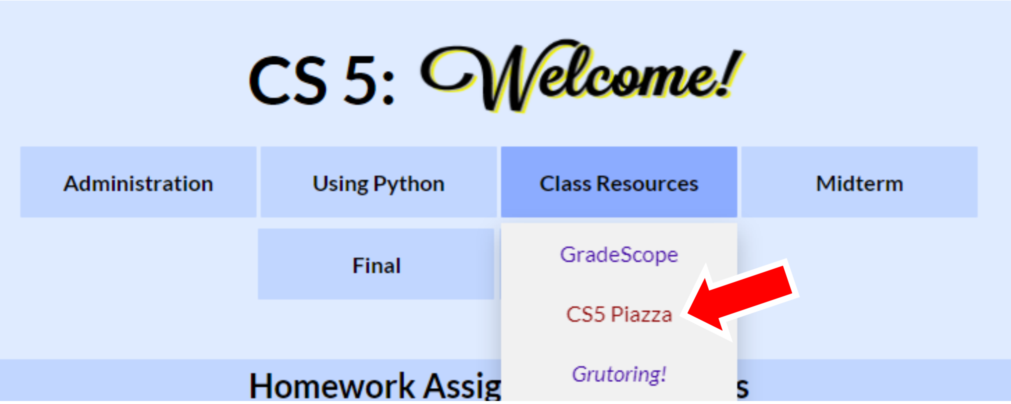
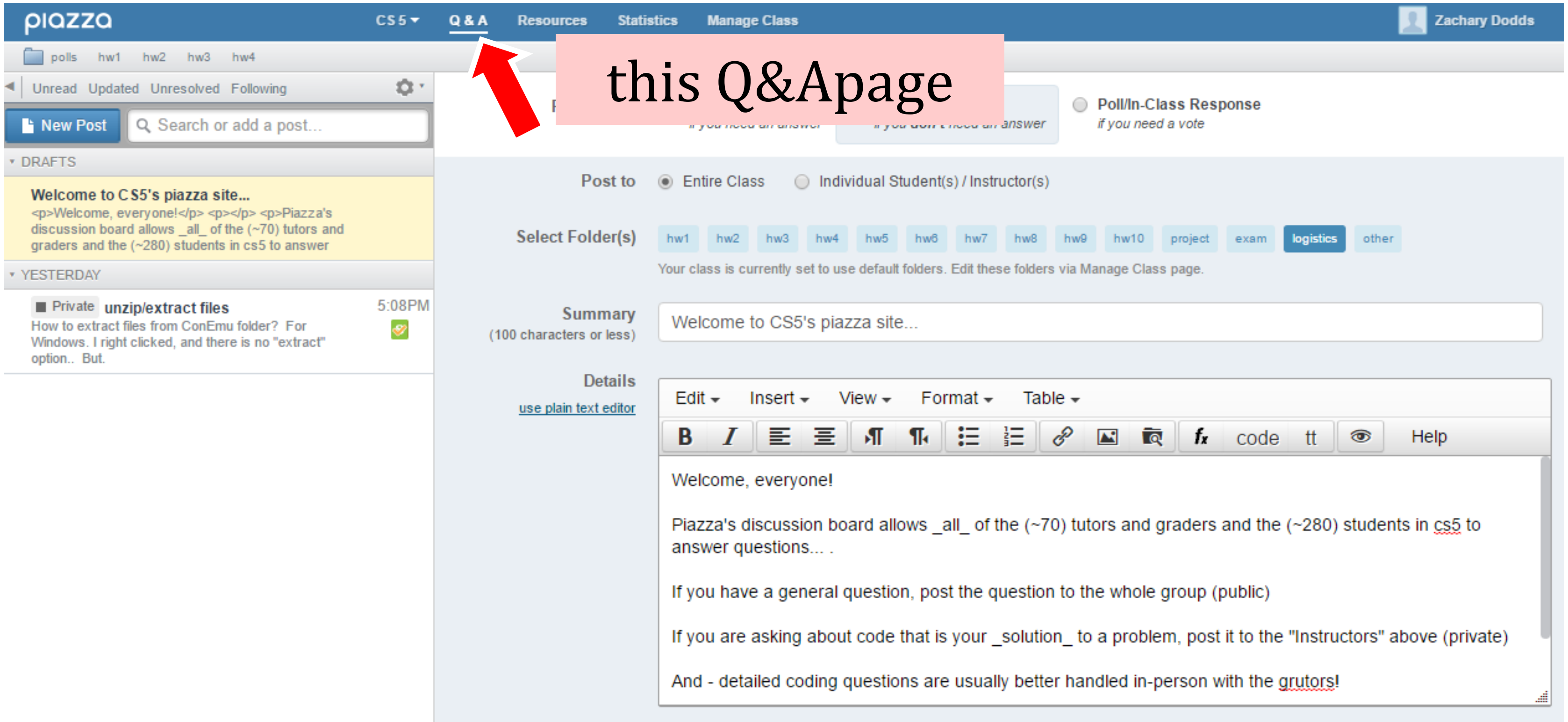
"Online" help: ***Piazza...***



for many questions,

Piazza is a great resource:



this Q&Apage

Welcome to C S5's piazza site...
<p>Welcome, everyone!</p> <p></p> <p>Piazza's discussion board allows _all_ of the (~70) tutors and graders and the (~280) students in cs5 to answer

Private **unzip/extract files** 5:08PM
How to extract files from ConEmu folder? For Windows. I right clicked, and there is no "extract" option.. But.

Post to ● Entire Class ○ Individual Student(s) / Instructor(s)

Select Folder(s): hw1 hw2 hw3 hw4 hw5 hw6 hw7 hw8 hw9 hw10 project exam **logistics** other

Your class is currently set to use default folders. Edit these folders via Manage Class page.

Summary (100 characters or less): Welcome to CS5's piazza site...

Details (use plain text editor):

Welcome, everyone!

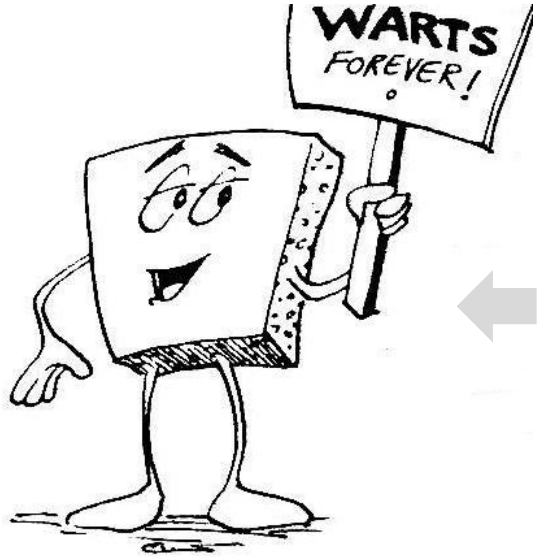
Piazza's discussion board allows _all_ of the (~70) tutors and graders and the (~280) students in cs5 to answer questions... .

If you have a general question, post the question to the whole group (public)

If you are asking about code that is your _solution_ to a problem, post it to the "Instructors" above (private)

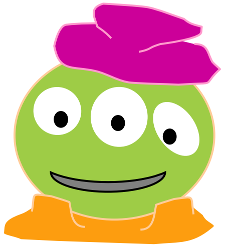
And - detailed coding questions are usually better handled in-person with the grutors!

Welcome back to CS 5 !



Wally

Average of these two?



Alien

Homework 0

due Tues. night (22:22:22)

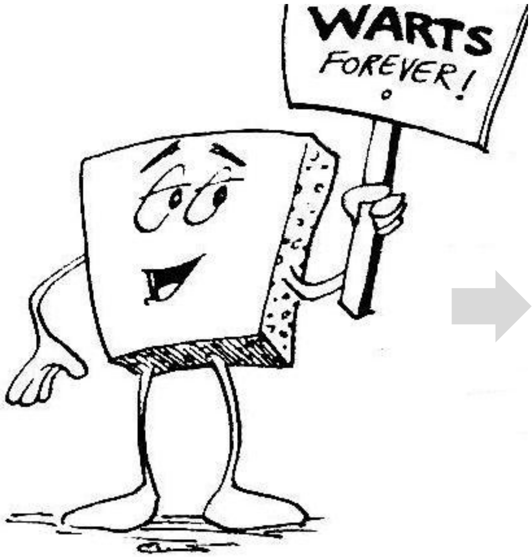
Problem 0: Reading + response...

Problem 1: Four-fours program: Can be done for lab...

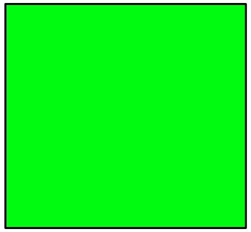
Problem 2: Rock-paper-scissors program (*Maybe* done already!)

Problems 3-4: Picobot! empty room (3) maze (4)

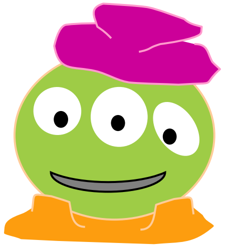
Welcome back to CS 5 !



Wally



Picobot!



Yes! I see the resemblance

Alien

Homework 0

due Mon. night (10:42pm)

- **Problem 0:** **Reading + response…**
- **Problem 1:** **Four-fours program: Can be done for lab…**
- **Problem 2:** **Rock-paper-scissors program (*Maybe* done already!)**
- **Problems 3-4:** **Picobot! empty room (3) maze (4)**

Problem 0 ?

Typically, an article on CS or an application...

Submit a one-paragraph response

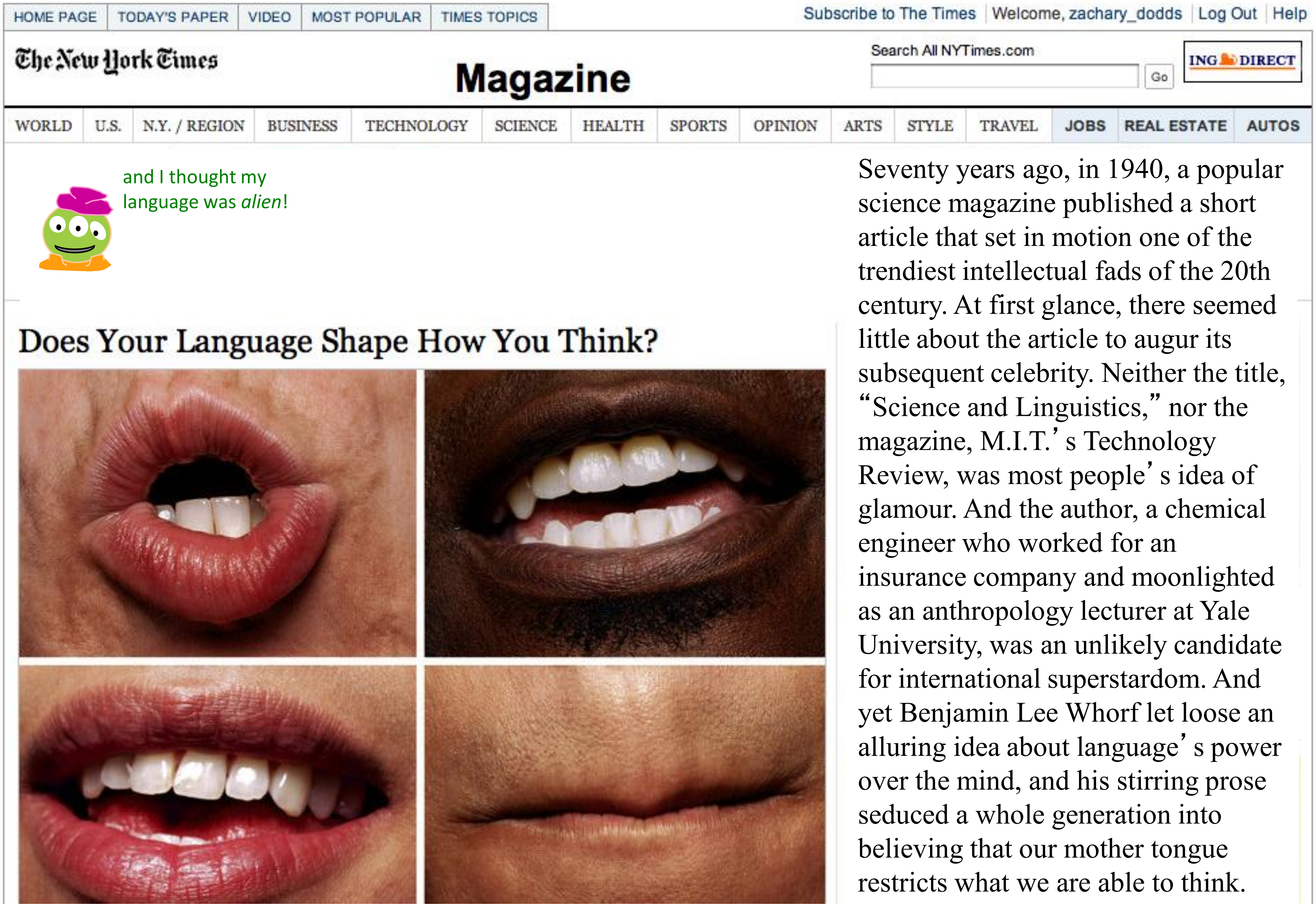
A few sentences that raise or address questions, using the article as a guide.

Small part (5 pts)

5 – insightful, careful
4 – thoughtful
3 – complete, on topic
0-2 – less than complete

Does Your Language Shape How You Think?

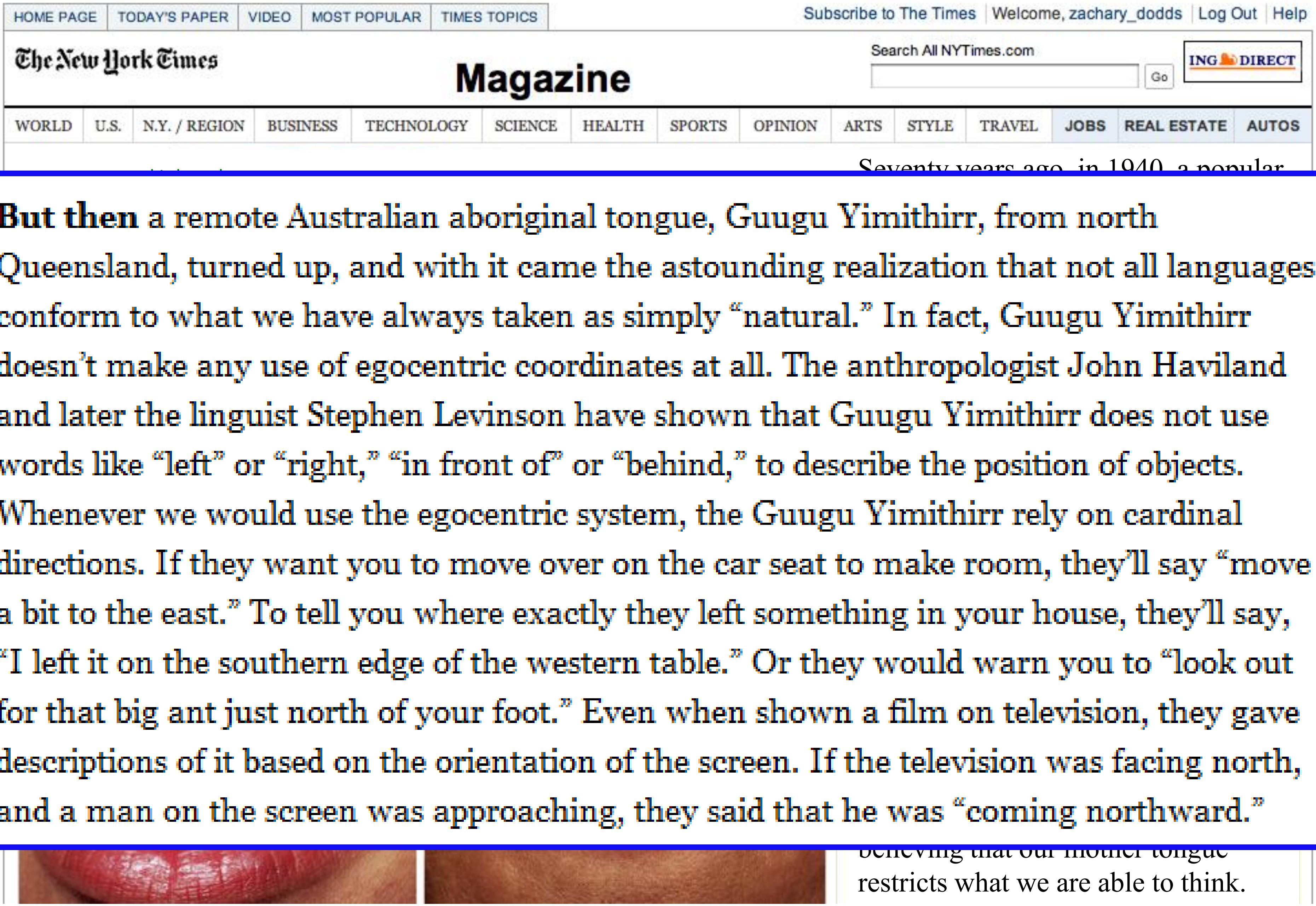
This week's article might not seem like CS at first...



and I thought my language was *alien*!

Does Your Language Shape How You Think?

Seventy years ago, in 1940, a popular science magazine published a short article that set in motion one of the trendiest intellectual fads of the 20th century. At first glance, there seemed little about the article to augur its subsequent celebrity. Neither the title, "Science and Linguistics," nor the magazine, M.I.T.' s Technology Review, was most people' s idea of glamour. And the author, a chemical engineer who worked for an insurance company and moonlighted as an anthropology lecturer at Yale University, was an unlikely candidate for international superstardom. And yet Benjamin Lee Whorf let loose an alluring idea about language' s power over the mind, and his stirring prose seduced a whole generation into believing that our mother tongue restricts what we are able to think.



HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS

Subscribe to The Times | Welcome, zachary_dodds | Log Out | Help

The New York Times

Magazine

Search All NYTimes.com

Go

ING DIRECT

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

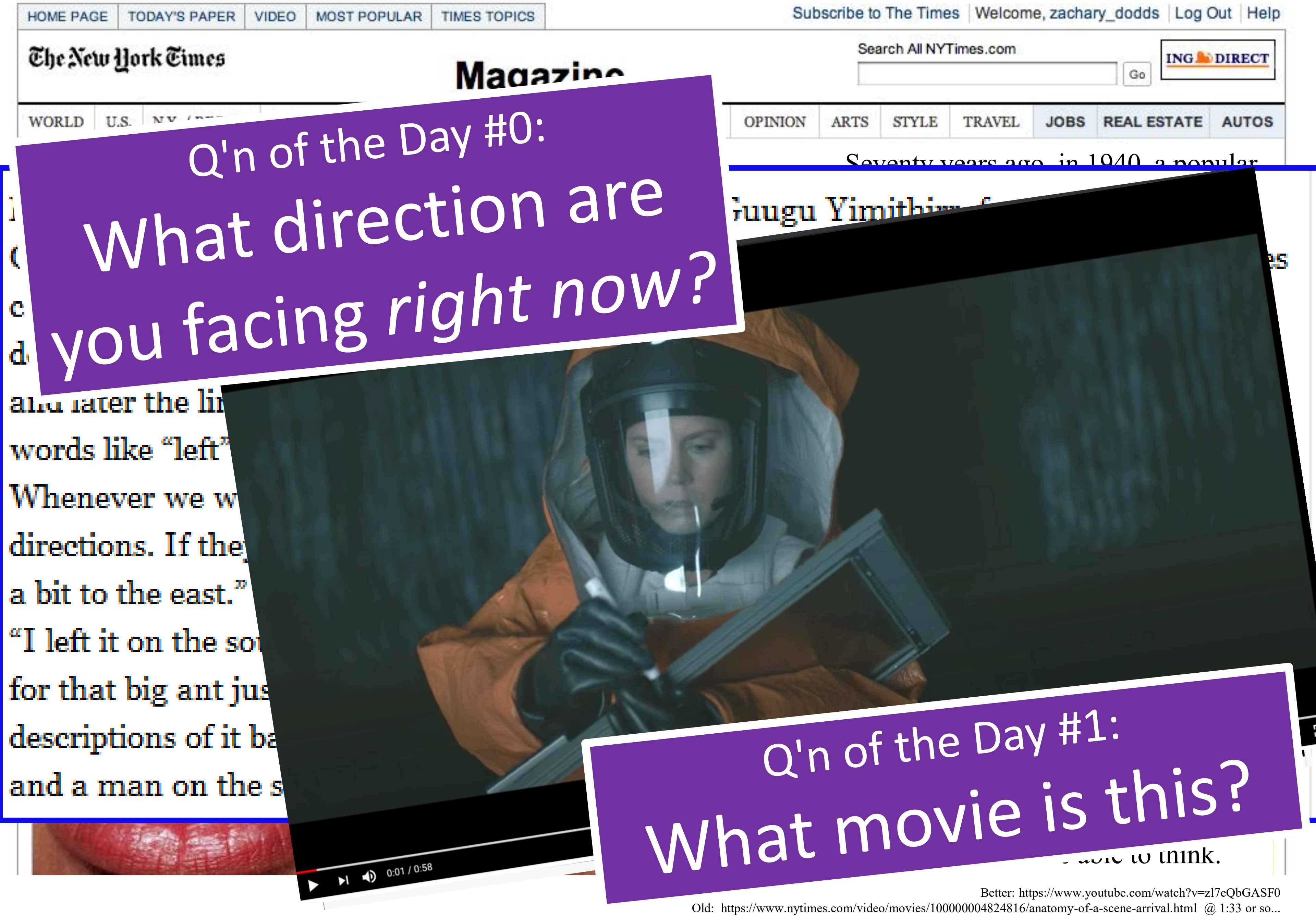
Seventy years ago, in 1940, a popular

But then a remote Australian aboriginal tongue, Guugu Yimithirr, from north Queensland, turned up, and with it came the astounding realization that not all languages conform to what we have always taken as simply "natural." In fact, Guugu Yimithirr doesn't make any use of egocentric coordinates at all. The anthropologist John Haviland and later the linguist Stephen Levinson have shown that Guugu Yimithirr does not use words like "left" or "right," "in front of" or "behind," to describe the position of objects. Whenever we would use the egocentric system, the Guugu Yimithirr rely on cardinal directions. If they want you to move over on the car seat to make room, they'll say "move a bit to the east." To tell you where exactly they left something in your house, they'll say, "I left it on the southern edge of the western table." Or they would warn you to "look out for that big ant just north of your foot." Even when shown a film on television, they gave descriptions of it based on the orientation of the screen. If the television was facing north, and a man on the screen was approaching, they said that he was "coming northward."

believing that our mother tongue restricts what we are able to think.

Better: https://www.youtube.com/watch?v=zl7eQbGASF0

Old: https://www.nytimes.com/video/movies/100000004824816/anatomy-of-a-scene-arrival.html @ 1:33 or so...



HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS

Subscribe to The Times | Welcome, zachary_dodds | Log Out | Help

The New York Times

Magazine

Search All NYTimes.com

Go

ING DIRECT

WORLD | U.S. | N.Y. | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

Seventy years ago, in 1940, a popular

Guugu Yimithirr

es

and later the li

words like "left"

Whenever we w

directions. If the

a bit to the east."

"I left it on the so

for that big ant jus

descriptions of it ba

and a man on the s

able to think.

Q'n of the Day #0:

What direction are you facing *right now*?

Q'n of the Day #1:

What movie is this?

Better: https://www.youtube.com/watch?v=zl7eQbGASF0

Old: https://www.nytimes.com/video/movies/100000004824816/anatomy-of-a-scene-arrival.html @ 1:33 or so...

Seventy years ago, in 1940, a popular

But then a remote Australian aborig tongue, Guugu Yimithirr, f

Queensland, tu th ges

con

does d

and the ling

words like "left" o

Whenever we wou

directions. If they

a bit to the east." T

"I left it on the sout

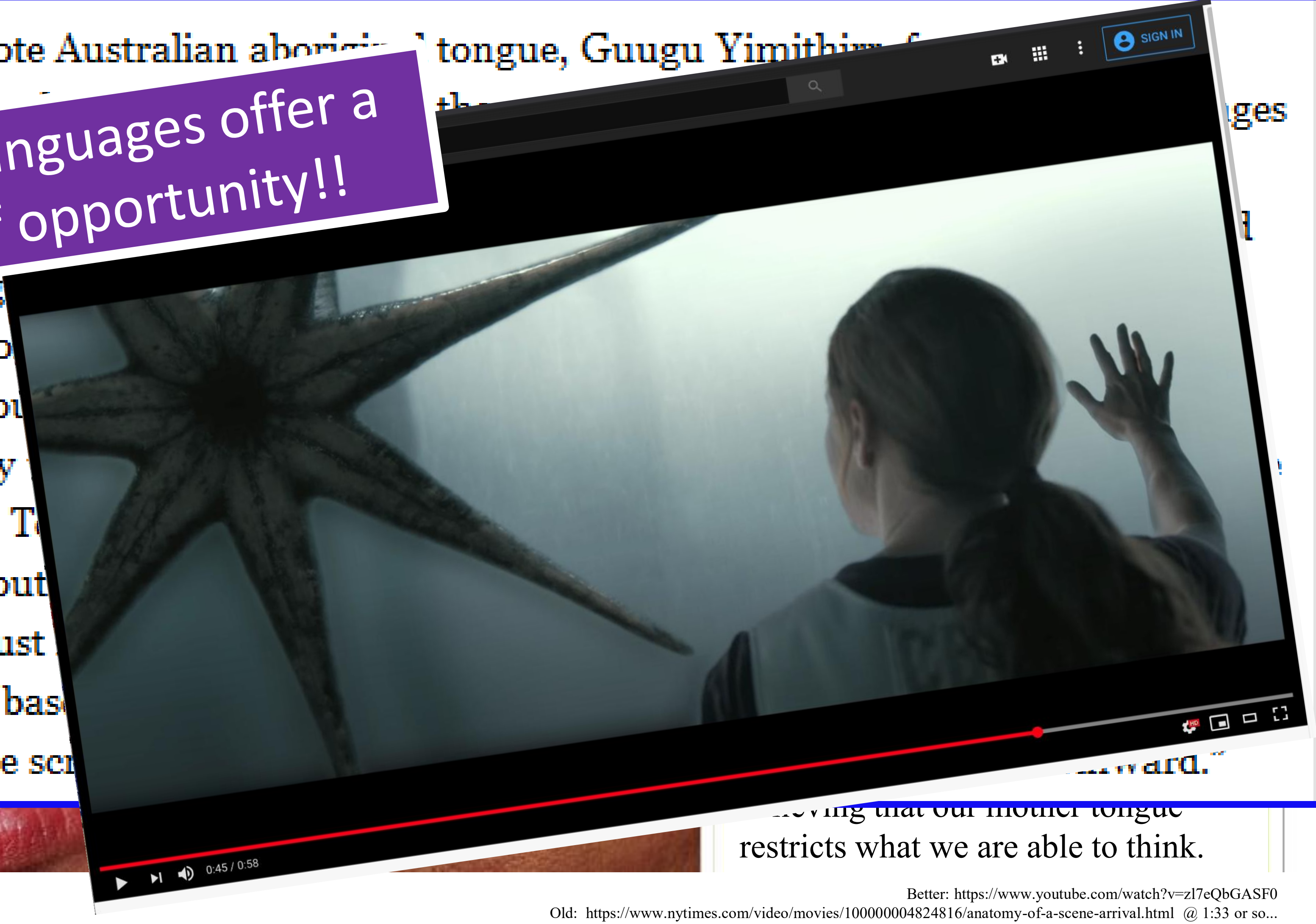
for that big ant just

descriptions of it bas

and a man on the scr ward."

lieving that our mother tongue restricts what we are able to think.

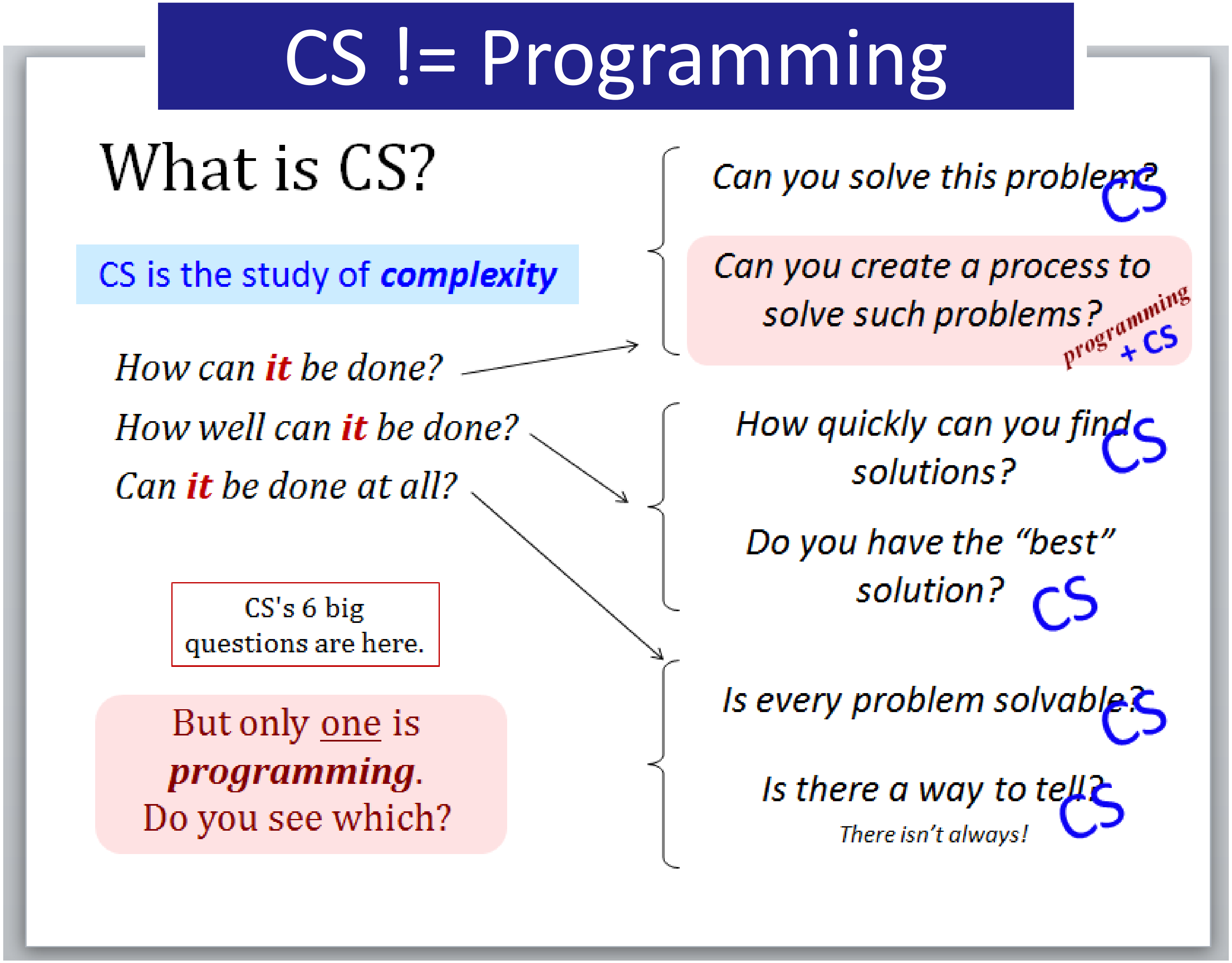
Alien languages offer a lot of opportunity!!



Better: https://www.youtube.com/watch?v=zl7eQbGASF0

Old: https://www.nytimes.com/video/movies/100000004824816/anatomy-of-a-scene-arrival.html @ 1:33 or so...

Last time…



CS != Programming

What is CS?

CS is the study of ***complexity***

*How can **it** be done?*

*How well can **it** be done?*

*Can **it** be done at all?*

CS's 6 big questions are here.

But only one is ***programming***.
Do you see which?

Can you solve this problem? CS

Can you create a process to solve such problems? programming + CS

How quickly can you find solutions? CS

Do you have the "best" solution? CS

Is every problem solvable? CS

Is there a way to tell? CS

There isn't always!

Last time…

A Big Idea

Information

(simple)

+ Rules

Composition

Complexity

What *is* programming ?

Programming as recipe-writing

vs.

Programming as learning a foreign language

1) Expect it to be different!

Baggage!

2) Practice, not memorization!

3) Immerse == Experiment!

ser/estar
go/went

What about the *Python* programming language ?

The ***foreign language*** of python…

syntax

How it looks

semantics

What it actually does

intent

What it should do

The *foreign language* of python...

syntax

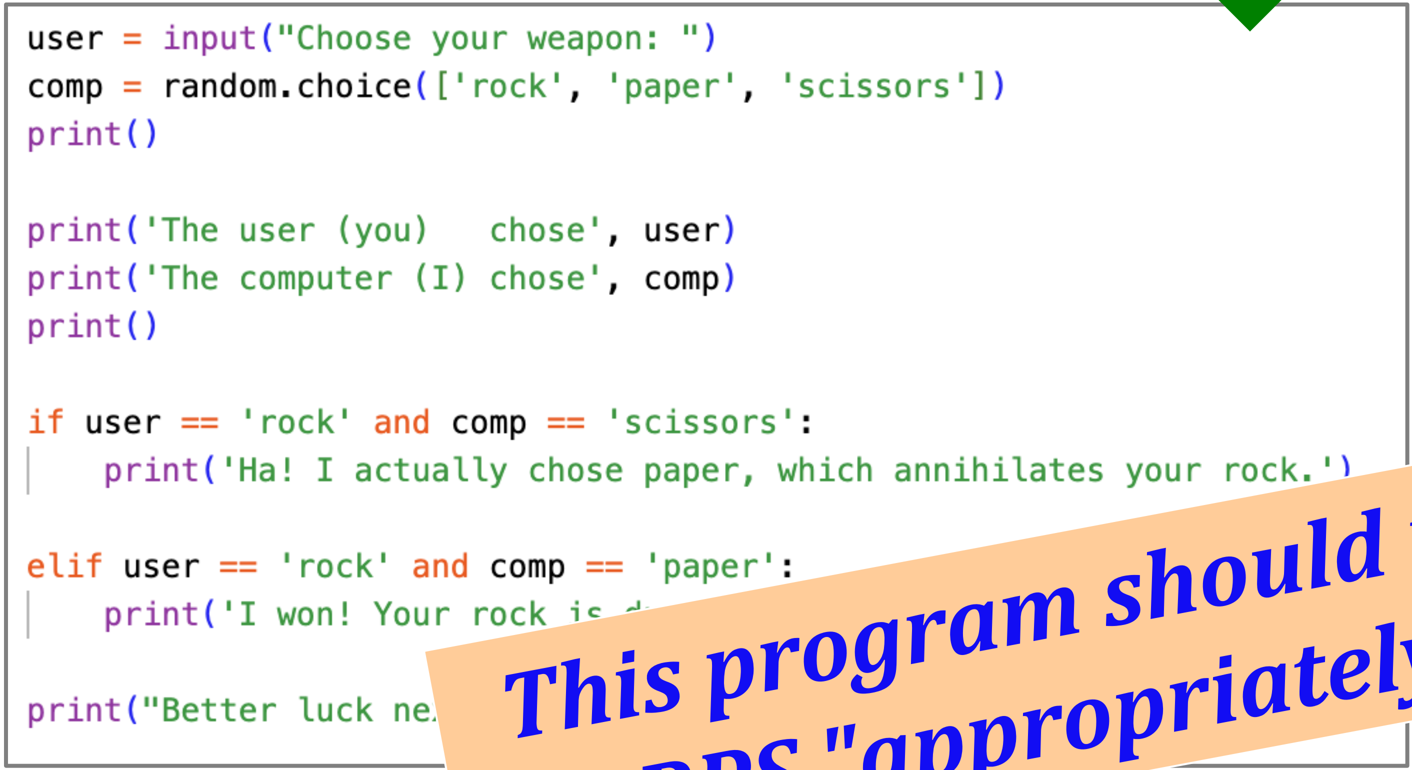
How it looks

semantics

What it actually does

intent

What it should do



```
user = input("Choose your weapon: ")
comp = random.choice(['rock', 'paper', 'scissors'])
print()

print('The user (you)   chose', user)
print('The computer (I) chose', comp)
print()

if user == 'rock' and comp == 'scissors':
    print('Ha! I actually chose paper, which annihilates your rock.')

elif user == 'rock' and comp == 'paper':
    print('I won! Your rock is d

print("Better luck ne
```

This program should play RPS "appropriately."

