LEDE Algorithms

Richard Dunks
Chase Davis

WEEK 1
CLASS 1

Goals for today

- Review basic course information
- Discuss algorithms and why they’re important
- Explain the basic structure of algorithms
- Create a basic algorithm in Python
- Review and discuss basic data structures in Python

Outcomes for today

- You will have a basic understanding of algorithms and why they’re important to study
- You will understand the basic structure of an algorithm
- You will have designed and implemented a basic algorithm in Python
- You will understand basic data structures in Python

INTRODUCTIONS

Name

Briefly describe your background

Tell us one thing you’d like to get out of this class

BACKGROUND ON THIS CLASS

Assumptions

- You’re all journalists or interested in journalistic tradecraft as it relates to data
- You’re not computer scientists
- You’re not interested in creating the next Facebook (unless it’s for journalists)
- You’re not great at math (don’t worry, I’m not either)
- You care more about the why than the how

Course Learning Objectives

- You will understand the basic structure and operation of algorithms
- You will understand the primary types of data mining algorithms, including techniques of supervised and unsupervised machine learning
- You will be practiced in implementing basic algorithms in Python

Course Learning Objectives

- You will be able to meaningfully explain and critique the use and operation of algorithms as tools of public policy and business
- You will understand the role algorithms play in the newsroom

Course Policies

- Attendance and Tardiness – Show up to every class and be on time
- Participation – Take an active role in your learning
- Late Assignments – Get your work in on time
- Office hours – I’m available when you need me via email (but ask the TAs)

Class Structure

- Lecture from 10 am to 1 pm
  - 3 - 50 minute blocks
  - 10 minute breaks (50 minutes past to top of hour)
- Lunch from 1 pm to 2 pm
- Lab from 2 pm to 5 pm
  - Open time to work on exercises
  - TA-led, instructor available as possible

HOW DO I TEACH?

I’ve written several blogposts on this:

The Missing Pedagogy in Computer Science - http://wp.me/p2PLpM-1Jr

Teaching with Intention - http://wp.me/p2PLpM-1Or

Some of the Topics We'll Be Covering

- Supervised learning
  - Linear regression
  - Decision Trees
- Unsupervised learning
  - Clustering
  - DBScan
- Natural Language Processing
- Reverse Engineering Algorithms

Discussion

In groups of 4-5, discuss the following questions:

- What is an algorithm?
- Why are they important?
- Why are we studying them in this class?

Write out the major themes that come up in your discussion on the Post-It notes provided and we'll discuss as a class

What is an Algorithm?

- A precise set of instructions required to accomplish a task
- Roughly broken down into three components:
  1. Inputs
  2. Operation
  3. Output
- Each step in the operation must be clearly defined and work the same with any input

