Compsci 101, Language, Lamdas, Libraries

Owen Astrachan

Kristin Stephens-Martinez

April 17, 2018

`W` is for …

- **World Wide Web**
  - Where http meets tcp/ip?
- **WiFi**
  - We need and use this every day
- **Windows**
  - From OS to …

PFTLFW

- Review work remaining in Compsci 101
  - Labs, APTs, Assignments, APT/Data Quiz
- Lab for this week and APT/Data quiz
  - What and when
- Python language and libraries
  - From operator.itemgetter to lambda

Work and Due Dates

- APT 6, due on Thursday 4/19
  - What's required? What's optional? Why?

- Input to the Green Dance/Creative assignment?
  - https://www.youtube.com/playlist?list=PLl6wh3hSLvVu3MdE70jV24NwrOaNTls5A
  - http://www.thegreendance.com/

Recommender Assignment

- Transform data into a common format
- Make recommendations using collaborative filtering

- http://bit.ly/101spring18-partners
  - April 18

woto

APT Data Quiz

- Read files, use dictionaries, answer questions
  - Lab is a model for this
  - Depending on when you take it, you may get a different data set and/or different questions

Data Science

Pandas

Data

- Using Python to answer questions about data
  - 4400+ lines of airport data, 2003-2016
  - ***What is the busiest airport?***

Code,Month,Year,Cancelled,Carriers,Delayed,Total,Ontime,City,Name
"TPA","January",2016,146,8,1095,5996,4748,"Tampa, FL"," Tampa International"

```
Code,Month,Year,Cancelled,Carriers,Delayed,Total,Ontime,City,Name
"ATL","June",2003,216,11,5843,30060,23974,"Atlanta, GA"," Hartsfield-Jackson Atlanta International"
"BOS","June",2003,138,14,1623,9639,7875,"Boston, MA"," Logan International"
"BWI","June",2003,29,11,1245,8287,6998,"Baltimore, MD"," Baltimore/Washington International Thurgood Marshall"
"CLT","June",2003,73,11,1562,8670,7021,"Charlotte, NC"," Charlotte Douglas International"
...
"SEA","January",2016,104,10,1274,9739,8330,"Seattle, WA"," Seattle/Tacoma International"
"SFO","January",2016,449,10,3825,13206,8912,"San Francisco, CA"," San Francisco International"
"SLC","January",2016,84,8,1175,8699,7426,"Salt Lake City, UT"," Salt Lake City International"
"TPA","January",2016,146,8,1095,5996,4748,"Tampa, FL"," Tampa International"
```

Defining and Solving the Problem

- Most average number of flights/month
  - Every line of the file is a month of data
  - Track total # flights, and count how many lines
  - Store and sort by total/count

- Header row reminder

Code,Month,Year,Cancelled,Carriers,Delayed,Total,Ontime,City,Name

  - Accumulate: **`total += int(line[6])`**

Tracking Two Quantities

- Similarities here compared to EatDrink APT
  - https://www2.cs.duke.edu/csed/pythonapt/eatdrinkcontest.html
- [total # flights, # months] and [total time, # tasks]
  - Track for each *TLAC* or each contestant
  - Use **`d[name] = [0,0]`** to initialize
  - Use **`d[name][1] += 1`** to update ... What?

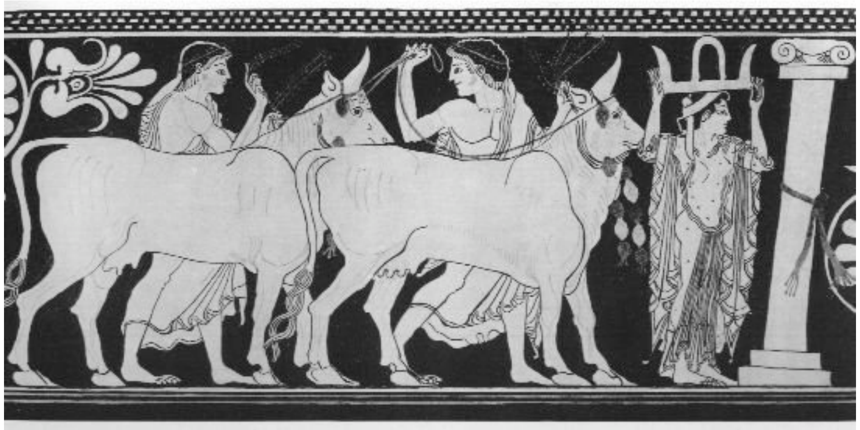
Reading Data

- Converting data from strings to ..., e.g., in APT
  - **`for one in data: x = one.split()`**
  - **`time = x[1].split(":")`**
- Still need to convert strings to ints, e.g., to get seconds in APTs

- JSON data avoids int/string/float conversion
  - Data automagically converted, see Assign 6

Completing this APT with help

- https://www2.cs.duke.edu/csed/pythonapt/eatdrinkcontest.html
- How do we break ties? Sorting first ..



