INTRO TO PYTHON LIBRARIES & MODULES

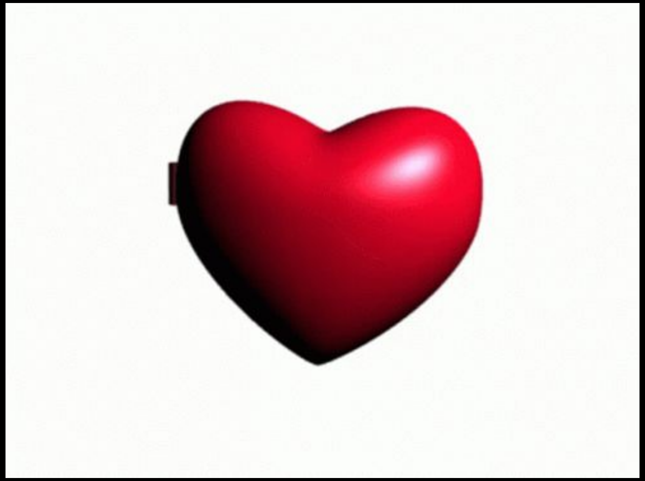
SJU ACM STUDENT CHAPTER

SJU ACM
Student Chapter

BEFORE WE START...

IMPORTANT REMINDERS:

- THIS LAB WILL INVOLVE THE USE OF PYTHON AND VISUAL STUDIO CODE (VSCODE/VSC), BOTH THROUGHOUT THE PRESENTATION AND THE CODE-ALONG LAB AT THE END.
- IF YOU DO NOT HAVE PYTHON AND/OR VSCODE SET UP ON YOUR MACHINE, THE FOLLOWING WILL BE A GUIDE ON HOW TO SET UP PYTHON AND VSCODE.



PYTHON/VSCODE SET UP

- STEP ONE:
  - CHECK IF YOU HAVE PYTHON ON YOUR MACHINE
    - WINDOWS: PRESS WIN+R, TYPE IN CMD, AND COMMAND PROMPT SHOULD POPUP UP. TYPE IN ‘PY —VERSION’
    - MAC OS: LAUNCH TERMINAL, TYPE IN ‘PYTHON3 —VERSION’
  - IF NOT ON YOUR MACHINE, PROCEED TO STEP TWO
- STEP TWO:
  - VISIT PYTHON.ORG/DOWNLOADS, CLICK ‘DOWNLOAD PYTHON 3.12.2 (THIS IS THE LATEST MAJOR RELEASE)
  - FOLLOW ALL ON-SCREEN INSTRUCTIONS ON PYTHON SETUP WIZARD
    - MAINTAIN ALL DEFAULT SETTINGS

PYTHON/VSCODE SET UP CONT.

- STEP THREE
  - VISIT CODE.VISUALSTUDIO.COM
    - WINDOWS: CLICK DOWNLOAD BUTTON
    - MACOS: CLICK THE ARROW NEXT TO DOWNLOAD BUTTON, AND DOWNLOAD MAC OS STABLE RELEASE
- STEP FOUR:
  - FOLLOW ON-SCREEN INSTRUCTIONS ON VSCODE SETUP WIZARD
    - NOTES:
      - YOU WILL GET AN OPTION TO CREATE A DESKTOP ICON, I RECOMMEND YOU CHECK THAT BOX
      - ONCE THE INSTALLATION PROCESS FINISHES, ENSURE TO CHECK THE BOX TO LAUNCH VSCODE

PYTHON/VSCODE SET UP CONT.

- STEP FIVE
  - ONCE INSIDE VSCODE, CUSTOMIZE YOUR EXPERIENCE TO YOUR LIKING (THIS CAN BE CHANGED LATER)
    - FOR NOW, ENSURE YOU SELECT THE COLOR THEME YOU WANT
- STEP SIX:
  - ON THE LEFT HAND SIDE, THERE IS A MENU WITH SEVEN ICONS (FIVE ON THE TOP HALF, TWO ON THE BOTTOM HALF)
  - CLICK ON THE LAST ICON IN THE TOP HALF OF THE MENU (EXTENSIONS)
- STEP SEVEN:
  - ONCE INSIDE EXTENSIONS, YOU WILL SEARCH FOR TWO THINGS THAT YOU NEED TO DOWNLOAD:
    - PYTHON (SHOULD BE IN THE POPULAR SECTION)
    - CODE RUNNER (HELPS YOU RUN YOUR CODE)
  - INSTALL BOTH, AND YOU SHOULD BE SET!

PRIMER ON PYTHON

PRIMER ON PYTHON

- PYTHON IS A HIGH-LEVEL, GENERAL-PURPOSE PROGRAMMING LANGUAGE.
    - IT IS DYNAMICALLY-TYPED AND GARBAGE-COLLECTED.
    - KNOWN FOR CLEAR SYNTAX, READABILITY, FLEXIBILITY, AND INDENTATION.
    - SUPPORTS MULTIPLE PROGRAMMING PARADIGMS:
        - STRUCTURED, OOP AND FUNCTIONAL PROGRAMMING
- FAST FACTS:
    - CREATOR: GUIDO VAN ROSSUM
    - FIRST RELEASE: FEBRUARY 20TH, 1991
    - LATEST RELEASE: PYTHON 3.12.2 (FEBRUARY 6TH, 2024)

PRIMER ON PYTHON CONT.

