Penn Libraries
University of Pennsylvania

PYTHON FOR HUMANISTS

November 8, 2022

Cynthia Heider, Public Digital Scholarship Librarian

Follow along with links at: **https://etherpad.wikimedia.org/p/P4H_Penn_2022**

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?*

Follow along with links at: **https://etherpad.wikimedia.org/p/P4H_Penn_2022**

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.

Back to Overview

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

Back to Overview

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!

GETTING STARTED

Python is an *interpreted* language.

Downloadable interpreters and Integrated Development Environments (IDEs)

Web-based IDEs

Interactive notebooks

Back to Overview

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’)
```

Back to Overview

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.

Back to Overview

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.

Back to Overview

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)
```

'READING' A PYTHON NOTEBOOK SCRIPT

Navigate to our JupyterLite environment to access the Notebooks at **https://upenndigitalscholarship.github.io/p4h2022/**

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?