CS 30 Discussion 1A

10.09.2020

Welcome to CS30 Discussion

- TA: Yichao (Joey) Zhou -- yichao.joey.zhou@gmail.com
- LA: Julia and Denise.

- Office Hours: Wednesday 10:00 am -- 12:00 pm
  - Online: https://ucla.zoom.us/j/95020786522.
  - For help, not answers

- Discussion Section: Friday 2:00 pm -- 4:00 pm
  - Review main concepts from class
  - Practice programming with new problems

About me

- I am a fourth-year PhD at ScAi Data Mining Lab, Department of Computer Science, working with Dr.Wei Wang and Dr.Kai-Wei Chang.
- Other courses I taught: CS145 (data mining) CS146 (machine learning), CS32
- Research Interests:
  - Natural Language Processing
  - Information Extraction, Text Mining
  - Clinical Case Report Mining
- https://yz-joey.github.io

	Homographic Puns
1.	Did you hear about the guy whose whole left side was cut off? He's **all right** now.
2.	I'd tell you a chemistry joke but I know I wouldn't get a **reaction**.

	Heterographic Puns
1.	The boating store had its best **sail (sale)** ever.
2.	I lift weights only on Saturday and Sunday because Monday to Friday are **weak (week)** days.

Pun Detection and Generation

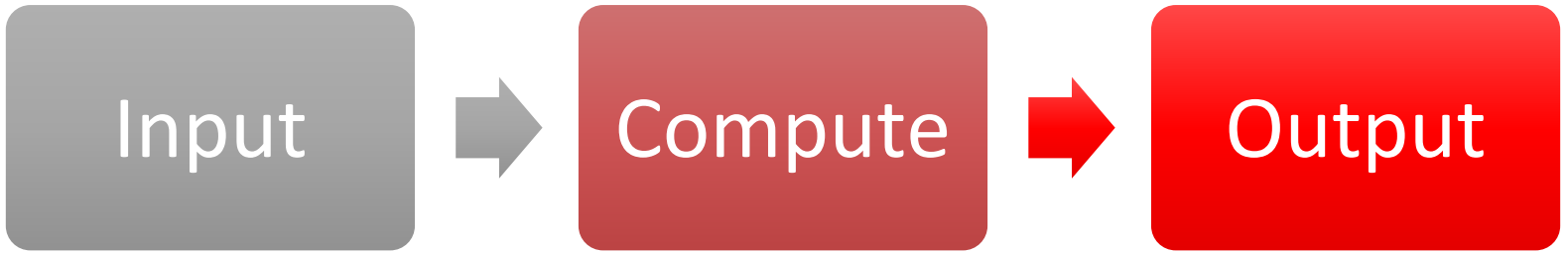
UCLA Samueli Computer Science

Materials

1. Course Zoom Link
2. Discussion and Office Hour Zoom Link
3. Course Videos, Slides, Homeworks CCLE
4. Asking Questions Piazza
5. Textbook, Reading Materials CS for All
6. Additional Materials Link

Work in groups in the discussion

Programming




GOOD NEWS EVERYONE
PROGRAMMING IS NOT HARD
imgflip.com

IDLE IDE

IDE stands for **Integrated Development Environment**. This is a **software environment** which usually consist of a software development package containing Code Editor, Build Automation, Tools and Debugger.

IDLE is the IDE provided with Python. Using IDLE is not a requirement for using Python. There are many other IDEs that can be used to write Python programs, not to mention a variety of text-based programmer's editors that many programmers prefer to IDEs.

Week 1 – Recap
Functions

- **A reusable unit of code**
- The key to functions (as to much of programming) is that it takes some input(s), and produces outputs.
- Like mathematical functions, the input is the data (variable), and the output depends on the concrete input

  - `def double (x): return 2*x`
  - x: parameter(variable)

Week 1 – Recap
Functions

- A reusable unit of code
- The key to functions (as to much of programming) is that it takes some input(s), and produces an output.

E.g.

```
def double (x):
    return 2*x
>>double(3)
```

1. x is an **argument** to the function
2. When you call the function you pass a **value** for the argument

Week 1 – Recap

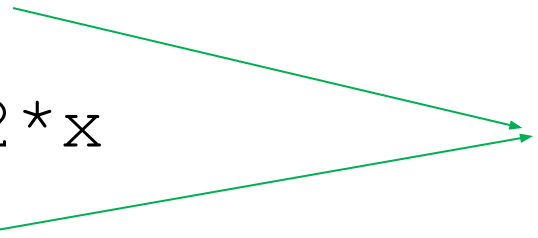
- **Data Types**: Int, String , Bool.
- **‘Strings’ versus variables**. Remember that quotes gives you literally exactly what is between the quotes; that's why they're called string *literals*. Ex:

```
>> num = 8
>> print( num )
>> print(‘num’)
```

Week 1 – Recap

- **Scope:** Every function has its own arguments and is different from other variables with same name
- **E.g.**

```
def double(x):
        return 2*x
def triple(x):
        return 3*x
```



These two x's are different

Week 1 – Common pitfalls

- **Function syntax**: Function definition contains the keyword “def”, name of the function and parameters in parentheses.

```
def nameOfFunction(parameters…):
    return ‘’
```

