SI 507 Lab #2

September 7/8

Schedule

- Go here again: bit.ly/3AGZz5g
    - Download codebook
- Logistics
- Material
    - functions as arguments
    - exercises
- Work on HW

Logistics

- Homework #1 is due on 9/12
- Lab
    - Recordings will be available on Canvas
    - Slides (with solutions) available in my resources doc after lab
- Surveys
    - Constant feedback survey
    - Finish get-to-know-you-survey!

Big Picture

how does lab material apply to homework, final project, and beyond???

- Functions
    - user-defined
        - scaffolding
        - commenting
    - built-in
        - `input()`
        - `print()`
    - as arguments
        - `sorted( lambda x: … )`
        - `filter( lambda x: … )`
        - `apply( lambda x: … )`
- Applications
    - HW1 (Tic Tac Toe)
    - Final project

Exercise: Walkthrough example

- You are analyzing Tweets from Twitter users
- Q: Which Tweets mention Donald Trump's Twitter account ("`@realDonaldTrump`") ? Output a list that contains **only** these Tweets
- Format: `[ (tweet_ID, tweet_source_user, text) … ]`

```
[
  (1, "this_is_alex", "I wonder what @realDonaldTrump thinks about this"),
  (151, "motivational_quotes", "inspirational quote for the day: love trumps hate"),
  (41, "i_heart_president_trump", "Thank you @realDonaldTrump for your amazing service
to this country!!!") ,
   …
]
```

Functions as arguments

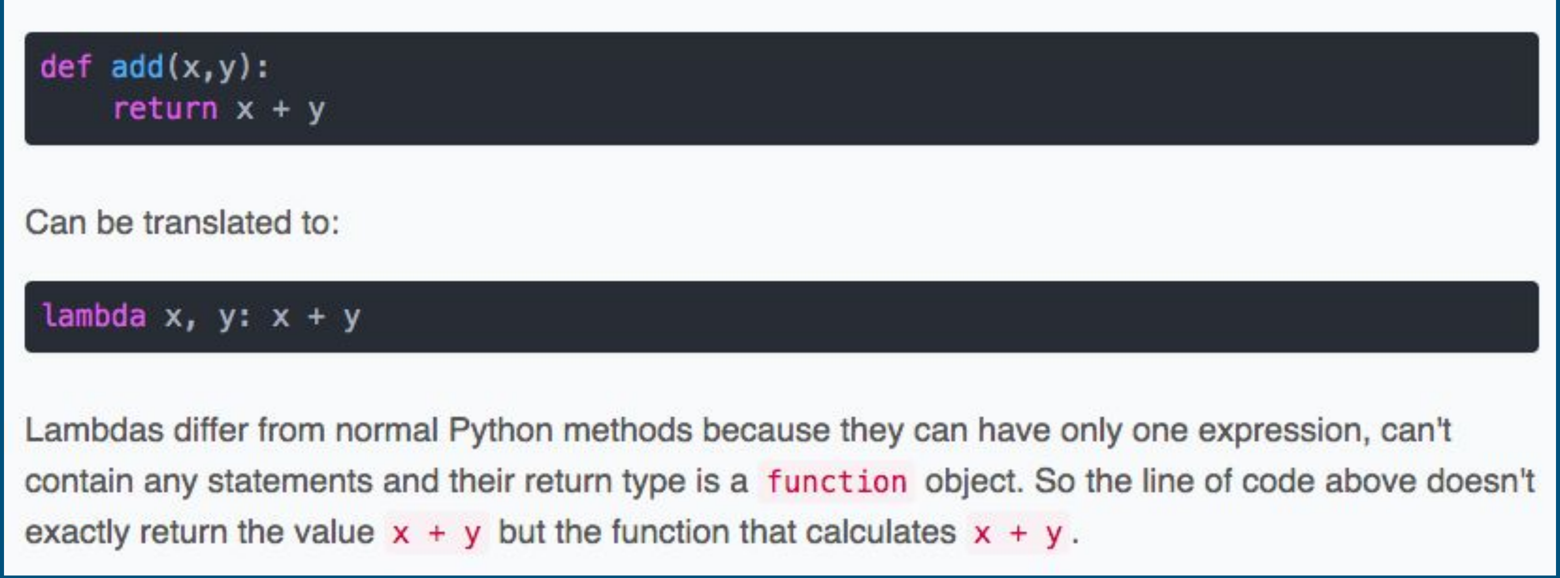
- Some functions take **functions as arguments**
  - `sort()`
  - **`filter()`**
  - `map()`
  - `reduce()`