APT WOTO

http://bit.ly/101spring18-april17-1

Eat Drink & Repeat™

Nancy Leveson: Software Safety

- (1995) AIAA Information Systems Award for "developing the field of software safety and for promoting responsible software and system engineering practices where life and property are at stake."
- Therac 25: Radiation Machine
  - https://hackaday.com/2015/10/26/killed-by-a-machine-the-therac-25/

- Software v. Safeware

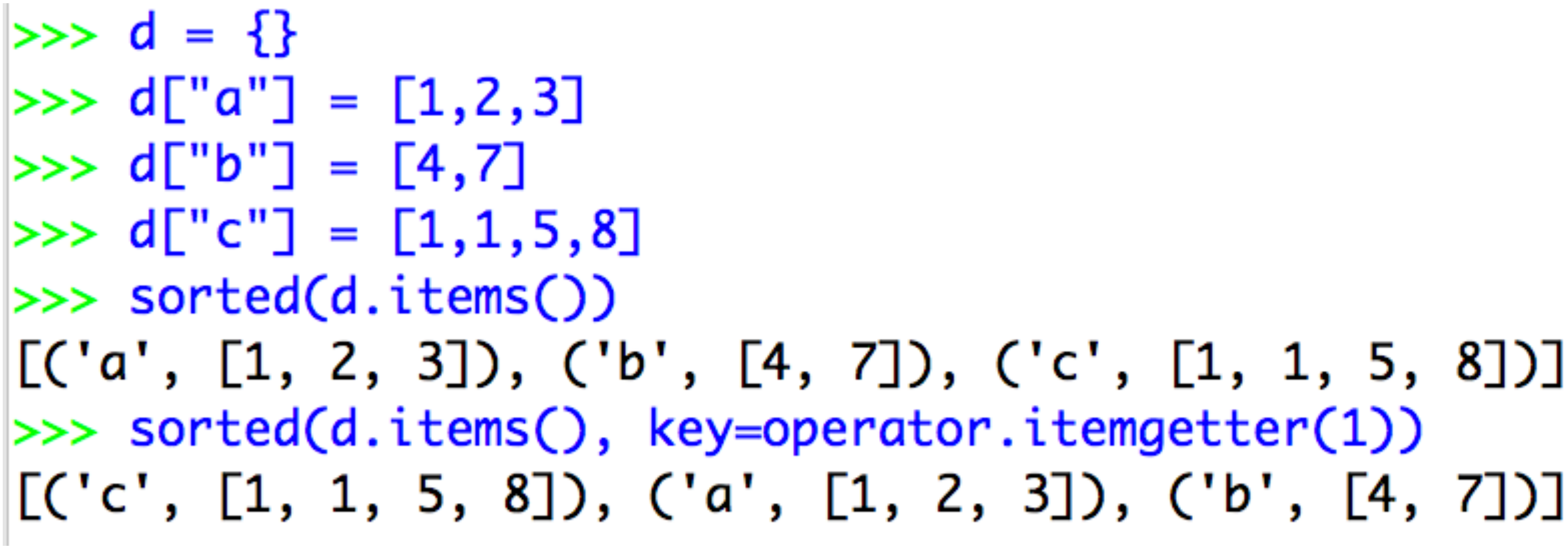
Abstraction

- What does `operator.itemgetter(1)` do?
  - Function that compares elements being sorted
  - Rather than use list/tuple, use function result

`("sam", [1,5,7,9], ...)` sort by list-length?