- PYTHON IS KNOWN FOR ITS COMPREHENSIVE STANDARD LIBRARY.
  - AS A RESULT, PYTHON IS SAID TO BE A “BATTERIES INCLUDED” LANGUAGE.
  - IT PROVIDES BROAD FUNCTIONALITY, WITH A VAST NUMBER OF MODULES AND PACKAGES THAT SUPPORT A VARIETY OF TASKS.
  - ELIMINATES NEED FOR EXTERNAL DEPENDENCIES.
  - INCLUDED WITH EVERY PYTHON INSTALLATION!
- EXAMPLES:
  - TEXT PROCESSING, DATA TYPES, MATH, FILE/DIRECTORY ACCESS, ETC.

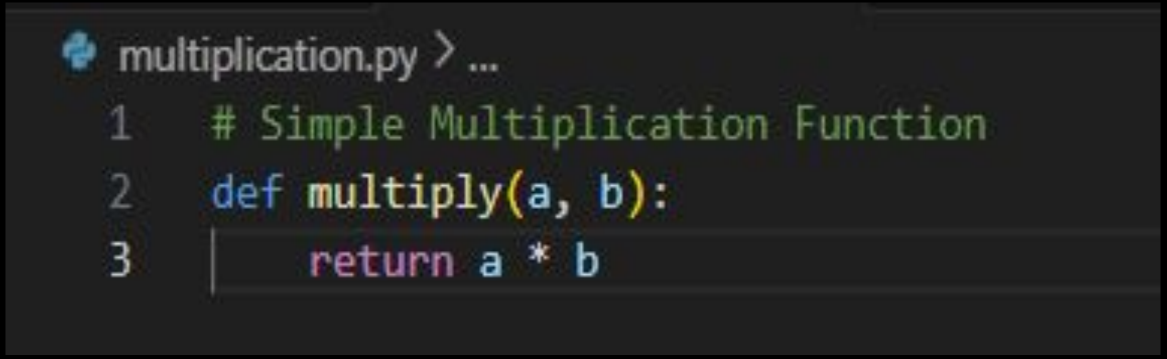
BASICS OF MODULES AND LIBRARIES

WHAT ARE MODULES?

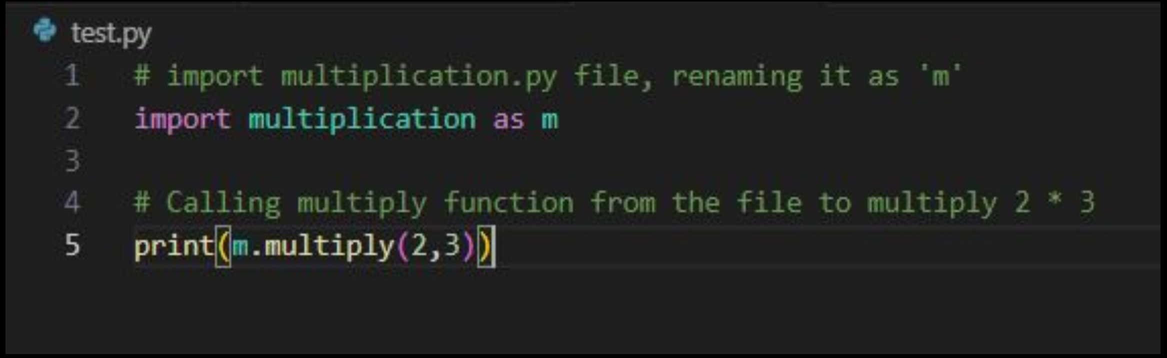
- A MODULE IN PYTHON IS A FILE CONTAINING PYTHON DEFINITIONS AND STATEMENTS.
- CAN DEFINE FUNCTIONS, CLASSES, AND VARIABLES THAT CAN BE USED IN OTHER PYTHON FILES.
- TYPICALLY USED TO BREAK DOWN LARGE PROGRAMS AND TASKS INTO SMALL AND MANAGEABLE FILES.
- TWO TYPES OF MODULES:
  - USER-DEFINED MODULES
  - BUILT-IN MODULES

USER-DEFINED MODULES

- HOW TO CREATE YOUR OWN MODULE:
    - FIRST, CREATE A NEW FILE, AND NAME IT APPROPRIATELY. ENSURE IT ENDS IN .PY.
    - NEXT, DEFINE A FUNCTION, CLASS, VARIABLE, ETC.
    - THEN, IN A NEW FILE, USE THE ‘IMPORT’ KEYWORD ON THE FIRST LINE OF CODE.
        - YOU ARE ABLE TO RENAME THE MODULE WHATEVER YOU WANT IN THE NEW FILE.
    - FINALLY, ACCESS THE FUNCTION(S) YOU DESIRE USING THE FOLLOWING SYNTAX:
        - MODULE_NAME.FUNCTION_NAME



```python
# Simple Multiplication Function
def multiply(a, b):
    return a * b
```



