Evolutionary Processes and Systems

Una-May O'Reilly, PhD
Erik Hemberg, PhD
The Alfa Group
Massachusetts Institute of Technology, USA

We are excited to be here!

http://sjsdblogs.com/jordansblog/files/2013/01/excited-face-cartoon-i0-2gi4cpx.png

http://vecto.rs/1024/vector-of-a-happy-cartoon-businessman-running-with-a-big-smile-on-his-face-by-ron-leishman-34700.jpg

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

How we learn

and

What we learn

are

Both Important

ALFA

ANYSCALE LEARNING FOR ALL

CSAIL

There is No Wrong Answer

http://jolynproject.files.wordpress.com/2012/08/bebold.jpg

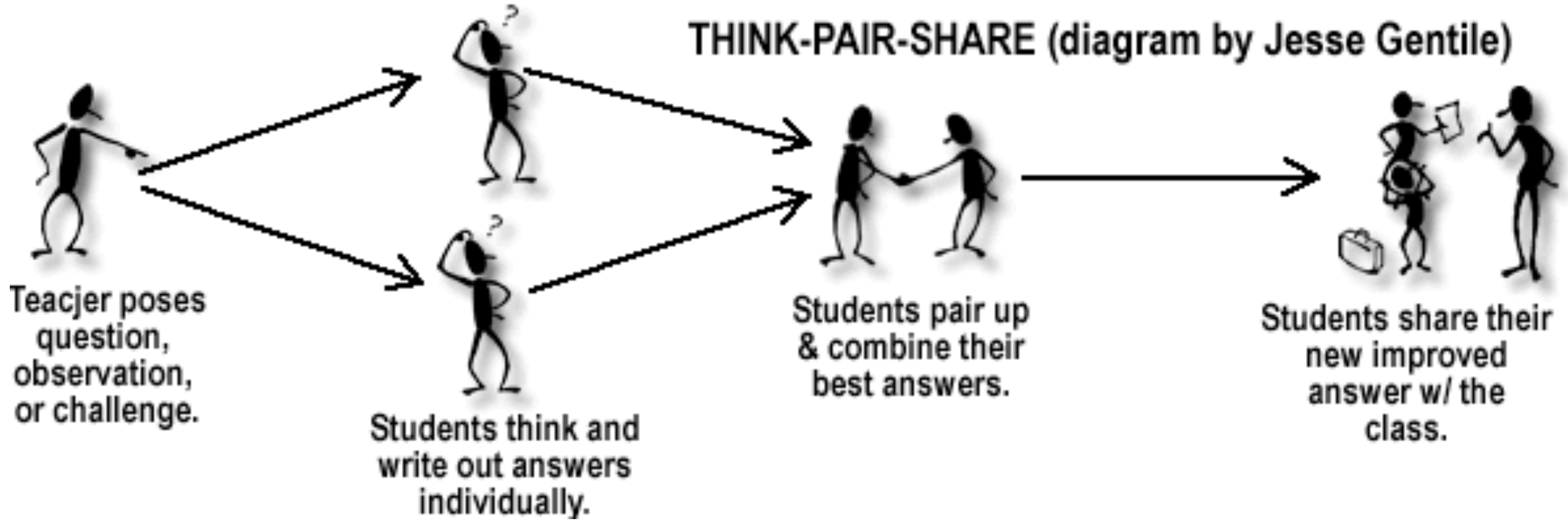
Be Bold!

http://jolynproject.files.wordpress.com/2012/08/bebold.jpg

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

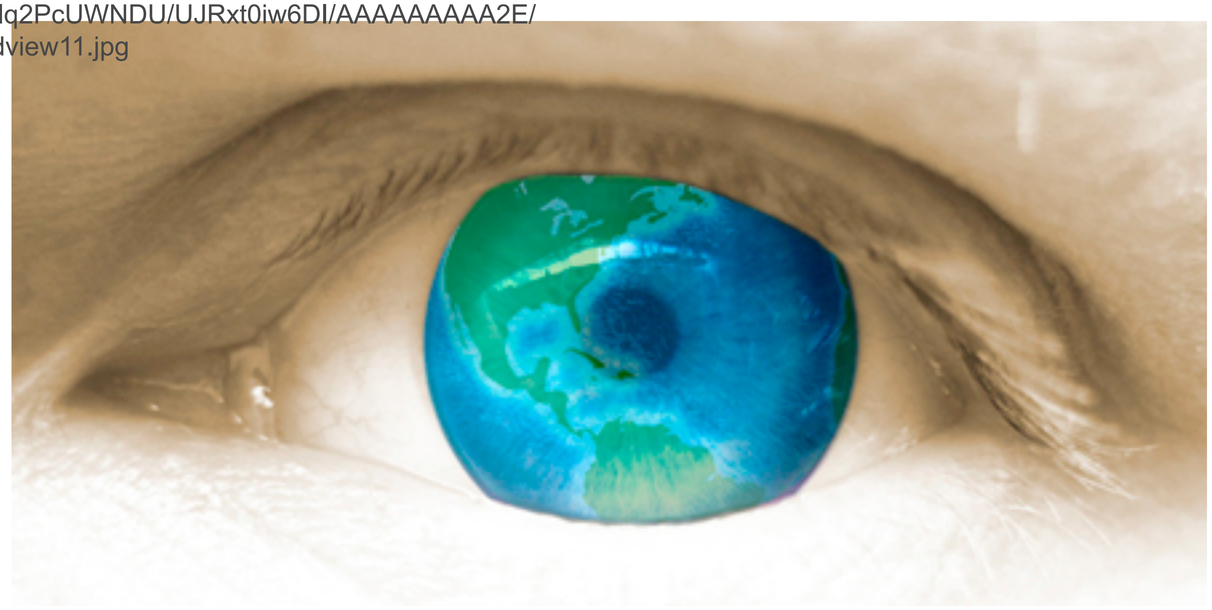
Active Learning


THINK-PAIR-SHARE (diagram by Jesse Gentile)
Teacjer poses question, observation, or challenge.
Students think and write out answers individually.
Students pair up & combine their best answers.
Students share their new improved answer w/ the class.

http://inforatiblog.files.wordpress.com/2012/11/lpthinkpairshare2.gif

We will expand our worldviews

http://3.bp.blogspot.com/-qlq2PcUWNDU/UJRxt0iw6DI/AAAAAAAAA2E/arLhmSrHVE8/s1600/worldview11.jpg



Definition

- WorldView: an intellectual perspective on the world or universe.