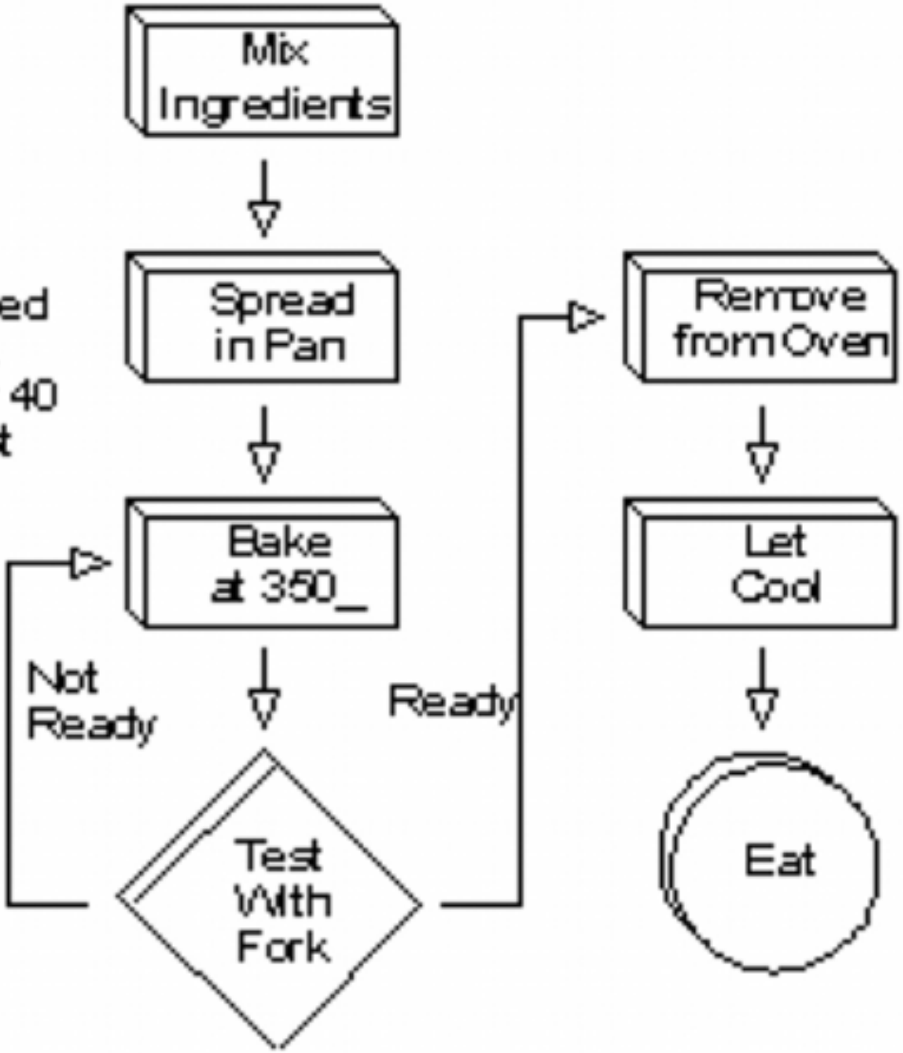
Recipe
CHOCOLATE CAKE

4 oz. chocolate
1 cup butter
2 cups sugar
3 eggs
1 tsp. vanilla
1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

```
Declare variables:
    chocolate   eggs       mix
    butter      vanilla
    sugar       flour
mix = melted  ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```



Inputs

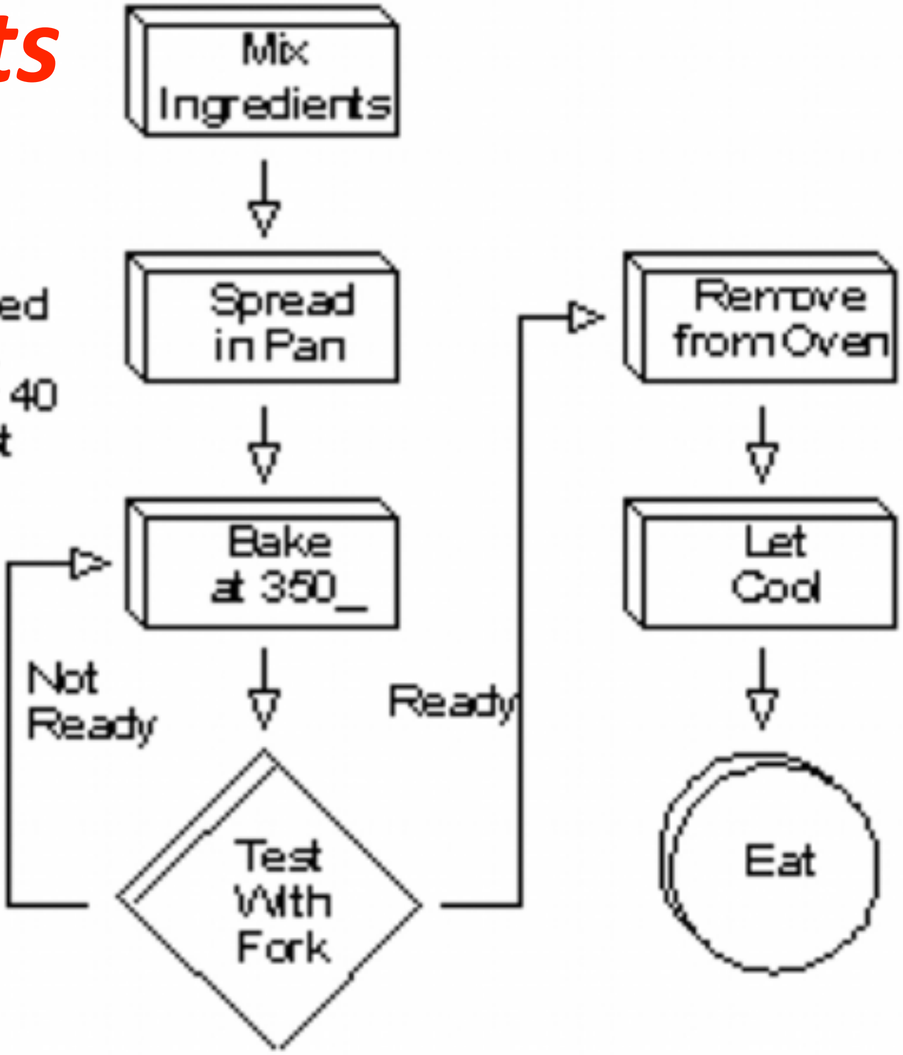
Recipe
CHOCOLATE CAKE

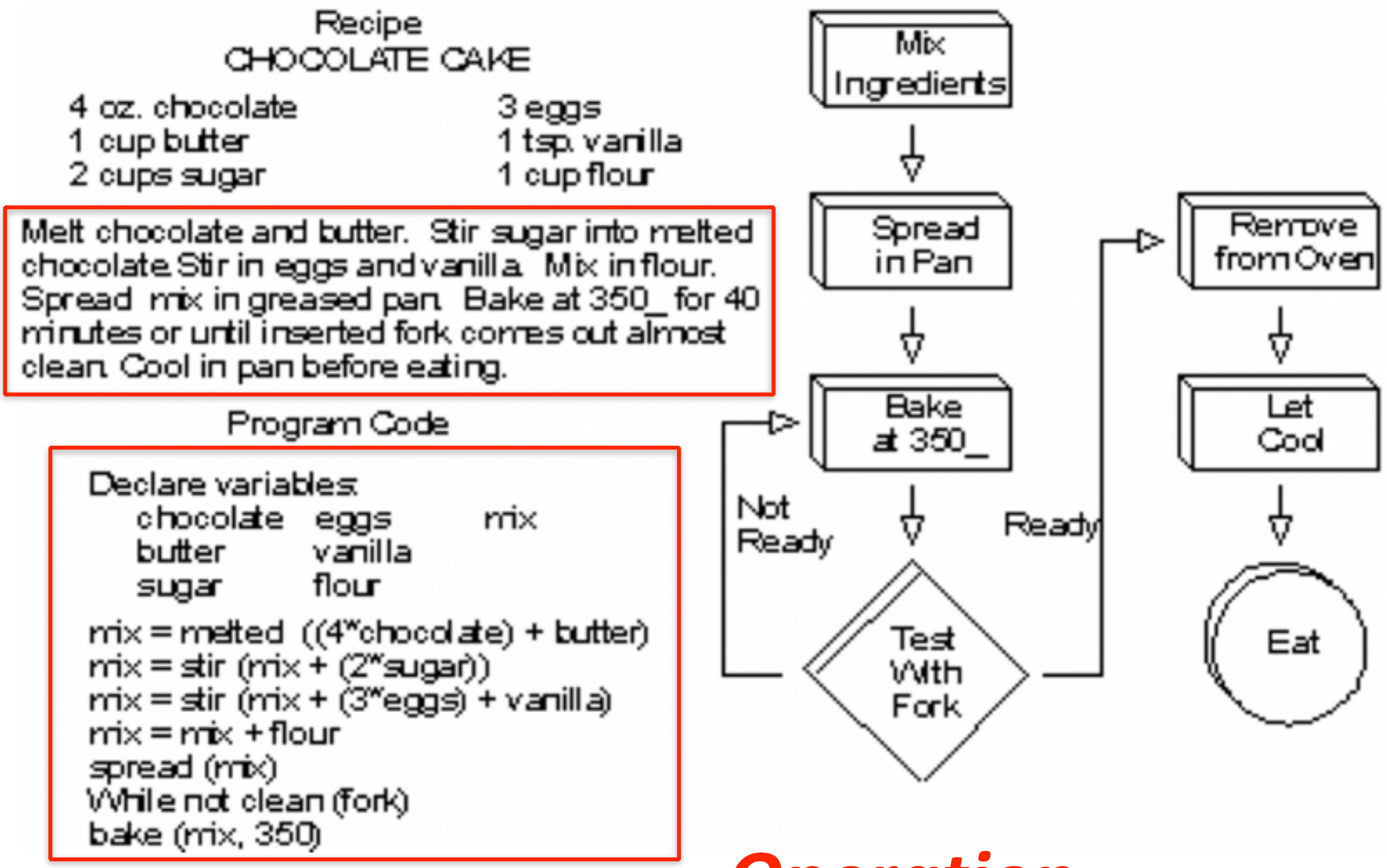
4 oz. chocolate	3 eggs
1 cup butter	1 tsp. vanilla
2 cups sugar	1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

```
Declare variables:
    chocolate   eggs       mix
    butter      vanilla
    sugar       flour
mix = melted ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```