The *foreign language* of python...

syntax

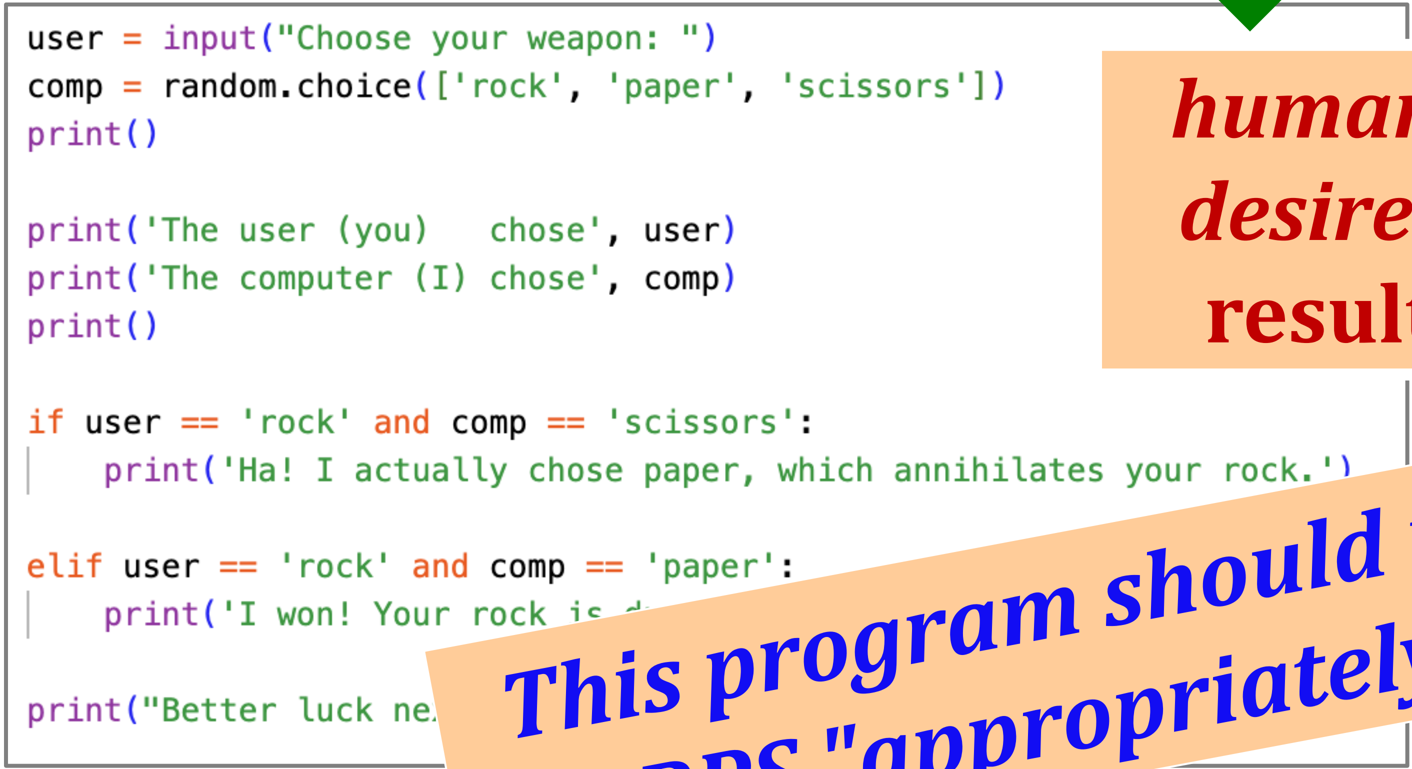
How it looks

semantics

What it actually does

intent

What it should do



```
user = input("Choose your weapon: ")
comp = random.choice(['rock', 'paper', 'scissors'])
print()

print('The user (you)   chose', user)
print('The computer (I) chose', comp)
print()

if user == 'rock' and comp == 'scissors':
    print('Ha! I actually chose paper, which annihilates your rock.')

elif user == 'rock' and comp == 'paper':
    print('I won! Your rock is

print("Better luck ne
```

***human-desired* result**

This program should play RPS "appropriately."

The *foreign language* of python…

syntax

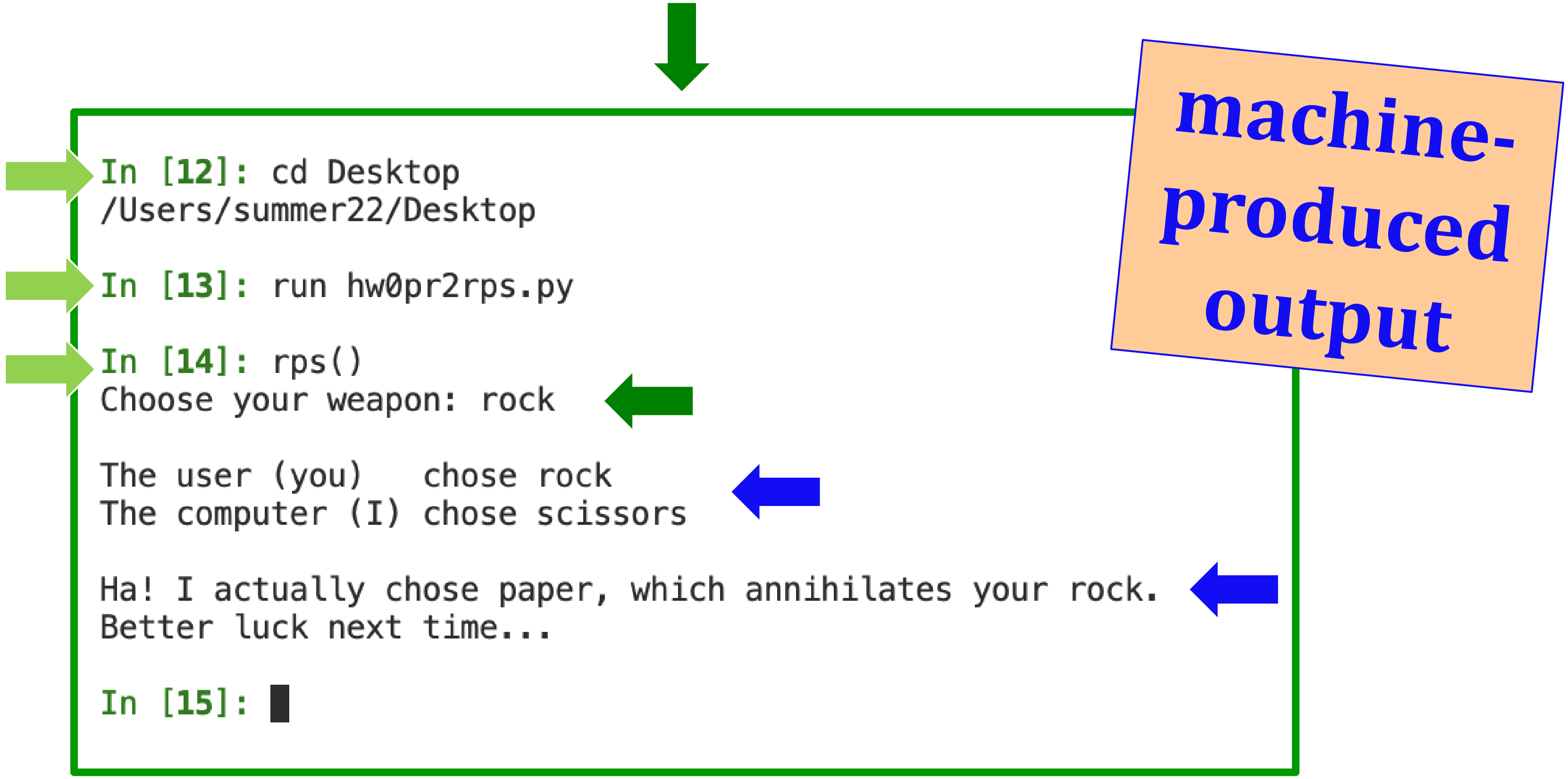
How it looks

semantics

What it actually does

intent

What it should do

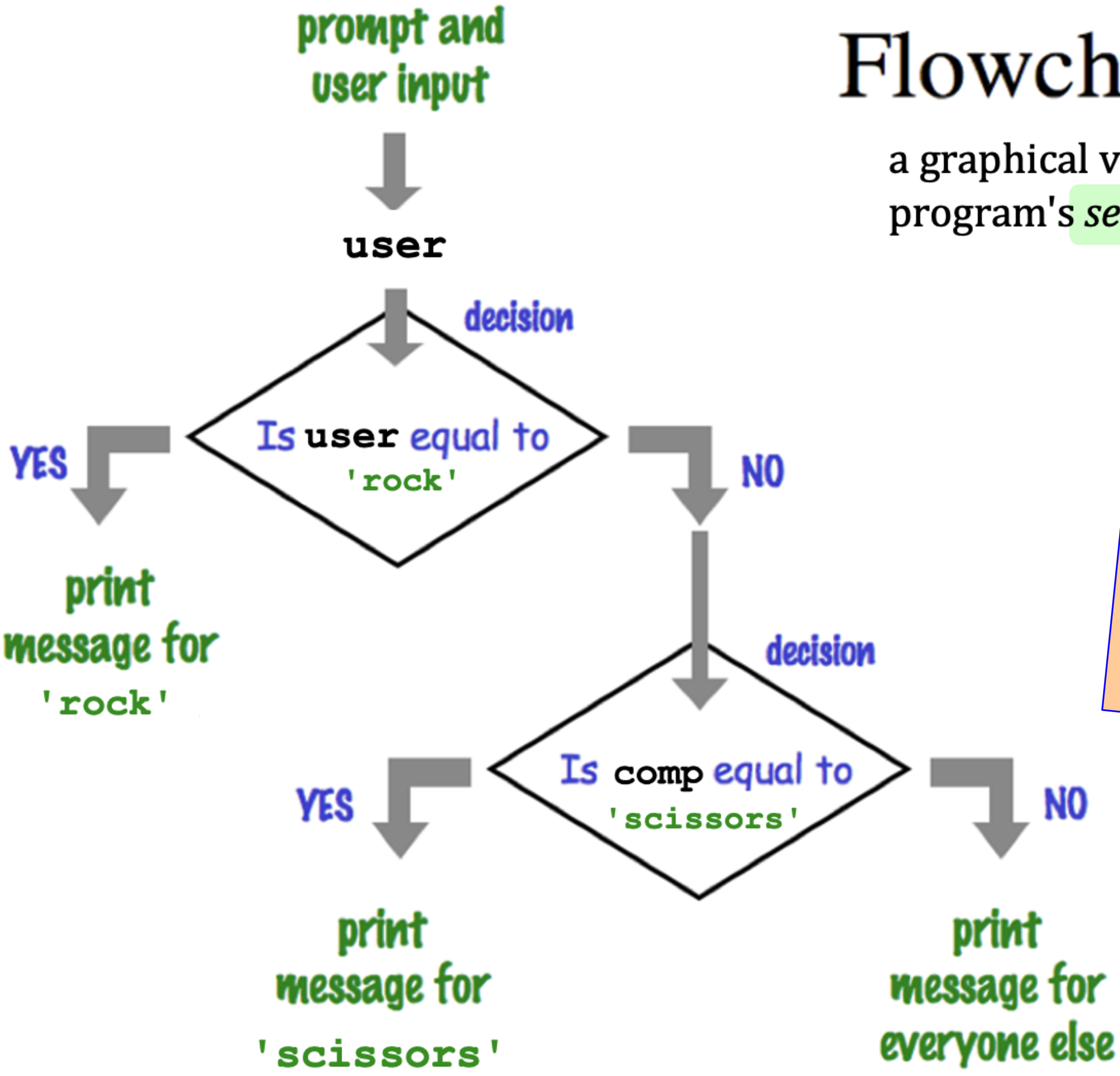


machine-produced output

```
In [12]: cd Desktop
/Users/summer22/Desktop

In [13]: run hw0pr2rps.py

In [14]: rps()
Choose your weapon: rock

The user (you)   chose rock
The computer (I) chose scissors

Ha! I actually chose paper, which annihilates your rock.
Better luck next time...

In [15]: 
```

Flowchart…

a graphical view of a
program's *semantics*



prompt and
user input

user

decision

Is user equal to
'rock'

YES

print
message for
'rock'

NO

decision

Is comp equal to
'scissors'

YES

print
message for
'scissors'

NO

print
message for
everyone else

machine-produced output

The *foreign language* of python…

syntax

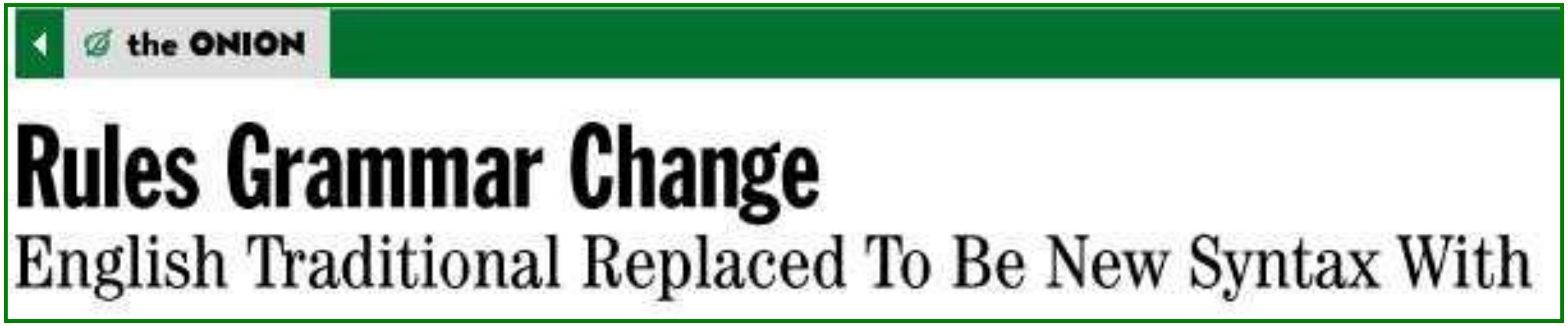
How it looks

semantics

What it actually does

intent

What it should do



the ONION

Rules Grammar Change

English Traditional Replaced To Be New Syntax With

How Python ***looks***!

- how punctuation is used
- the language keywords used
- use of whitespace

- peculiarities of formatting
- how behavior is affected …

The *foreign language* of python…

syntax

How it looks

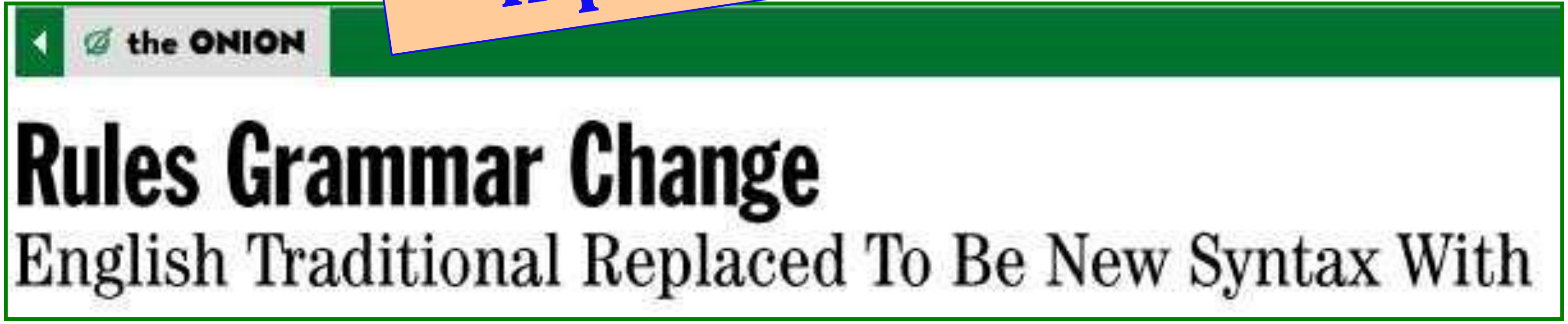
human-typed input

ntics

ally does

intent

What it should do



Rules Grammar Change

English Traditional Replaced To Be New Syntax With

How Python ***looks***!

- how punctuation is used
- the language keywords used
- use of whitespace

- peculiarities of formatting
- how behavior is affected …

The *challenge* of programming...