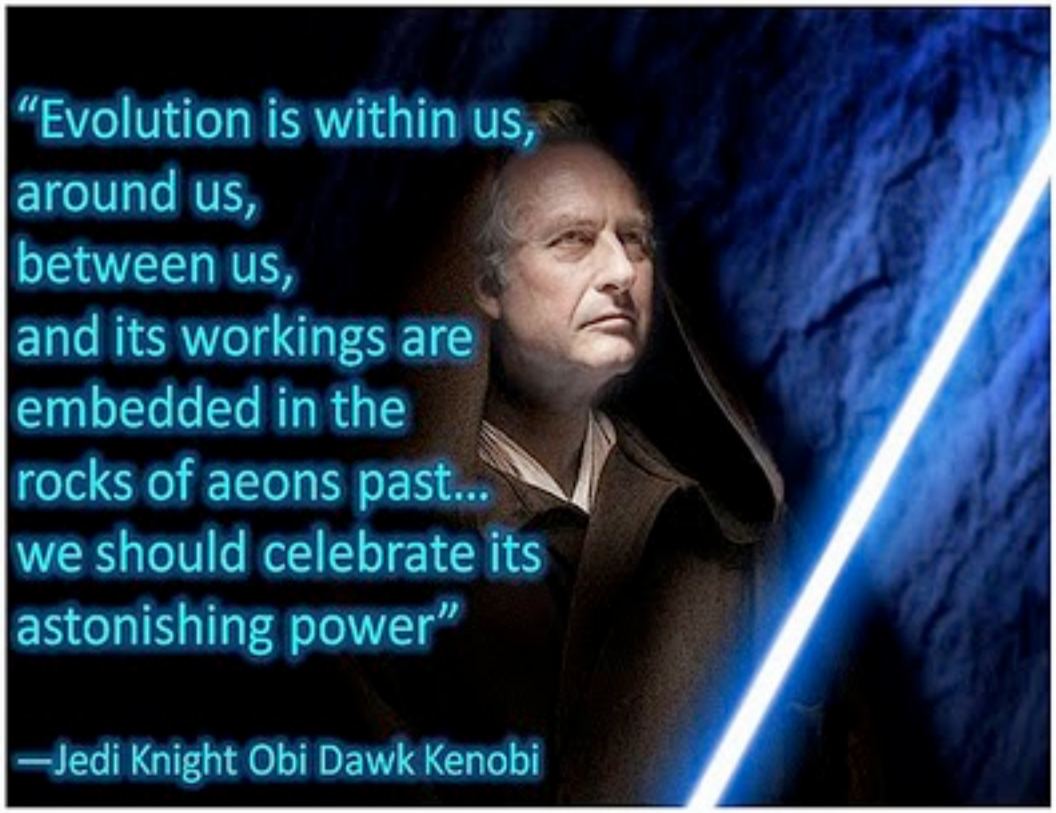
ALFA ANYSCALE LEARNING FOR ALL

CSAIL

We seek to expand our Scientific World View

http://www.truthandscience.net/earth_analysis.gif

We seek to develop our Darwinian World View



Let's Get Started

About You

Turn in a piece of paper with:

- Name, Major
- Name your strongest programming language:
  - Python, C, Java, AppInventor?
- What is your skill level?
  - Expert (I wrote the book), Average, Beginner

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

About Us

Dr O'Reilly

unamay@csail.mit.edu

Dr Hemberg

hembergerik@csail.mit.edu

Dr. Una-May

MIT School of Engineering

ALFA ANYSCALE LEARNING FOR ALL

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

Dr. Erik

MIT School of Engineering

ALFA
ANYSCALE LEARNING FOR ALL

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

All of Us

- Dr O'Reilly
  - unamay@csail.mit.edu
- Dr Hemberg
  - hembergerik@csail.mit.edu
- `meijuan `yan(Carah)
  - mjyan@stu.edu.cn
- Xiaozhong Peng (David)
- Qian Liao (Sunny)

ANYSCALE LEARNING FOR ALL

CSAIL

Assessment

- No grades
- Certificate of achievement if the following are completed:
  - Participation (oral discussions)
  - Completion of programming tasks (technical)
  - Reflective journal to be turned in at end of module (written)
    - Daily entry:
      - What have I learned?
      - What would I have done differently?
      - What should my instructors do differently?
  - Evolutionary Gems Presentation (oral presentation)

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

Lecture and Lab Schedule

- **Monday**
  - **Lecture 8:00-8:50; Break 8:50-9:00; discussion part of lecture: 9:00-9:50**
- **Tuesday**
  - **Lab 19:00-21:00**
- **Wednesday**
  - **Lecture 8:00-8:50; Break 8:50-9:00; discussion part of lecture: 9:00-9:50**
- **Thursday**
  - **Lab 19:00-21:00**
- **Friday**
  - **Lecture 8:00-8:50; Break 8:50-9:00; discussion part of lecture: 9:00-9:50; Module wrap up: 10:00-10:50.**

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

MIT Staff Office hours
IN SOFTWARE Engineering LAB 302

- **Monday 1400 – 1700 (just Dr. Hemberg)**
  - **You will have a programming exercise**
- **Tuesday 0800 – 1000**
  - **Check in on programming exercise**
  - **Demonstrate it running**
- **Tuesday 18:00-19:00**
  - **To demonstrate first programming task**
- **Wednesday 1400 – 1700, 1930 - 2130**
- **Thursday 1800 - 1900**