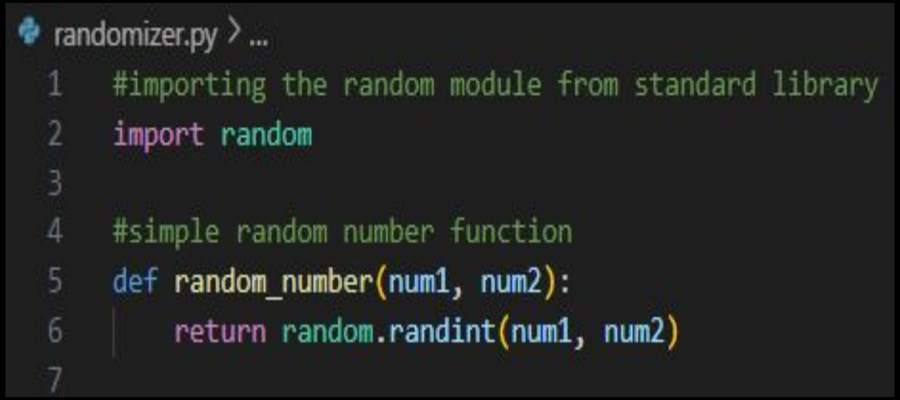
```python
# import multiplication.py file, renaming it as 'm'
import multiplication as m

# Calling multiply function from the file to multiply 2 * 3
print(m.multiply(2,3))
```

BUILT-IN MODULES

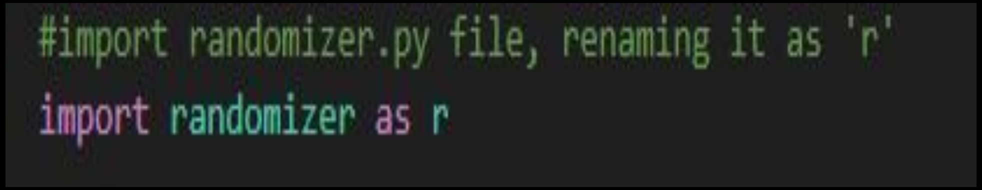
- PYTHON COMES EQUIPPED WITH A LIBRARY OF STANDARD MODULES.
    - OVER 200 MODULES IN THE LIBRARY
    - SOME ARE BUILT INTO THE INTERPRETER, OTHERS ARE NOT.
    - MOST MODULES ARE AVAILABLE TO USERS, BUT SOME ARE DESIGNED FOR SPECIFIC OPERATING SYSTEMS/PLATFORMS/SOFTWARE.
        - WINREG, WINSOUND: DESIGNED FOR WINDOWS
        - POSIX, PWD, SYSLOG: DESIGNED FOR UNIX/LINUX

EXAMPLES OF MOST-OFTEN USED MODULES

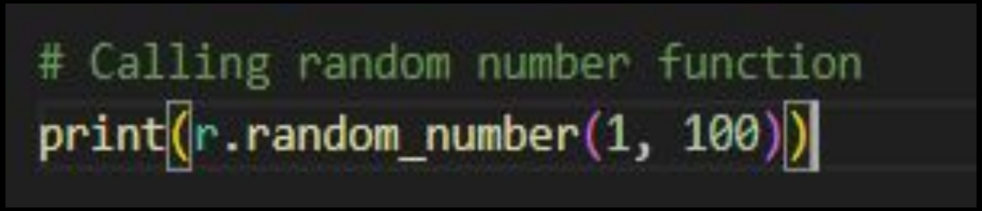
- OS:
    - PROVIDES WAY TO USE OS-DEPENDENT FUNCTIONALITY, E.G. NAVIGATING FILE SYSTEM
- SYS
    - USED TO MANIPULATE THE PYTHON RUNTIME ENVIRONMENT
- DATETIME
    - SUPPLIES CLASSES FOR MANIPULATING DATES & TIME, ESSENTIAL FOR TIME-BASED OPS
- MATH
    - OFFERS ACCESS TO MATHEMATICAL FUNCTIONS
- RANDOM
    - PROVIDES TOOLS FOR RANDOM SELECTIONS



```python
#importing the random module from standard library
import random

#simple random number function
def random_number(num1, num2):
    return random.randint(num1, num2)

```



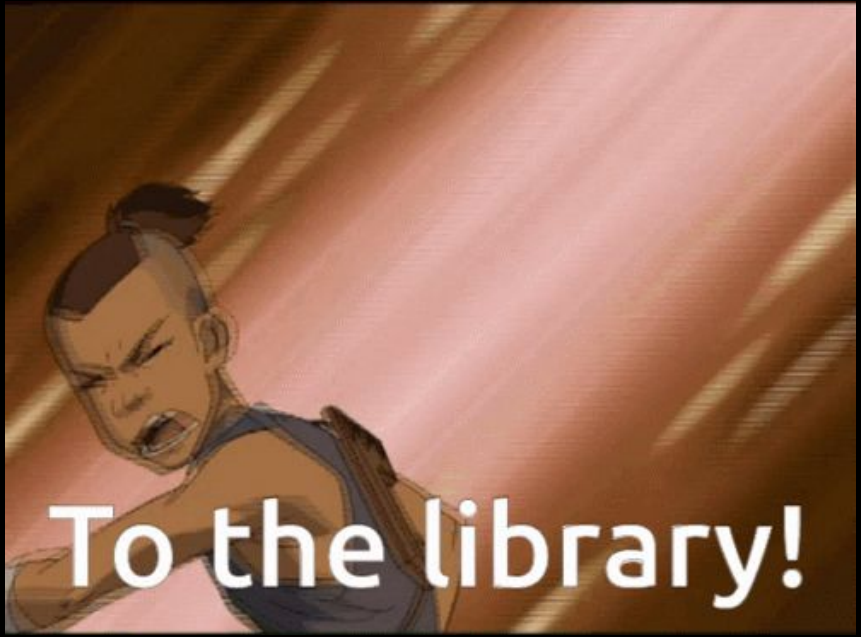
```python
#import randomizer.py file, renaming it as 'r'
import randomizer as r
```



```python
# Calling random number function
print(r.random_number(1, 100))
```

WHAT ARE LIBRARIES?