Recipe

CHOCOLATE CAKE

4 oz. chocolate
1 cup butter
2 cups sugar

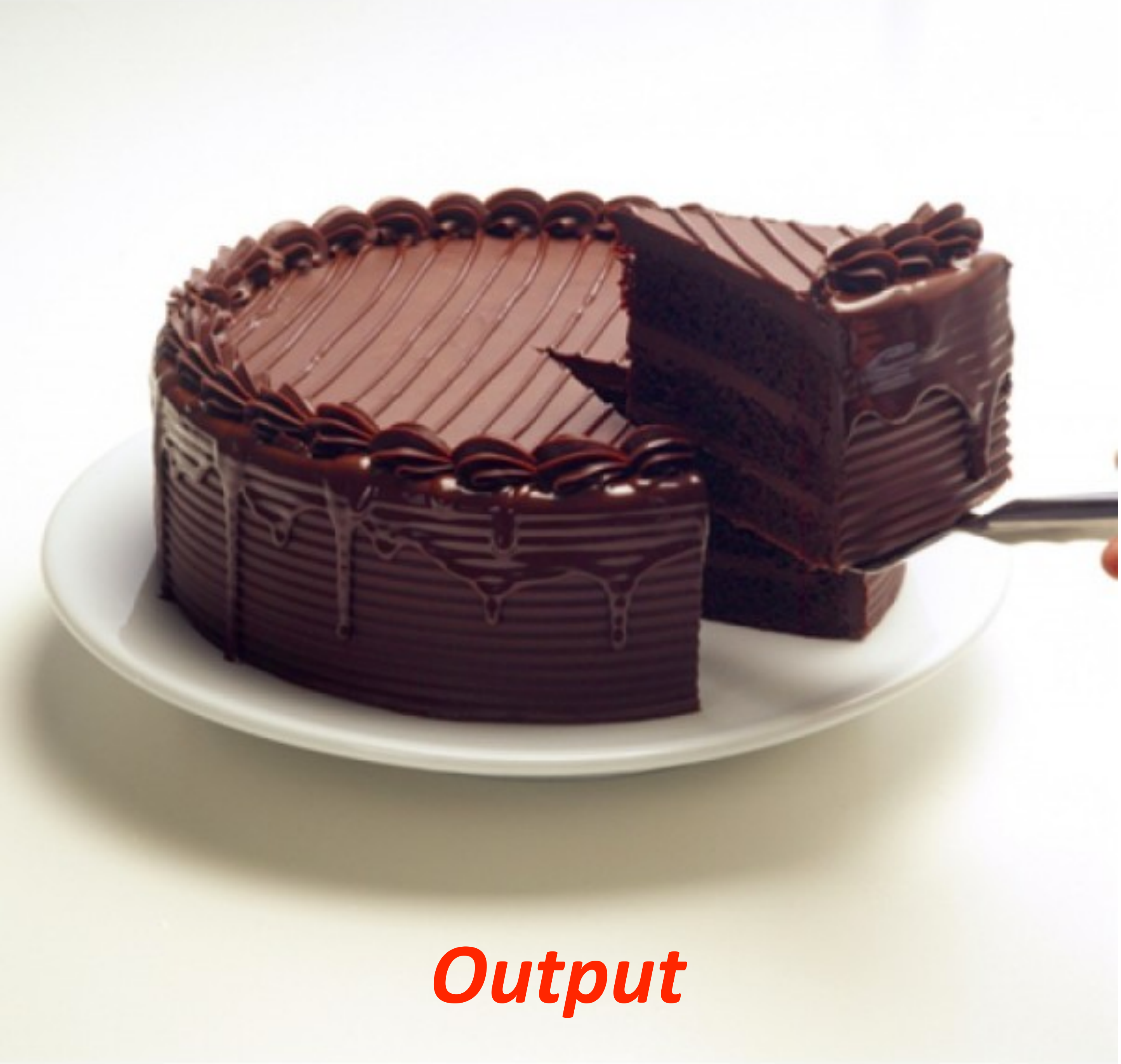
3 eggs
1 tsp. vanilla
1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate. Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

```
Declare variables:
     chocolate    eggs        mix
     butter       vanilla
     sugar        flour
mix = melted ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```

Operation



Output

Another Example

$$\begin{array}{r} 786 \\ +\ 467 \\ \hline \end{array}$$

10 MIN BREAK

ALGORITHMS IN THE NEWSROOM

10 MIN BREAK

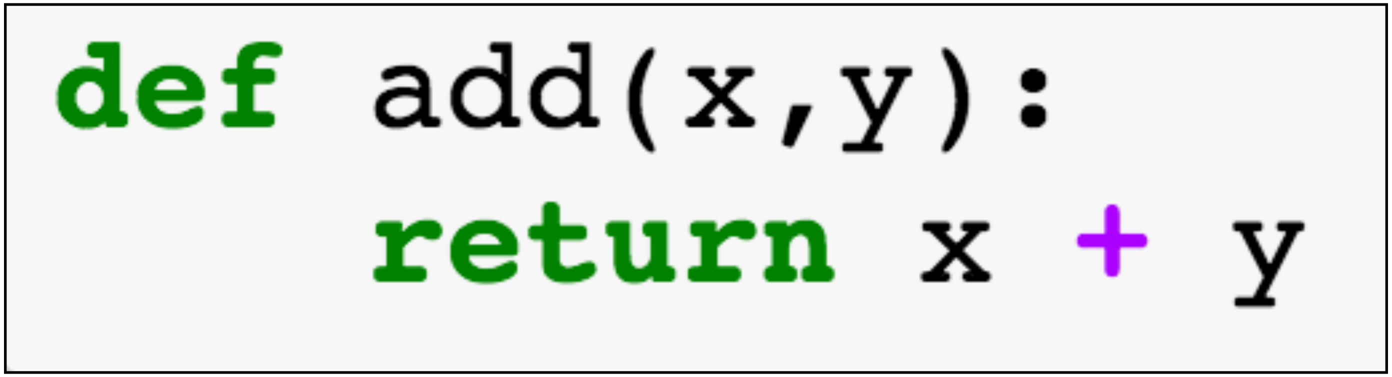
What is a function?

- Block of organized, reusable code
- Ideally perform a single action
- Make your code more modular

When do I write a function?

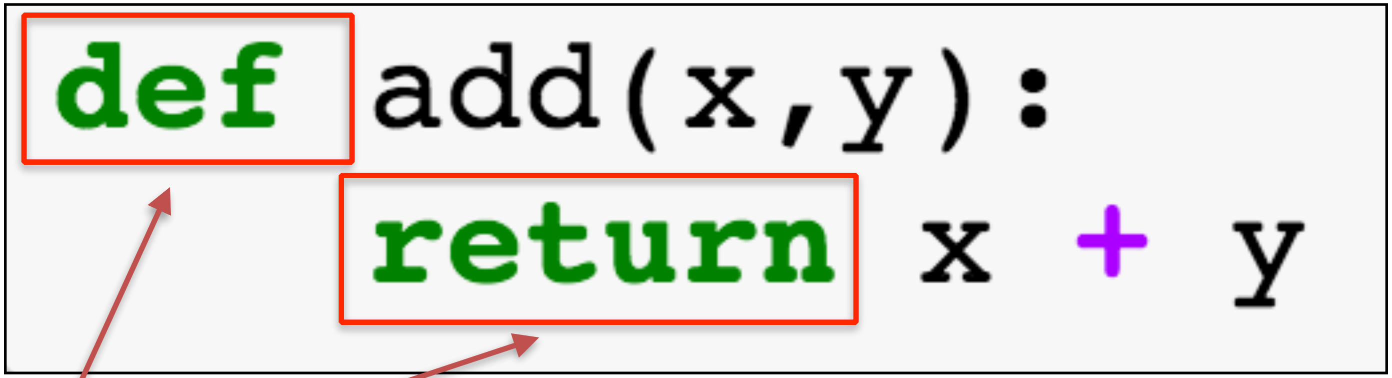
- If you find yourself repeating the same code sequence, it’s time to write a function
- **D**on’t **R**epeat **Y**ourself (DRY)
- Helps make your code more readable and easier to maintain

How do I write a function in Python?



```
def add(x,y):
    return x + y
```

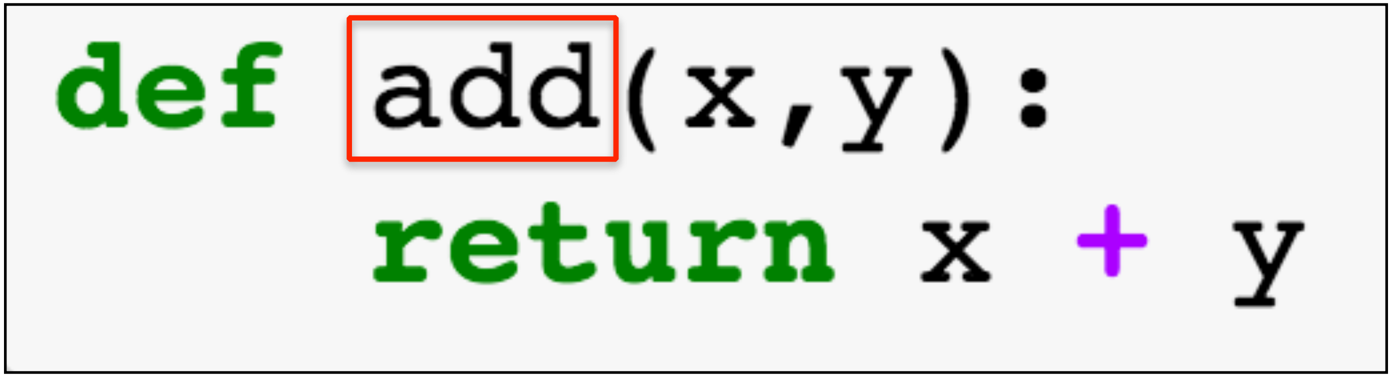
How do I write a function in Python?



