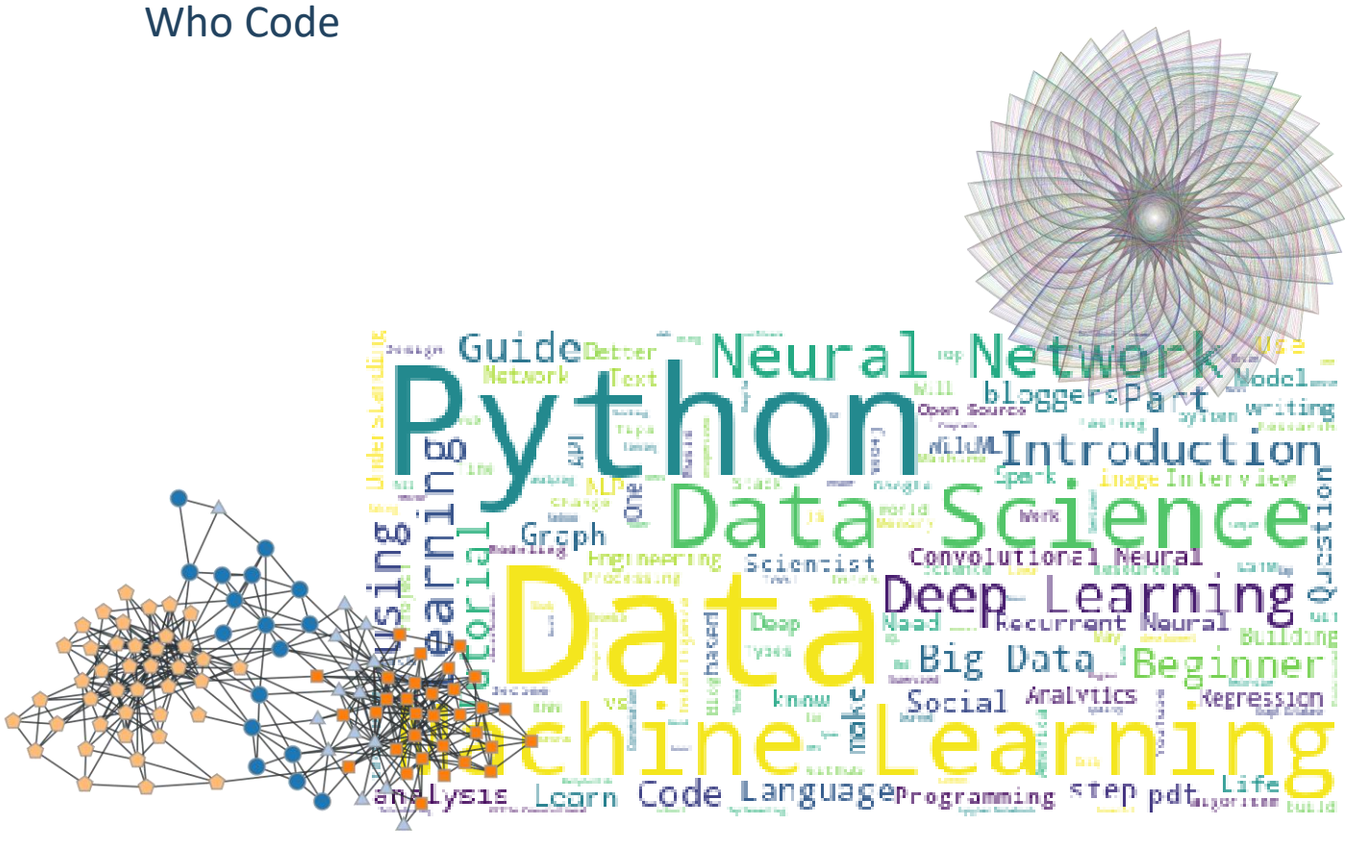
Python Pizza Party!

Presented by Women Who Code



https://github.com/nelariddle/IntroToPython

Agenda

1. Functions
2. Variables, datetime library
3. Recursive functions
4. Lists and loops
5. Conditionals, random numbers
6. Strings and dictionaries
7. Web scraping
8. File reading/writing

Objective

Learn about the many capabilities of Python while practicing coding thinking.