Simple Sorting

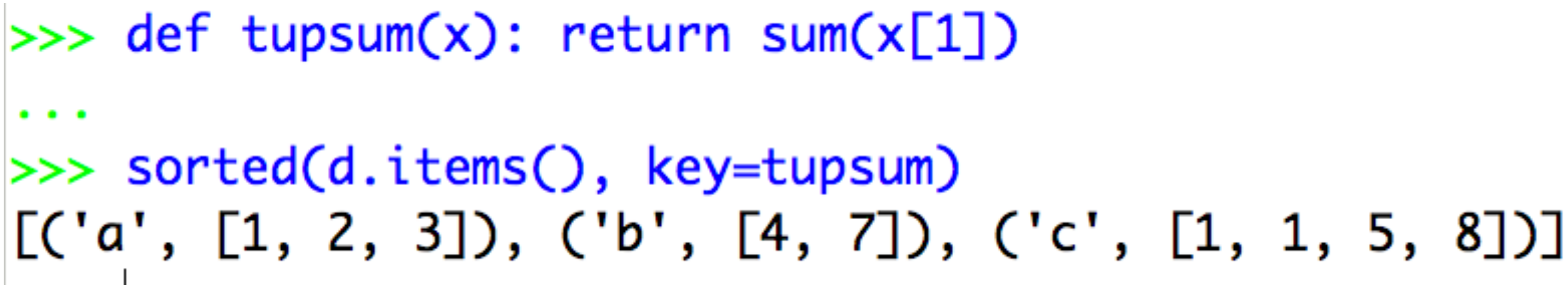
- Explaining what each line here means ...
    - How are tuples or lists sorted?



```
>>> d = {}
>>> d["a"] = [1,2,3]
>>> d["b"] = [4,7]
>>> d["c"] = [1,1,5,8]
>>> sorted(d.items())
[('a', [1, 2, 3]), ('b', [4, 7]), ('c', [1, 1, 5, 8])]
>>> sorted(d.items(), key=operator.itemgetter(1))
[('c', [1, 1, 5, 8]), ('a', [1, 2, 3]), ('b', [4, 7])]
```

Sorting by sum of tup[1]: a list

- How do we apply a function to each tuple?



```
>>> def tupsum(x): return sum(x[1])
...
>>> sorted(d.items(), key=tupsum)
[('a', [1, 2, 3]), ('b', [4, 7]), ('c', [1, 1, 5, 8])]
```

- Why is this the order?
  - Why is **`('c', [1,1,5,8])`** last?
  - What is the parameter passed to **`tupsum`**?

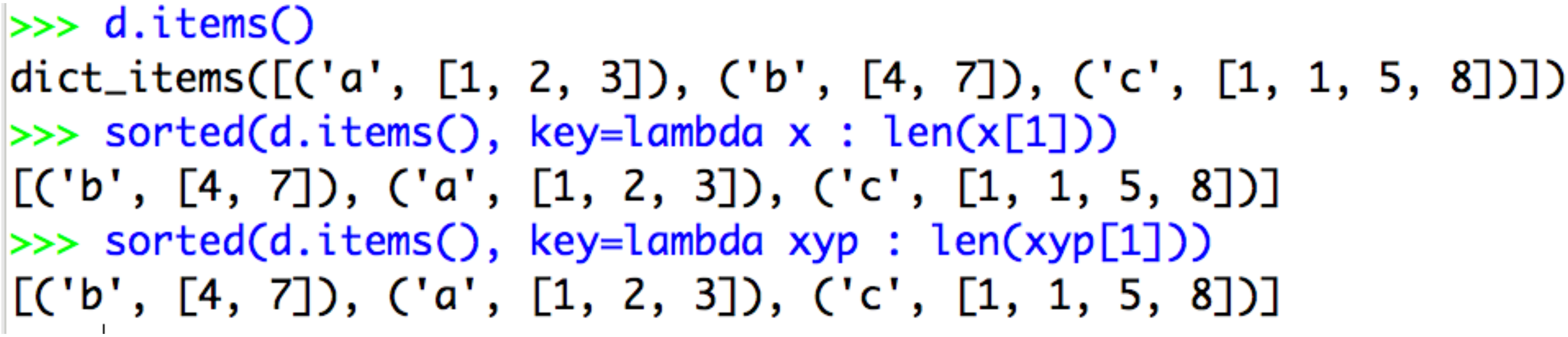
Creating anonymous functions

- Avoid the need to create/label a function just used once, e.g., for sorting
  - How do we define function **`tupsum`**? Where?
  - What if we only need it to sort, called a million times for sorting, but never used again?