```python
def add(x,y):
    return x + y
```

Keywords

How do I write a function in Python?

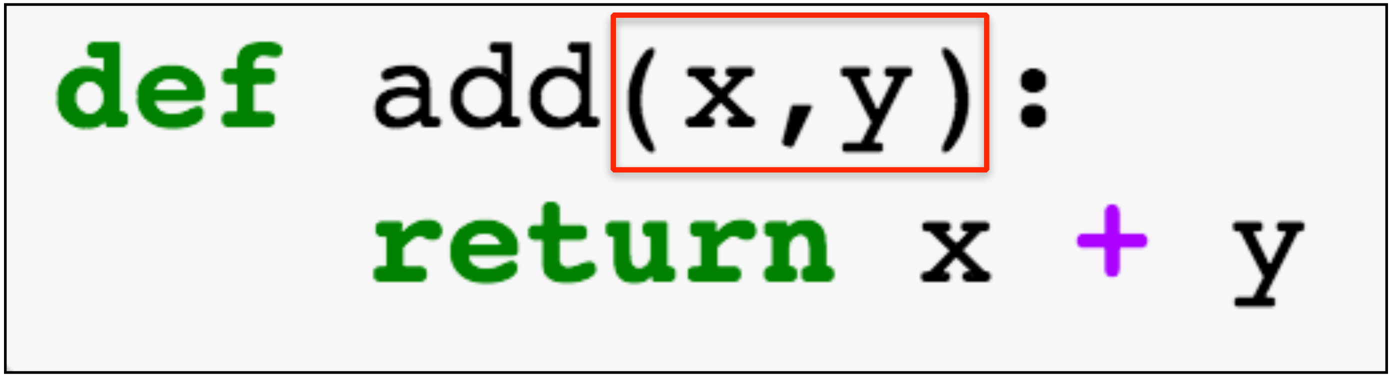


Function name

```
def add(x,y):
    return x + y
```

How do I write a function in Python?

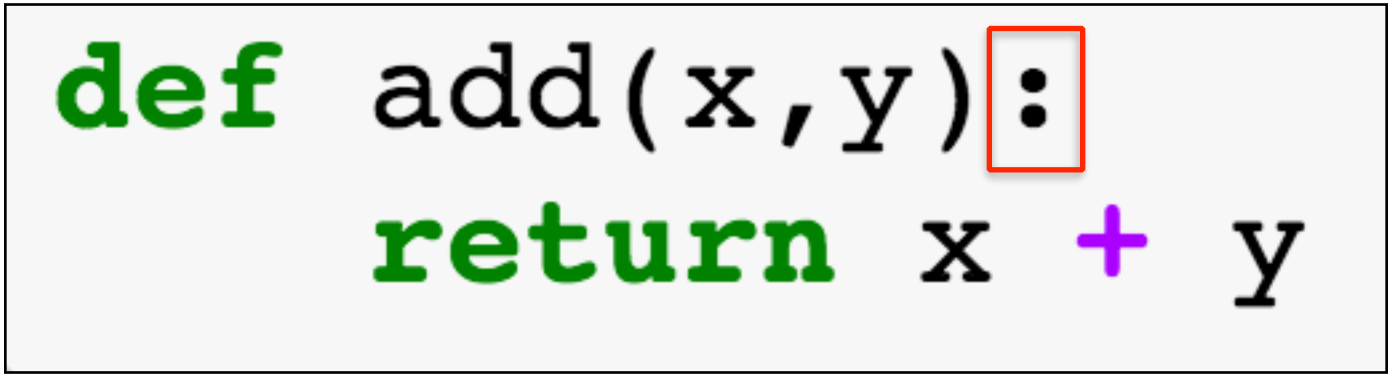
Arguments (function inputs)



```
def add(x,y):
    return x + y
```

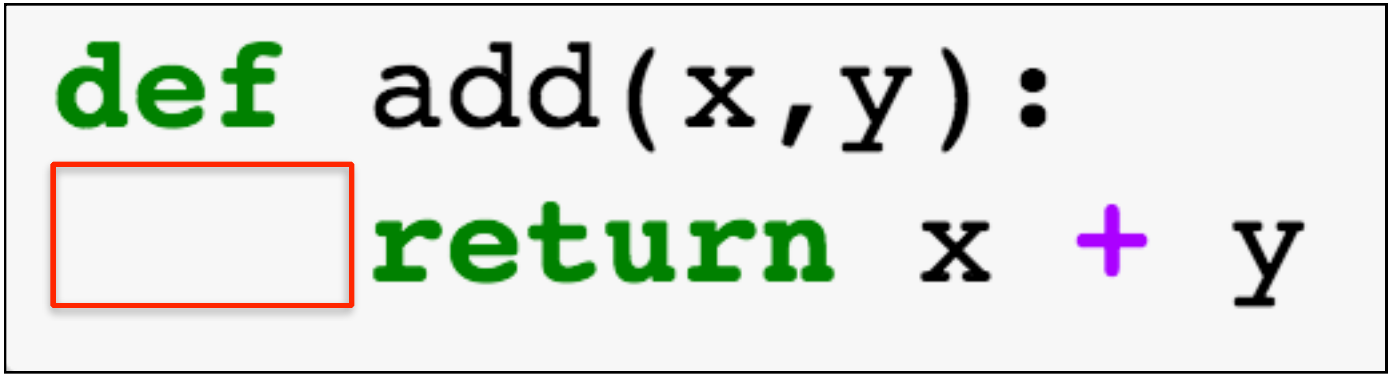
How do I write a function in Python?

Semi-colon



```python
def add(x,y):
    return x + y
```

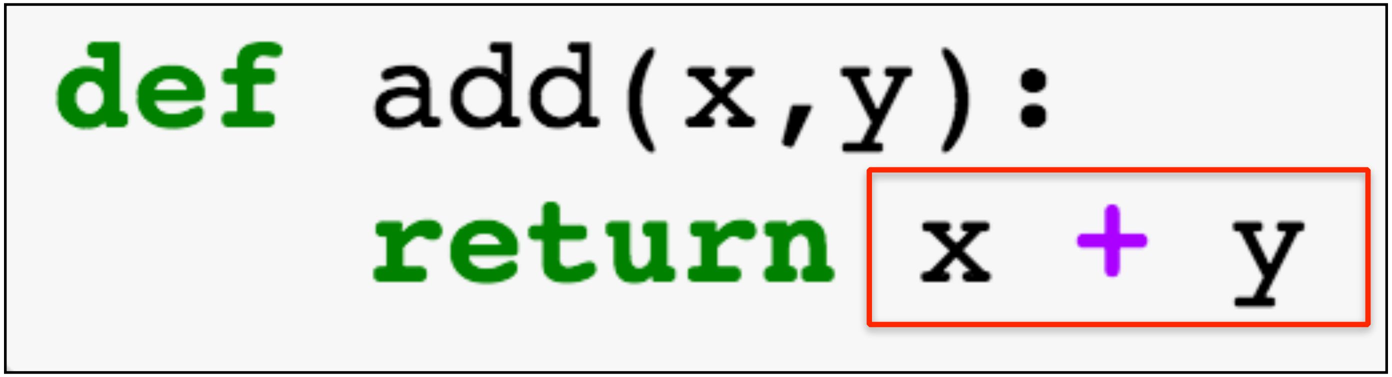
How do I write a function in Python?



