

CompSci 101

Stable Sorting, Lambda

```
f = lambda x : x[1]  
sorted(lst, key=f)
```

Yesenia Velasco
Susan Rodger
March 29, 2022

3/29/22

CompSci 101, Spring 2022

1

S is for ...



- **Software**
 - Joy, sorrow, fun, changing the world
- **System and sys**
 - Connecting to the machine at different levels
- **Sorting**
 - From hat to tim to more

3/29/22

CompSci 101, Spring 2022

2

Shaundra Daily

- Professor of the Practice, Duke University
- B.S. Florida State, Electrical Eng
- PhD Media Arts/Sciences – MIT
- Combines Dance with Robotics
- Focuses on technologies, programs and curricula to support Diversity, Equity and Inclusion in STEM Fields



3/29/22

CompSci 101, Spring 2022

3

Announcements

- Assignment 5 due Today!
- APT-6 due Thurs. March 31
- APT-7 out Thursday, Due April 7
- Lab 10 Friday
 - There is a prelab, it is out!
- Coming up...
 - APT Quiz 2 – April 7-10
 - Exam 4 – April 12

3/29/22

CompSci 101, Spring 2022

4

PFTD

- Sorting in Python and sorting in general
 - How to use .sort and sorted, differences
 - Key function – change how sorting works
 - Lambda – create anonymous functions
- Stable sorting
 - How to leverage when solving problems
 - Why Timsort is the sort-of-choice (! quicksort)

3/29/22

CompSci 101, Spring 2022

5

Review Sort: Items Same Length

- Use key=function argument and reverse=True
 - What if we want to write our own function?

```
a = ['red', 'orange', 'green', 'blue', 'indigo', 'violet']
print(sorted(a))
```

```
print(sorted(a, key=len))
```

```
print(sorted(a, key=len, reverse=True))
```

3/29/22

CompSci 101, Spring 2022

6

Sorting Examples

```
a = [4, 1, 7, 3]
b = sorted(a)
a.sort()
a = ['Q', 'W', 'B', 'F']
b = sorted(a)
c = sorted(a, reverse = True)
a = ['hello', 'blue', 'car']
b = sorted(b, key=len)
```

3/29/22

CompSci 101, Spring 2022

8

More Sorting Examples

```
a = [ [2, 2, 34], [2, 6, 7, -1], [1, 2, 3] ]
b = sorted(a)
c = sorted(a, key = len)
d = sorted(a, key=max)
e = sorted(a, key=min)
```

3/29/22

CompSci 101, Spring 2022

10

WOTO-1 Basic Sorting

<http://bit.ly/101s22-0329-1>

3/29/22

CompSci 101, Spring 2022 12

Anonymous Functions

- Useful when want “throw-away” function
 - Our case mainly sort
- Syntax: **lambda PARAMETERS: EXPRESSION**
 - PARAMETERS – 0 or more comma separated
 - EXPRESSION – evaluates to something

3/29/22

CompSci 101, Spring 2022 20

The power of lambda

- We want to create a function “on-the-fly”
 - aka anonymous function
 - aka “throw-away” function
- ```
In[7]: a
Out[7]: ['red', 'orange', 'green', 'blue', 'indigo', 'violet']
In[8]: sorted(a,key=lambda x : x.count("e"))
Out[8]: ['indigo', 'red', 'orange', 'blue', 'violet', 'green']
```
- Why ‘indigo’ first and ‘green’ last?
    - What about order of ties? Later today! Stable

3/29/22

CompSci 101, Spring 2022 19

## Why is lambda used?

- It doesn’t matter at all could use zeta? iota? ...
  - [https://en.wikipedia.org/wiki/Alonzo\\_Church](https://en.wikipedia.org/wiki/Alonzo_Church)
- Lisp and Scheme have lambda expressions
- Guido van Rossum, learned to live with lambda



3/29/22

CompSci 101, Spring 2022 21

# 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(num) for num in p]
[2, 4, 6, 8]
```

3/29/22

CompSci 101, Spring 2022 22

## Syntax and Semantics of Lambda

- Major use: single variable function as key

```
fruits = ['banana', 'apple', 'lemon', 'kiwi', 'pineapple']
b = sorted(fruits)
```

```
c = min(fruits)
```

```
d = max(fruits)
```

Syntactic sugar  
(makes the medicine go down)

- Syntactic sugar for a normal function definition

```
def f(x):
 return x[1]
sorted(lst, key=f)
```

```
f = lambda x : x[1]
sorted(lst, key=f)
```

```
sorted(lst, key=lambda x : x[1])
```

```
>>> 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 sparky : len(sparky[1]))
[('b', [4, 7]), ('a', [1, 2, 3]), ('c', [1, 1, 5, 8])]
```

Parameter  
name does  
not matter

3/29/22

CompSci 101, Spring 2022 23

## Syntax and Semantics of Lambda (2)