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

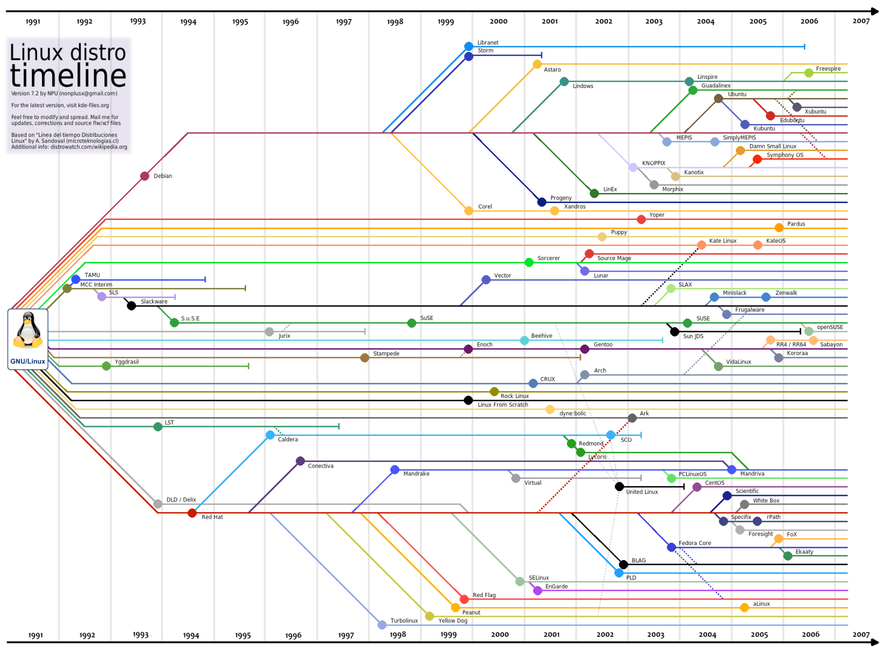
Syllabus for Students

- L1.1 (Monday 8:00 -8:50)
- L1.2 (Monday 9:00 – 9:50)
  - Journal exercise given
    - Due Friday
  - Coding exercise given
    - Due Tuesday, 19:00
- Lab 1 (Tuesday 19:00-21:00)
  - 2nd coding exercise given
    - Due Friday, 8:00 am
- L2.1 (Weds, 8:00- 8:50)
  - Oral Exercise given
    - Due Friday, 8:00 am
- L 2.2 (Weds, 9:00- 9:50)
- Lab 2 (Thurs, 19:00-21:00)
  - Time to work on 2nd coding exercise
- L3.1 (Fri, 8:00 – 8:50)
  - Check exercises
- L3.2 (Fri, 9:00-9:50)
- WRAP UP (Fri, 10-10:50)

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

Evolution in Action



- There are OS, one kind is Linux, who know about Linux? It's interesting because it's open source

  Started in 1991

  Quick expansion of version

  Versions expanded into multiple versions, a little different

  Some become extinct – die off because user groups favor some over others

  At present we have versions which are in some ways incompatible – package handling RedHat and Debian, directory layout

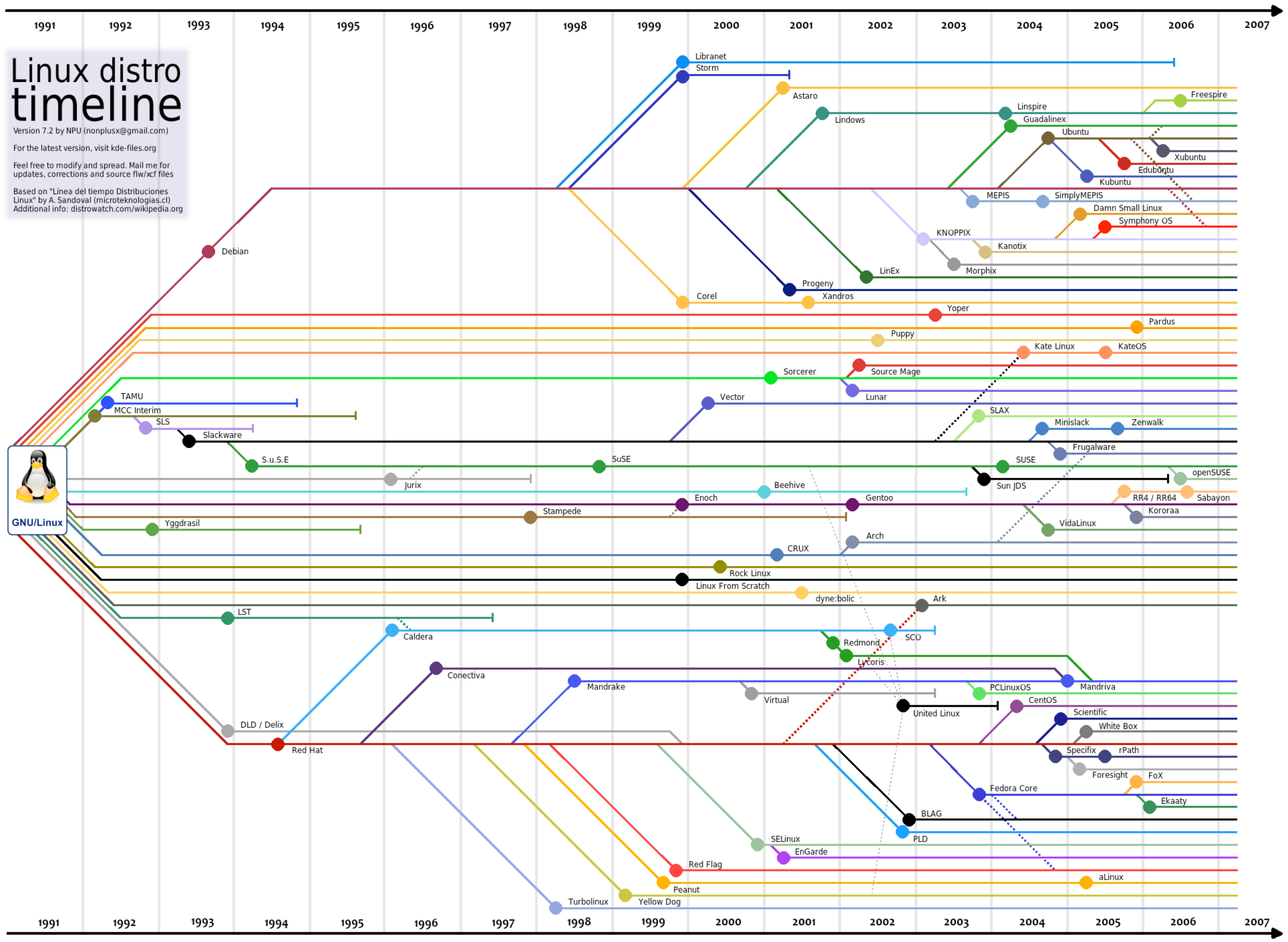
  can't always port code running on RedCode to Debian

  So over time, I would argue that OS's and Linux have evolved
- There's an evolutionary process going on here
- Can you think of your own examples of evolved systems? FROM ANYWHERE! That are different from each other

ANYSCALE LEARNING FOR ALL

CSAIL

http://cdn.techpp.com/wp-content/uploads/2013/02/44218-linuxdistrotimeline-7.2.png