```python
def add(x,y):
    return x + y
```

Indentation

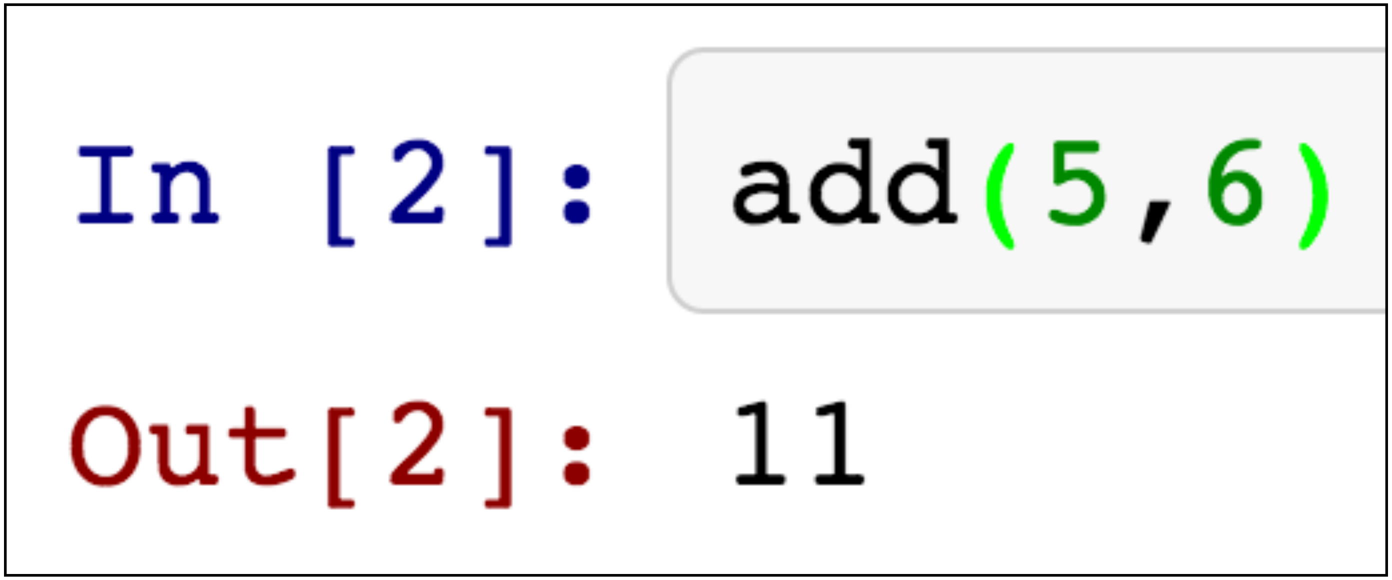
How do I write a function in Python?



```
def add(x,y):
    return x + y
```

Function result

How do I write a function in Python?



```
In [2]: add(5,6)

Out[2]: 11
```

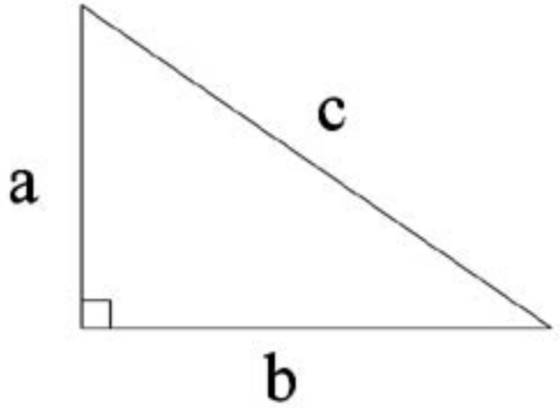
WHY ARE FUNCTIONS IMPORTANT IN AN ALGORITHMS CLASS?

NOW IT’S YOUR TURN

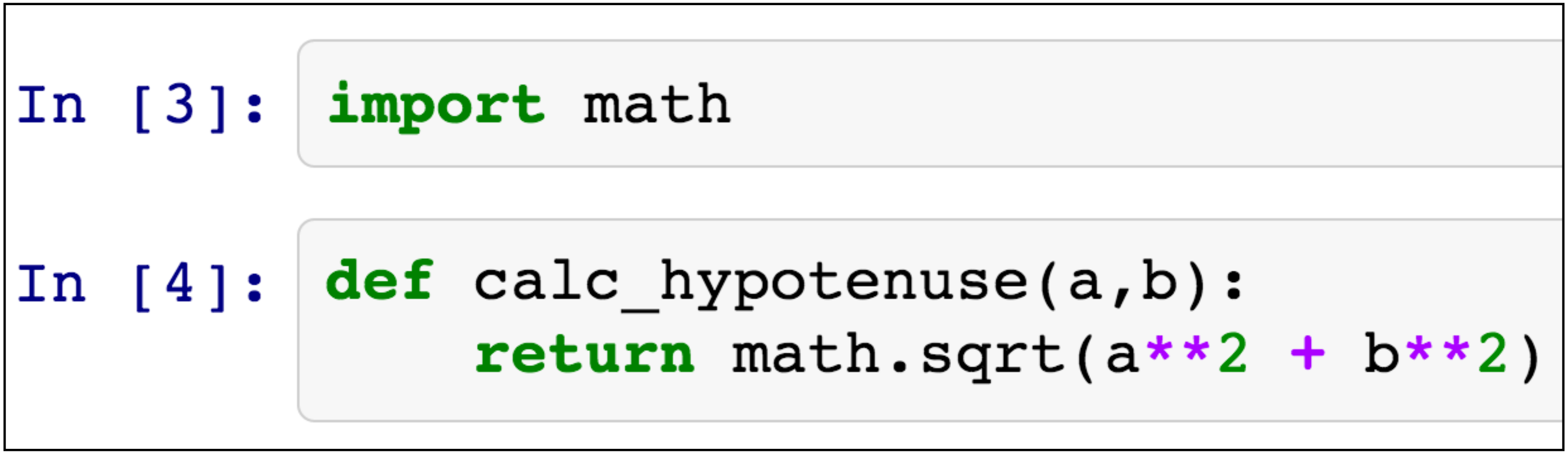
WRITE A FUNCTION TO CALCULATE THE HYPOTENUSE OF A RIGHT TRIANGLE GIVEN THE TWO LEGS

Remember:

$$a^2 + b^2 = c^2$$



Possible Solution



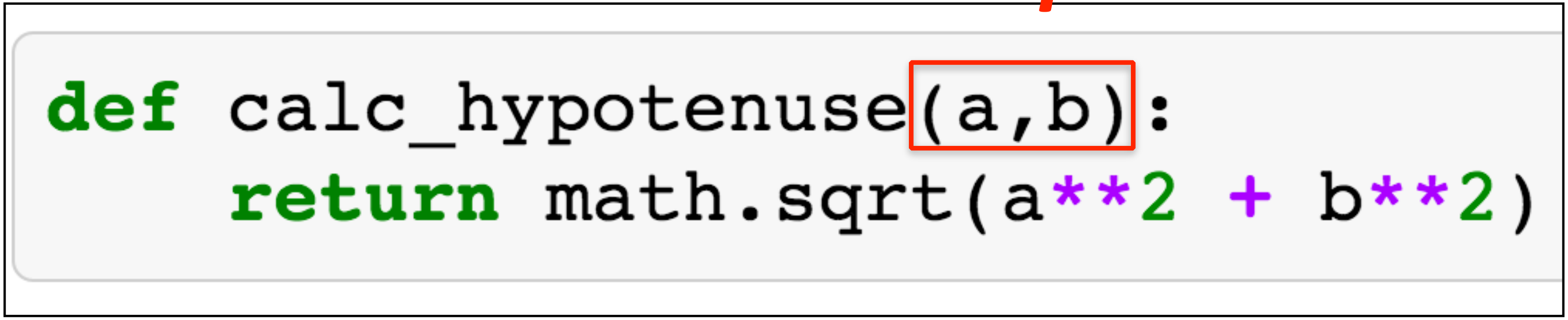
```
In [3]: import math