- What is lambda?
  - Syntactic sugar for a normal function definition
  - https://docs.python.org/3/tutorial/controlflow.html

Create a function on the fly

- Keyword lambda
  - lambda args : args-expression
  - **`def flen(t) : return len(t[1])`**
  - Use **`key=flen`** to sort, equivalent



```
>>> d.items()
dict_items([('a', [1, 2, 3]), ('b', [4, 7]), ('c', [1, 1, 5, 8])])
>>> sorted(d.items(), key=lambda x : len(x[1]))
[('b', [4, 7]), ('a', [1, 2, 3]), ('c', [1, 1, 5, 8])]
>>> sorted(d.items(), key=lambda xyp : len(xyp[1]))
[('b', [4, 7]), ('a', [1, 2, 3]), ('c', [1, 1, 5, 8])]
```

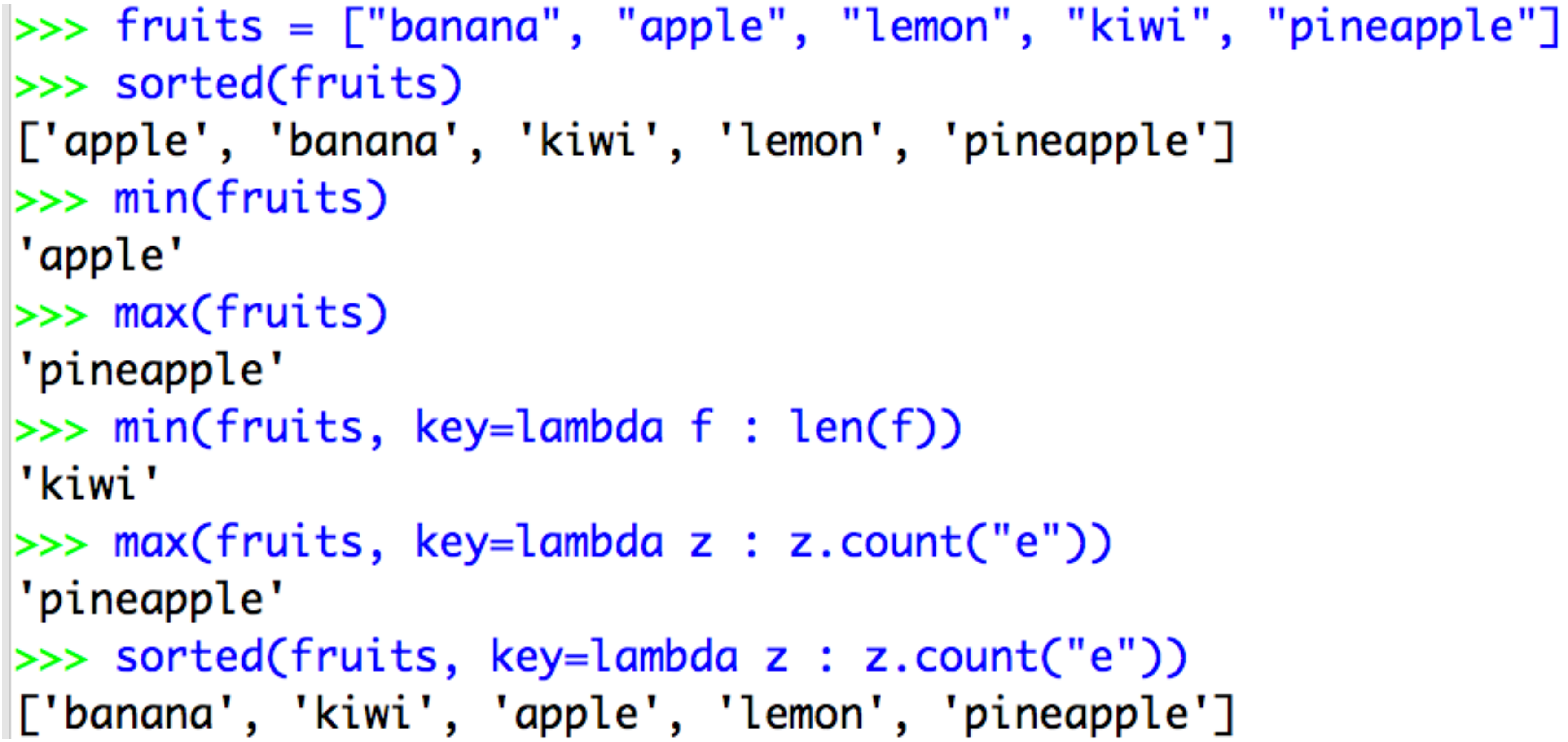
Why is lambda used?