Exercise: Walkthrough example

- You are analyzing Tweets from Twitter users
- Q: Which Tweets mention Donald Trump's Twitter account ("`@realDonaldTrump`") ?
- Format: `[ (tweet_ID, tweet_source_user, text) … ]`
- **Use `filter()`**

```
[
  (1, "this_is_alex", "I wonder what @realDonaldTrump thinks about this"),
  (151, "motivational_quotes", "inspirational quote for the day: love trumps hate"),
  (41, "i_heart_president_trump", "Thank you @realDonaldTrump for your amazing service
to this country!!!") ,
   …
]
```

Lambda Functions (syntax)

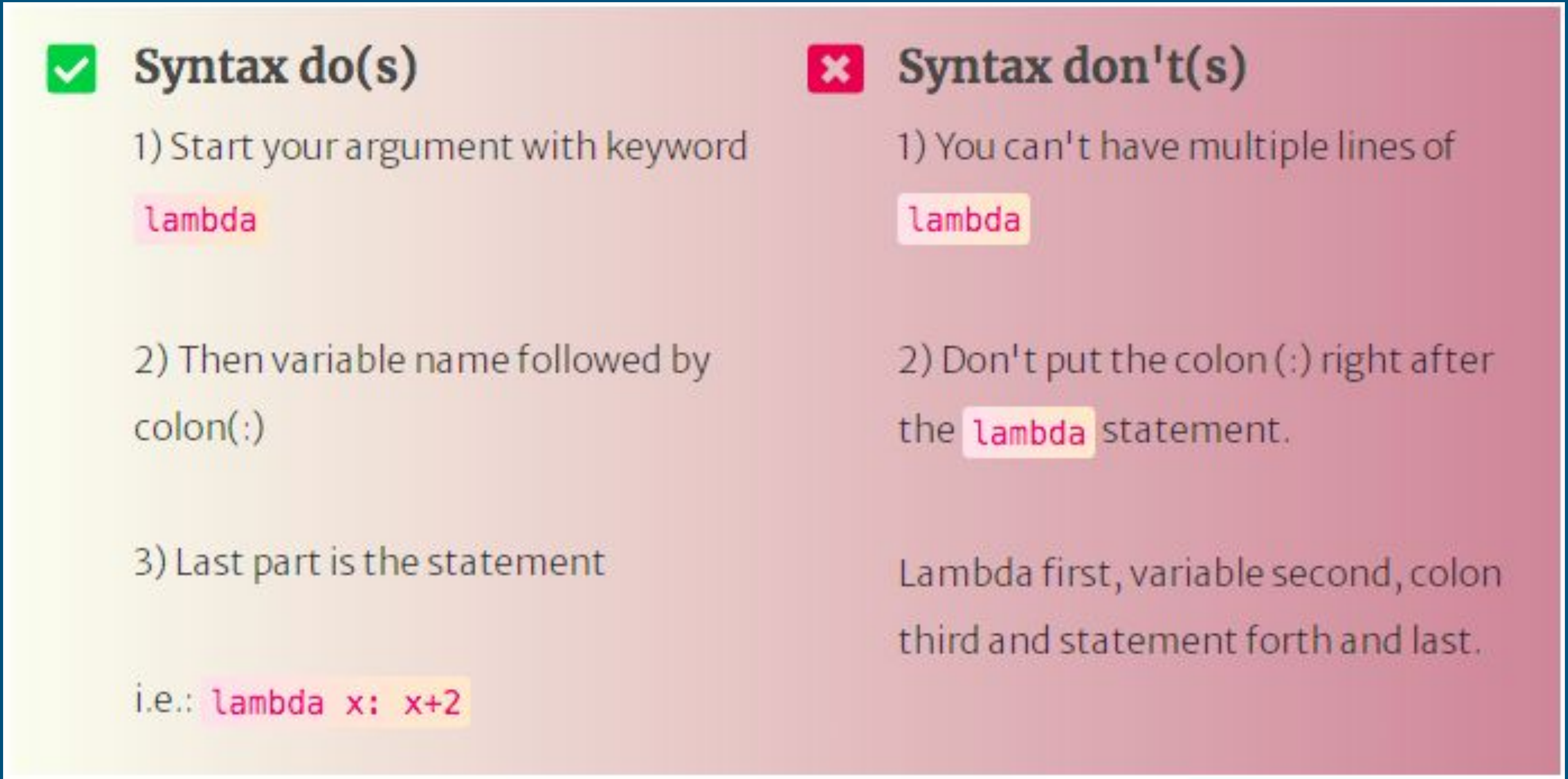


```
def add(x,y):
    return x + y
```