Look deep into my eyes...

syntax

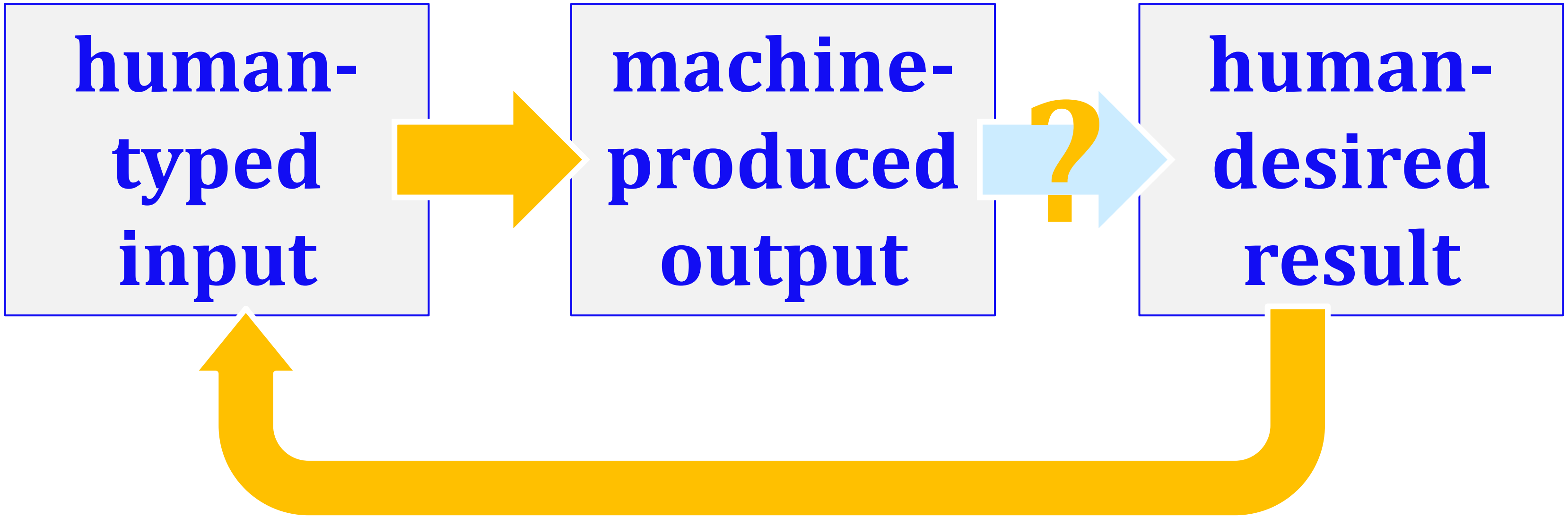
How it looks

semantics

What it actually does

intent

What it should do



human-typed input

machine-produced output

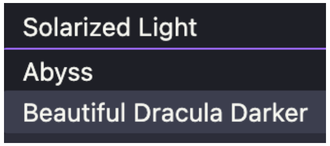
human-desired result

How we learn…

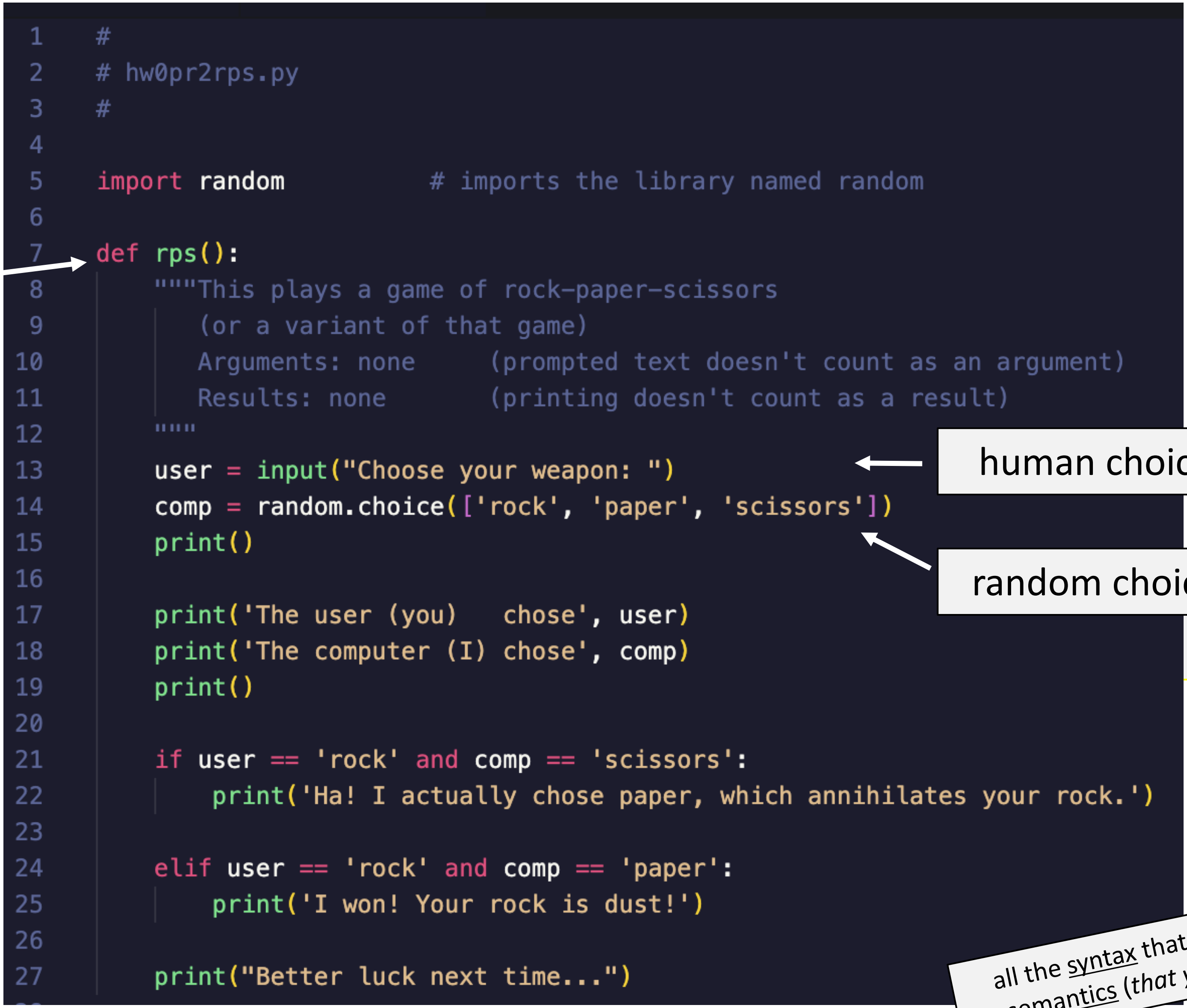
~~High Level Principles?~~

Concrete Examples!

hw0pr2rps: *RPS...*



A *function*!



```
#
# hw0pr2rps.py
#

import random          # imports the library named random

def rps():
    """This plays a game of rock-paper-scissors
       (or a variant of that game)
       Arguments: none     (prompted text doesn't count as an argument)
       Results: none       (printing doesn't count as a result)
    """
    user = input("Choose your weapon: ")
    comp = random.choice(['rock', 'paper', 'scissors'])
    print()

    print('The user (you)   chose', user)
    print('The computer (I) chose', comp)
    print()

    if user == 'rock' and comp == 'scissors':
        print('Ha! I actually chose paper, which annihilates your rock.')

    elif user == 'rock' and comp == 'paper':
        print('I won! Your rock is dust!')

    print("Better luck next time...")
```

human choices!

random choices!

all the syntax that creates the semantics (that you *intend*!)

(1) Find and correct as many errors as you can in this code:

"Quiz!"

Name(s) ______________________________

(we need to find your name on the roster!)

Syntax challenge!

```
import random

user = input( "Choose your weapon! " )
comp = random.choice( ['rock','paper','scissors") ]
print('user (you) chose:', 'user')
print('comp (me!) chose:'  comp)

if user == rock and comp = 'paper'
    print('The result is, YOU LOSE.'
   print('unless you're a CS5 grader, then YOU WIN :)')
```

(2) This one line does ***three*** things... what are they?

try to use one word for each step!

(a)

(b)

(c)

Syntax challenge!

(1) Find and correct as many errors as you can here...

(2) This line is doing ***three*** things... what are they?

every block of code must line up!

```
import random
```

set-equals always uses ONE equals sign

using one word for each step!

(a) prints!

(b) gets!!

(c) sets!!!

(a) prints the "weapon" prompt
(b) gets user's input from the kbd
(c) assigns that input to the variable <u>user</u>

```
user = input( "Choose your weapon! " )
comp = random.choice( ['rock','paper','scissors'] )
print('user (you) chose:', user)
print('comp (me!) chose:', comp)
```

`user` is a variable, not a string!

match brackets, parens and single/double quotes !

The comma prints a space and does NOT go to the next line.

test-equals uses TWO equals signs

`'rock'` is a string, not a variable

test-equals

```
if user == 'rock' and comp == 'paper':
    print('The result is, YOU LOSE.')
    print('unless you're a CS5 grader, then YOU WIN! :)')
```

a colon starts a new block

every block of code must line up!

matching parenthesis!

flattering - or flouting - graders is encouraged!

there's one more left...

Syntax challenge!

(1) Find and correct as many errors as you can here...

(2) This line is doing ***three*** things... what are they?

using one word for each step!

(a) prints!

(b) gets!!

(c) sets!!!

(a) prints the "weapon" prompt
(b) gets user's input from the kbd
(c) assigns that input to the variable <u>user</u>

every block of code must line up!

set-equals always uses ONE equals sign

```
import random

user = input( "Choose your weapon! " )
comp = random.choice( ['rock','paper','scissors'] )

print('user (you) chose:', user)

print('comp (me!) chose:', comp)

if user == 'rock' and comp == 'paper':
    print('The result is, YOU LOSE.')
    print('unless you\'re a CS5 grader, then YOU WIN! :)')
```

`user` is a variable, not a string!

match brackets, parens and single/double quotes !

The comma prints a space and does NOT go to the next line.

test-equals uses TWO equals signs

`'rock'` is a string, not a variable

test-equals

a colon starts a new block

matching parenthesis!

every block of code must line up!

a backslash handles special characters

flattering - or flouting - graders is encouraged!

```
print("how we'd print a single quote!")
```

Syntax challenge!

(1) Find and correct as many

re...

they?

' prompt
the kbd
ble user

every block of code must line up!

```
im

us

co

pr

pr

if
```

every block of code must line up!

a b

ns and
uotes !

:)')

flattering - or flouting - graders is encouraged!

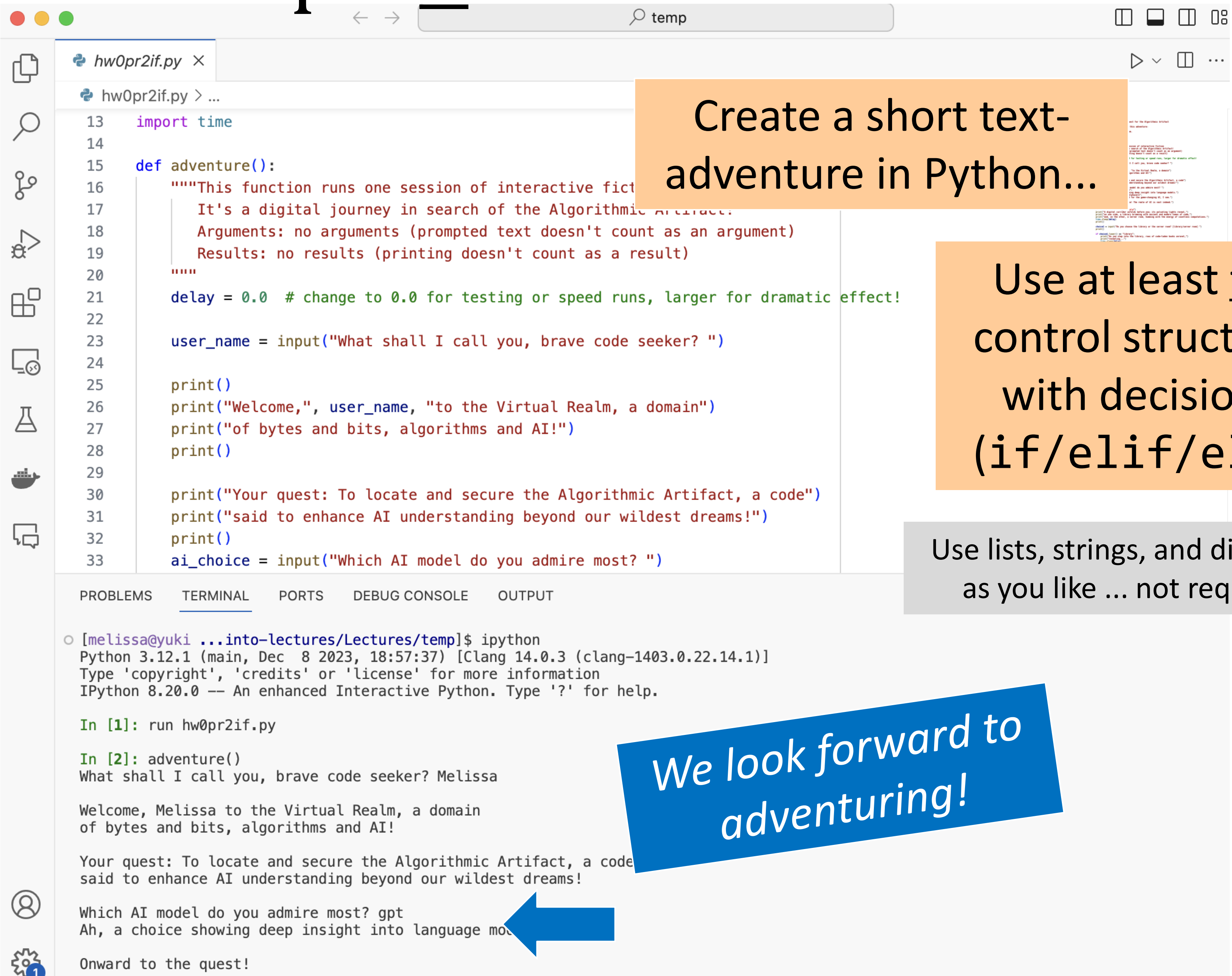
Pass 'em to your right!

(take a photo, if you'd like!)

"End of the line>" if you can grab, stack and pass forward: Thank you!!!

```
print("how we'd print a single quote!")
```

hw0pr2if: Interactive Fiction ☺



```python
import time

def adventure():
    """This function runs one session of interactive fict
       It's a digital journey in search of the Algorithmic
       Arguments: no arguments (prompted text doesn't count as an argument)
       Results: no results (printing doesn't count as a result)
    """
    delay = 0.0  # change to 0.0 for testing or speed runs, larger for dramatic effect!

    user_name = input("What shall I call you, brave code seeker? ")

    print()
    print("Welcome,", user_name, "to the Virtual Realm, a domain")
    print("of bytes and bits, algorithms and AI!")
    print()

    print("Your quest: To locate and secure the Algorithmic Artifact, a code")
    print("said to enhance AI understanding beyond our wildest dreams!")
    print()
    ai_choice = input("Which AI model do you admire most? ")
```

```
[melissa@yuki ...into-lectures/Lectures/temp]$ ipython
Python 3.12.1 (main, Dec  8 2023, 18:57:37) [Clang 14.0.3 (clang-1403.0.22.14.1)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.20.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: run hw0pr2if.py

In [2]: adventure()
What shall I call you, brave code seeker? Melissa

Welcome, Melissa to the Virtual Realm, a domain
of bytes and bits, algorithms and AI!

Your quest: To locate and secure the Algorithmic Artifact, a code
said to enhance AI understanding beyond our wildest dreams!

Which AI model do you admire most? gpt
Ah, a choice showing deep insight into language mo

Onward to the quest!
```

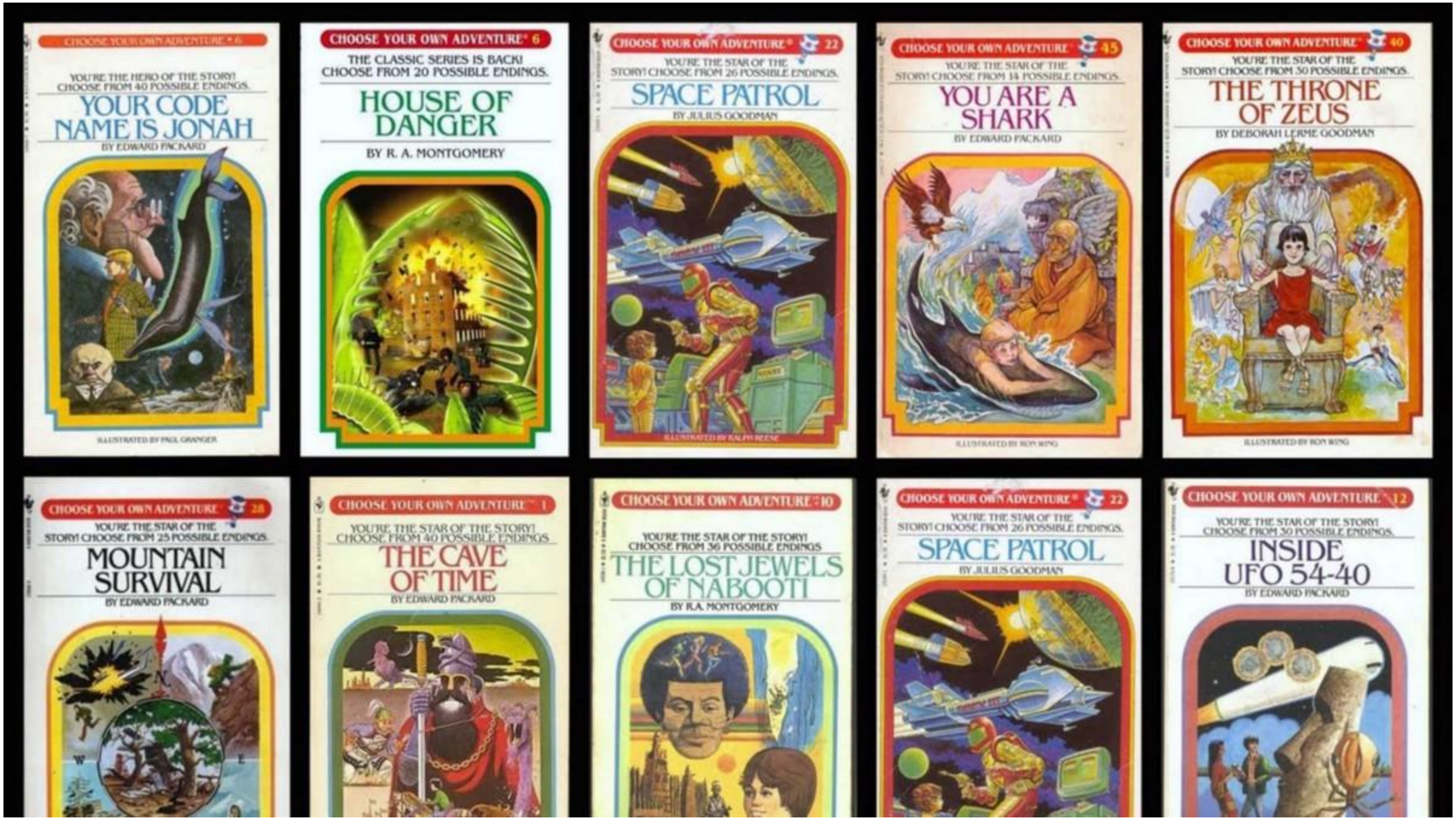
Create a short text-adventure in Python...