ANYSCALE LEARNING FOR ALL

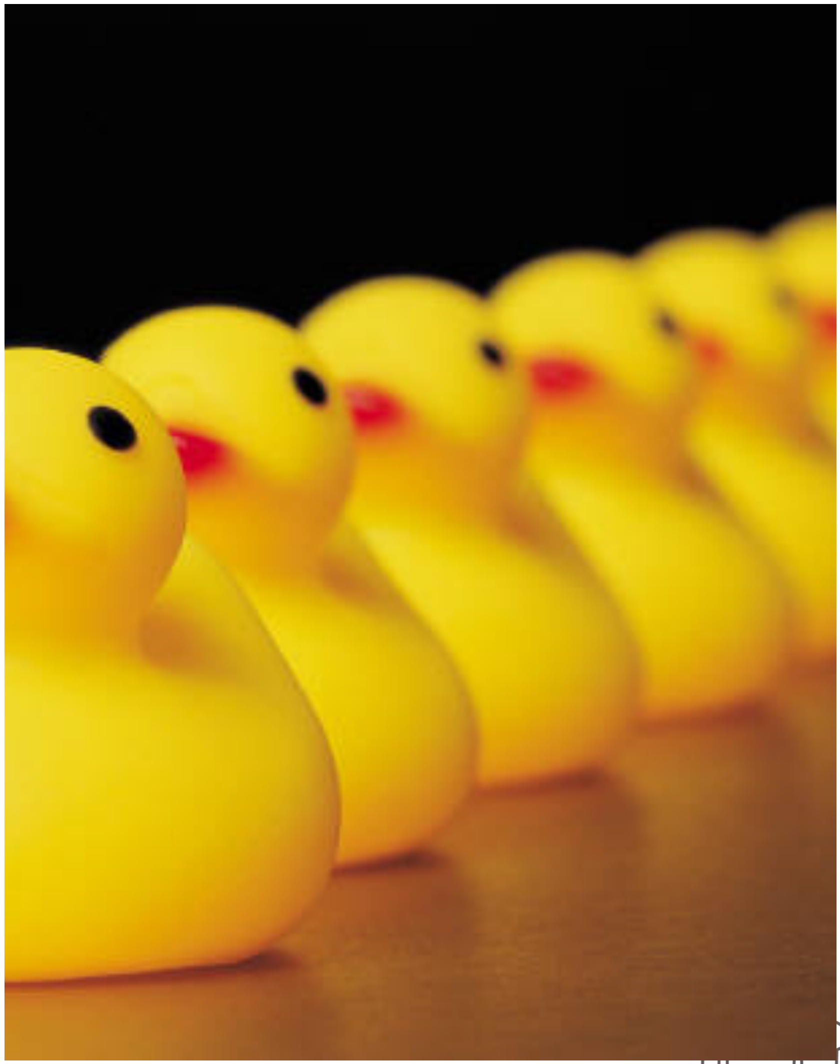
CSAIL

Our First Class Exercise

Part 1: Individual reflection

- Give examples of evolution in action
- Questions to ask
  - What is it that evolves?
  - How does it evolve?
    - How has it been evolving?
    - Which parts have been evolving?
  - When do things evolve?
    - When has it been evolving?
  - Why is this an evolutionary process?

- The aim is to find and discuss what you consider evolutionary processes and then figure out the common properties
- 10 minutes



Evolution in Action

Part 2: 10 minutes

- Pair up
- Discuss your individual examples
- Decide on 2 good examples of evolutionary processes

Part 3: Team up 1 more time [10 minutes]

- Merge into groups of 6
- Discuss the examples from the pairs
- Decide on 2 examples that the group will present
  - Pick a presenter
  - Help them with White board presentation plan
  - After break: present to other half of class – 3 minutes

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

Evolution in Action Follow Up

Use these questions to guide today's journal entry

- Be brief
- Sketch or bullets

Can you answer the following questions?

- What is the difference between evolution and trial and error?
  - Examples of each
- Why do things evolve?
- How do things evolve?
- When do things evolve?

