

Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try <https://etherpad.wikimedia.org>).

Users are expected to follow our code of conduct: [https://docs.carpentries.org/topic\\_folders/policies/code-of-conduct.html](https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html)

All content is publicly available under the Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>

---

## Instructors & Helpers

Kayleigh Bohemier  
Catherine DeRose  
Barbara Esty  
Jake Kara  
Rashelle Nagar  
Jennifer Snow

## Workshop page

<https://nesclio.github.io/2021-06-10-yale-online/>

#####

## Introductions

Name | Department/Research area

(during the breakout rooms section, put a !! at the end of the line if you need assistance)

- 1.Ed Hendrickson
- 2.Collin Moret
- 3.Rachel Sperling, Yale Library (Marx), shadowing this session to find out more about Library Carpentry and to learn a little more about data management. Does guacamole count as a condiment?
4. Alex Beltran Jr, Yale Information Technology Services, I wanted to be able to organize and evaluate my data through code to make things easier for future projects, and my favorite food condiment is chipotle mayonaise
- 5.Zaritza

6. Kappy Mintie, Institute for the Preservation of Cultural Heritage, interested in learning Python basics.  
Love hot sauce
- 7.
8. Max Behrendt, Divinity School Library, Interested in learning Python and Excel, Tabasco
9. Jude Yang, DASHRS, Yale Library, interested in learning basics Python and other tools.
10. Ansley Joe, Yale University Library Preservation/Conservation, particularly interested in OpenRefine,  
I've been really into sriracha ketchup lately
- 11.
- 12.
- 13.

#####

## SPREADSHEETS

What kind of operations/tasks do you do in spreadsheets?

Basic math operation such as addition, subtraction, division and multiplication for numbers.

Computing the mean of a large set of data

Entering data for future manipulation

list material information (title, author, publication, price, etc) for ordering and tracking (record)

Organizing/categorizing information

Keeping track of budget (addition, subtraction)

Which ones do you think spreadsheets are good for?

Keeping data organized and doing simple searches or tabulations

Keeping track of the data recorded and always having the set data available and defined.

What is something you have accidentally done in a spreadsheet? (Or, alternatively, something that is hard to do!)

Hard to search through data, esp across multiple areas

I accidentally deleted all of the space gaps between data entries.

#####

## Python, Part 1

Colab notebook: <https://colab.research.google.com/drive/13yQwUstbUoz7zPkhFYtQwgskNfq2fQp?usp=sharing>

Dataset: <https://yale.box.com/v/internet-users>

#####

## Python, Part 2

Intro to Python lesson (starting with for loops): <https://librarycarpentry.org/lc-python-intro/>

Colab notebooks:

- - functions & scope:  
<https://colab.research.google.com/drive/1SngXNJ0OADfN6pS5OWyqqLuVfxBV8NAL?usp=sharing>
- - for loops:  
[https://colab.research.google.com/drive/1\\_gHGfQhvCxu8AT7\\_x30N2Zys7WVnArS?usp=sharing](https://colab.research.google.com/drive/1_gHGfQhvCxu8AT7_x30N2Zys7WVnArS?usp=sharing)
- - conditions: <https://colab.research.google.com/drive/1XXrCqNDd9ai0GBZ5yKuGJq0hl-4bsa7A?usp=sharing>
- - challenge!  
<https://colab.research.google.com/drive/1YOeDrzL0Kh6WqQH5FklLalQvZnAkMFFu?usp=sharing>

```
def sum(a,b):  
    output = a + b  
    print(output)
```

```
def add_two_numbers(a,b):  
    result = a + b  
    return result
```

## Starter code

```
def classify_number(number):  
  
    # Your code here. Delete the `pass` and start coding  
    pass
```

```
## Type checking demo  
value = 1  
if type(value) == str:  
    print("This is a string")  
elif type(value) == float:  
    print("This is a float")  
elif type(value) == int:  
    print("This is an int")  
else:  
    print("This is not a string")
```

```
def classify_number(number):
    print(type(number))
    if type(number) == int or type(number) == float:
        if number < 10:
            return 'This number is small'
        elif number > 10 and number < 100:
            return 'This number is medium'
        elif number > 100:
            return 'This number is large'
        else:
            return 'Something went wrong'
    else:
        return 'This is not an int or float'
classify_number('true')
```

```
def classify_number(number):
    if (type(number) == float) or (type(number) == int):
        pass
    else:
        print ("Whoops, bad data")
    if number <= 15:
        print("This is a small number")
    if (number > 15) and (number < 50):
        print("This is a medium number")
    if number >= 50:
        print("This is a large number")
```

```
classify_number(3)
```