In [4]: def calc_hypotenuse(a,b):
            return math.sqrt(a**2 + b**2)
```

You’ve built an algorithm!

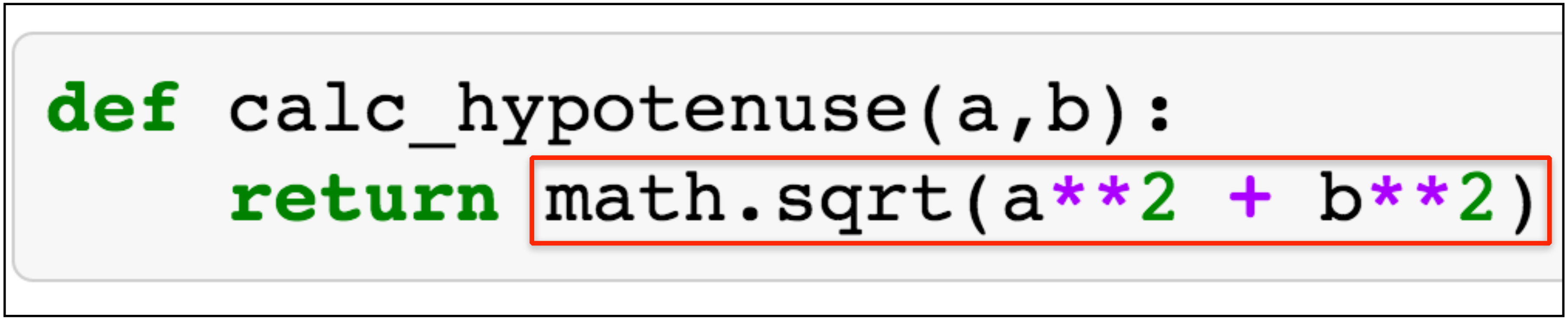
Three Elements of an Algorithm

1. Input



```python
def calc_hypotenuse(a,b):
    return math.sqrt(a**2 + b**2)
```

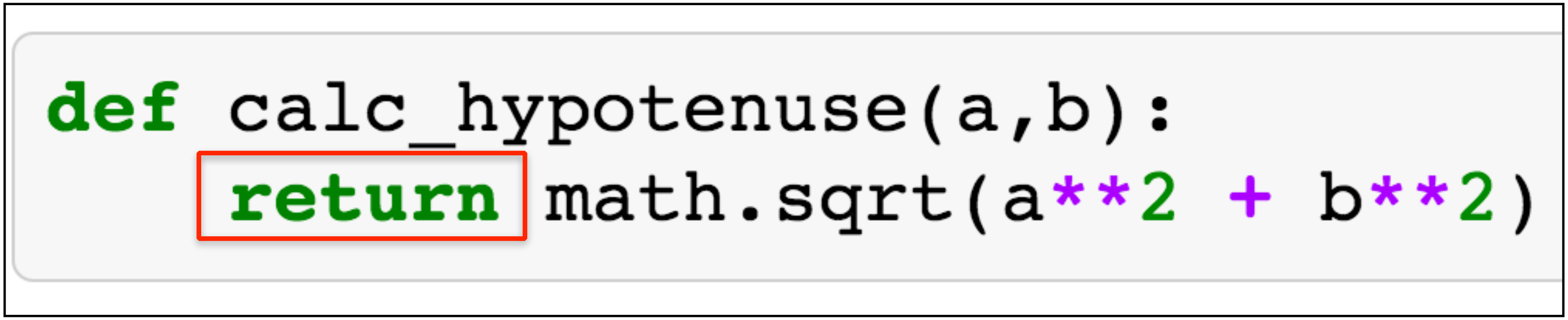
Three Elements of an Algorithm



```
def calc_hypotenuse(a,b):
    return math.sqrt(a**2 + b**2)
```

2. Operation

Three Elements of an Algorithm