Use at least <u>five</u> control structures with decisions: (`if/elif/else`)

Use lists, strings, and dictionaries as you like ... not required ...

We look forward to adventuring!

What *is* programming ?

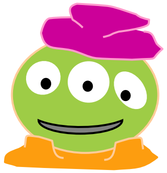

CHOOSE YOUR OWN ADVENTURE 6
YOU'RE THE HERO OF THE STORY! CHOOSE FROM 40 POSSIBLE ENDINGS.
YOUR CODE NAME IS JONAH
BY EDWARD PACKARD
ILLUSTRATED BY PAUL GRANGER
CHOOSE YOUR OWN ADVENTURE 6
THE CLASSIC SERIES IS BACK! CHOOSE FROM 20 POSSIBLE ENDINGS.
HOUSE OF DANGER
BY R. A. MONTGOMERY
CHOOSE YOUR OWN ADVENTURE 22
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 26 POSSIBLE ENDINGS.
SPACE PATROL
BY JULIUS GOODMAN
ILLUSTRATED BY RALPH REESE
CHOOSE YOUR OWN ADVENTURE 45
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 14 POSSIBLE ENDINGS.
YOU ARE A SHARK
BY EDWARD PACKARD
ILLUSTRATED BY RON WING
CHOOSE YOUR OWN ADVENTURE 40
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 30 POSSIBLE ENDINGS.
THE THRONE OF ZEUS
BY DEBORAH LERME GOODMAN
ILLUSTRATED BY RON WING
CHOOSE YOUR OWN ADVENTURE 28
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 25 POSSIBLE ENDINGS.
MOUNTAIN SURVIVAL
BY EDWARD PACKARD
CHOOSE YOUR OWN ADVENTURE 1
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 40 POSSIBLE ENDINGS.
THE CAVE OF TIME
BY EDWARD PACKARD
CHOOSE YOUR OWN ADVENTURE 10
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 36 POSSIBLE ENDINGS.
THE LOST JEWELS OF NABOOTI
BY R.A. MONTGOMERY
CHOOSE YOUR OWN ADVENTURE 22
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 26 POSSIBLE ENDINGS.
SPACE PATROL
BY JULIUS GOODMAN
CHOOSE YOUR OWN ADVENTURE 12
YOU'RE THE STAR OF THE STORY! CHOOSE FROM 30 POSSIBLE ENDINGS.
INSIDE UFO 54-40
BY EDWARD PACKARD

It's an adventure!

Another language!

Let's ***not only*** add another language…

… *but also make it **half the hw**!*



Even with three eyes, I must be misreading this!

Last time…

A Big Idea

Information

(simple)

+ Rules

Composition

Complexity

Another language *already*?

Python

General-purpose language

you might see 50% by the end of the term

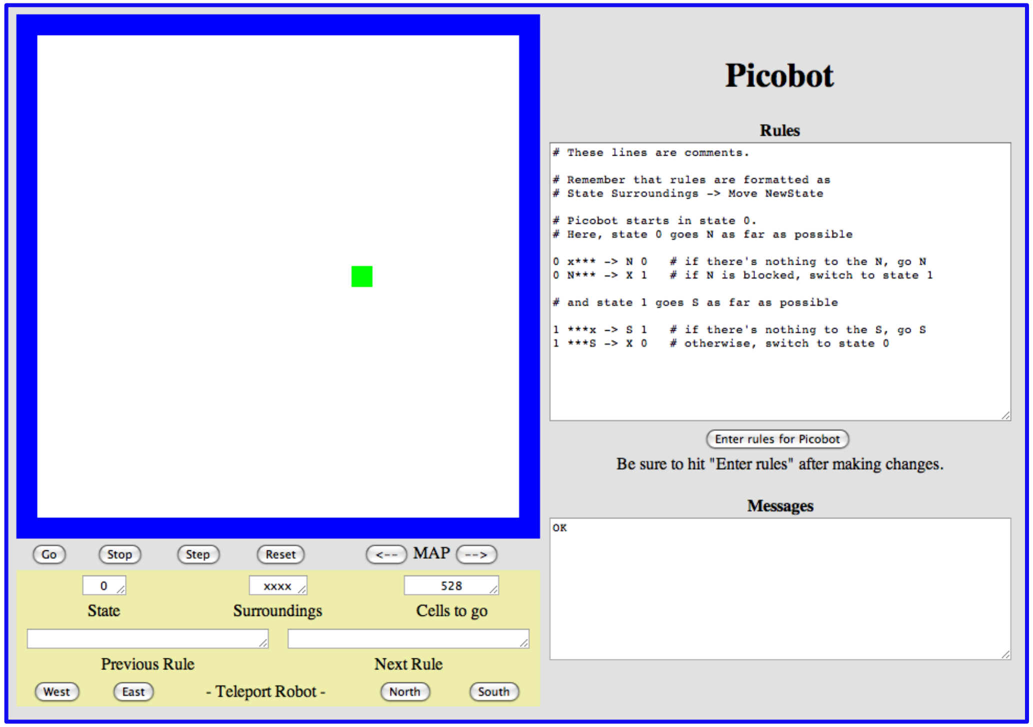
even then, <1% of its libraries!

Picobot

Special-purpose language

you'll see 100% in the next 10 minutes

Picobot!



```
# These lines are comments.

# Remember that rules are formatted as
# State Surroundings -> Move NewState

# Picobot starts in state 0.
# Here, state 0 goes N as far as possible

0 x*** -> N 0   # if there's nothing to the N, go N
0 N*** -> X 1   # if N is blocked, switch to state 1

# and state 1 goes S as far as possible

1 ***x -> S 1   # if there's nothing to the S, go S
1 ***S -> X 0   # otherwise, switch to state 0
```

Be sure to hit "Enter rules" after making changes.

OK

State: 0
Surroundings: xxxx
Cells to go: 528

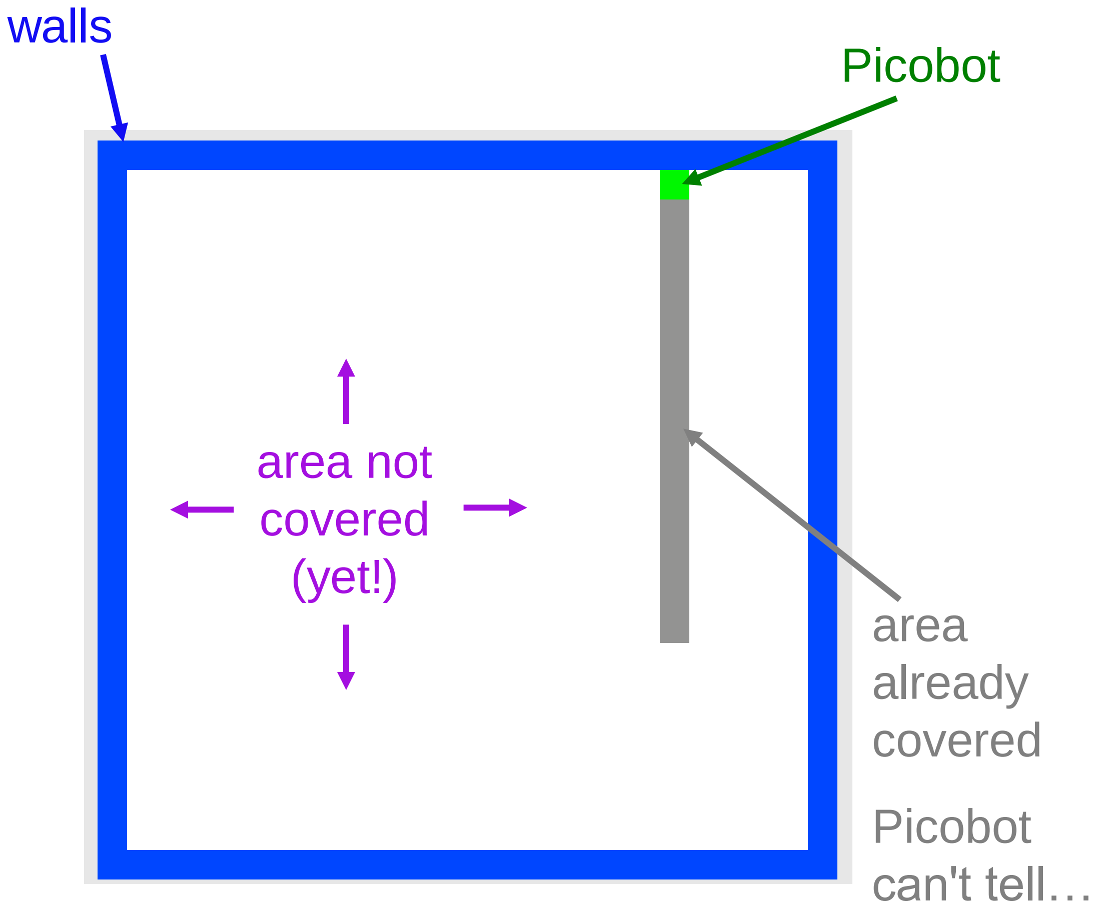
The Picobot simulator

www.cs.hmc.edu/picobot

Demo!

HW problems 3 and 4: Picobot!

Goal: full-room coverage with only ***local sensing***…



Inspiration?

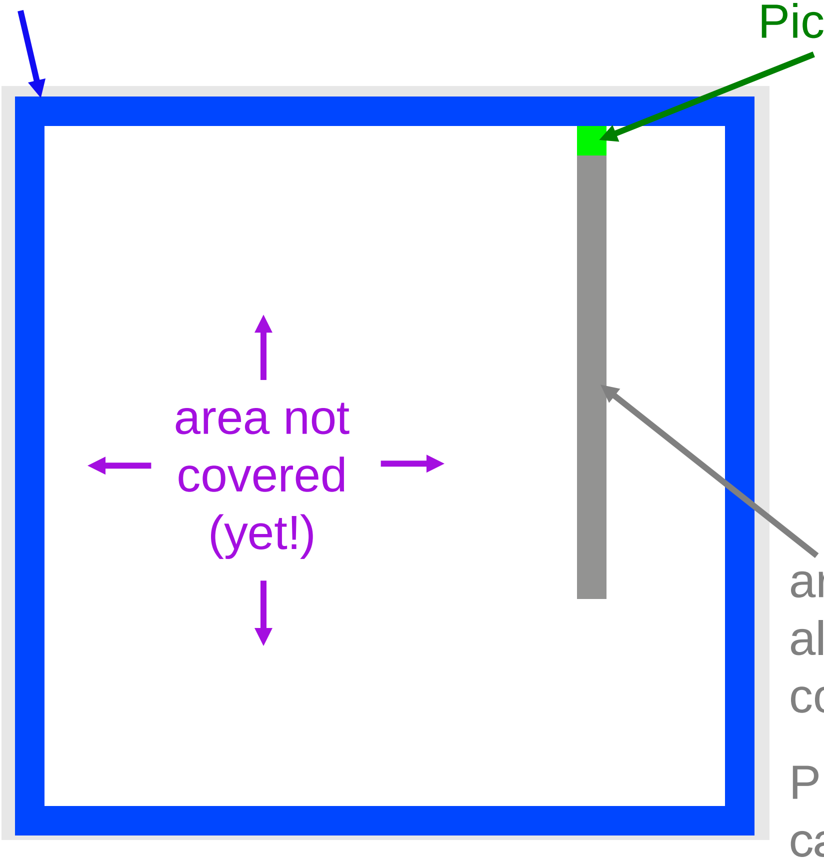
HW problems 3 and 4: Picobot!

Goal: full-room coverage with only ***local sensing***…

The Roomba!

can't tell "vacuumed" from "unvacuumed" area

https://www.youtube.com/watch?v=uGI8Od22WM4



walls

Picobot

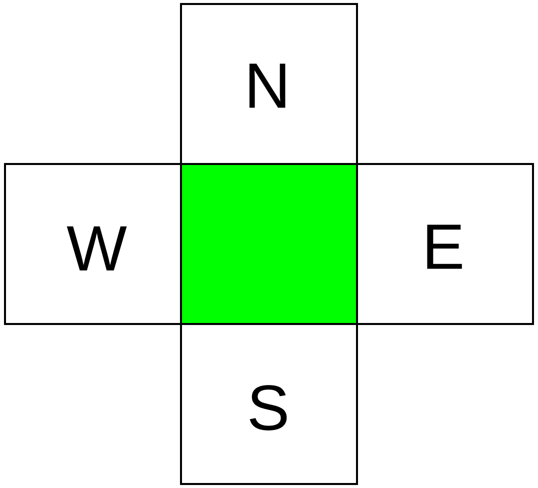
area not covered (yet!)

area already covered

Picobot can't tell…

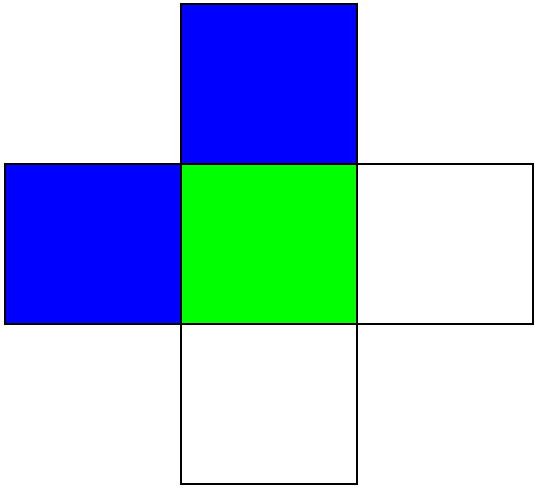
Let's see it!

Surroundings



Picobot can only sense things directly to the N, E, W, and S

For example, here its surroundings are



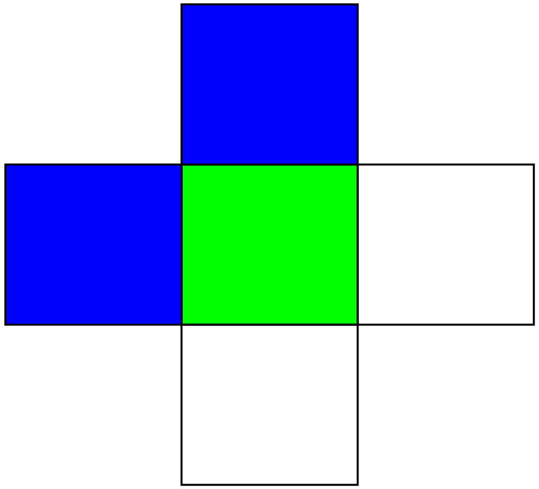
Nx**W**x

N E W S

Surroundings are always in NEWS order.

What are these surroundings?

How many surroundings are there, total?

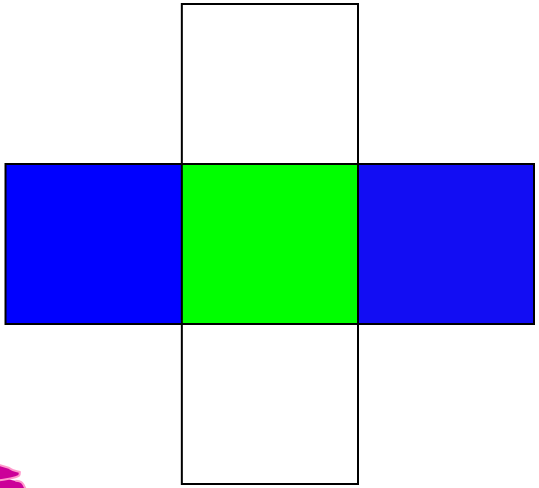


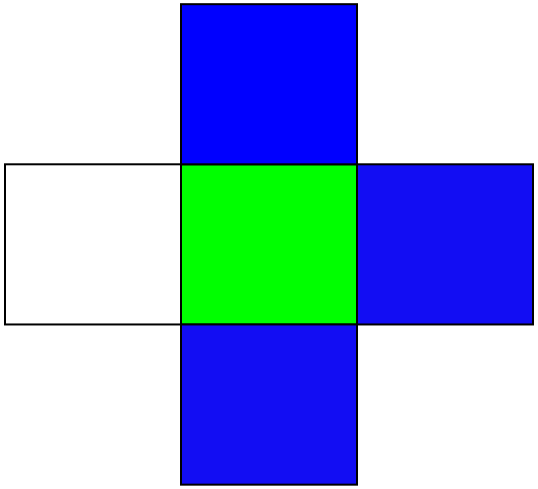
Surroundings are always in NEWS order.

NxWx

N E W S

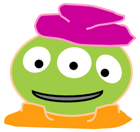
What are these two?





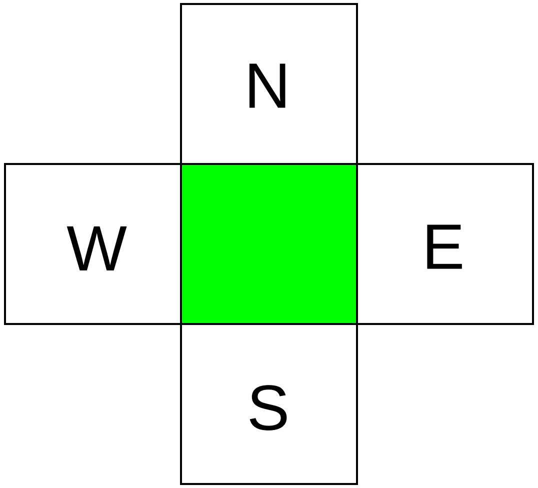
N E W S

N E W S



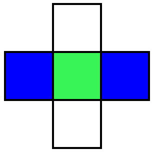
Wow - this one is disgusting!

Surroundings

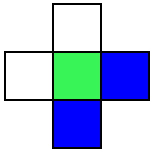


How many ***distinct*** surroundings are there?

