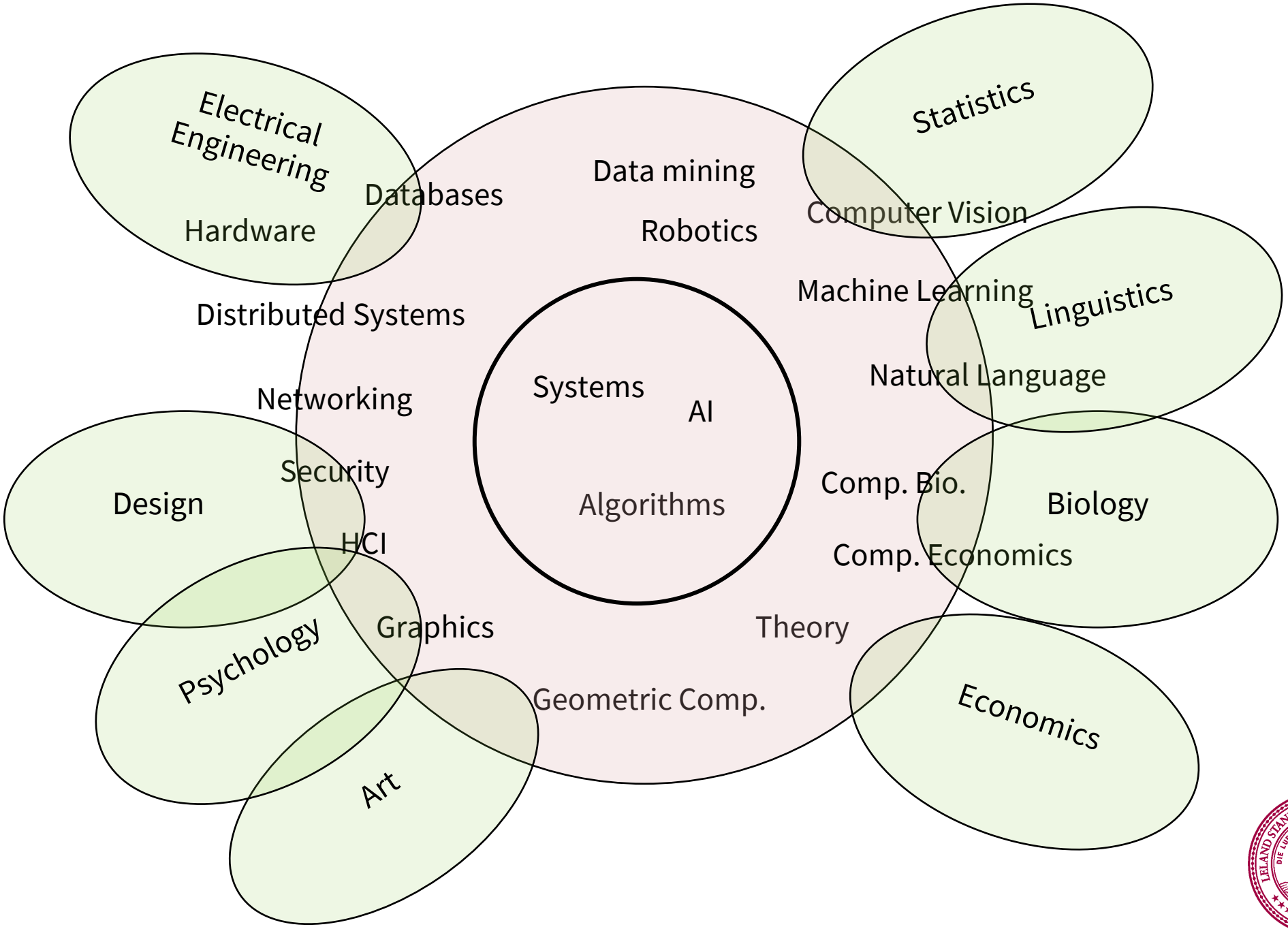
CS Major Allows Exploration



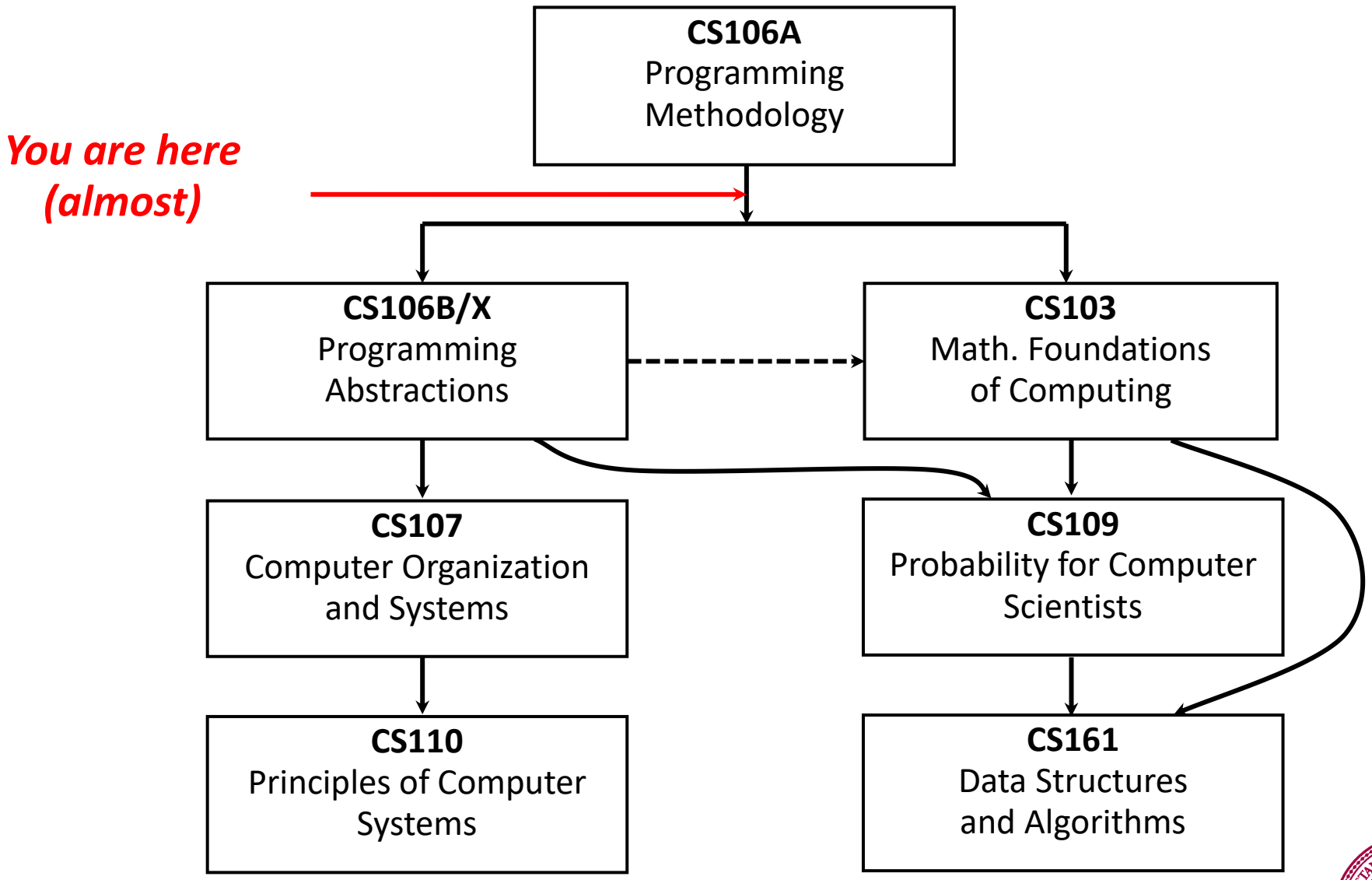
…in a Diverse Set of Areas



The "Big Tent" of Computer Science



CS Core Course Sequence

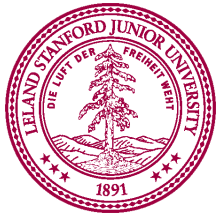


Track Areas

- Artificial Intelligence
- Theory
- Systems
- Computer Engineering
- Human-Computer Interaction
- Graphics
- Information
- Biocomputation
  - Incorporates many pre-medical school requirements
- Unspecialized
- *Individually Designed*

Sample of CS Research Areas

- Artificial Intelligence
  - Robotics, machine learning, computer vision, …
- Computational Biology
  - Bioinformatics, genomics, drug design, …
- Graphics
  - Animation, modeling, motion capture, architecture, …
- Databases and information systems
  - Web search, transaction management, data integrity, …
- Security
  - Cryptography, secure protocols, …
- Systems
  - Network design, cloud computing, virtualization, …
- Human-Computer Interaction
  - Interface design, user-centric computing, …