Can be translated to:

```
lambda x, y: x + y
```

Lambdas differ from normal Python methods because they can have only one expression, can't contain any statements and their return type is a `function` object. So the line of code above doesn't exactly return the value `x + y` but the function that calculates `x + y`.



✅ Syntax do(s)

1) Start your argument with keyword `lambda`

2) Then variable name followed by colon(:)

3) Last part is the statement

i.e.: `lambda x: x+2`

❎ Syntax don't(s)

1) You can't have multiple lines of `lambda`

2) Don't put the colon (:) right after the `lambda` statement.

Lambda first, variable second, colon third and statement forth and last.

Exercise: Walkthrough example

- You are analyzing Tweets from Twitter users
- Q: Which Tweets mention Donald Trump's Twitter account ("`@realDonaldTrump`") ?
- Format: `[ (tweet_ID, tweet_source_user, text) … ]`
- Use `filter()` **with `lambda` function**

```
[
  (1, "this_is_alex", "I wonder what @realDonaldTrump thinks about this"),
  (151, "motivational_quotes", "inspirational quote for the day: love trumps hate"),
  (41, "i_heart_president_trump", "Thank you @realDonaldTrump for your amazing service
to this country!!!") ,
   …
]
```

Functions as arguments (recap)

- Recap
  - Underlying concept: **Functions as arguments**
  - Make your code a bit "cleaner" and faster to write: **Lambda functions**
- Tips
  - When in doubt, write a regular function
  - If it will be re-used, write a regular function
  - Be aware of the function's inputs, outputs

Exercise 1 (sort)

- A student's grades on different subjects are given. The student would like help prioritizing which subject's homework to work on. Ideally, their worst subject should be worked on first, followed by their second subject, … . Their best subject should be worked on last
- Q: Reorder the list to match this ordering
- Format: [(subject_name, grade), …]
- Use sort()