- IN SHORT, A LIBRARY IN PYTHON IS AN UMBRELLA TERM THAT REFERS TO A REUSABLE CHUNK OF CODE.
  - TYPICALLY CONSISTS OF A COLLECTION OF *RELATED* MODULES AND PACKAGES
  - IN OTHER WORDS, LIBRARIES ARE A COLLECTION OF PACKAGES.
- LIBRARIES HELP DEVS SHARE REUSABLE CODE WITH THE COMMUNITY.
  - THIS ELIMINATES THE NEED TO WRITE CODE FROM SCRATCH, WHICH IS VERY USEFUL FOR LARGE-SCALE PROJECTS!
  - THIS IS DONE BY CREATING A SET OF FUNCTIONS THAT ARE RELATED TO THE SAME AREA.
- FUN FACT:
  - THERE CURRENTLY ARE OVER 137,000 PYTHON LIBRARIES!
    - ROOKIE NUMBERS!

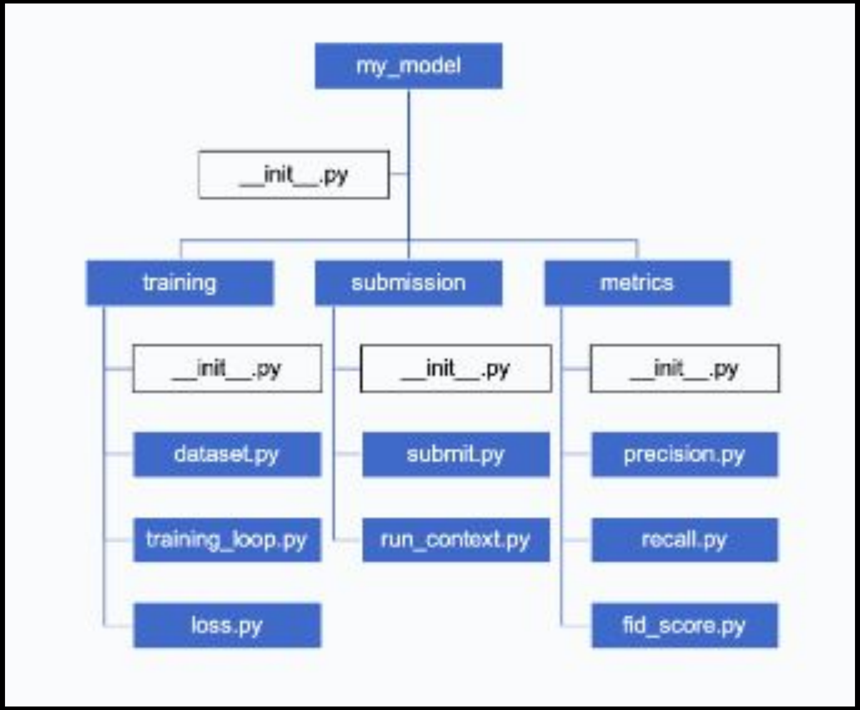


POPULAR PYTHON LIBRARIES

- DATA ANALYSIS & SCIENCE:
  - PANDAS & NUMPY: DATA STRUCTURES, DATA ANALYSIS, MATH FUNCTIONS
- MACHINE LEARNING & AI:
  - TENSORFLOW & PYTORCH: FLEXIBILITY, SPEED, EFFICIENCY IN ML & AI DEVELOPMENT
- DATA VISUALIZATION:
  - MATPLOTLIB & SEABORN: HIGH-LEVEL INTERFACES FOR VISUALIZING AND DRAWING STATISTICAL GRAPHICS
- SCIENTIFIC COMPUTING:
  - SCIPY & SUMPY: SCIENCE COMPUTING, ALGORITHMS, & MATHEMATICAL TOOLS
- NETWORKING & INTERNET:
  - REQUESTS & BEAUTIFUL SOUP: HTTP LIBRARY, WEB SCRAPING

