

# PYTHON FOR HUMANISTS

November 22, 2024

Cynthia Heider, Public Digital Scholarship Librarian

Follow along with links at: <https://github.com/upenndigitalscholarship/p4h>

# OVERVIEW



- *Introduction*
- *What is Python?*
- *Why use Python?*
- *What can humanists do with Python?*
- *Basics:* Getting started
- *Basics:* Variables
- *Basics:* Data Types
- *Basics:* Lists
- *Next steps on your Python journey*
- *What more do you want to learn?*

# WHAT IS PYTHON?

Python is a free, popular, open-source programming language that has lots of uses for the humanities.

*Our workshop learning goals  
Participants can expect to be able to...*

- set up a Python coding environment suitable for use with their own computer.
- recognize potential use cases for Python and its libraries as humanistic research tools.
- demonstrate familiarity with fundamental Python syntax and concepts such as variables, data types, lists, and conditionals.
- identify available resources for practicing skills and creating projects with Python.

# ADVANTAGES TO USING PYTHON FOR PROGRAMMING

- Open source
- Has stood the test of time
- Lots of support
- Available for everyone
- "General purpose"
- Human-friendly
- Easy to learn

# *THINGS HUMANISTS CAN DO WITH PYTHON*

Gather information

Parse documents and  
pulling out relevant  
information

Exploring patterns

Source analysis

Visualizing data in  
charts, graphs,  
infographics, maps,  
and more

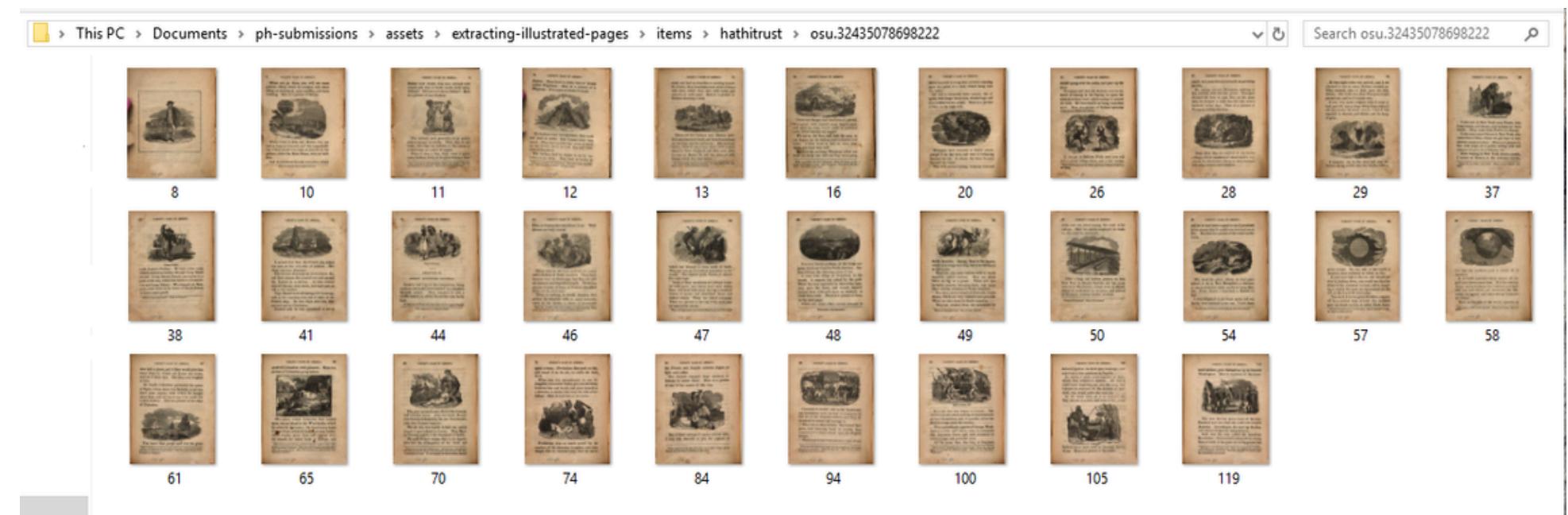
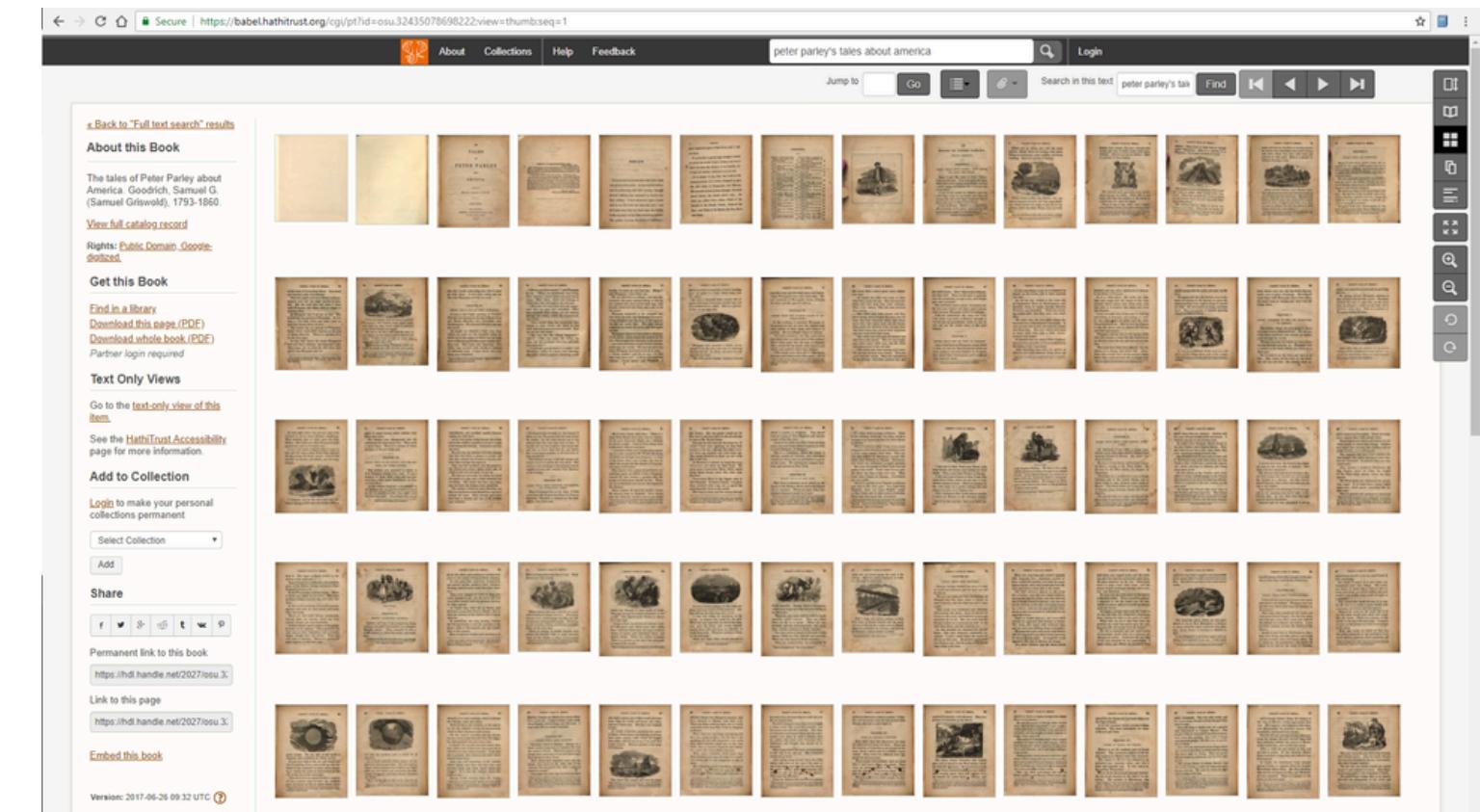
Building interfaces to  
work with source  
materials and data

Math - it does any  
calculations for you  
:)

Your idea here!

