Lesson 5:
**fstrings and the
"main" function
and chili**
will begin at
12:03 CDT

Please unmute me
(you should hear music)

Next Steps
in Python:
Lunch Lessons

———

with Colby Witherup Wood

and Austin Alleman

Next Steps in Python: fstrings and the "main" function

- 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/fstrings. 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/fstrings

Summer chili 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: fstrings and the "main" function

- 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/fstrings. 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/fstrings

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

Building strings out of _variables and strings_

I do this a lot.

```
new_filename = old_filename + "_ver2"



print("The mean size is " + str(mean(size_list)))



id = run_num + "_" + condition + "_" + date
```

Building strings out of _variables and strings_

Python has had several methods to make this a shorter process. Each new method has made the process a little bit shorter.

```
new_filename = old_filename + "_ver2"
```

```
new_filename = "{}_ver2".format(old_filename)
```

fstrings

```python
new_filename = f"{old_filename}_ver2"
```

fstrings syntax

```
new_filename = f"{old_filename}_ver2"
```

fstrings syntax

`new_filename = f"{old_filename}_ver2"`

No space between f and the first quotation mark

fstrings syntax

`new_filename = f"{old_filename}_ver2"`

Whatever text we want in our string

fstrings syntax

```
new_filename = f"{old_filename}_ver2"
```

fstrings syntax

```
new_filename = f"{old_filename}_ver2"
```

A variable

fstrings

```
old_filename = "my_file"
new_filename = f"{old_filename}_ver2"
print(new_filename)
```
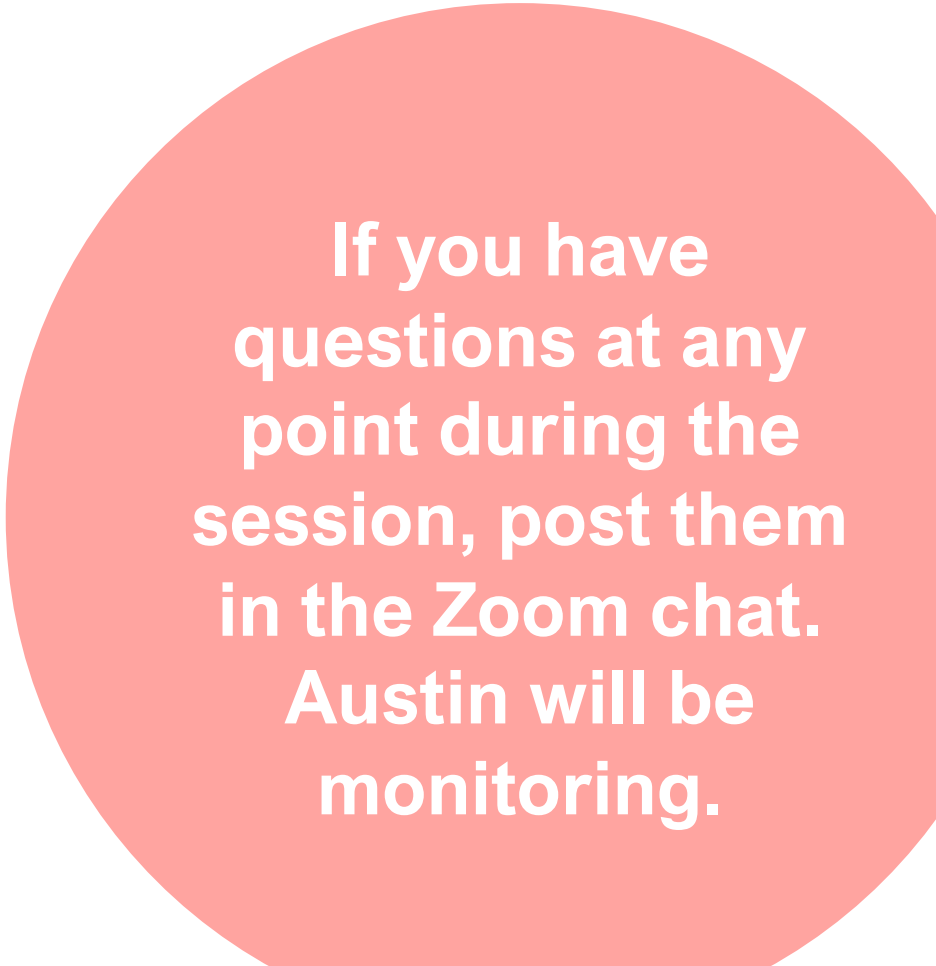
my_file_ver2

fstrings

```
name = "Colby"
adj = "delicious"
hb = f"Happy Birthday, {name}! Have a {adj} day."
print(hb)
```

Happy Birthday, Colby! Have a delicious day.

fstrings

Open up the Jupyter notebook fstrings.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. Austin will be monitoring.