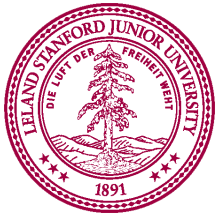


Sampling of Career Paths

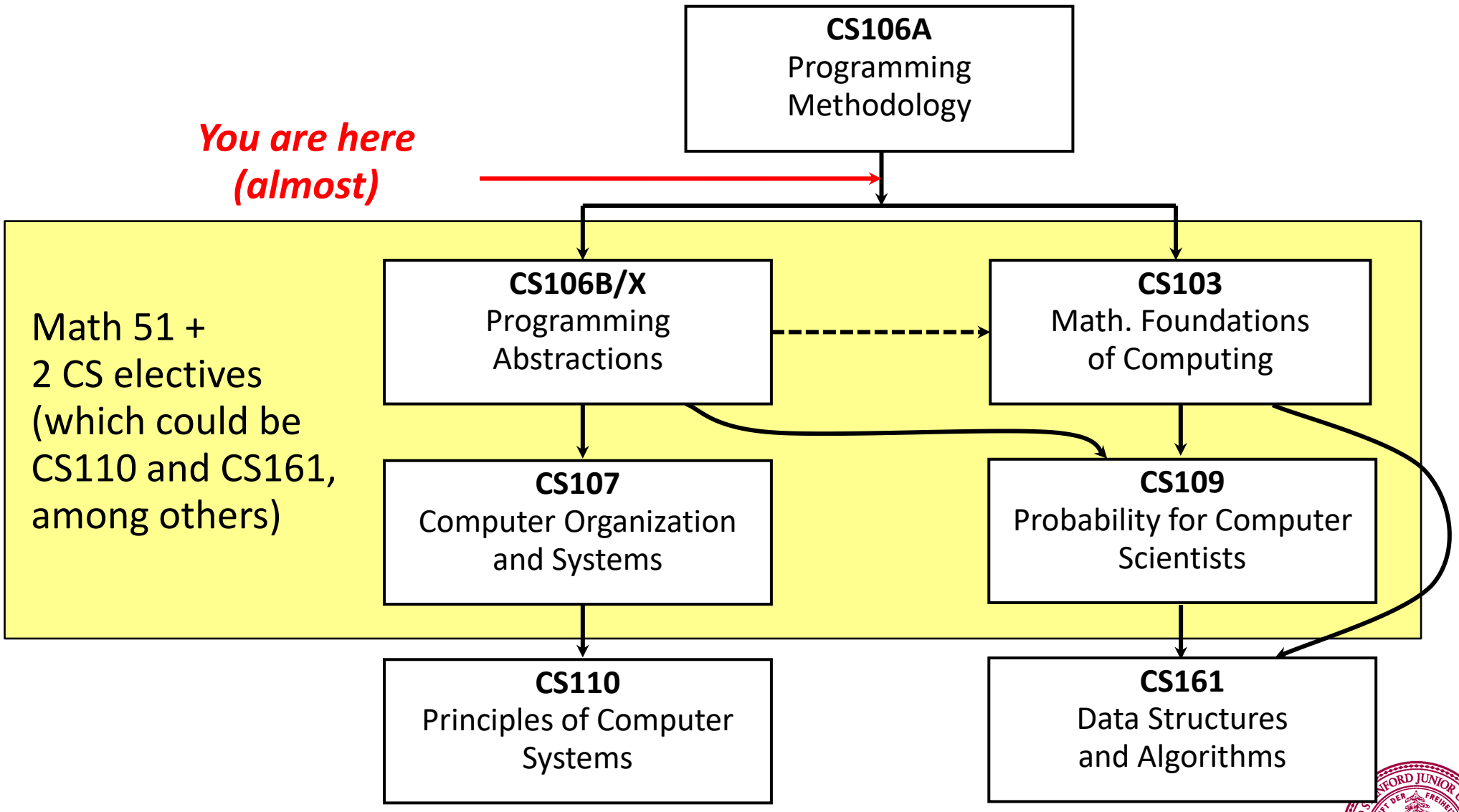
- High-tech industry
  - Research and development
  - Engineering management
  - Product management
- Entrepreneurship (consider Mayfield Fellows Program)
  - Start-ups (over 2,500 companies found by Stanford community)
  - Venture capital
- Graduate and professional schools
  - Graduate school → Academia/research/teaching
  - Law school → Public policy (consider CS181/182)
  - Business school → Management/entrepreneurship
- Teaching (consider CS198)

CS Minor

- Math through Math 51
- Required:
  - CS106B
  - CS107
  - CS103
  - CS109
- Two additional CS elective courses



CS Minor


CS106A
Programming
Methodology
You are here
(almost)
Math 51 +
2 CS electives
(which could be
CS110 and CS161,
among others)
CS106B/X
Programming
Abstractions
CS103
Math. Foundations
of Computing
CS107
Computer Organization
and Systems
CS109
Probability for Computer
Scientists
CS110
Principles of Computer
Systems
CS161
Data Structures
and Algorithms

Related Majors

- Math and Computational Science (Data Science)
  - Math, CS, Statistics, and MS&E, …
  - Tracks in Biology, Engineering, Statistics
- Electrical Engineering
  - Hardware, information science, analog/physical systems, …
  - Tracks in: areas above as well as Bio-EE, Green-EE, Music-EE
- Symbolic Systems
  - CS, Philosophy, Linguistics, Psychology. …
  - Tracks in: Logic, AI, Cognitive Science, Computer Music, Decision-Making, Human-Computer Interaction, Learning, Natural Language, Neurosciences, Philosophical foundations

Everyone is Welcome

A Now Some Fun Movies…

- Lighthouse
- Fireball
- Curtain
- Robot