Next Steps in Python: LIST COMPREHENSIONS

with Colby Witherup Wood

and special guest, Dan Turner

This is a special REPEAT of **NSIP: Lunch Lesson 1!**

WE WILL BEGIN AT 12:02 PM CENTRAL

Next Steps in Python: List Comprehensions

Two ways to access the materials:

1. Go to www.github.com/agithasnoname/list. Click on the green "Code" button, and then Download Zip. Open Anaconda Navigator and choose either Jupyter Lab or Jupyter Notebook. Navigate to the folder you downloaded. Open list.ipynb.
2. Go to colab.research.google.com, select GitHub, search for agithasnoname/list. Select list.ipynb.

If my internet breaks, everyone gets a 10-minute break!

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

Two ways to access the materials:

1. Go to www.github.com/agithasnoname/list. Click on the green "Code" button, and then Download Zip. Open Anaconda Navigator and choose either Jupyter Lab or Jupyter Notebook. Navigate to the folder you downloaded. Open list.ipynb.
2. Go to colab.research.google.com, select GitHub, search for agithasnoname/list. Select list.ipynb.

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

What is a list?

- One of the most commonly used objects in Python
- A collection of objects
- In order
- Designated by square brackets
- Objects are separated by commas
- Great for looping through

```
food_games = ["BurgerTime", "Food Fight", "Pac-Man", "Order Up!",
              "Cooking Mama", "Kool-Aid Man", "Pizza Tycoon"]
```

What is a list comprehension?

- A way to make a new list and add certain items to that list in one line of code
- Replaces making an empty list and writing a `for` loop to `append` items into the empty list

Why use a list comprehension?

- Faster to type
- Neater - turns something big and clunky into something straightforward
- Faster to run

When should you use a list comprehension?

- If you are `appending` to an empty list using a `for` loop
- AND, the list is the only output of your `for` loop

Two ways of saying the same thing:

1. For every item on the shopping list, if the item is available at the store, I will add the item to my cart.

2. I will add the item to my cart for every item on the shopping list, if the item is available in the store.

```
store_list = ["oranges", "milk", "bar soap", "pears",
              "canned artichokes", "frozen spinach",
              "frozen burrito", "ice cream", "peanuts"]

shop_list = ["oranges", "bread", "frozen spinach",
             "ice cream", "toilet paper"]
```

With paper and pencil, write Python code to make a new list that only includes the items in `shop_list` that are also in `store_list`.

```
my_cart = []
for item in shop_list:
     if item in store_list:
          my_cart.append(item)
```

```
my_cart = []
for i in shop_list:
    if i in store_list:
        my_cart.append(i)
```

`i` is a common Python name for a variable that is only used inside a loop

```
my_cart = []
for i in shop_list:
     if i in store_list:
          my_cart.append(i)


my_cart = [i for i in shop_list if i in store_list]
```

```
my_cart = []
for i in shop_list:
    if i in store_list:
        my_cart.append(i)
```

```
my_cart = [i for i in shop_list if i in store_list]
```

```
my_cart = []
for i in shop_list:
    if i in store_list:
        my_cart.append(i)
```

```
my_cart = [i for i in shop_list if i in store_list]
```

```
my_cart = []
for i in shop_list:
    if i in store_list:
        my_cart.append(i)
```

```
my_cart = [i for i in shop_list if i in store_list]
```

```
my_cart = []
for i in shop_list:
    if i in store_list:
        my_cart.append(i)
```

```
my_cart = [i for i in shop_list if i in store_list]
```

7 uses of list comprehension

Open up the jupyter notebook list.ipynb and then return to Zoom.

For each use, 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: Lunch Lessons

Next week:

Lunch Lesson 7:

pandas loc, iloc, and at

Monday, July 20, NOON – 1 PM CENTRAL

Remember to register on Eventbrite

to receive the Zoom link

Need help with list comprehensions in your own code?

We're here to help.

bit.ly/rcsconsult