5-second challenge

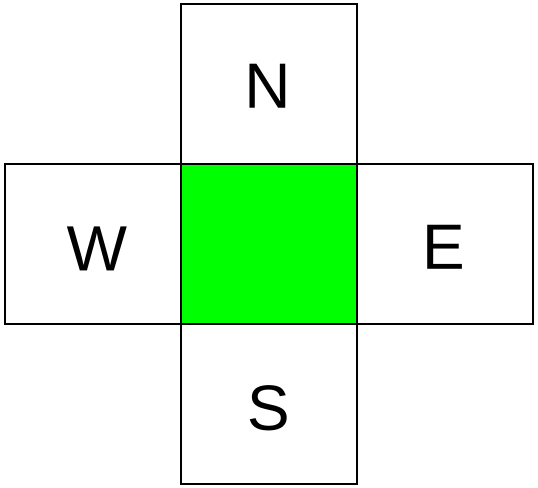


xEWx



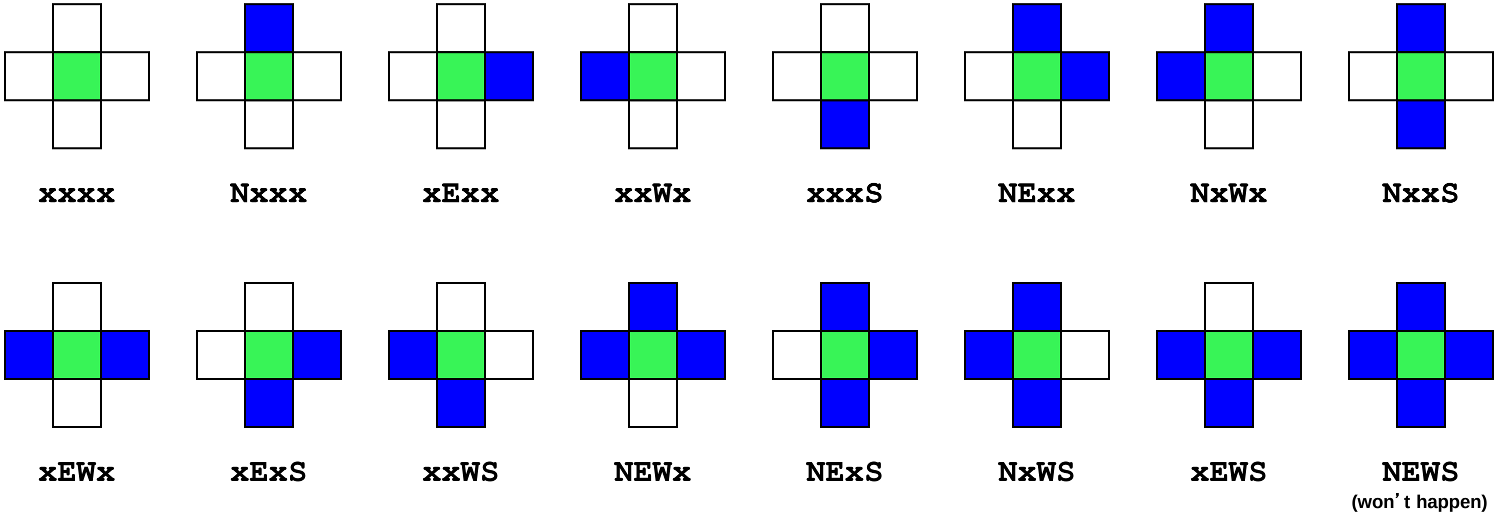
xExS

Surroundings

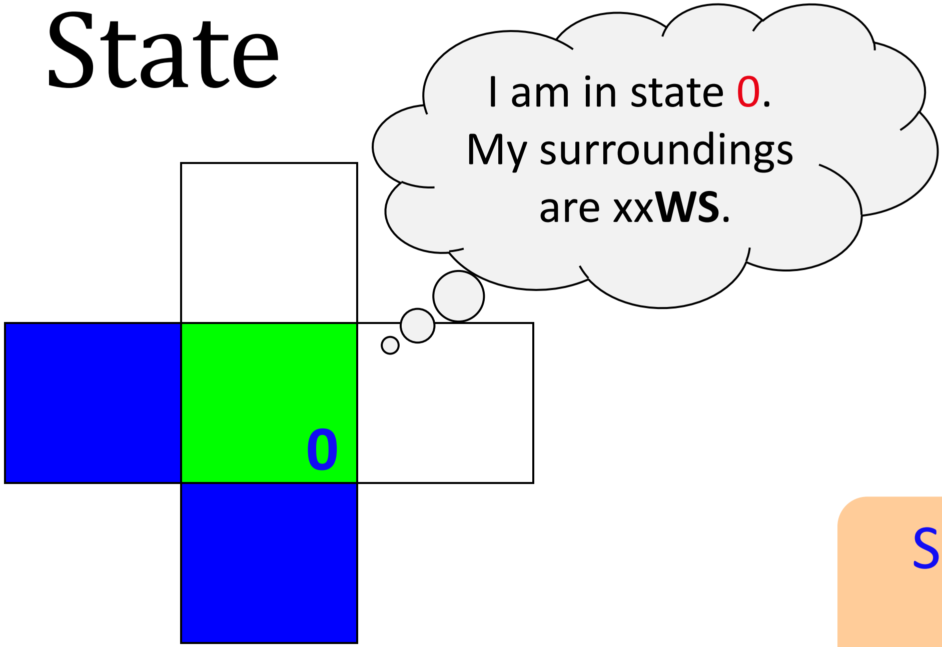


How many distinct surroundings are there?

2^4 == 16 possible



State



Picobot's memory is a single number, called its state.

State is the *internal context* of a computation, i.e., its *subtask*.

Picobot always starts in state 0.

State and **`surroundings`** represent everything Picobot knows about the world

self-contained
but not simplistic

Picobot programming ~ *rules*

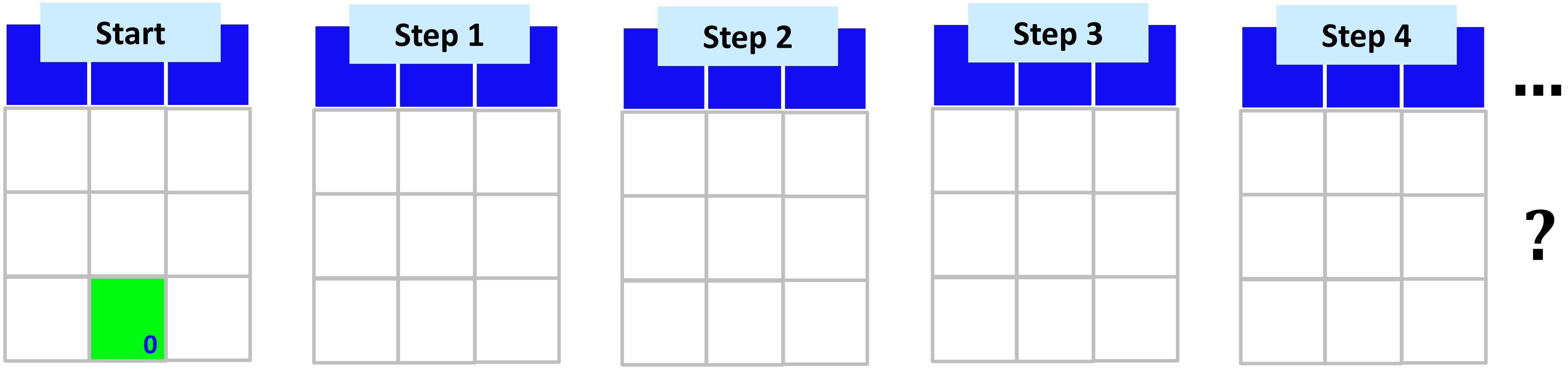
	current state	surroundings		step	new state
rule A	0	Nxxx	→	S	0
rule B	0	xxxx	→	N	0

these two rules are a complete Picobot program

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Picobot programming ~ ***rules***

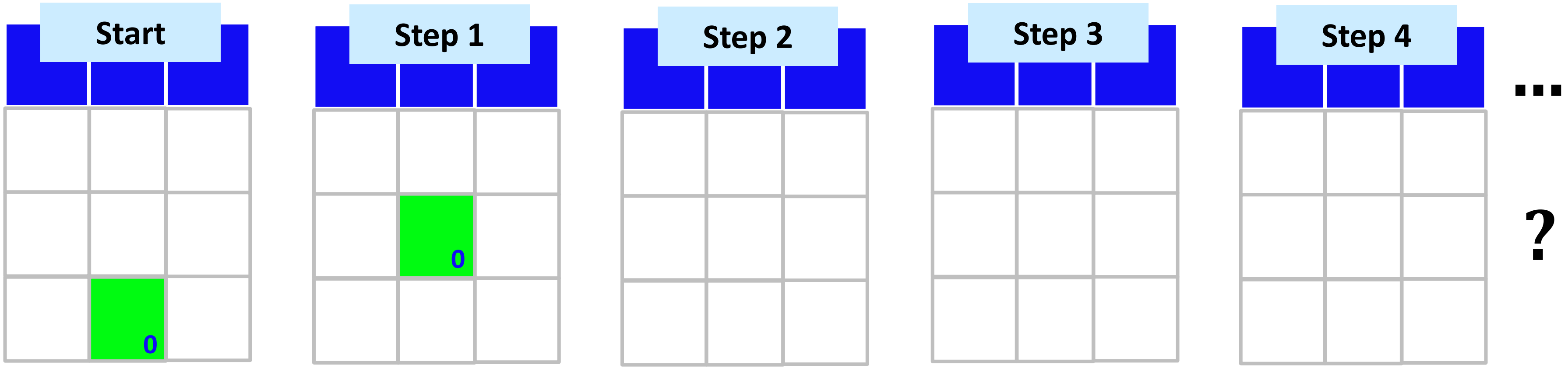
	current state	surroundings		step	new state
rule A	**0**	**Nxxx**	→	**S**	**0**
rule B	**0**	**xxxx**	→	**N**	**0**

these two rules are a complete Picobot program

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Picobot programming ~ ***rules***

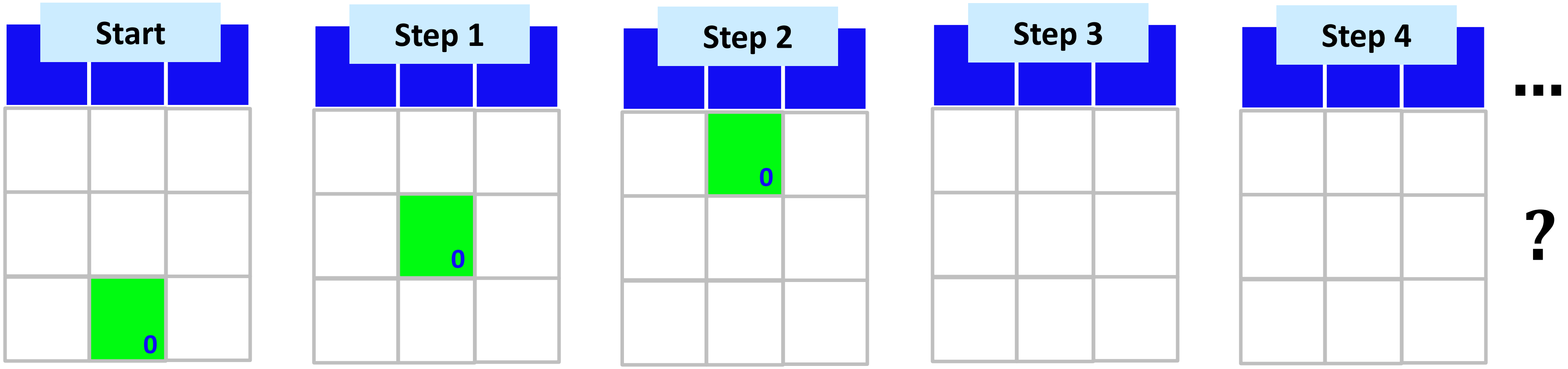
	current state	surroundings		step	new state
rule A	0	Nxxx	→	S	0
rule B	0	xxxx	→	N	0

these two rules are a complete Picobot program

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Picobot programming ~ ***rules***

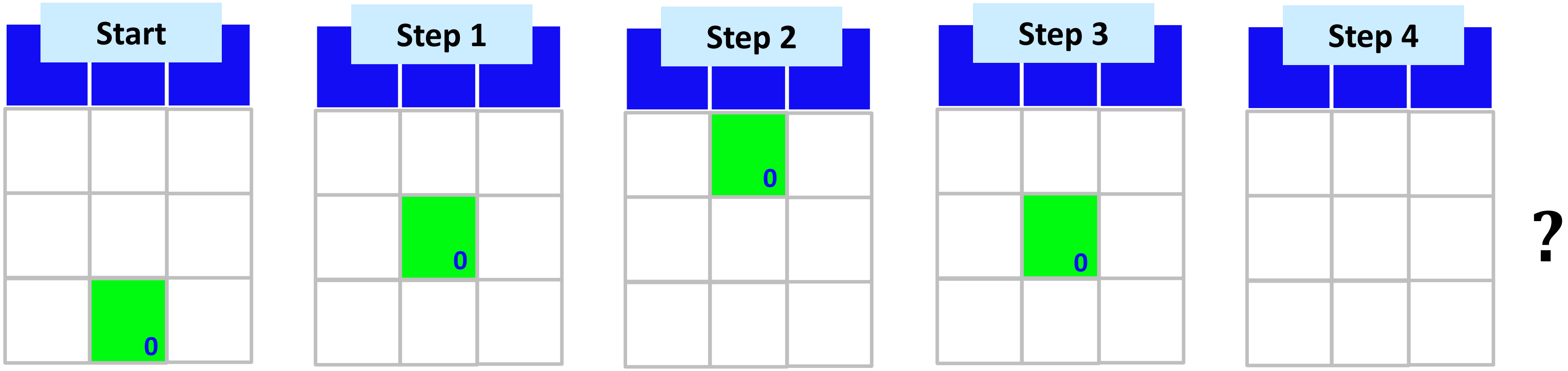
	current state	surroundings		step	new state
rule A	0	Nxxx	→	S	0
rule B	0	xxxx	→	N	0

these two rules are a complete Picobot program

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Picobot programming ~ ***rules***

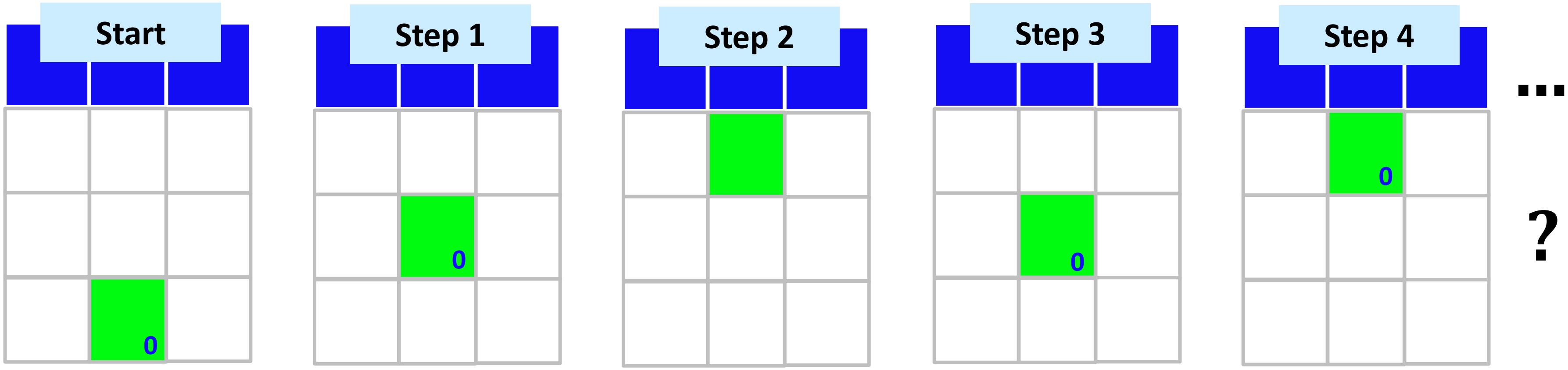
	current state	surroundings		step	new state
rule A	0	Nxxx	→	S	0
rule B	0	xxxx	→	N	0

these two rules are a complete Picobot program

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Picobot programming ~ ***rules***

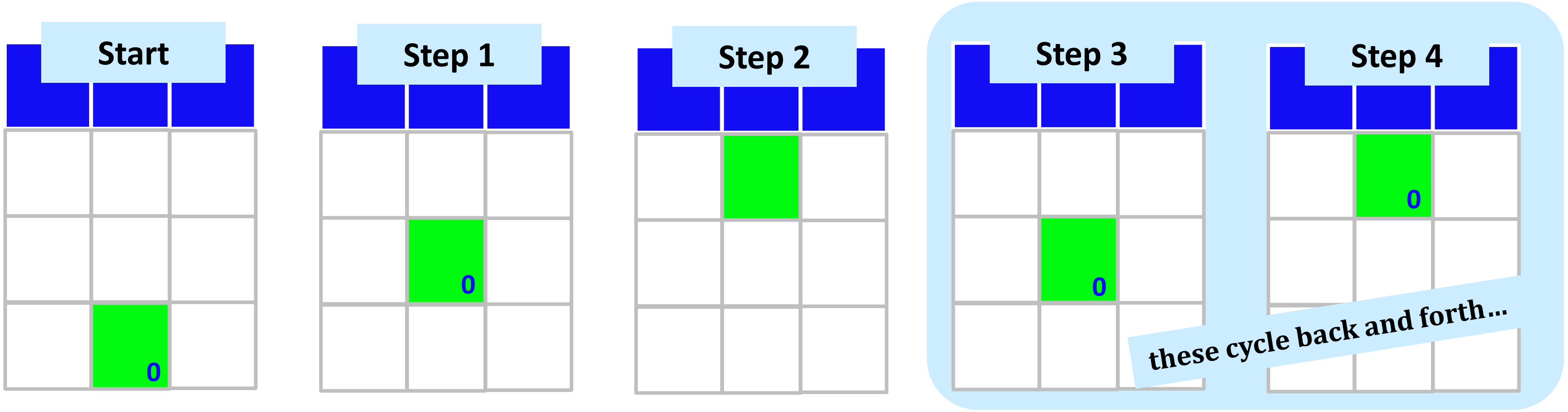
	current state	surroundings		step	new state
rule A	0	Nxxx	→	S	0
rule B	0	xxxx	→	N	0

these two rules are a complete Picobot program

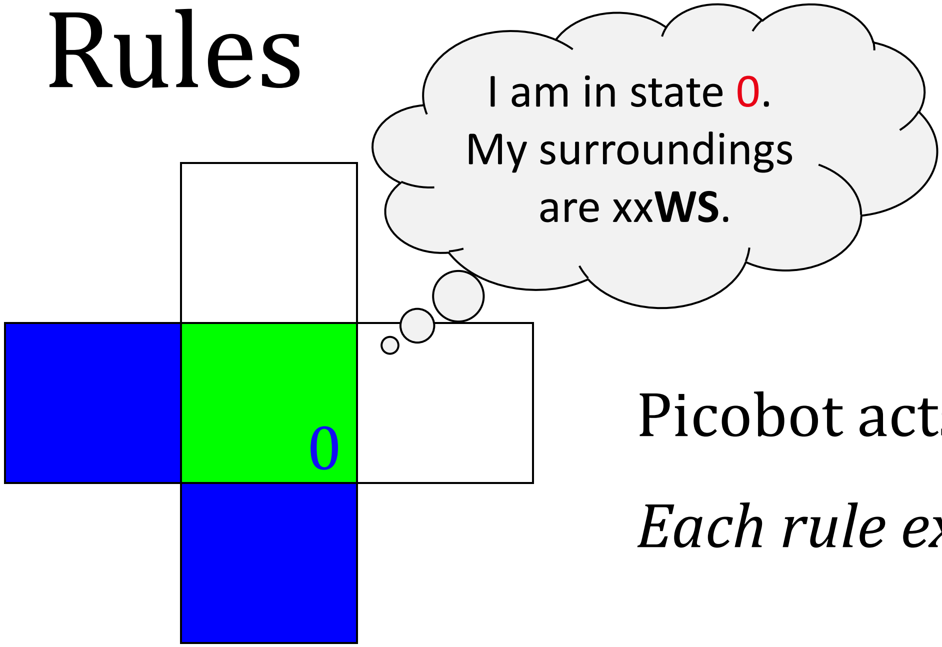
Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.