```
def calc_hypotenuse(a,b):
    return math.sqrt(a**2 + b**2)
```

3. Output

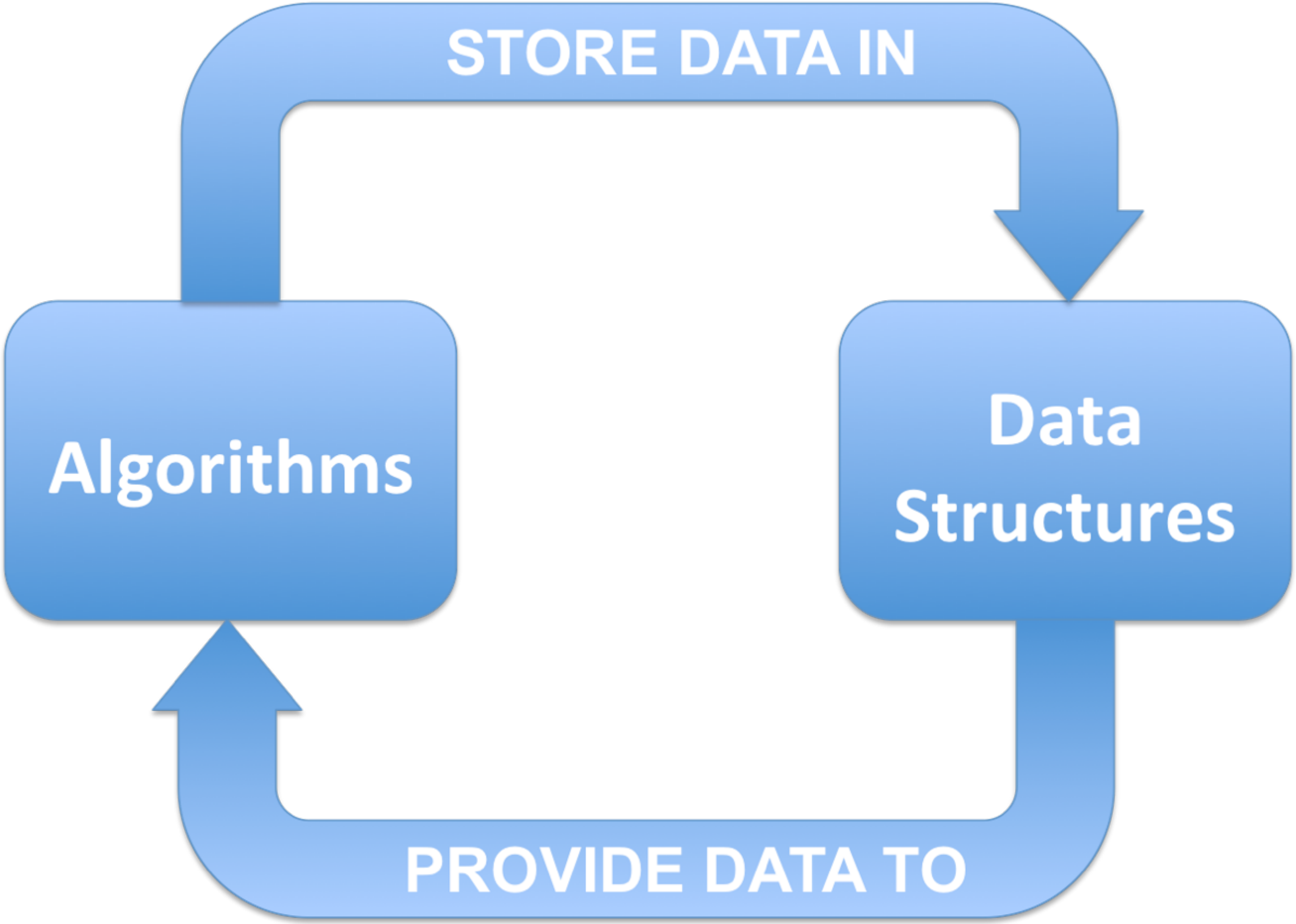
WE’LL EXPLORE THESE MORE LATER IN THE COURSE

DATA STRUCTURES IN PYTHON

What is a Data Structure?

- A way of organizing data in a computer program
- The method of organizing data differs across types, allowing for different applications
- Important to use the right data structure for the proper task

Algorithms and Data Structures



STORE DATA IN

Algorithms

Data Structures

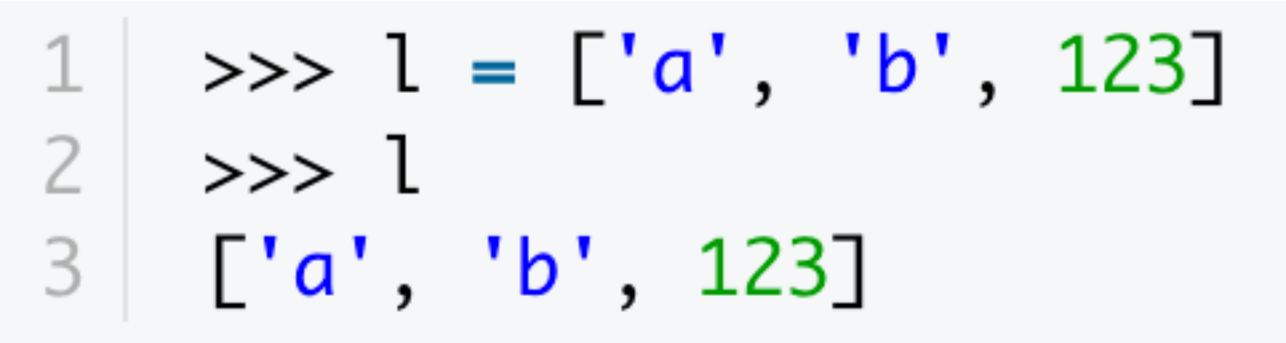
PROVIDE DATA TO

DATA STRUCTURES IN PYTHON

How many can you list?

Lists

- The most versatile data structure in Python
- Can contain multiple data types
- Can be sliced or added to
- Are mutable (can be changed)
- Access individual items by index



```
>>> l = ['a', 'b', 123]
>>> l
['a', 'b', 123]
```

List functions

- append(*object*) - add object to list
- extend(*list*) - extend original list with elements of specified in list
- insert(*index, object*) - insert object at specified index
- pop(*[index]*) - pop element from list at index and return (default to last value in list)
- remove(*value*) - delete first occurrence of value from list (no return)

List functions

- reverse - reverse the order of the list in place
- sort() - sort the list
- count(*value*) - count number of occurrences of value
- index(*value*) - find index of first value and return

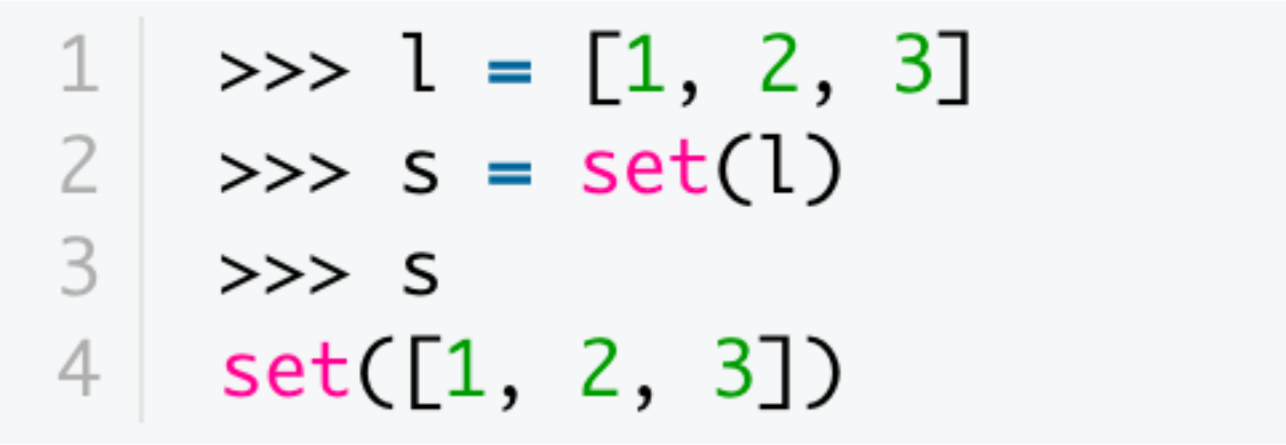
For more information, see documentation
https://docs.python.org/2/library/functions.html#list

When to use lists

- When order matters
- When you can look up the value using a simple numerical index
- When your data might be changed, removed, or extended
- When your data doesn’t need to be unique

Sets

- An unordered collection with no duplicate values
- Created using the keyword “set”
- More limited in the types of objects that can be included (must be hashable, no lists)



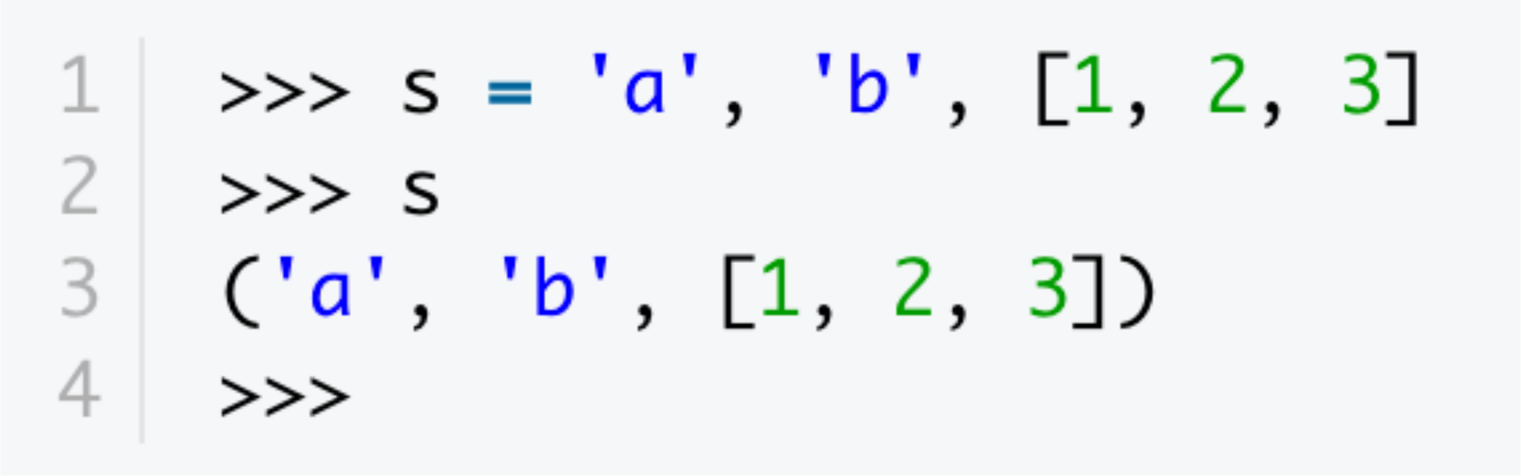
```
>>> l = [1, 2, 3]
>>> s = set(l)
>>> s
set([1, 2, 3])
```

When to use sets

- When you only need unique values
- When the data types you’re working with are relatively basic (hashable)
- When your data changes
- When you need to manipulate your sets mathematically (set supports operations like union, intersection, difference, etc)

Tuples

- Immutable (unchangable) data structure similar to a list
- Can contain elements of different types (don’t need to be hashable)



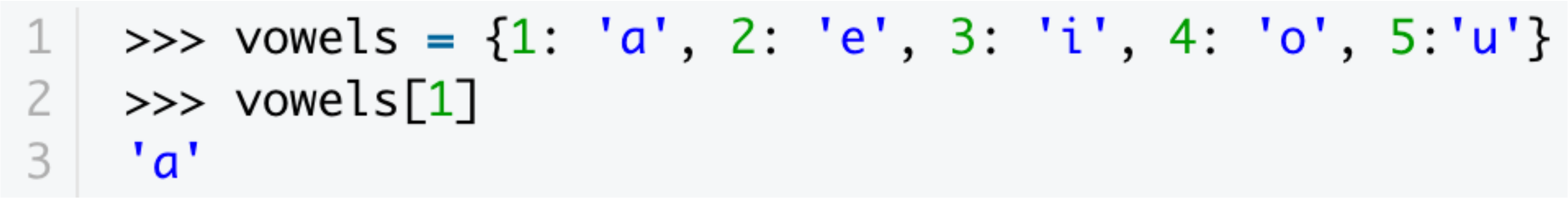
```
>>> s = 'a', 'b', [1, 2, 3]
>>> s
('a', 'b', [1, 2, 3])
>>>
```

When to use tuples

- When your data doesn’t change
- When performance is important (tuples provide better performance because of their immutability)

Dictionaries

- Stores data in key-value pairs
- Provides lookup based on custom keys (instead of numerical indexes)
- Keys must be unique