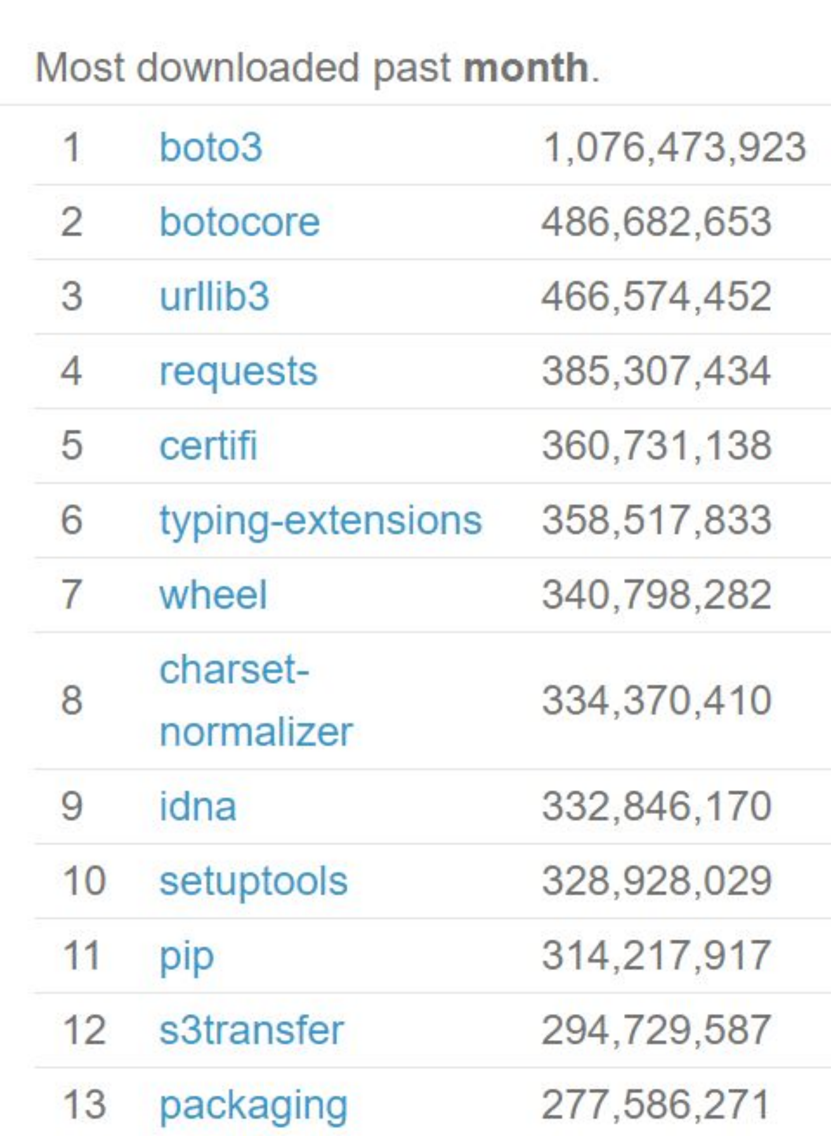
WHAT ARE PACKAGES?

- PACKAGES IN PYTHON ARE DIRECTORIES OF A COLLECTION OF MODULES
  - PACKAGES ALLOW FOR HIERARCHICAL STRUCTURING OF THE MODULE NAMESPACE.
  - IN THE SAME WAY WE ORGANIZE OUR FILES INTO FOLDERS AND SUBFOLDERS ON A HARD DRIVE, WE CAN ORGANIZE MODULES INTO PACKAGES AND SUBPACKAGES.
- EACH PACKAGE MUST INCLUDE AN INITIALIZATION FILE FOR IT TO BE CONSIDERED A PACKAGE.
  - FILE NAME: '__INIT__.PY'
- FUN FACT:
  - THERE ARE OVER 497,000 PYTHON PACKAGES, ACCORDING TO PYPISTATS.ORG (AS OF 2/14/2024)
    - ALL-PRO NUMBERS!



POPULAR PACKAGES

- NUMPY/PANDAS
  - PREVIOUSLY MENTIONED; ARE PACKAGE/LIBRARY HYBRIDS
- PYTEST
  - PROVIDES A VARIETY OF MODULES TO TEST NEW CODE.
    - INCL. SMALL UNIT/COMPLEX FUNCTIONAL TESTS
- BOTO3/BOTOCORE
  - THESE ARE PART OF THE AMAZON WEB SERVICES SOFTWARE DEVELOPMENT KIT (AWS SDK).
  - MAKES IT EASY TO INTEGRATE PYTHON WITH AWS SERVICES!
- PIP
  - THE RECOMMENDED TOOL FOR INSTALLING PYTHON PACKAGES
- SETUPTOOLS
  - HELPS TO EASILY DOWNLOAD, BUILD, INSTALL, UPGRADE, AND UNINSTALL PYTHON PACKAGES.
  - REQUIRES A FEW DEPENDENCIES FOR IT TO WORK