Rules



I am in state 0.
My surroundings
are xx**WS**.

Picobot acts through a **set of rules**

Each rule expresses ***your intent*** *for Picobot!*

current state	surroundings		step	new state
0	**xxWS**	→	**N**	**0**

syntax

If Picobot's in state
0 *seeing* **xxWS***,*

Then move **N***orth, and*
"change" to state **0***.*

semantics

Wildcards

I only care about **NORTH being EMPTY** 🟩

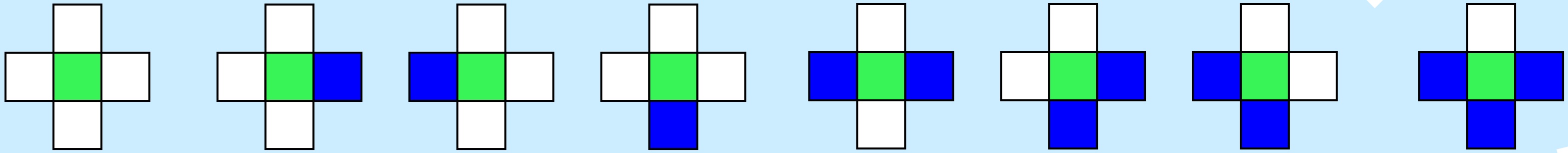
Asterisks * are wild cards.
They match walls ***or*** empty space:

current state	surroundings		direction	new state
0	**x*****	→	**N**	**0**

N must be empty

EWS may be wall ***or*** empty space

8 surroundings in one rule

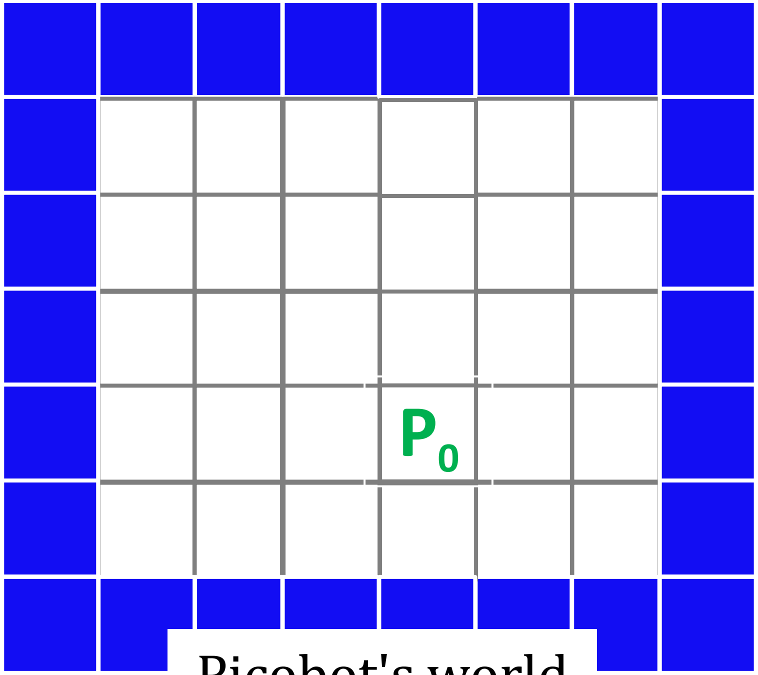


that's it!

The Rule is ***One step per rule***

One rule to rule them all?

That's *precious*!



Picobot's world

	state	surr.		move	new state
rule (A)	0	N***	->	W	1
rule (B)	0	x***	->	N	0
rule (C)	1	***x	->	S	1

more rules ________________________________

Write a rule so Picobot returns back upwards...

1. Run Picobot! Which rule **A, B,** or **C** runs ***first***? _______

   1a. How many times does **rule (A)** run? _________

   1b. How many times does **rule (B)** run? _________

   1c. How many times does **rule (C)** run? _________

2. Picobot stops when no rule matches. ***Where does it stop?***

3. Add a rule so that Picobot continues ***back upwards!***

Hint: Use a step of **x** to stay in place ...

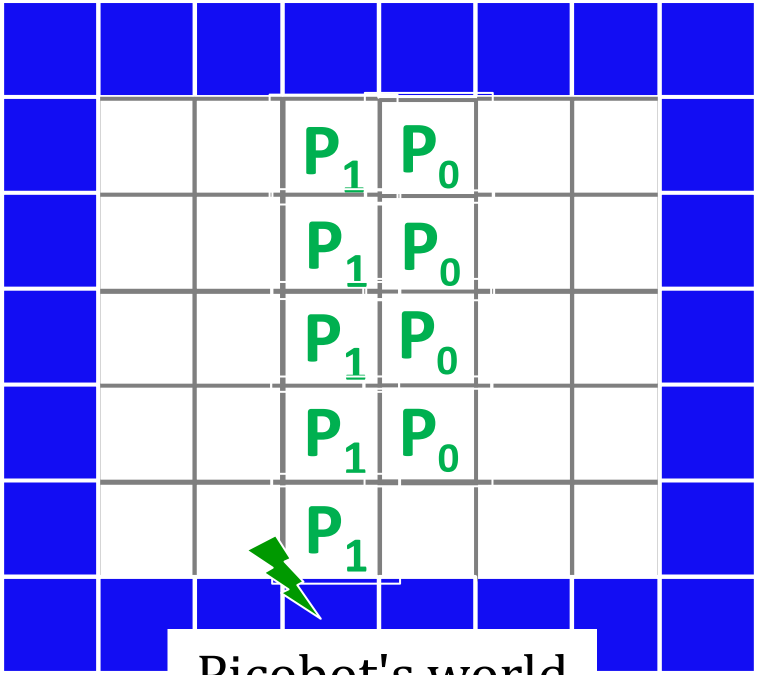
Extra #1 Rule A has a bug! What is it?
Extra #2 Add rules to finish exploring the empty room *from any starting point...*
Extra #3 ***How to do this in only 6 rules total?!***

Solutions!

The Rule is *One step per rule*

One rule to rule them all?

That's *precious*!



Picobot's world

error!

	state	surr.	move	new state
rule (A)	0	N***	-> W	1
rule (B)	0	x***	-> N	0
rule (C)	1	***x	-> S	1
more rules	1	***S	-> x	0

Write a rule so Picobot returns back upwards...

1. Run Picobot! Which rule **A, B,** or **C** runs ***first***? ***B***

   1a. How many times does **rule (A)** run? ***1***

   1b. How many times does **rule (B)** run? ***3***

   1c. How many times does **rule (C)** run? ***4***

2. Picobot stops when no rule matches. ***Where does it stop?*** ***see above!***

3. Add a rule so that Picobot continues ***back upward!*** ***see above!***

   Hint: Use a step of **x** to stay in place ...

Extra #1 Rule A has a bug! What is it? should be **N*x***

Extra #2 Add rules to finish exploring the empty room *from any starting point...* hw0pr3

Extra #3 ***How to do this in only 6 rules total?!*** extra!

Warning! *What's wrong here?*

state	surroundings		direction	new state
0	x***	→	S	0
0	***x	→	N	0

these two rules are a **broken** Picobot program!

Notes

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

Warning! *What's wrong here?*

state	surroundings	direction	new state
0	**x*****		
0	*****x**		

These two situations COULD BE the same!

these two rules are a broken Picobot program!

Notes

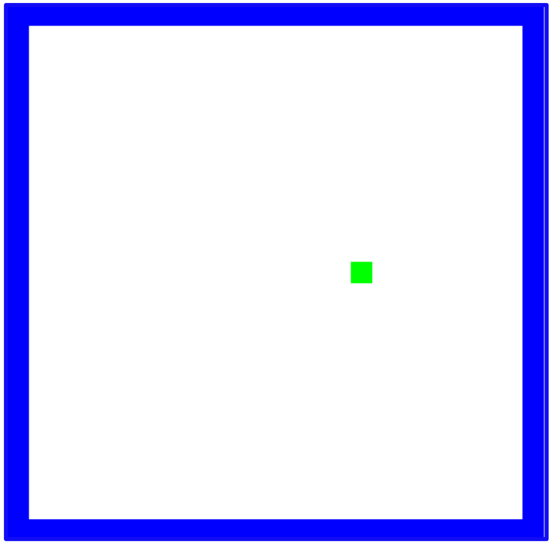
Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

There can only be **ONE** rule per situation!

and a "situation" is *state and surroundings*

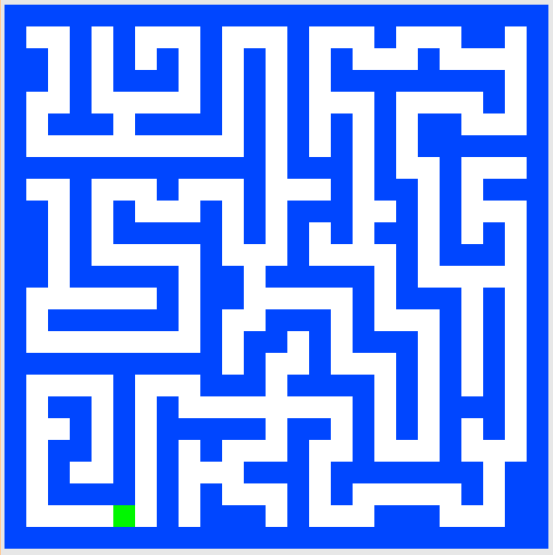
CS ~ *Complexity* Science



problem 3

Shortest Picobot program:

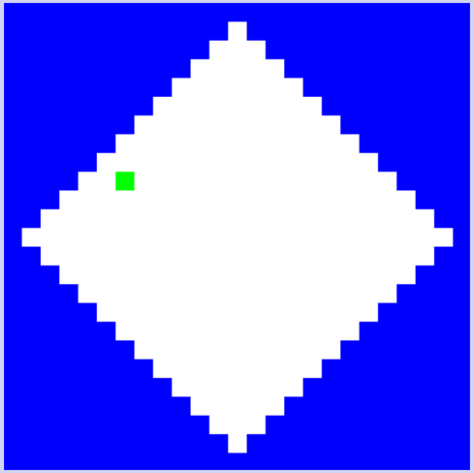
6 rules



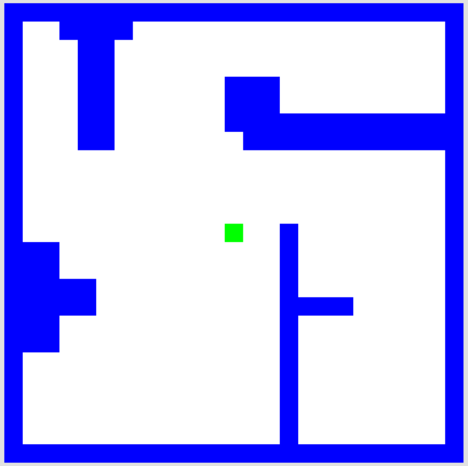
problem 4

Shortest Picobot program:

8 rules

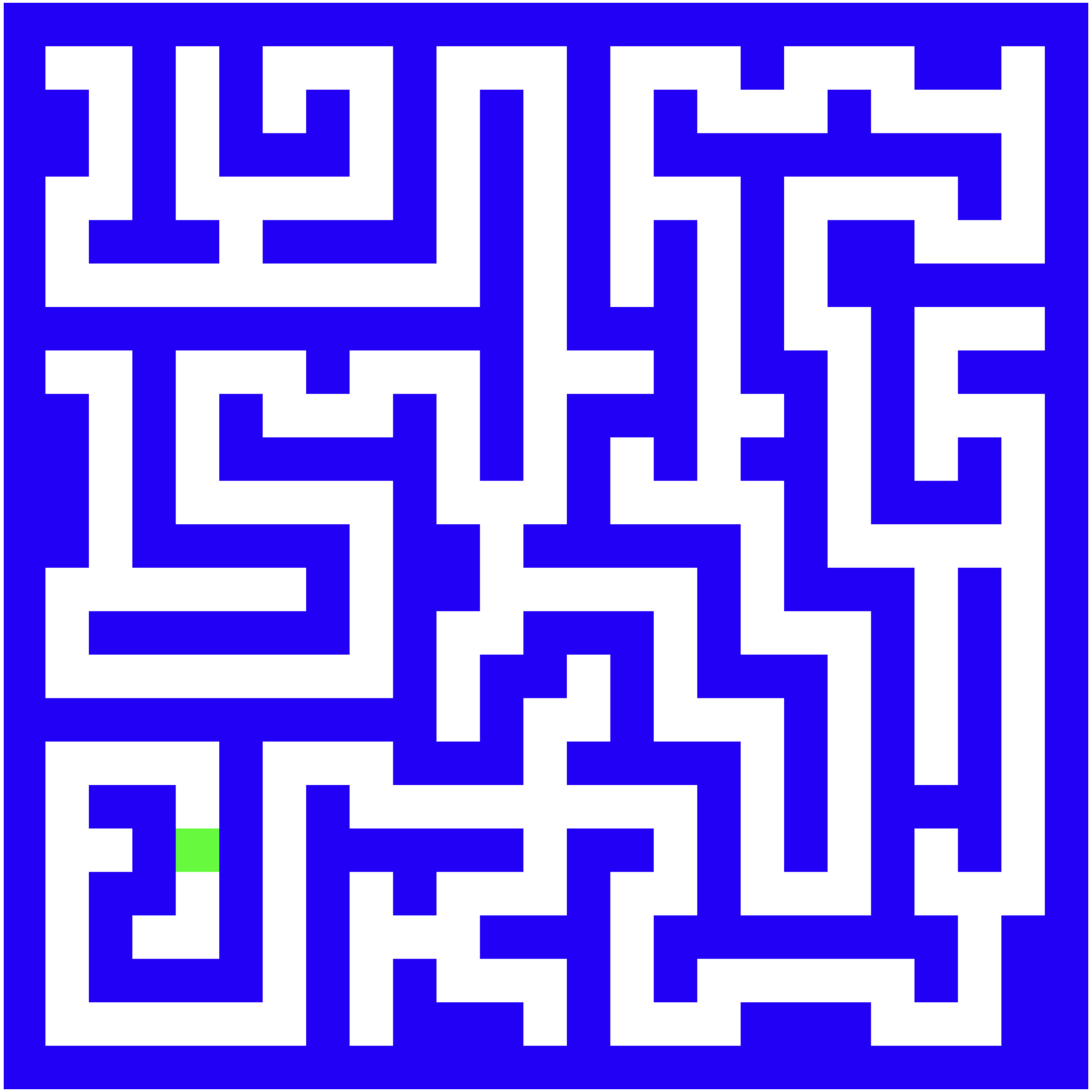


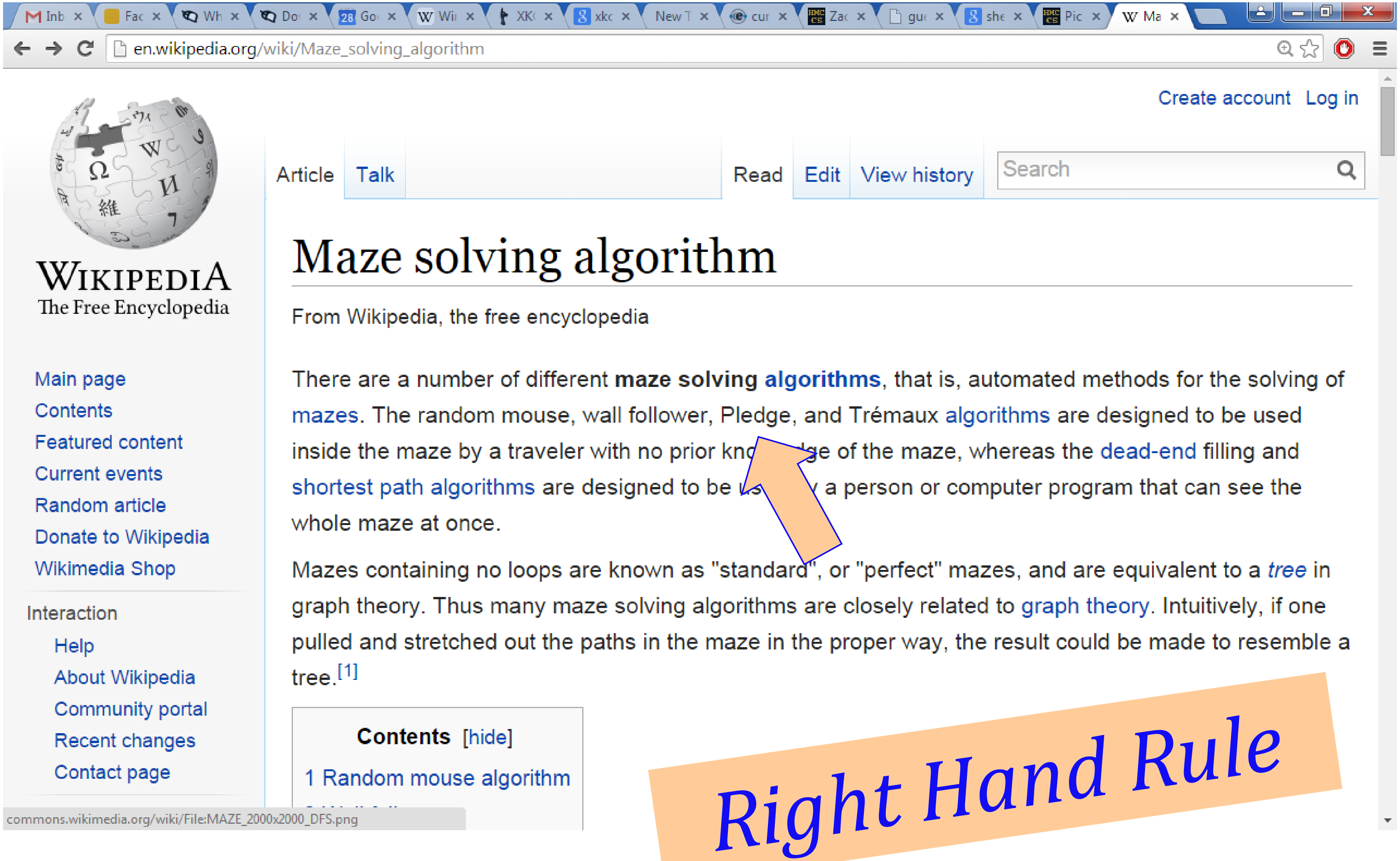
pr. 5 (extra!)



pr. 6 (extra!)

