Next Steps in Python: Lunch Lessons

with Colby Witherup Wood

and Dan Turner

Lesson 2:
***args and**
****kwargs**
and cookies
will begin at
12:03 CDT

Please unmute me
(you might hear music)

Next Steps in Python: List Comprehensions

- On Zoom, unmute me ("Colby Witherup Wood") and choose Speaker View
- Two ways to access the materials:
  1. Go to http://www.github.com/agithasnoname/args_kwargs. Click on the green "Clone or download" button, and then Download Zip. Open Anaconda Navigator and choose either Jupyter Lab or Jupyter Notebook. Navigate to the folder you downloaded.
  2. Go to colab.research.google.com, select GitHub, search for and select agithasnoname/args_kwargs

No Bake Cookie recipe included in the GitHub repo.

Lunch Lessons are brought to you by **NUIT Research Computing Services**

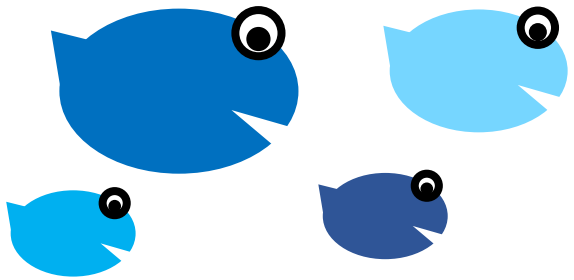
Have a programming or data question about your research? **We're here to help.** bit.ly/rcsconsult

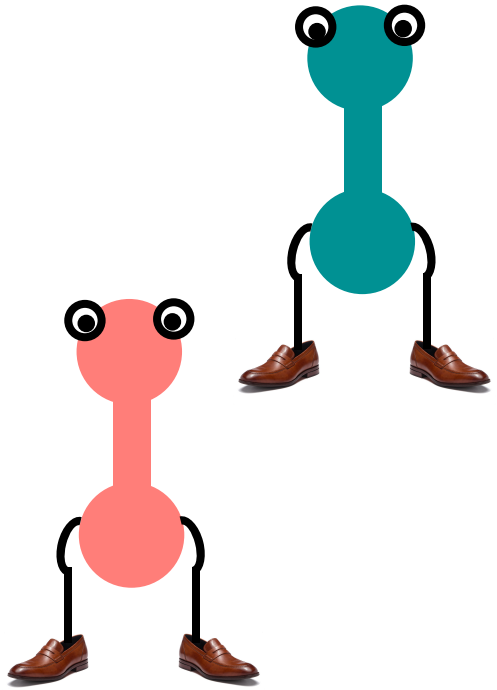
Next Steps in Python: List Comprehensions

- On Zoom, unmute me ("Colby Witherup Wood") and choose Speaker View
- Two ways to access the materials:
    1. Go to http://www.github.com/agithasnoname/args_kwargs. Click on the green "Clone or download" button, and then Download Zip. Open Anaconda Navigator and choose either Jupyter Lab or Jupyter Notebook. Navigate to the folder you downloaded.
    2. Go to colab.research.google.com, select GitHub, search for and select agithasnoname/args_kwargs

If you are having trouble opening the notebook on your own computer, try the google colab option.

*args and **kwargs





*args and **kwargs

arguments and keyword arguments

*args and **kwargs

Today's goal: Participants should be able to recognize *args and **kwargs in code and know how to use them

functions

- Built-in functions
- User-defined functions
  - function definition:

```
def add_two(a, b):
    c = a + b
    return c
```