- It doesn't matter at all
    - https://en.wikipedia.org/wiki/Alonzo_Church
    - Lisp and Scheme have lambda expressions
    - Guido, the BDFL, learned to live with lambda

λ

Syntax and Semantics of Lambda

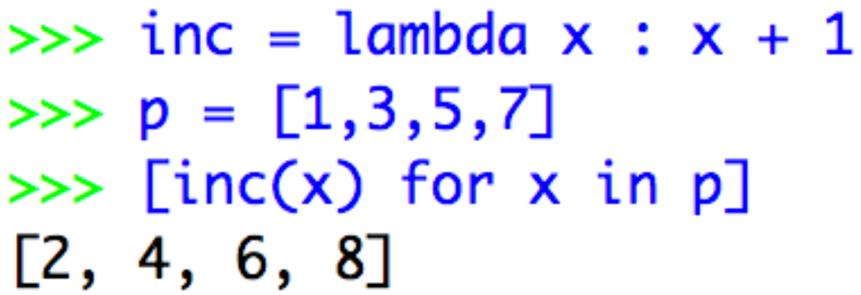
- Major use: single variable function as key



```
>>> fruits = ["banana", "apple", "lemon", "kiwi", "pineapple"]
>>> sorted(fruits)
['apple', 'banana', 'kiwi', 'lemon', 'pineapple']
>>> min(fruits)
'apple'
>>> max(fruits)
'pineapple'
>>> min(fruits, key=lambda f : len(f))
'kiwi'
>>> max(fruits, key=lambda z : z.count("e"))
'pineapple'
>>> sorted(fruits, key=lambda z : z.count("e"))
['banana', 'kiwi', 'apple', 'lemon', 'pineapple']
```

What is a lambda expression?

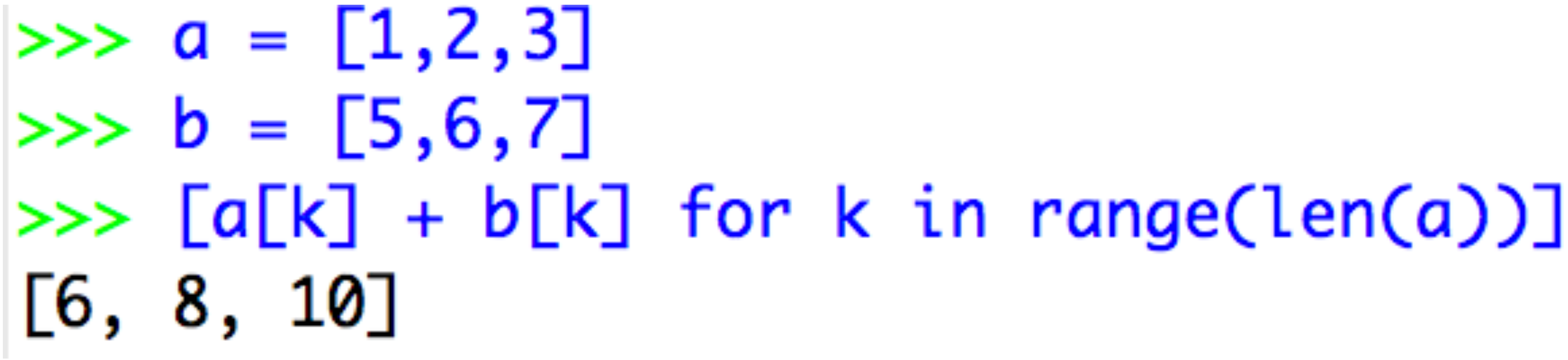
- It's a function object, treat like expression/variable
  - Like list comprehensions, access variables



```
>>> inc = lambda x : x + 1
>>> p = [1,3,5,7]
>>> [inc(x) for x in p]
[2, 4, 6, 8]
```

What about List Comprehensions?

- List comprehensions and lambda expressions have access to enclosing scope: closures
    - Advice: don't do this, create named function



```
>>> a = [1,2,3]
>>> b = [5,6,7]
>>> [a[k] + b[k] for k in range(len(a))]
[6, 8, 10]
```

WOTO

http://bit.ly/101spring18-april17-2

Questions