# EXAMPLES OF HUMANISTIC PYTHON PROJECTS

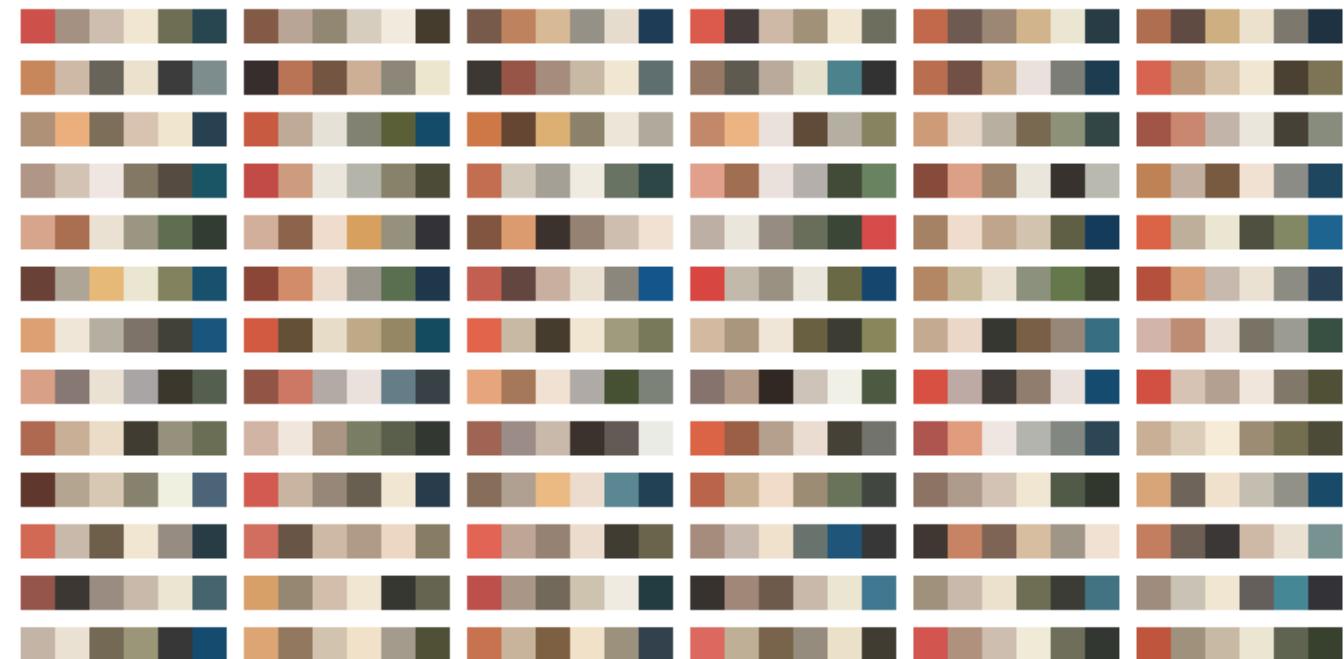
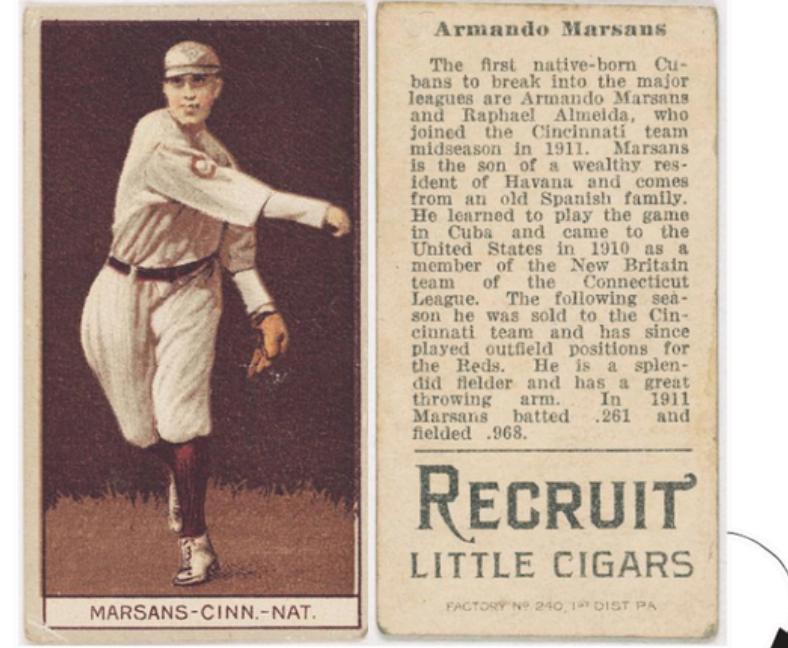
## Parsing Documents



From Stephen Krewson, "Extracting Illustrated Pages from Digital Libraries with Python," Programming Historian 8 (2019), <https://doi.org/10.46430/phen0084>.