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

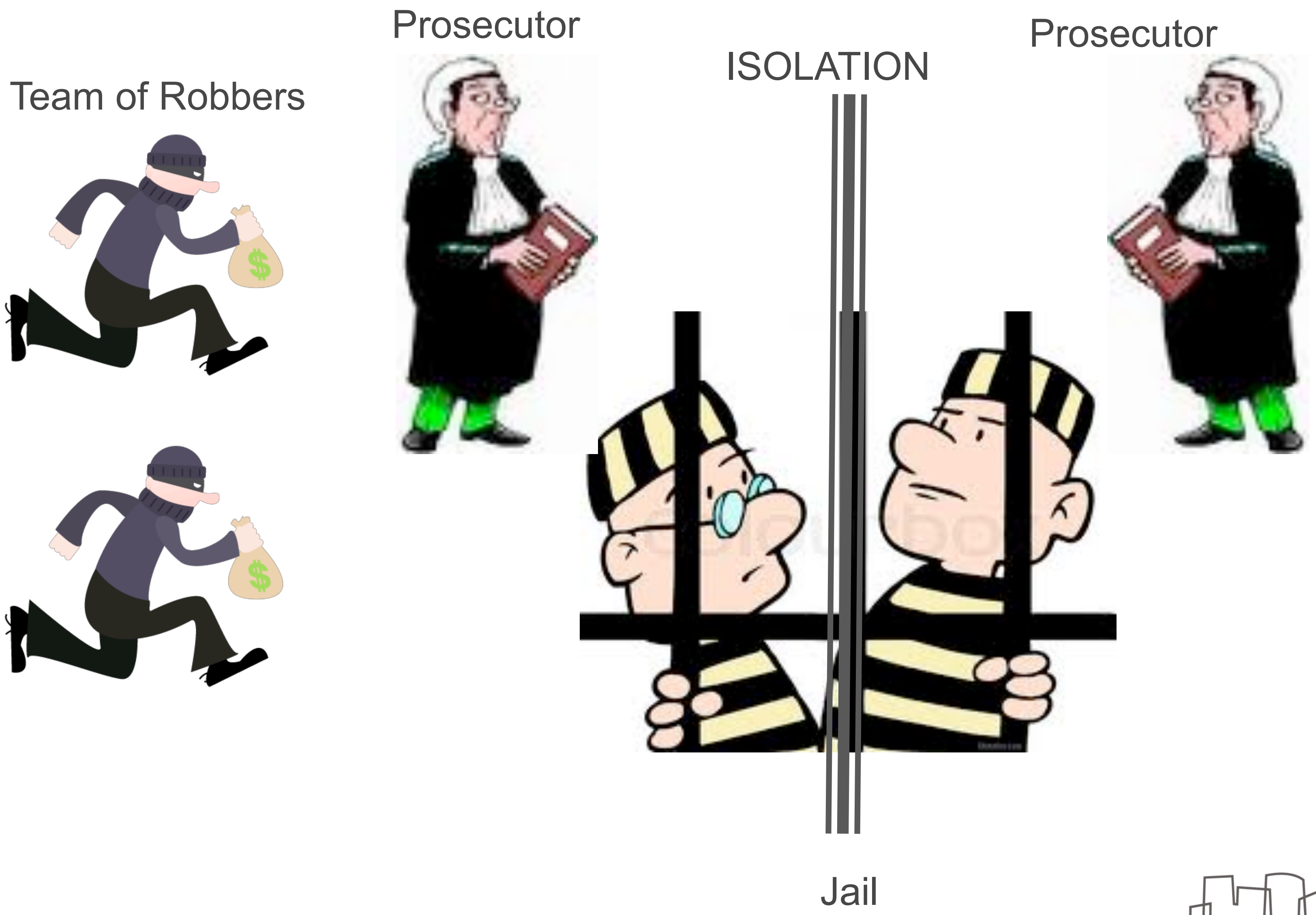
Prisoner's Dilemma

- Alice and Bob have been arrested for robbing a bank. Agree not to say anything if caught
- They are put in isolation
- Both are selfish
- Prosecutor says: maximum penalty is 3 years
  - Confess or be Silent
  - To Alice:
    - If you confess and Bob is silent
      - You go free and he spends 3 years in jail
    - If Bob confesses and you are silent
      - He goes free and you spend 3 years in jail
    - If you both confess
      - Both get 2 years
    - If you both stay silent
      - Both get 1 year
    - I will offer Bob same terms
- If you are Bob or Alice, what will you?

ANYSCALE LEARNING FOR ALL

CSAIL

Prisoner’s Dilemma



ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

Prisoner’s Dilemma Formalized – Cooperate, Defect

- **Prosecutor says:**
  - **Confess or be Silent**
  - **Maximum penalty: 3 years**
  - **To Alice:**
    - » **If you confess and Bob is silent**
      - **You go free and he spends 3 years in jail**
    - » **If Bob confesses and you are silent**
      - **He goes free and you spend 3 years in jail**
    - » **If you both confess (defect)**
      - **Both get 2 years**
    - » **If you both stay silent (cooperate)**
      - **Both get 1 year**
    - » **I will offer Bob same terms**
- **If you are Bob or Alice, what will you?**

	Bob is Silent	**Bob Confesses**
Alice is Silent	Each 1 year	Alice – 3 years Bob Free
Alice Confess	Alice Free Bob: 3 years	Each 2 years

	Bob Cooperates With Alice	**Bob Defects on Alice**
Alice cooperates With Bob	Each 1 year	Alice – 3years Bob Free
Alice defects on Bob	Alice Free Bob: 3 years	Each 2 years

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

PD: T>R>P>S

	Bob Cooperates With Alice	**Bob Defects on Alice**
Alice cooperates With Bob	Each 1 years	Alice – 3 years Bob Free
Alice defects on Bob	Alice Free Bob: 3 years	Each 2 years

	Cooperate	**Defect**
Cooperate	R,R	S,T
Defect	T,S	P,P

T: Temptation, if you defect and partner cooperates maximum reduction

R: Reward for cooperating with partner

P: Punishment for each betraying partner (defecting)

S: Sucker, if you cooperate and partner defects minimum reduction

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

PD: What to Do?

Consider Bob cooperates, what should Alice do to be selfish?

- If Alice cooperates -> 1 year
- If Alice defects -> free
- So Alice should defect

Consider Bob defects, what should Alice to be selfish?

- if Alice cooperates -> 3 years
- if Alice defects -> 2 years
- So Alice should defect

Ergo, whatever Bob does, Alice should defect.

But Bob sees same situation as Alice…so Bob should defect

Bob and Alice both defect → each serves 2 years

If only Alice changes action, Alice will do worse

If only Bob changes, Bob will do worse

Bob and Alice are isolated, can't talk …if they could, they would cooperate and get 1 year each

This is a NASH EQUILIBRIUM – doesn't it disturb you?

	Bob Cooperates With Alice	**Bob Defects on Alice**
Alice cooperates With Bob	Each 1years	Alice – 3 years Bob Free
Alice defects on Bob	Alice Free Bob: 3 years	Each 2 years

	Cooperate	**Defect**
Cooperate	R,R	S,T
Defect	T,S	P,P

ANYSCALE LEARNING FOR ALL

CSAIL

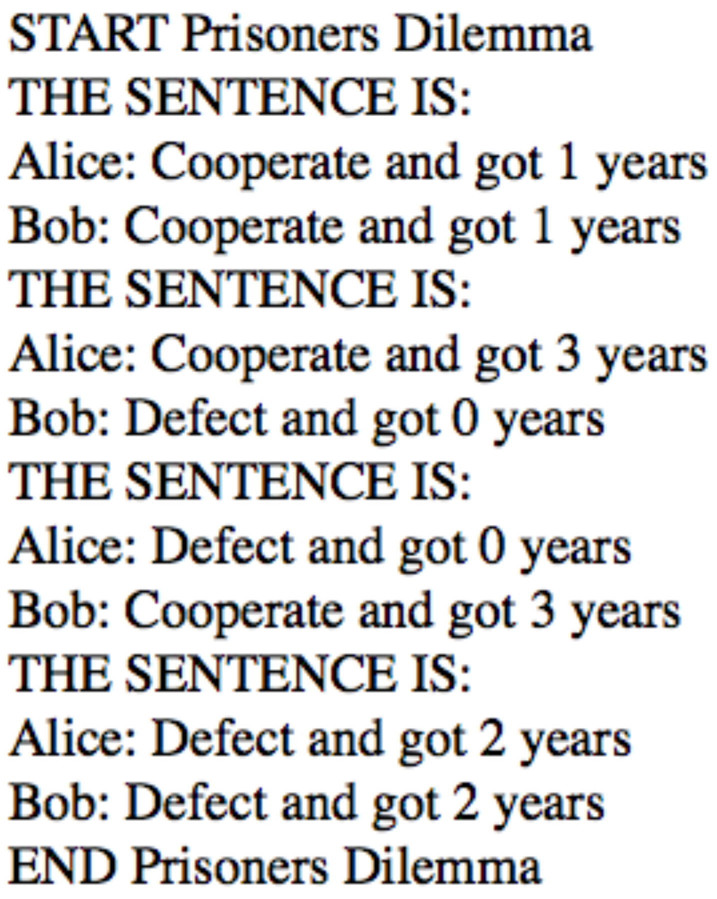
Let's Program Prisoner's Dilemma

Your program:

- **Computes sentence for Alice and Bob in 4 combinations**
  - **(C, C) : R,R**
  - **(C, D): S,T**
  - **(D, C): T,S**
  - **(D, D): P, P**

Program must have info in payoff matrix

Expected output



```
START Prisoners Dilemma
THE SENTENCE IS:
Alice: Cooperate and got 1 years
Bob: Cooperate and got 1 years
THE SENTENCE IS:
Alice: Cooperate and got 3 years
Bob: Defect and got 0 years
THE SENTENCE IS:
Alice: Defect and got 0 years
Bob: Cooperate and got 3 years
THE SENTENCE IS:
Alice: Defect and got 2 years
Bob: Defect and got 2 years
END Prisoners Dilemma
```

ANYSCALE LEARNING FOR ALL

CSAIL

Programming Homework Guidelines

- Work alone or in pairs
- Work on programming today after class
  - Dr. Hemberg is available for office hours
    - » Monday 14:00 – 17:00 in Software Eng Lab, 302
- Show whatever solution you have at Tuesday office hours(8:00 to 10:00)
  - Doesn't have to work
    - » We will help you finish
- **At 9am we will change the game slightly**
  - Program this until the lab at 1900.

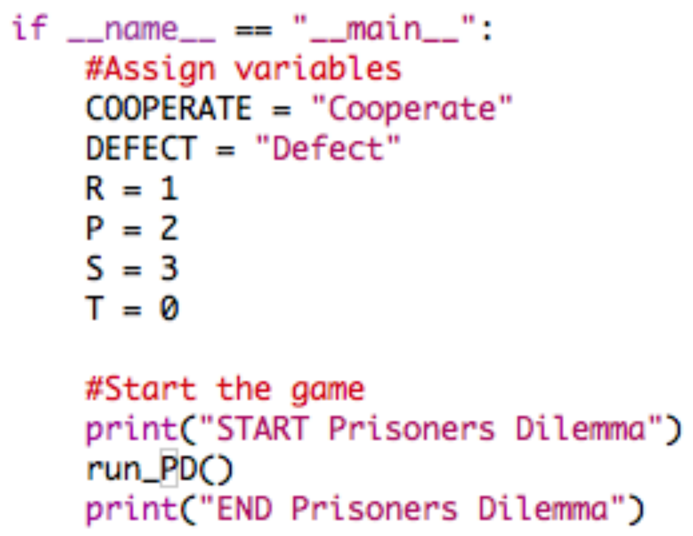
PD_skeleton.py - main

- Initialize variables
  - Variables:

    COOPERATE="cooperate"

    DEFECT="defect"

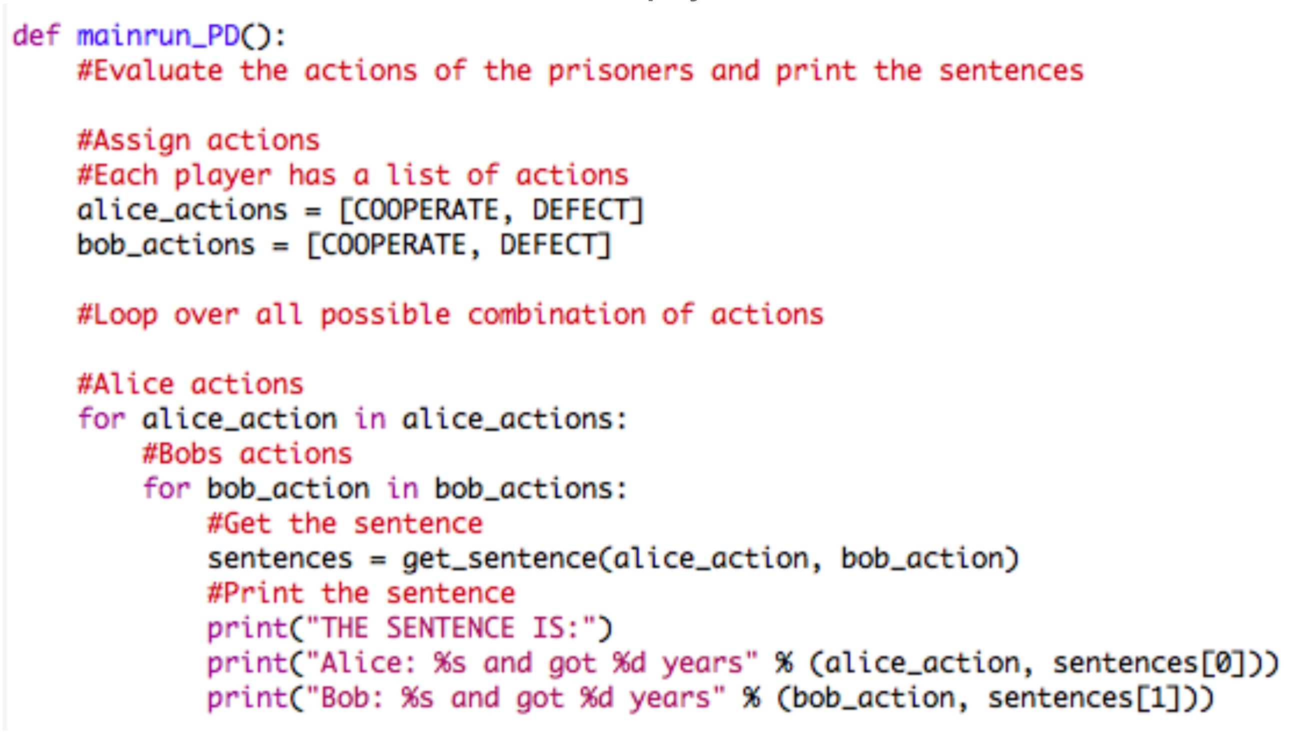
    P = 2,R = 1,S=3,T=0
- Start the game
- Run a function called run_PD()
- End the game



```
if __name__ == "__main__":
    #Assign variables
    COOPERATE = "Cooperate"
    DEFECT = "Defect"
    R = 1
    P = 2
    S = 3
    T = 0

    #Start the game
    print("START Prisoners Dilemma")
    run_PD()
    print("END Prisoners Dilemma")
```

PD_skeleton.py run_PD()