Functions

1. A way to process (optional) inputs and return (optional) outputs
2. Also called a subroutine, method, or procedure
3. Examples
    a. Print your name
    b. Area of a triangle
    c. Pythagorean theorem (will need **import math**)
4. Try it!
    a. Function to print your name and age
    b. Function to find area of a circle
5. Tips
    a. Docstring
    b. Comments (single line and multiline)

Variables and Datetime

1. A **variable** is a way to store information for later use
    a. Built-in types: string, int, list, dict, etc.
    b. Created with **assignment**
    c. **var_name = var_value**
2. The library **datetime** allows us to concisely store dates
    a. First, we need **from datetime import datemine**
    b. We can create a datetime variable with **datetime(year, month, day)**
    c. Get the current time with **datetime.now()**
    d. Subtract dates with **–**
3. Example!
    a. Write a function that finds the time since a date (the time difference between now and then). Use it to calculate your age!

Recursive Functions

1. Usually used in a mathematical context where a problem has an identical subproblem
2. Example
   a. Factorial

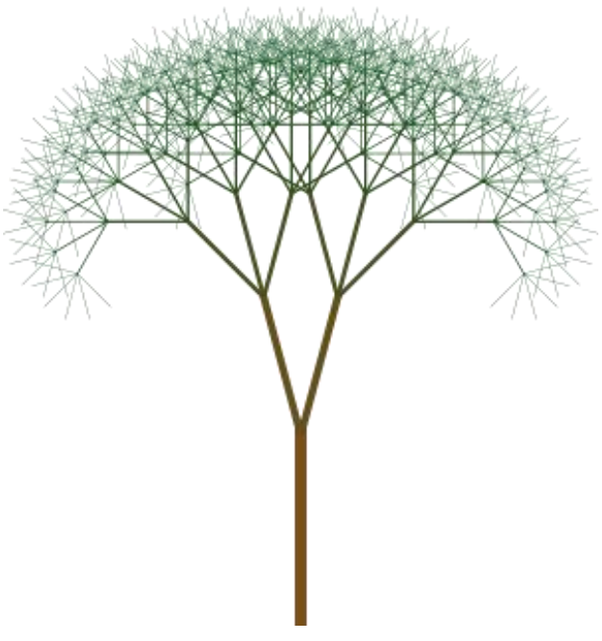
$$1! = 1$$

$$2! = 2(1) = 2$$

$$3! = 3(2)(1) = 6$$

$$4! = 4(3)(2)(1) = 24$$

$$5! = 5(4)(3)(2)(1) = 120$$

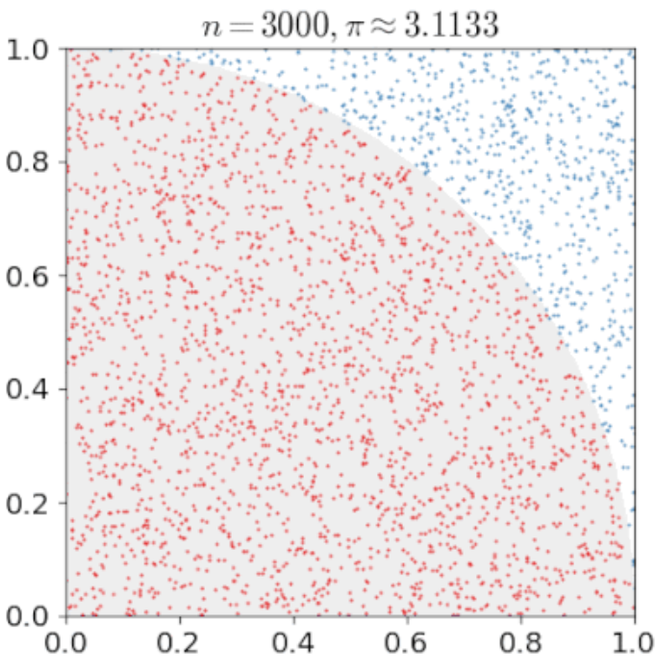


Lists and loops

1. Lists simply store several items in brackets
    a. Create a list by surrounding comma-separated values with []
    b. **my_list = [1,2,3]**
    c. **epic_list = [“nela”, “stuti”, “dikshita”]**
2. Loops can be used to execute something a certain number of times, or iterate over a collection of items
    a. **for thing in stuff:** where stuff is the overarching list, and thing refers to an individual item
    b. Must tab all lines inside the loop
3. Example
    a. Printing a list of names