```
>>> vowels = {1: 'a', 2: 'e', 3: 'i', 4: 'o', 5:'u'}
>>> vowels[1]
'a'
```

When to use a dictionary

- When you need to lookup values by a custom key
- When you need a fast way to lookup values
- When your data needs to be modified

Things to remember with dictionaries

- Key-value pairs aren’t stored in order (use collections.OrderedDict if you need to key order is important)
- collections.defaultdict is a more flexible implementation for creating a dictionary and adding values

FOR MORE INFORMATION AND EXAMPLES

http://code.tutsplus.com/articles/advanced-python-data-structures--net-32748

TROUBLESHOOTING

MORE ON USING PYTHON

Python for Data Analysis Appendix Python Language Essentials:

https://www.dropbox.com/s/vm9wu100coj7dfh/pda_python_essentials.pdf?dl=0

WRAP-UP

Readings

- Miller, Claire Cain, “When Algorithms Discriminate,” New York Times, 9 July 2015, http://nyti.ms/1KS5rdu
- O’Neil, Cathy, “Algorithms And Accountability Of Those Who Deploy Them”, http://bit.ly/1CAxkV7
- Elkus, Adam, “You Can’t Handle the (Algorithmic) Truth”, http://slate.me/1O7s2RU
- Diakopoulos, Nicholas, "Algorithmic Accontability Reporting: On the Investigation of Black Boxes” http://bit.ly/1L304bF

Exercises

1. Write a function that takes in a list of numbers and outputs the mean of the numbers using the formula for mean. Do this without any built-in functions like sum(), len(), and, of course, mean()

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{\sum x_i}{n}$$

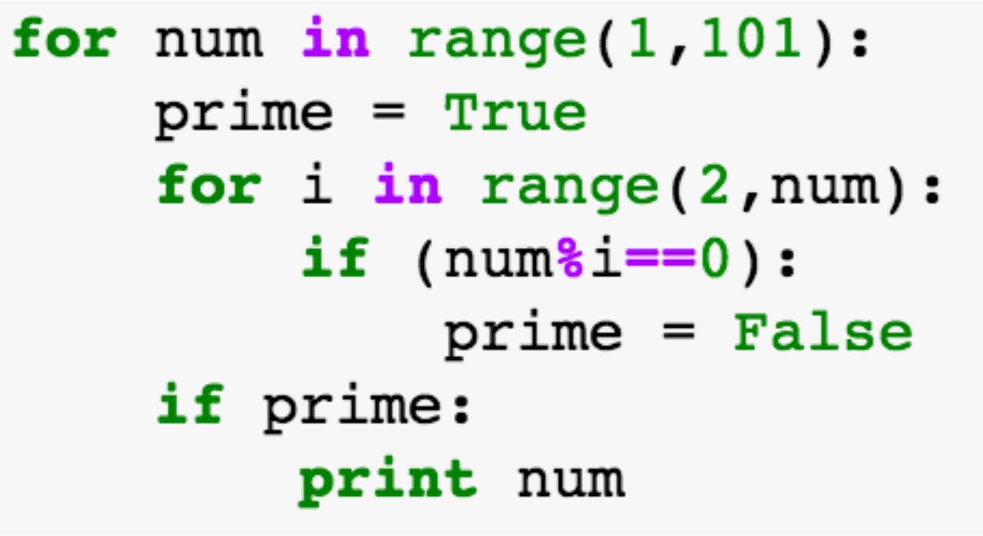
Exercises

2. Create your own version of the Mayoral Excuse Machine (http://dnain.fo/1CCHKml) in Python that takes in a name and location, selects an excuse at random and prints an excuse (“Sorry, Richard, I was late to City Hall to meet you, I had a very rough night and woke up sluggish”).

   Use the “excuses.csv” in the Github repository. Extra credit if you print the link to the story as well.

Exercises

3. Modify the code below (in Exercise3.ipynb) that prints every prime number between 1 and 100 to only print every other prime number. Extra credit if you can modify the code to speed it up.



```
for num in range(1,101):
    prime = True
    for i in range(2,num):
        if (num%i==0):
            prime = False
    if prime:
        print num
```

Exercises

4. The code in Exercise4.ipynb is meant to search for New York Times articles on gay marriage and look at the mean and median word count, but the code has some problems. Follow the instructions in the notebook to fix the code and submit your fixed code.

Exercises

5. Watch this video on how Yelp determines whether to recommend a review:
   https://youtu.be/PniMEnM89iY

   Based on the video, think about the features necessary for the algorithm to determine whether to recommend a review and write a short blogpost on the class Tumblr discussing what features you think Yelp is using and how they might quantifying what they’re trying to measure