- **Run_PD function**
  - **Initialize the actions**
  - **First loop (outer)**
    - » **over all possible combinations of actions for Alice**
  - **Second loop (inner)**
    - » **over all possible combinations of actions for Bob**
    - » **Get sentence for the actions according to payoff matrix**
    - » **Print the sentence for each player**



```python
def mainrun_PD():
    #Evaluate the actions of the prisoners and print the sentences

    #Assign actions
    #Each player has a list of actions
    alice_actions = [COOPERATE, DEFECT]
    bob_actions = [COOPERATE, DEFECT]

    #Loop over all possible combination of actions

    #Alice actions
    for alice_action in alice_actions:
        #Bobs actions
        for bob_action in bob_actions:
            #Get the sentence
            sentences = get_sentence(alice_action, bob_action)
            #Print the sentence
            print("THE SENTENCE IS:")
            print("Alice: %s and got %d years" % (alice_action, sentences[0]))
            print("Bob: %s and got %d years" % (bob_action, sentences[1]))
```

PD_skeleton.py
get_sentence(action_alice, action_bob) function

- **Two arguments**
  - **action_alice, action_bob**
- **Uses If statements**
  - **For example**

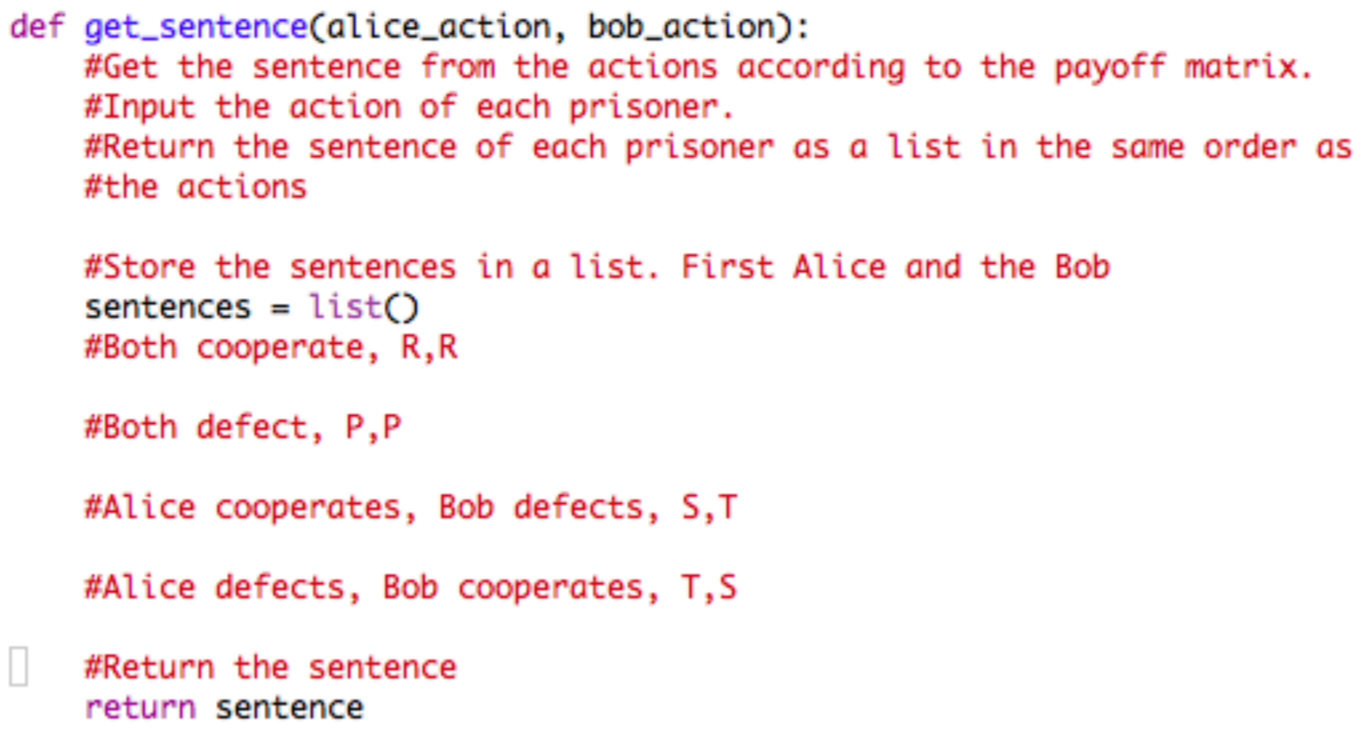
```
If action of Alice ==COOPERATE and action of bob ==COOPERATE:
        Sentences[0]=R
        Sentences[1]=R
```

- **Don’t forget to indent!**
- **Returns sentences**
  - **list of Alice’s sentence and Bob’s sentence**

ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

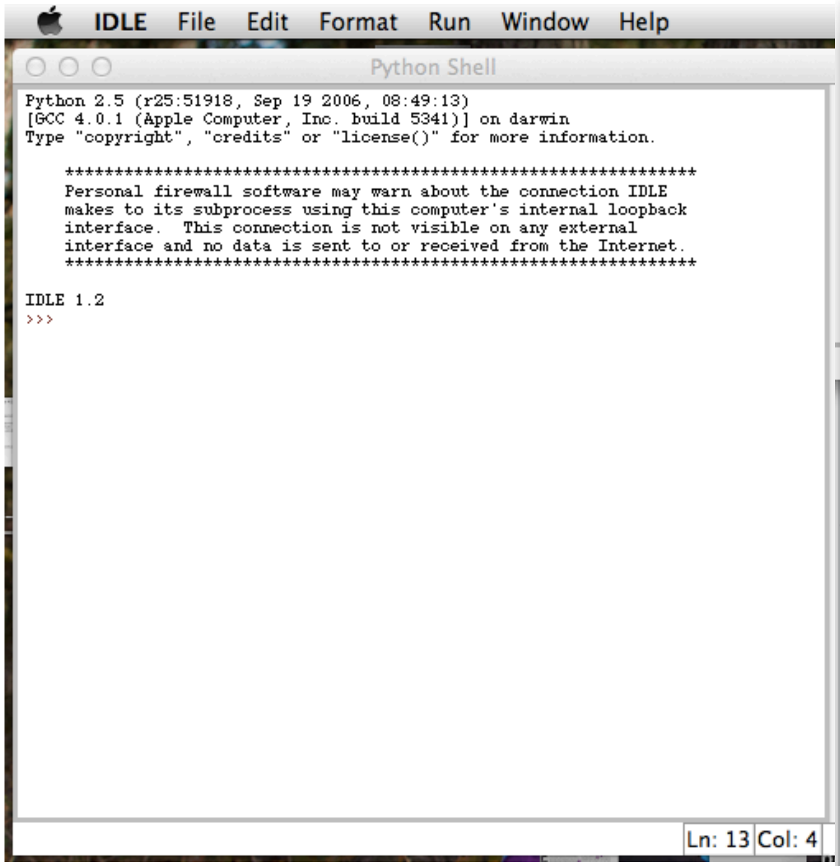
get_sentences Template in file PD_skeleton.py