# EXAMPLES OF HUMANISTIC PYTHON PROJECTS

## Identifying Patterns

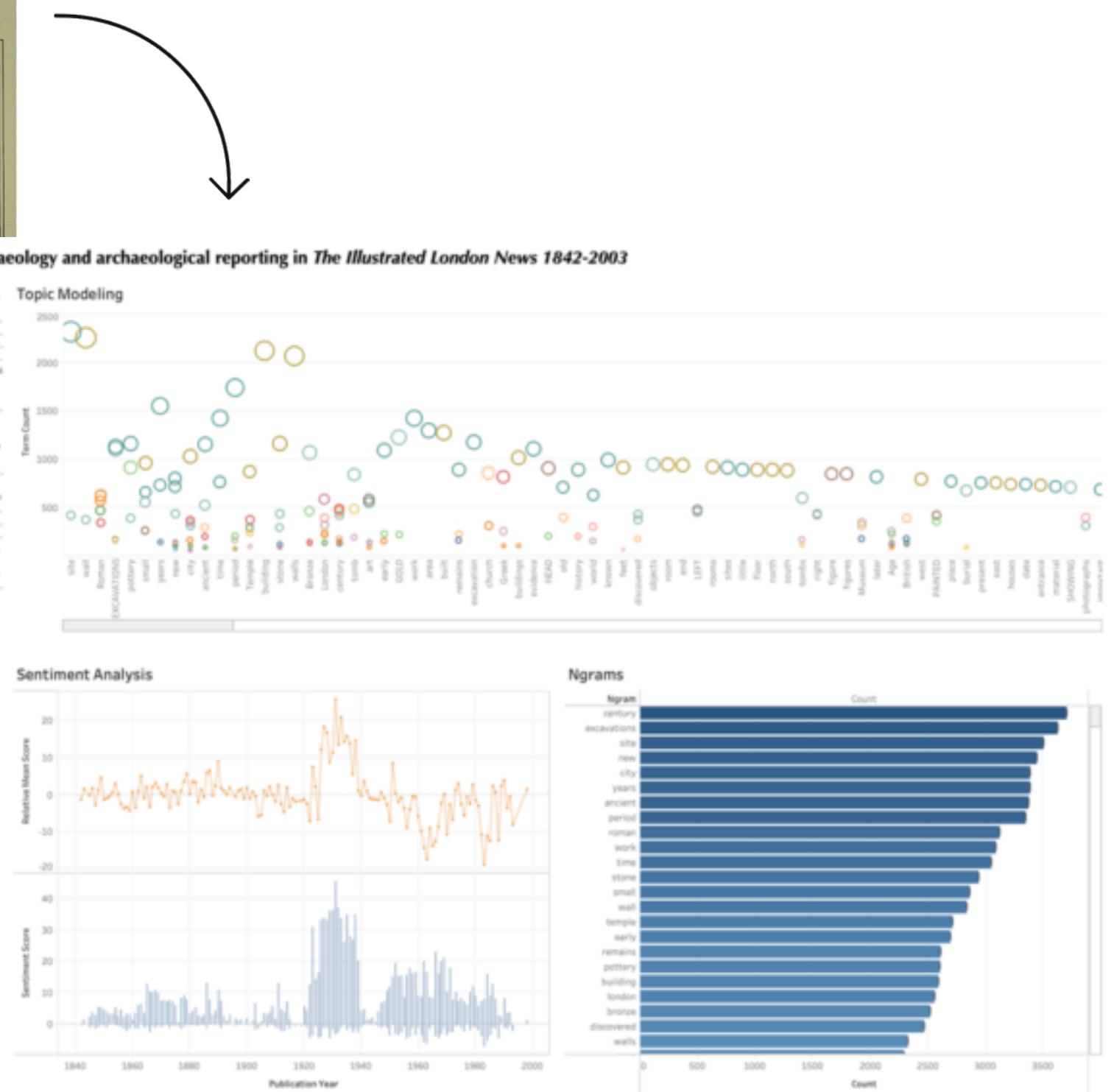


View of colors derived from the Library of Congress Baseball Cards digital collection

From Meghan Ferriter, "From Code to Colors: Working with the loc.gov JSON API," The Signal (2018), <https://blogs.loc.gov/thesignal/2018/01/from-code-to-colors-working-with-the-loc-gov-json-api/>.

# EXAMPLES OF HUMANISTIC PYTHON PROJECTS

## Source Analysis and Data Visualization



From Sarah Ketchley, "Tracking Archaeology in The Illustrated London News" Gale (2021), <https://blog.gale.com/tracking-archaeology-in-the-illustrated-london-news/>.

# GETTING STARTED

Python is an *interpreted* language.

Downloadable interpreters and Integrated Development Environments (IDEs)

Web-based IDEs

Interactive notebooks

# *CREATING A PYTHON NOTEBOOK SCRIPT*

Navigate to our  
JupyterLite environment  
to access the Notebooks at  
**<https://upenndigitalscholarship.github.io/p4h/>**

## *Objective*

- Identify available resources for practicing skills and creating projects with Python.