"main" function and scripts

- Jupyter notebooks are only one way to run Python code

- Scripts are another – they are text files of Python code (ending in .py) that have a self-contained chunk of code

"main" function and scripts

- Scripts are a fundamental part of coding
  - you can combine them with other programs into pipelines
  - you can run them behind the scenes
  - you can run them on a computer like Quest
- They can also be called a program

docstring

function definitions

main function definition
(contains function calls)

main function call

```python
1  """
2  This is a script to convert km to miles.
3  """
4
5  def get_km():
6      i = 0
7      while type(i) != float:
8          i = input("Enter number of kilometers now. Only nu
9          try:
10             i = float(i)
11         except ValueError:
12             pass
13     return i
14
15 def convert(km):
16     j = km * 0.621371
17     return j
18
19 def main():
20     km = get_km()
21     miles = convert(km)
22     print(f"{km} kilometers is {round(miles, 4)} miles.")
23
24 if __name__ == "__main__":
25     main()
```

Why you should use a main function in your scripts

1. Organization – easier to read, code won't get overlooked because it's hidden randomly throughout your script

Why you should use a main function in your scripts

2. There are two ways to run a script:

    1. In its entirety from the command line

    2. In pieces – you can use individual functions from a script

A "main" function lets you use the script both ways.

"main" function and scripts

A script runs from beginning to end.

If you don't use `if __name__ == "__main__"` and a main function, you will not be able to use your code in pieces, only in its entirety

docstring

function definitions
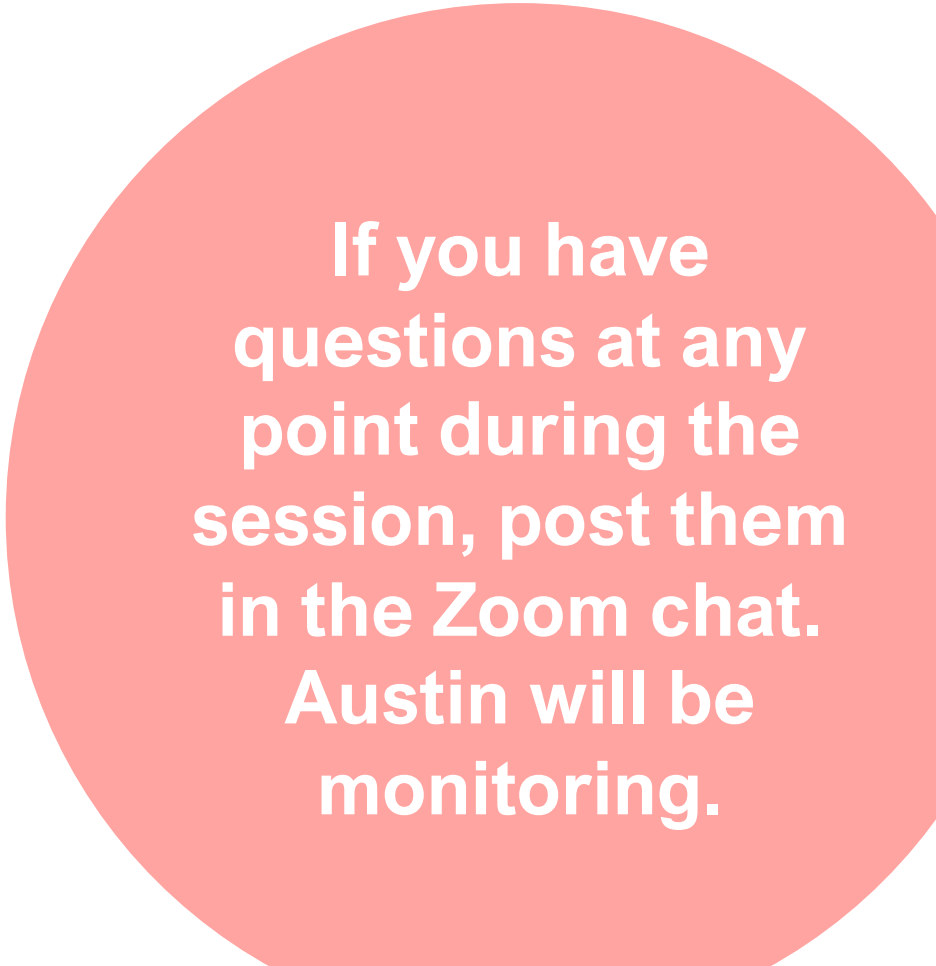
main function definition
(contains function calls)

main function call

```python
1   """
2   This is a script to convert km to miles.
3   """
4
5   def get_km():
6       i = 0
7       while type(i) != float:
8           i = input("Enter number of kilometers now. Only nu
9           try:
10              i = float(i)
11          except ValueError:
12              pass
13      return i
14
15  def convert(km):
16      j = km * 0.621371
17      return j
18
19  def main():
20      km = get_km()
21      miles = convert(km)
22      print(f"{km} kilometers is {round(miles, 4)} miles.")
23
24  if __name__ == "__main__":
25      main()
```

"main" function

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

I am first going to run the script in its entirety from the command line, and then we will use the Jupyter notebook.

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

Next Steps in Python

Next week: treasures from the standard library

Don't forget to register on eventbrite for each week

Need help
using fstrings or
writing scripts?

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