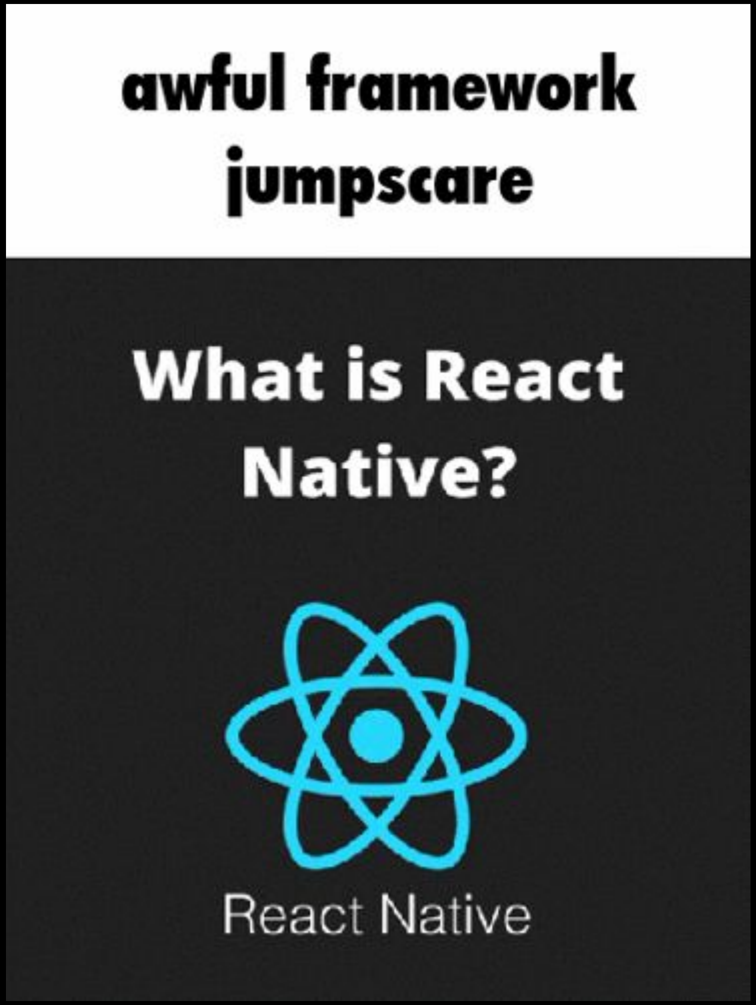


Most downloaded past **month**.

1	boto3	1,076,473,923
2	botocore	486,682,653
3	urllib3	466,574,452
4	requests	385,307,434
5	certifi	360,731,138
6	typing-extensions	358,517,833
7	wheel	340,798,282
8	charset-normalizer	334,370,410
9	idna	332,846,170
10	setuptools	328,928,029
11	pip	314,217,917
12	s3transfer	294,729,587
13	packaging	277,586,271

WHAT ARE FRAMEWORKS?

- FRAMEWORKS IN PYTHON ARE A SPECIAL COLLECTION OF MODULES AND PACKAGES THAT HELP PROGRAMMERS FAST TRACK THE DEVELOPMENT PROCESS.
    - USUALLY MORE COMPLEX THAN LIBRARIES
    - CONTAIN PACKAGES THAT PERFORM SPECIFIC OPERATIONS
        - BASIC FLOW & ARCHITECTURE OF APPLICATION
- ANALOGY OF HOUSE CONSTRUCTION:
    - PYTHON FRAMEWORKS PROVIDE YOU WITH ALL THE ESSENTIAL BUILDING BLOCKS OF CONSTRUCTION
        - FOUNDATION, WALLS, WINDOWS, ROOF, ETC.
    - THEN, DEVS BUILD THE APPLICATION AROUND THE FOUNDATION AND ADD FURNITURE, APPLIANCES, ETC.
- THREE TYPES OF FRAMEWORKS:
    - FULL-STACK
    - MICRO
    - ASYNCHRONOUS



TYPES OF FRAMEWORKS

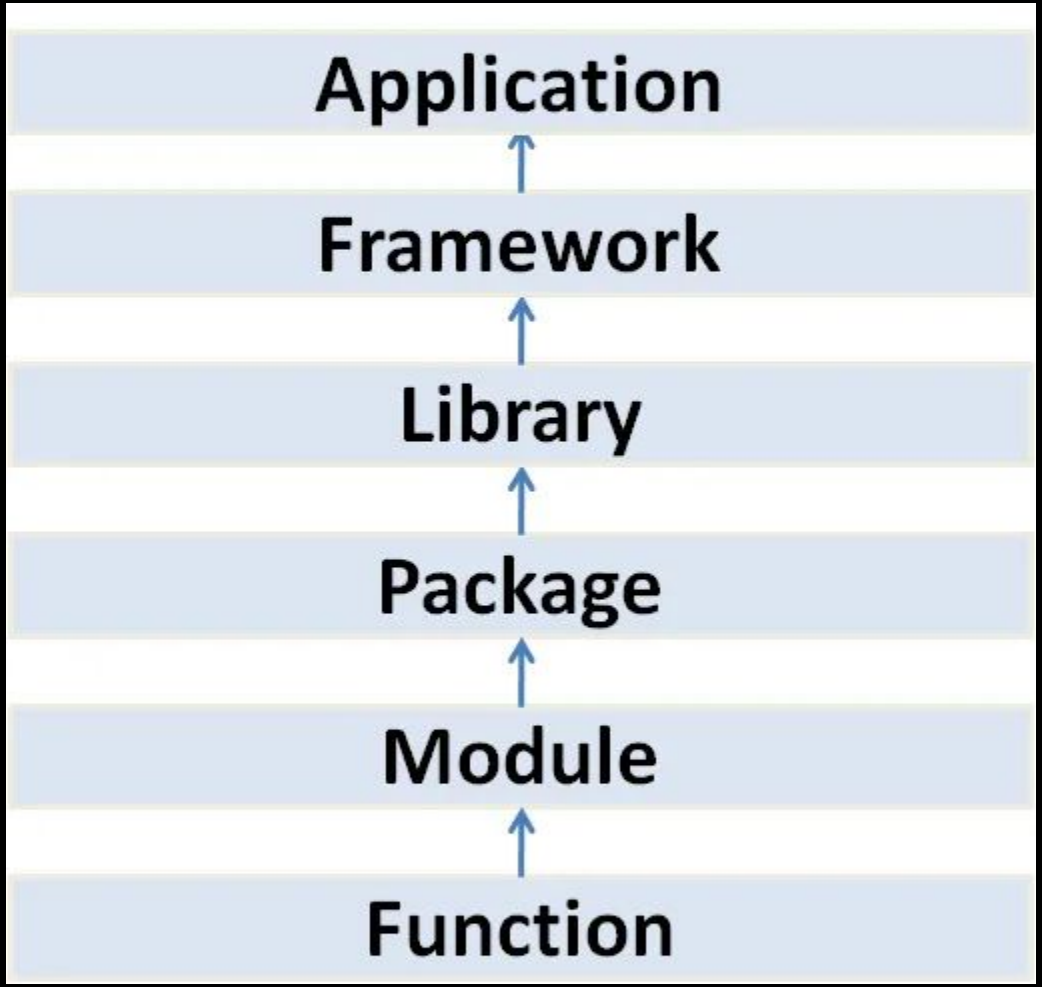
- FULL-STACK:
    - HAVE ALL WEB DEV REQUIREMENTS
        - FORM GENERATORS, FORM VALIDATION, TEMPLATE LAYOUTS
        - CORE FEATURES OF ANY FULL-STACK FRAMEWORK
- MICRO
    - REQUIRE LOTS OF CODE AND ADDITIONAL REQUIREMENTS TO BE ADDED MANUALLY
        - DOESN'T PROVIDE THE SPECIFIC TOOLS PROVIDED BY FULL-STACK FRAMEWORKS
- ASYNCHRONOUS
    - SUPPORTS HIGH CONCURRENCY
        - LARGE SET OF CONCURRENT CONNECTIONS
    - USES ASYNCIO LIBRARY TO RUN THE PROCESS