Python notebooks are valuable tools for practicing Python with instant validation. We'll create one to practice with in the rest of this workshop.

# VARIABLES

*Use variables to store values.*

*A variable is a kind of “sticky note.”*

---

- Variables are names for values.
- In Python the = symbol assigns the value on the right to the name on the left.
- The variable is created when a value is assigned to it.

# VARIABLES

*Use variables to store values.*

*A variable is a kind of “sticky note.”*

---

Variable names:

- cannot start with a digit
- cannot contain spaces, quotation marks, or other punctuation
- may contain an underscore (typically used to separate words in long variable names)

# *SHOWING YOUR WORK WITH THE 'PRINT' FUNCTION*

## **Use print to display values**

- Python has a built-in function called print that prints things as text.
- Call the function (i.e., tell Python to run it) by using its name.
- Provide values to the function (i.e., the things to print) in parentheses.
- To add a string to the printout, wrap the string in single quotations.
- The values passed to the function are called ‘arguments’

## **Example**

```
>>>print(first_name, 'is', age, 'years  
old')
```

# VARIABLES

*Use variables to store values.*

*A variable is a kind of “sticky note.”*

---

You can also use them for calculations!

Example:

```
>>> age = age + 3  
>>> print('Age in three years:', age)
```

# *EXPLORING DATA TYPES*

**Integers (whole numbers)**  
**Floats (decimals)**  
**Strings (text)**  
**Booleans (true/false)**

## *Practical Implications*

- Every value has a type.
- Types control what operations can be done on values.
- Strings can be added and multiplied.
- Strings have a length (but numbers don't).
- Must convert numbers to strings or vice versa when operating on them.
- Can mix integers and floats freely in operations.
- Variables only change value when something is assigned to them.

# LISTS

**Lists are a common data structure to hold an ordered sequence of elements.**

---

Each element can be accessed by an index.

Note that Python indexes start with 0 instead of 1:

**EXAMPLE:**

```
>>> numbers = [1, 2, 3]
```

```
>>> numbers[0]
```

# 'FOR' LOOPS WITH LISTS

A for loop can be used to access the elements in a list or other Python data structure one at a time:

Example

```
for num in numbers:
```

```
...     print(num)
```

```
...
```

```
1
```

```
2
```

```
3
```

## *Indentation*

Indentation is very important in Python!  
Note that the second line in the example is indented.

Just like three chevrons >>> indicate an interactive prompt in Python, the three dots ... are Python's prompt for multiple lines. This is Python's way of marking a block of code.

# LISTS

Lists are a common data structure to hold an ordered sequence of elements.

---

Other things you can do with lists:

Add items to the end of a list with append:

```
>>> numbers.append(4)  
>>> print(numbers)  
[1, 2, 3, 4]
```

# LISTS

Lists are a common data structure to hold an ordered sequence of elements.

---

Other things you can do with lists:

Sort items in a list with sort:

```
>>>Ages = [28, 19, 60, 80, 23]  
>>>ages.sort()  
>>>print(ages)
```

# USEFUL VOCABULARY

---

- = Assign variable
  - \_ Temporary function in a variable
  - . Attribute of a variable
  - [ ] A list of items that can be changed
  - “ ” Defines a string
  - ( ) A list of items that can't be changed
  - ( ) “Calls” or activates an activity
  - # Starts a comment
-

# 'READING' A PYTHON NOTEBOOK SCRIPT

Navigate to our JupyterLite environment to access the Notebooks at <https://upenndigitalscholarship.github.io/p4h/>

## *Objective*

- Demonstrate familiarity with fundamental Python syntax and concepts such as variables, data types, lists, and conditionals.

Which elements do we see in the notebook? How are they being used?

# *LEARNING MORE*

Penn Libraries  
Libguide for Python

Dataquest subscription  
through Penn Libraries  
(free!).

Penn Libraries'  
statistical software  
consultants

Penn Libraries'  
Carpentries sessions

Python for Humanities  
interactive textbook by  
Felipe Álvarez de Toledo

Introduction to Cultural  
Analytics & Python open  
interactive textbook by  
Melanie Walsh

Exploratory.  
Programming for the  
Arts and Humanities  
by Nick Montfort

Humanities Data  
Analysis: Case Studies  
with Python by Karsdorp,  
Kestemont, and Riddell