```
fruits = ['banana', 'apple', 'lemon', 'kiwi', 'pineapple']
```

```
e = min(fruits, key=lambda f: len(f))
```

```
g = max(fruits, key=lambda z: z.count('e'))
```

```
h = sorted(fruits, key=lambda z: z.count('e'))
```

3/29/22

CompSci 101, Spring 2022 24

3/29/22

CompSci 101, Spring 2022 26

## Review: CSV and Sort for top artists

- Using two sorts to get top artists

```
31 print('\nTop 5 artists: ')
32 sortbycount = sorted([(a[1], a[0]) for a in counts.items()])
33 sortedArtists = [(a[1], a[0]) for a in sortbycount]
34 for artist in sortedArtists[-5:]:
35 print(artist)
```

- Reverse tuples to sort
- Reverse tuples to print

```
Top 5 artists:
('John, Elton', 21)
('Who', 24)
('Rolling Stones', 36)
('Led Zeppelin', 38)
('Beatles', 51)
```

3/29/22

Compsci 101, Spring 2022 28

## Top 5 Artists

- Instead of intermediary list, use **lambda**
- Instead of **[-5:]**, use **reverse=True**

```
31 print('\nTop 5 artists: ')
32 sortbycount = sorted([(a[1], a[0]) for a in counts.items()])
33 sortedArtists = [(a[1], a[0]) for a in sortbycount]
34 for artist in sortedArtists[-5:]:
35 print(artist)

37 print("repeat it")
38 sortedArtists = sorted(counts.items(), key=lambda item: item[1], reverse=True)
39 for tup in sortedArtists[:5]:
40 print(tup)
```

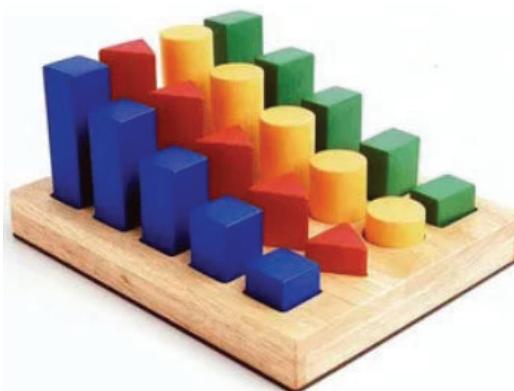
Output slightly different. Why?

```
repeat it
('Beatles', 51)
('Led Zeppelin', 38)
('Rolling Stones', 36)
('Who', 24)
('Eagles', 21)
```

3/29/22

Compsci 101, Spring 2022 29

## WOTO-2 Sorting <http://bit.ly/101s22-0329-2>



3/29/22

Compsci 101, Spring 2022 30

## How is the sorting happening?

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

>>> sorted(d.items(), key=lambda x: x[1])

>>> sorted(d.items(), key=lambda x: x[1][-1])
```

3/29/22

Compsci 101, Spring 2022 31

# How to do some “fancy” sorting

- lambda PARAMETER : EXPRESSION
- Given data: list of tuples: (first name, last name, age)  
[('Percival', 'Avram', 51),  
 ('Melete', 'Sandip', 24), ...]
- Think: What is the lambda key to sort the following?  
`sorted(data, key=lambda z : (z[0], z[1], z[2]))`
  - Sort by last name, break ties with first name
  - Sort by last name, break ties with age
  - Alphabetical by last name, then first name, then reverse age order

3/29/22

CompSci 101, Spring 2022 35

# Creating Tuples with lambda

- Sort by last name, break ties with first name
- Sort by last name, break ties with age
- Alphabetical by last name, then first name, then reverse age order

3/29/22

CompSci 101, Spring 2022 36

# Leveraging the Algorithm

- Can't sort by creating a tuple with lambda, use:
  - Pattern: Multiple-pass *stable* sort – first sort with last tie breaker, then next to last tie breaker, etc. until at main criteria
- Sort by index 0, break tie in reverse order with index 1  
[('b', 'z'), ('c', 'x'), ('b', 'x'), ('a', 'z')]
- *Stable* sort respects original order of "equal" keys

3/29/22

CompSci 101, Spring 2022 38

# Stable sorting: respect "equal" items

- Women before men, each group height-sorted
  - First sort by height



3/29/22

CompSci 101, Spring 2022 41

## Understanding Multiple-Pass Sorting

```
> data
[('f', 2, 0), ('e', 1, 4), ('a', 2, 0),
 ('c', 2, 5), ('b', 3, 0), ('d', 2, 4)]
> a0 = sorted(data, key = lambda x: x[0])
> a0

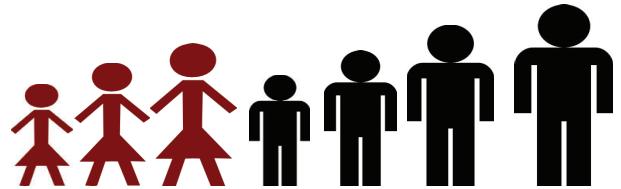
> a1 = sorted(a0, key = lambda x: x[2])
> a1

> a2 = sorted(a1, key = lambda x: x[1])
> a2
```

3/29/22

CompSci 101, Spring 2022 43

## WOTO-3 Multipass Sorting <http://bit.ly/101s22-0329-3>



3/29/22

CompSci 101, Spring 2022 47