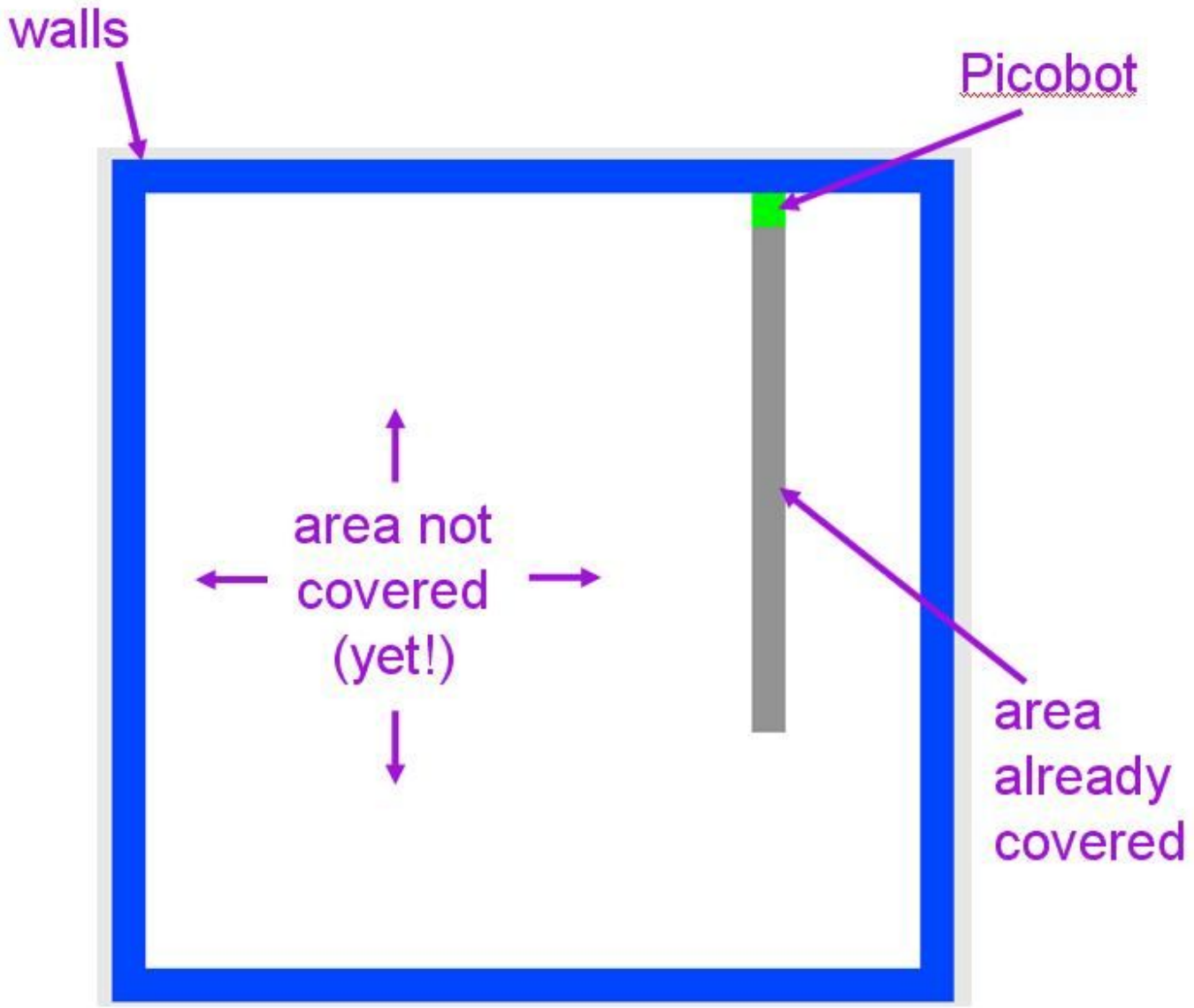
- **Indentation**: Python’s syntax requires you to indent your code, the following code will result in IndentationError: expected an indented block

```
def nameOfFunction(parameters…):
return ‘’
```

Picobot

- Picobot is a little roomba-like robot
  - Can sense its immediate environment
  - Can have different "states"
  - environment + state -> movement + new state
- Why Picobot?
  - Not very similar to a traditional programming language
  - Very good at expressing some *algorithms*

Picobot



Goal: whole-environment coverage with only *local sensing*…

Rules Format

- Rules consist of a few parts:
  - < current state >
  - < N > < E > < W > < S >
  - The "->" transition
  - < next direction >
  - < new state >
- Some special characters:
  - "*" means "Either a wall or free space"
  - NEWS means a wall
  - "x" before the arrow (confusingly!) means "free space"
  - "X" after the arrow means "do nothing"

1 xE** -> N 3

- Example:
  - 1 xE** -> N 3
  - In English this means: "In state 1, if Picobot senses a wall to its east and a free space to its north, then it should go north and change to state 3, no matter what is to its south or west.

States

- States introduce memory
- Use states to represent when Picobot needs to "decide" to do something *based on what it has previously done*
- i.e. no matter where it starts, make PB travel to the southern wall in a straight line, then travel in a circle around the walls

A practice problem

- Make Picobot travel to the southeast corner of the board in straight lines
  0 *x** -> E 0
  0 *E** -> S 1
  1 ***x -> S 1
- And a row, using states:
  0 *x** -> E 0
  0 *E** -> X 1
  1 **x* -> W 1
- What if?
  0 *x** -> E 0
  0 xE** -> W 0

- How do we make Picobot travel in a zigzag pattern across the (empty) board?
  - From any given location, PicoBot should take one step north, then one step east, then one step north ad infinitum until it reaches a wall and can't go any farther
  - It should never take two consecutive steps in the same direction

Worksheet