  - function call:

```
add_two(5, 6)
```

11

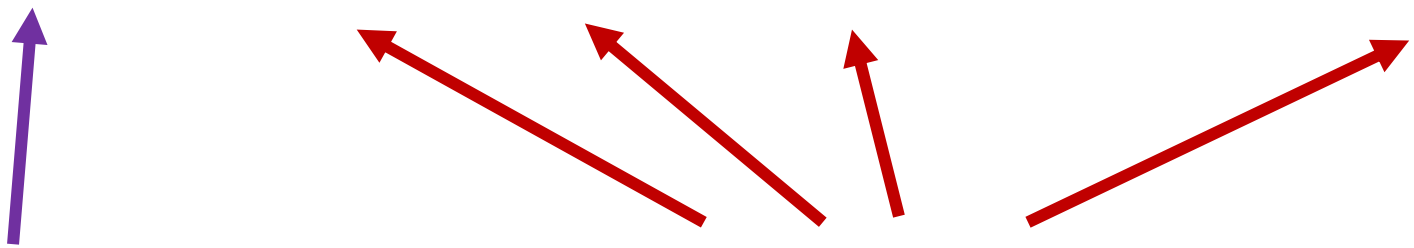
functions

- some take only a definite number of arguments
  - `abs(-16)` finds the absolute value of a number
- some can take one or multiple arguments
  - `print("something")` we usually only use one

functions

from the documentation:

print(**objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)



arguments –
always go first

keyword arguments –
always labeled with =

functions

from the documentation:

print(**objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)

the * before "objects" means that you can include more than one argument in this parameter – in fact, any number of arguments (this is what is often referred to as *args)

```
print("Hello world")
print("Hello", "world")
```

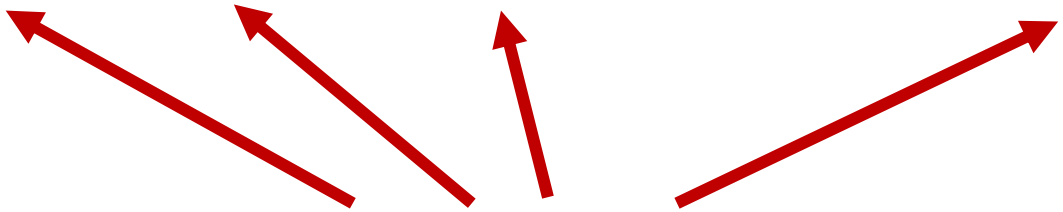
Hello world
Hello world

separate multiple *args with a comma

functions

from the documentation:

print(**objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)



keyword arguments - these are the default parameters, but you can change them

functions

from the documentation:

print(**objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)

keyword arguments include a keyword=argument

```
print("Hello", "world")
```
Hello world

(defaults to separating *args with a space)

```
print("Hello", "world", sep=" (!) ")
```
Hello (!) world

*args and **kwargs

Open up the Jupyter notebook args_kwargs.ipynb and then return to Zoom.

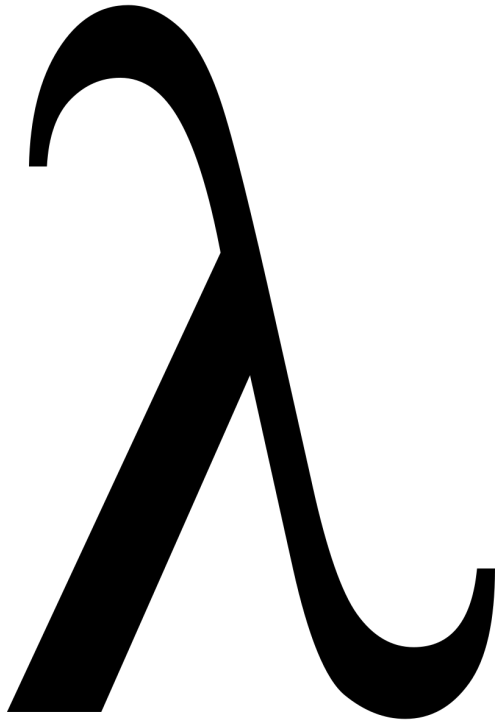
I will go over an example in the notebook (you can watch me on Zoom), and then you will have time to do an exercise in your own version of the notebook.

If you have questions at any point during the session, post them in the Zoom chat. Dan will be monitoring.

Next Steps in Python

Next week: Lambda functions and lunch

Don't forget to register on eventbrite for each week



Need help with *args and **kwargs in your own code?

We're here to help.
bit.ly/rcsconsult

Have any feedback or suggestions for other Lunch Lesson topics?

Have an easy lunch recipe that you would like to share?

colby.witherup@northwestern.edu

For more Python resources, check out our blog