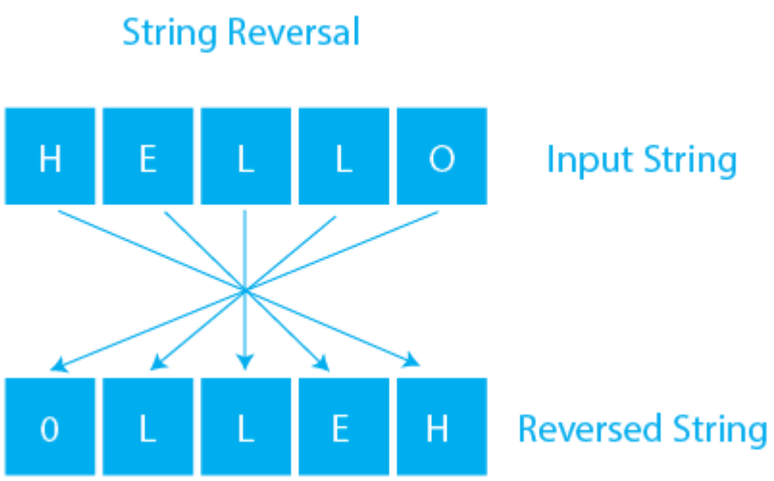
Conditionals, random numbers

1. We can use the random library to generate random numbers
   a. **random.randint(a,b)**
   b. **random.random()**
2. **for i in range(n):** a for loop that repeats code n times
3. Can increment a variable with **+=**
4. Conditionals execute code based on a certain condition
   a. **if condition:** where condition evaluates to true or false
   b. Must use **==** for comparison
5. Example
   a. Approximating pi



Strings and dictionaries

1. A dictionary is used for storing pairs of items (item:price, character:wand, etc)
    a. Can be accessed with name_of_dict[key]
2. A string is a way to store letters and words
    a. **dessert = “apple pie”**
3. We can loop through strings using the **for** loop as well
    a. **for char in string:**
4. Example
    a. NATO alphabet
5. Try it!
    a. Reverse a string with a for loop



Web scraping

1. Fun for hobbyists and useful for students/researchers!
2. Let's retrieve the national debt with web scraping:
    a. Google "national debt pgpf"
    b. Right-click > View Page Source
    c. Ctrl-F > 33,675…
    d. We need to access the **span** tag surrounding it
3. Try it!
    a. Find the per-person debt (use the **div** instead of span)
    b. Write a function that converts a dollar amount to an int using for loop and if statement
    c. Find the current US population!

File reading

1. Files are useful for processing lots of data we don't want directly in the code
2. We will use **with open("filename.txt") as file:**
    a. To read individual lines, we write **for line in file:**
    b. To read words within a line, we write **for word in line.split():**
    c. Let's add the words to a dictionary to keep track of counts!
3. Challenge: can you generate counts of 2-grams (2 adjacent words)? 3-grams?

File writing

1. We write to files if we want the output of our code to be used by other programs/languages
2. Again use **with open**
    a. This time we will create **writer = csv.writer(file)** to access the file and **writer.writerow()** to write to the file
    b. **writerow()** takes a list of everything to write on the line

Further learning…

1. Codecademy
2. w3schools.com
3. “Automate the Boring Stuff with Python”
4. YouTube
5. ChatGPT/Google Bard
    a. “create a study plan for learning python for…”
    b. “how to perform sentiment analysis in python”
    c. “explain this error I’m getting”
6. Request topic-specific workshops

Join our Teams for event resources!

Team code:
3gvaw0u