POPULAR FRAMEWORKS

- FLASK
  - A MICRO FRAMEWORK THROUGH WHICH DEVS CAN BUILD A SOLID WEB APP FOUNDATION FROM
    - LIGHTWEIGHT/MODULAR DESIGN, READILY ADAPTABLE
- DJANGO
  - A FULL-STACK FRAMEWORK, KNOWN FOR HELPING DEVELOP RICH WEB APPS
    - BUILT-IN LIBRARIES, FREE-TO-USE FEATURES, DATABASE SUPPORT
- BOTTLE
  - A MICRO FRAMEWORK, CREATES A SINGLE SOURCE FILE FOR EVERY DEVELOPED APP
    - API DEV, NO DEPENDENCIES, BUILT-IN HTTPS SERVER, DATABASE SUPPORT
- CHERRYPY
  - OPEN-SOURCE, OOP, MICRO FRAMEWORK
    - ONE OF THE OLDEST PYTHON FRAMEWORKS
    - RUNS ON ANDROID, FLEXIBILITY, ROBUST CONFIGURATION SYSTEM, ETC
- FALCON
  - A MICRO FRAMEWORK USED TO BUILD WEB APIS
    - NEEDS DEPENDENCIES, USED WIDELY AT LINKEDIN, OPENSTACK, ETC.
    - UPFRONT EXCEPTION HANDLING, HIGHLY OPTIMIZED CODE BASE, EXTENSIBLE

KNOWING THE DIFFERENCE

- PACKAGES AND LIBRARIES ARE OFTEN USED INTERCHANGEABLY. HOWEVER, IT IS IMPORTANT TO UNDERSTAND THE DIFFERENCE BETWEEN THEM.
- GENERALLY SPEAKING:
  - A PACKAGE IS A COLLECTION OF MODULES
  - A LIBRARY IS A COLLECTION OF PACKAGES
- ALL OF THE TERMS DISCUSSED SO FAR (FUNCTION, MODULE, PACKAGE, LIBRARY, AND FRAMEWORK) ARE ULTIMATELY COGS WORKING TOGETHER, HAND-IN-HAND, TO CREATE AN APPLICATION.

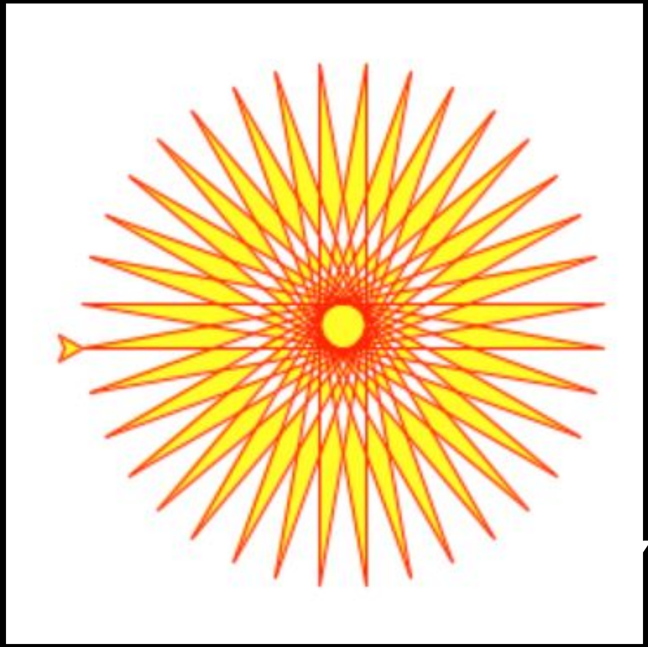


SOURCE: SWATHI ARUN, HTTPS://MEDIUM.COM/PYTHONEERS/6-MUST-KNOW-WORDS-IN-PYTHON-AC87AB420AB7

LAB TIME!