Maze strategies?





Maze solving algorithm

From Wikipedia, the free encyclopedia

There are a number of different **maze solving algorithms**, that is, automated methods for the solving of mazes. The random mouse, wall follower, Pledge, and Trémaux algorithms are designed to be used inside the maze by a traveler with no prior kno[illegible]ge of the maze, whereas the dead-end filling and shortest path algorithms are designed to be u[illegible]y a person or computer program that can see the whole maze at once.

Mazes containing no loops are known as "standard", or "perfect" mazes, and are equivalent to a *tree* in graph theory. Thus many maze solving algorithms are closely related to graph theory. Intuitively, if one pulled and stretched out the paths in the maze in the proper way, the result could be made to resemble a tree.[1]

Contents [hide]

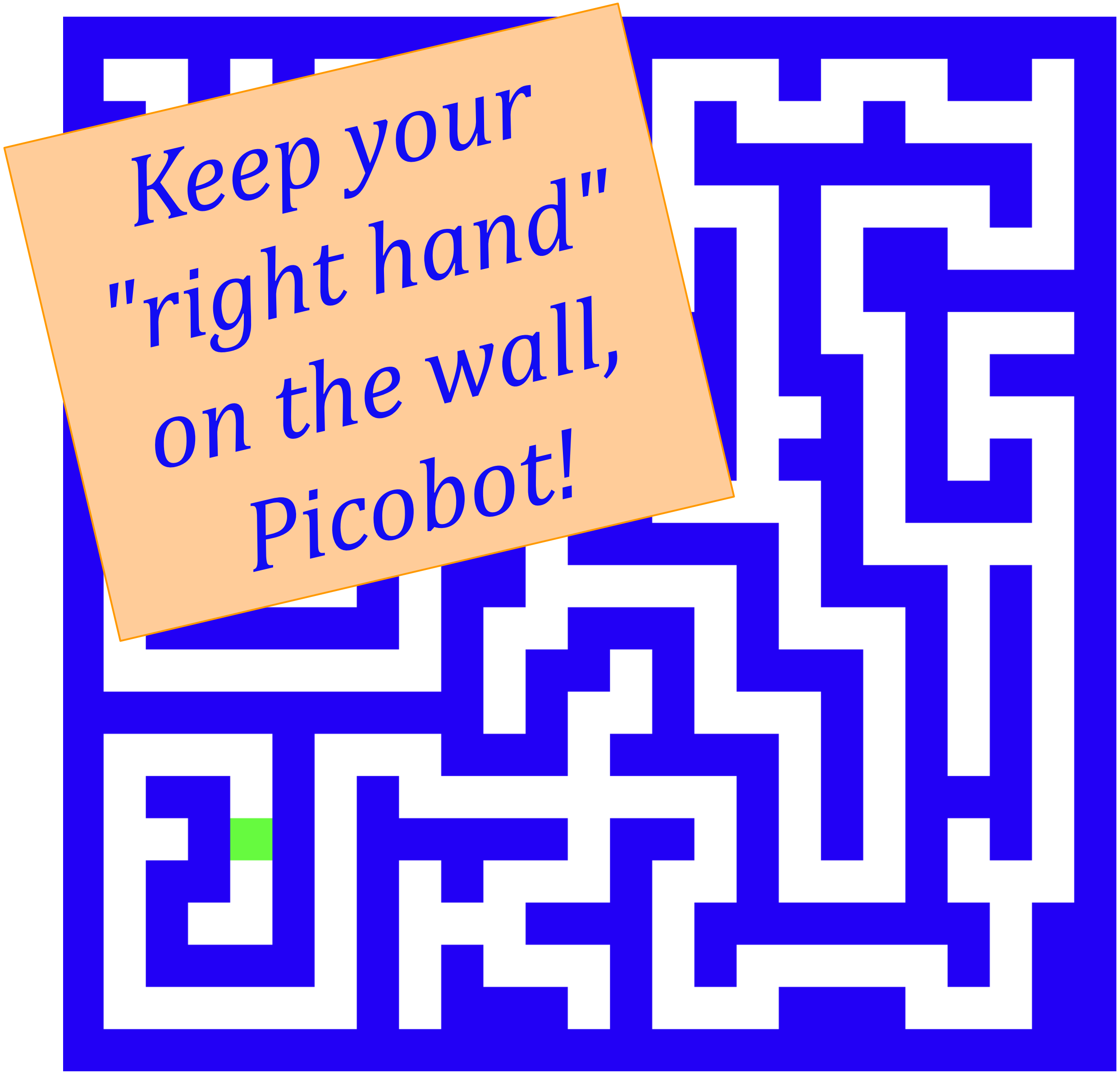
1 Random mouse algorithm

commons.wikimedia.org/wiki/File:MAZE_2000x2000_DFS.png

Right Hand Rule

Maze strategies?

Right Hand Rule

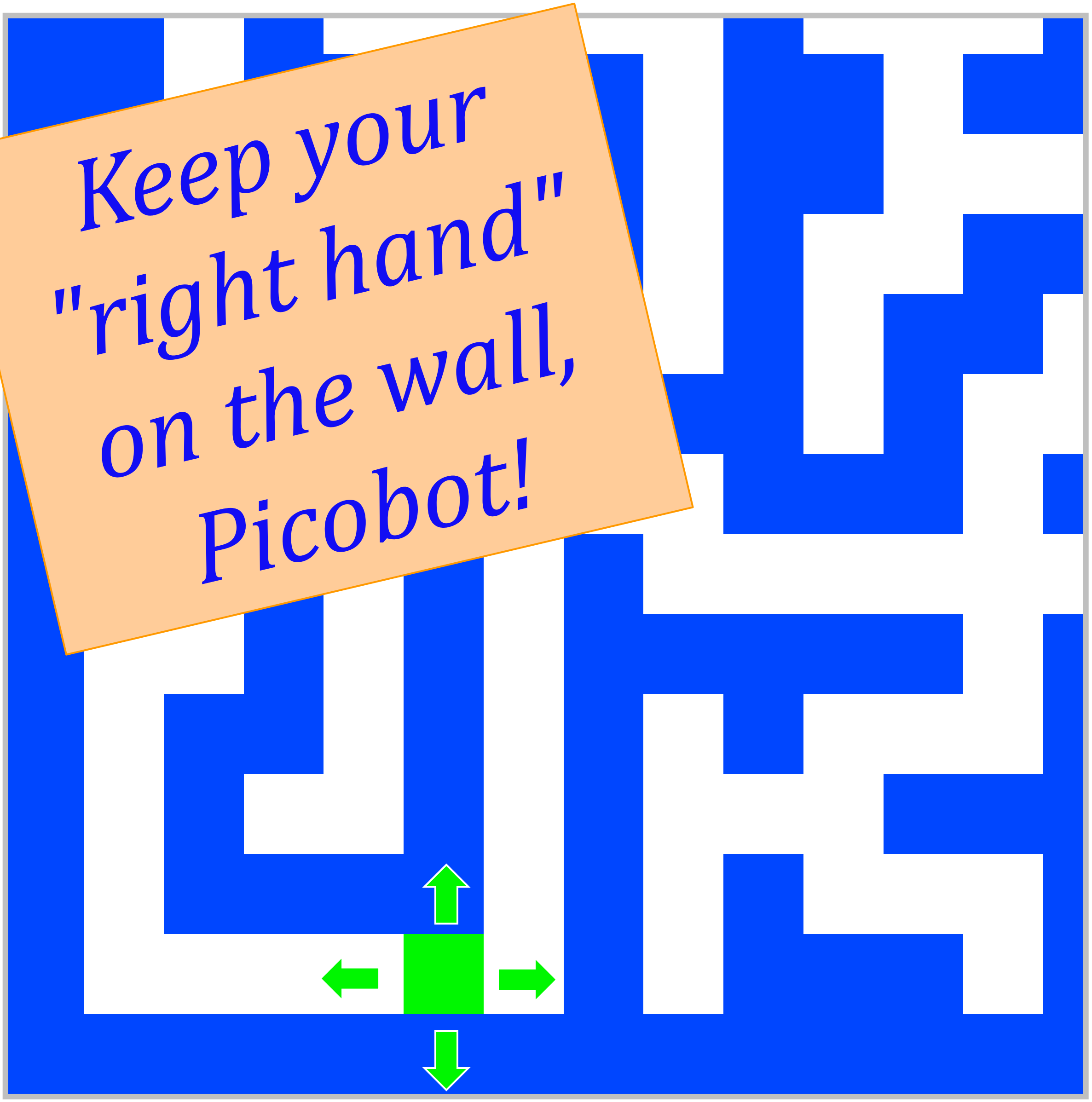


Keep your "right hand" on the wall, Picobot!

Why might this be *difficult* for Picobot?

Maze strategies?

Right Hand Rule



Keep your "right hand" on the wall, Picobot!

	facing	to the right
State **0**	N	E
State **1**	E	S
State **2**	W	N
State **3**	S	W

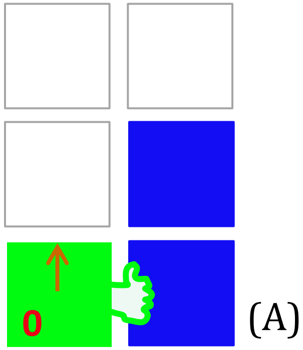
We'll need to use state to represent the ***direction Picobot is facing***.

Suppose Picobot wants to traverse a maze ***with its right hand <u>always</u> on the wall…***

(A) **CORRIDOR** rule

If you're facing N with a wall at right and space ahead then go forward”

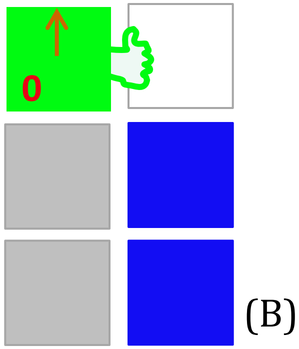
```
0      xE**      ->      N      0
```



(B) **INTERSECTION** rule

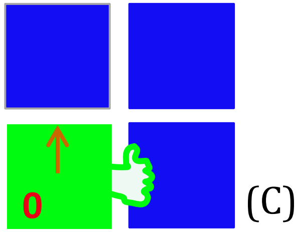
“If you're facing North and lose the wall, then get over to the wall now!”

```
0                ->
```



(C) **DEAD END** rule

Write 1 rule to tell Picobot to do the right thing if it hits a dead end.



Repeat this IDEA for all four states, representing all four ***facing directions.***

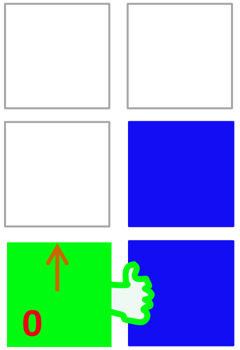
Suppose Picobot wants to traverse a maze ***with its right hand <u>always</u> on the wall...***

(A) **CORRIDOR** rule

If you're facing N with a wall at right and space ahead then go forward"

```
0     xE**     ->     N     0
```

state 0 means "still facing north"



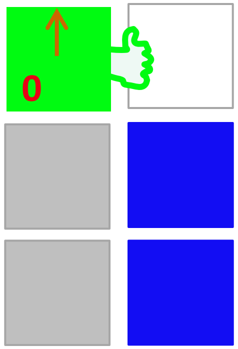
(A)

(B) **INTERSECTION** rule

"If you're facing North and lose the wall, then get over to the wall now!"

```
0     *x**     ->     E     1
```

state 1 means "now facing east"



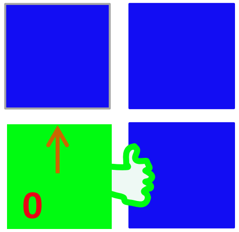
(B)

(C) **DEAD END** rule

Write 1 or 2 rules to tell Picobot to do the right thing if it hits a dead end.

```
0     NE**     ->     X     2
```

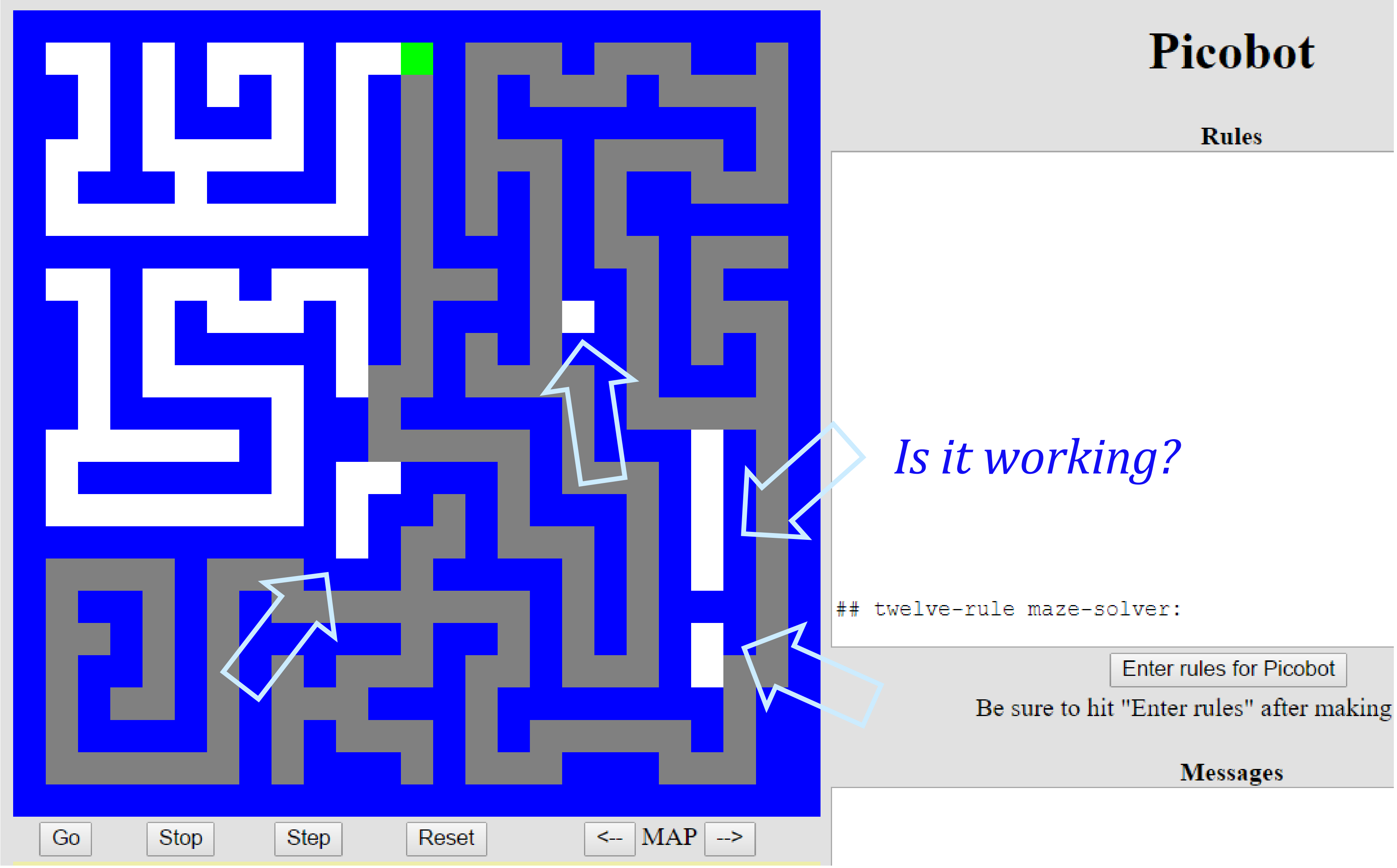
state 2 means "now facing west"



(C)

Repeat this IDEA for all four states, representing all four ***facing directions.***

Hooray!?!



Is it working?

Picobot

Rules

```
## twelve-rule maze-solver:
```

Be sure to hit "Enter rules" after making

Messages

Southward!

Northward!

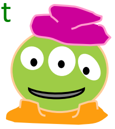
- Onward -

Westward!

Eastward!

hw0

You are not alone!



I can attest to that!

Come to tutoring hours!

Post questions to piazza...

Happy Picobotting!

And, good luck with the ***adventure*** *of Python!*