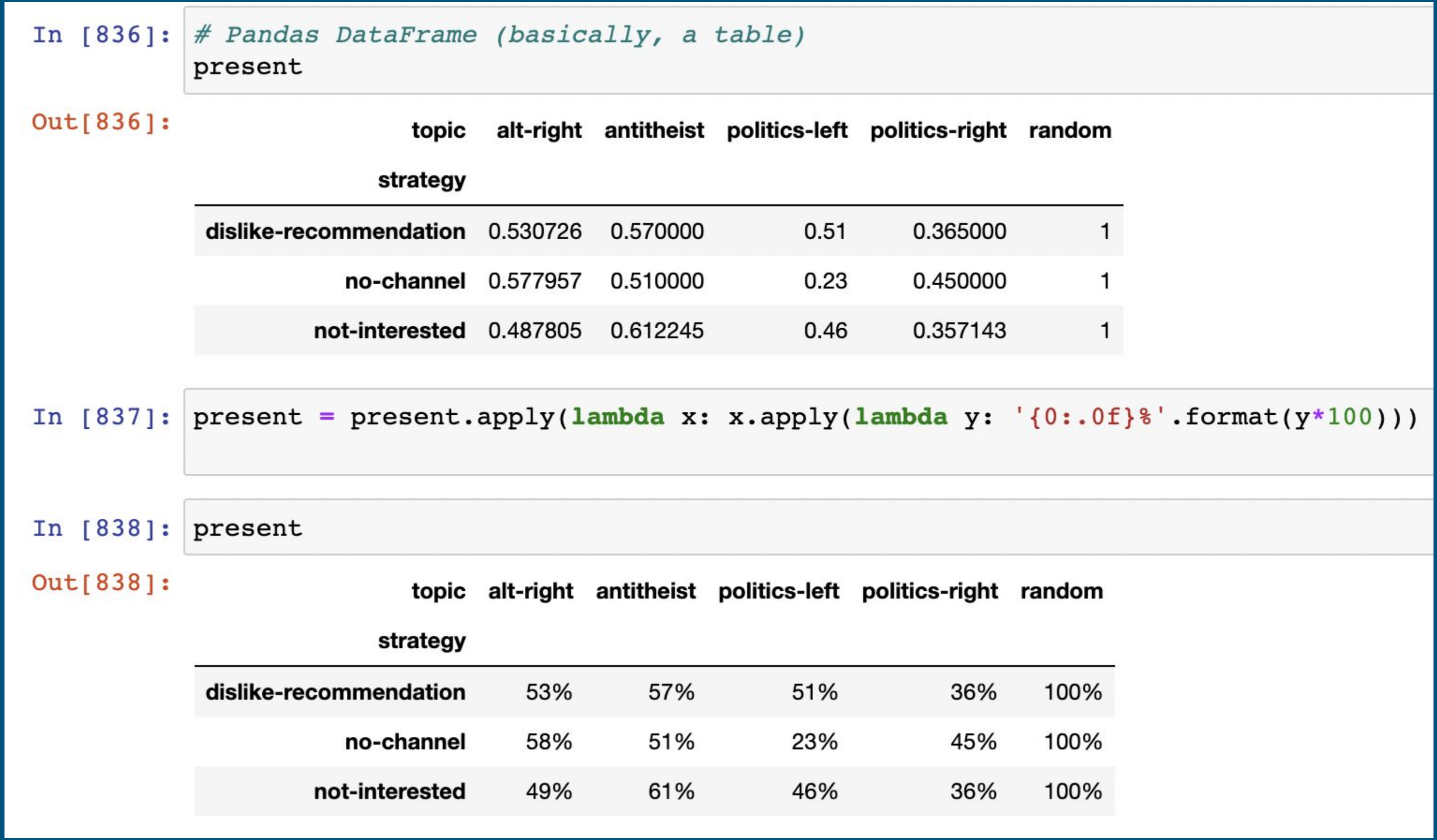
[(‘English’, 88), (‘Science’, 90), (‘Maths’, 97), (‘Social Sciences’, 82), …]

Exercise 2 (map)

- U of M departments within LSA are given. We'd like to get their acronyms (e.g. "School of Information" → "SI")
- Q: Convert this list to a list of acronyms
- Format: [department_name, …]
- Use map()

["School of Information", "Museum of Anthropological Archaeology", "Center for the Study of Complex Systems", …]

Example from Work



In [836]:

```
# Pandas DataFrame (basically, a table)
present
```

Out[836]:

topic	alt-right	antitheist	politics-left	politics-right	random
strategy					
dislike-recommendation	0.530726	0.570000	0.51	0.365000	1
no-channel	0.577957	0.510000	0.23	0.450000	1
not-interested	0.487805	0.612245	0.46	0.357143	1

In [837]:

```
present = present.apply(lambda x: x.apply(lambda y: '{0:.0f}%'.format(y*100)))
```

In [838]:

```
present
```

Out[838]:

topic	alt-right	antitheist	politics-left	politics-right	random
strategy					
dislike-recommendation	53%	57%	51%	36%	100%
no-channel	58%	51%	23%	45%	100%
not-interested	49%	61%	46%	36%	100%

"In the table, **apply** the following function to every row x:

in the row x, **apply** the following to every element y:

given the decimal y, format it to the percentage string version"

Example from Work (2)

[TODO]

Homework

Tic-Tac-Toe

You can use the rest of this time to work on HW and/or ask questions.