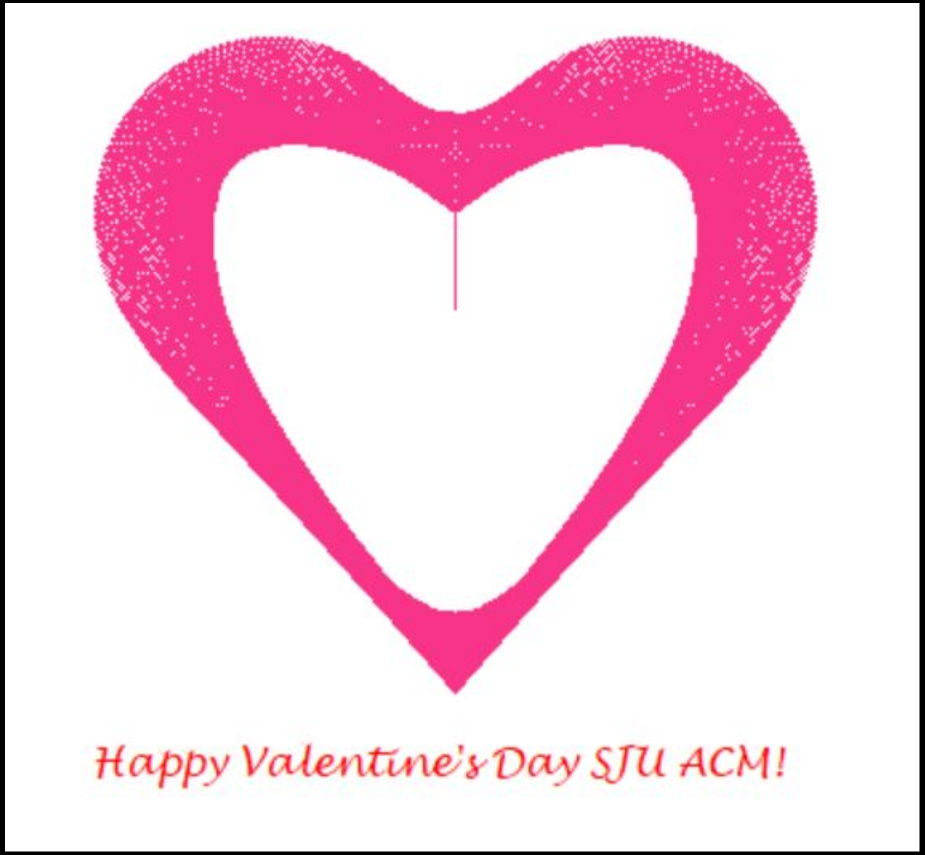
PRESENTING: PYTHON TURTLE!

- PYTHON TURTLE IS A STANDARD LIBRARY DESIGNED TO MAKE LEARNING PROGRAMMING CONCEPTS ENJOYABLE.
  - IDEAL FOR BEGINNERS
  - USES VIRTUAL TURTLE THAT CAN BE CONTROLLED TO DRAW SHAPES AND COMPLEX PATTERNS
- IT IS AN IMPLEMENTATION OF THE EPONYMOUS GEOMETRIC DRAWING TOOLS DEVELOPED FOR THE LOGO PROGRAMMING LANGUAGE.
  - DEVELOPED BY WALLY FEURZEIG, SEYMOUR PAPERT, AND CYNTHIA SOLOMON IN 1967
- TO USE TURTLE:
  - ENSURE IT IS INSTALLED ON YOUR MACHINE (USE PIP INSTALL PYTHONTURTLE)



WHAT WILL WE BE DOING?

- WE WILL BE DRAWING A HEART FOR VALENTINE'S DAY!
  - MAKE SURE TO FOLLOW ALONG TO FIND OUT HOW YOU CAN MAKE THIS:



PYTHON RESOURCES

PYTHON RESOURCES

- PYPI
  - HTTPS://PYPI.ORG/

- PYPI STATS
  - HTTPS://PYPISTATS.ORG/

- PYTHON - OFFICIAL WEBSITE
  - HTTPS://WWW.PYTHON.ORG/

- PYTHON DOCUMENTATION
  - HTTPS://DOCS.PYTHON.ORG/3/

- PYTHON TURTLE DOCUMENTATION
  - HTTPS://DOCS.PYTHON.ORG/3/LIBRARY/TURTLE.HTML

- THE BEGINNER’S GUIDE TO PYTHON TURTLE - REAL PYTHON
  - HTTPS://REALPYTHON.COM/BEGINNERS-GUIDE-PYTHON-TURTLE/

- AWS SDK FOR PYTHON (BOTO3)
  - HTTPS://AWS.AMAZON.COM/SDK-FOR-PYTHON/

PYTHON RESOURCES

- DEEP DIVE: CREATE AND PUBLISH YOUR FIRST PYTHON LIBRARY - JOFFREY BIENVENU
  - HTTPS://TOWARDSDATASCIENCE.COM/DEEP-DIVE-CREATE-AND-PUBLISH-YOUR-FIRST-PYTHON-LIBRARY-F7F618719E14

- DIFFERENCE BETWEEN PYTHON MODULES, PACKAGES, LIBRARIES, AND FRAMEWORKS - KATERYNA KOIDAN
  - HTTPS://LEARNPYTHON.COM/BLOG/PYTHON-MODULES-PACKAGES-LIBRARIES-FRAMEWORKS/

THANK YOU!