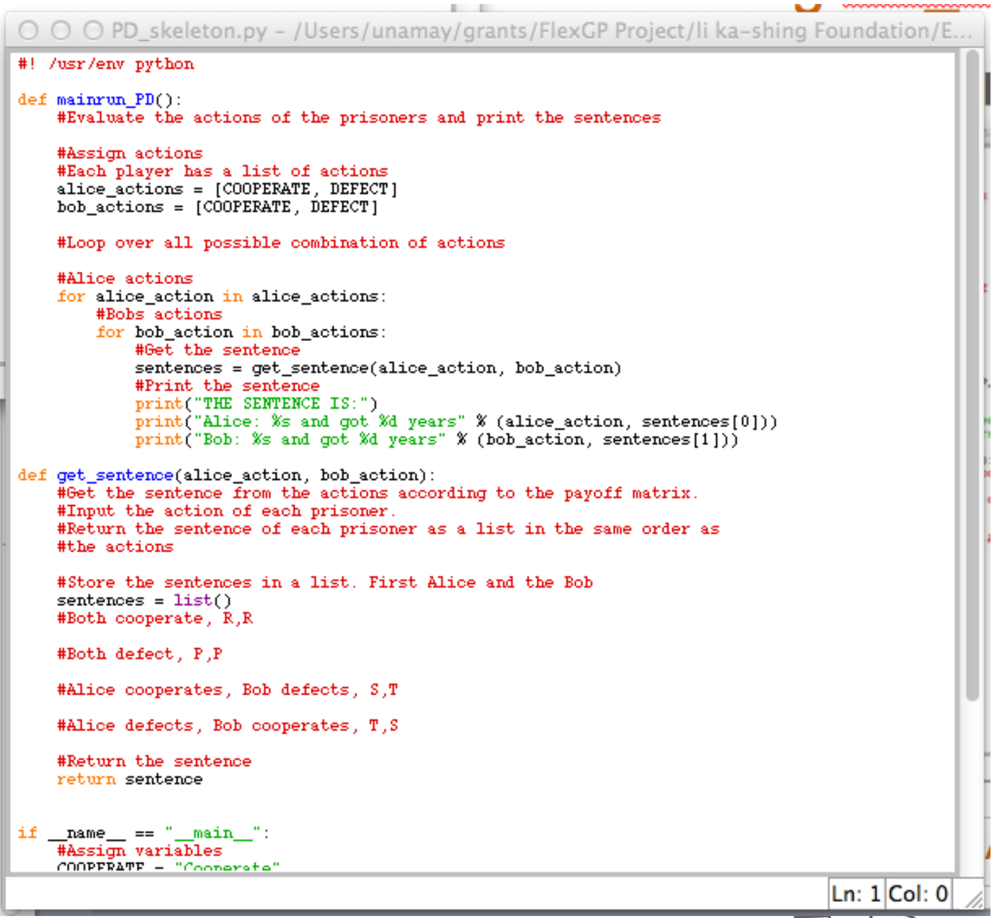
```
def get_sentence(alice_action, bob_action):
    #Get the sentence from the actions according to the payoff matrix.
    #Input the action of each prisoner.
    #Return the sentence of each prisoner as a list in the same order as
    #the actions

    #Store the sentences in a list. First Alice and the Bob
    sentences = list()
    #Both cooperate, R,R

    #Both defect, P,P

    #Alice cooperates, Bob defects, S,T

    #Alice defects, Bob cooperates, T,S

    #Return the sentence
    return sentence
```

Editing and Running PD_skeleton.py s

- **Use IDLE to edit and use Run on menu to execute your program**



```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "copyright", "credits" or "license()" for more information.

    ****************************************************************
    Personal firewall software may warn about the connection IDLE
    makes to its subprocess using this computer's internal loopback
    interface.  This connection is not visible on any external
    interface and no data is sent to or received from the Internet.
    ****************************************************************

IDLE 1.2
>>>
```


PD_skeleton.py - /Users/unamay/grants/FlexGP Project/li ka-shing Foundation/E...
Ln: 1 Col: 0

```
#! /usr/env python

def mainrun_PD():
    #Evaluate the actions of the prisoners and print the sentences

    #Assign actions
    #Each player has a list of actions
    alice_actions = [COOPERATE, DEFECT]
    bob_actions = [COOPERATE, DEFECT]

    #Loop over all possible combination of actions

    #Alice actions
    for alice_action in alice_actions:
        #Bobs actions
        for bob_action in bob_actions:
            #Get the sentence
            sentences = get_sentence(alice_action, bob_action)
            #Print the sentence
            print("THE SENTENCE IS:")
            print("Alice: %s and got %d years" % (alice_action, sentences[0]))
            print("Bob: %s and got %d years" % (bob_action, sentences[1]))

def get_sentence(alice_action, bob_action):
    #Get the sentence from the actions according to the payoff matrix.
    #Input the action of each prisoner.
    #Return the sentence of each prisoner as a list in the same order as
    #the actions

    #Store the sentences in a list. First Alice and the Bob
    sentences = list()
    #Both cooperate, R,R

    #Both defect, P,P

    #Alice cooperates, Bob defects, S,T

    #Alice defects, Bob cooperates, T,S

    #Return the sentence
    return sentence


if __name__ == "__main__":
    #Assign variables
    COOPERATE = "Cooperate"
```

ANYSCALE LEARNING FOR ALL

CSAIL

Python Syntax Quick Reference!

- Indentation is NECESSARY
  - DON'T forget!!!!
- "String"
- Comments

```
#Clearly and concisely
```

- Loop (note indentation)

```
for item in list:
        do_something()
```

- Conditional (note indentation!)

```
if X:
        do_something()
else:
        do_something_else()
```

- List Syntax

```
items = list() #empty
items = [1,2] #initial assignment
item_0 = list[0] #access
Items.append(1) #append item to list
```

- Print a variable

```
variable = 9
print("Print this integer variable %d" % (variable))
```

ALFA ANYSCALE LEARNING FOR ALL

CSAIL

Python Syntax Quick Reference! (-2)

- **Program entry function**

```
if __name__ == "__main__":
```

- **Functions (indent on lines after def)**

```
def function_name(argument_1, argument_2):
    print("Argument 1 %s and Argument 2 %s" % (argument_1, argument_2))
    return argument_1, argument_2
```