IT Automation with Google

1. Crash Course on Python.

- **Regex Cheat Sheet**
  i. \d - matches a single digit character [0-9]
  ii. \w - matches any alphabet, digit, or underscore
  iii. \s - matches a white space character (space, tab, enter)
  iv. . - matches any character (except for newline character)
  v. ^ - the string starts with a character
  vi. $ - the string ends with a character
  vii. * - zero or more occurrences
  viii. + - one or more occurrences
  ix. ? - one or no occurrence
  x. {} - exactly the specified number of occurrences
  xi. | - either or
  xii. [abcd] - matches either a, b, c or d
  xiii. [a-z0-9] - matches one of the characters from a-z or 0-9
  xiv. [\w] - matches an alphabet, digit, or underscore
  xv. findall() - Returns a list that contains all matches
  xvi. search() - Returns a 'match object' if there is a match in the string
  xvii. split() - Returns a list of string that has been split at each match
  xviii. sub() - Replaces the matches with a string
- **Dictionary Cheat sheet**
  i. .keys() returns the keys (the first object in the key-value pair),
  ii. .values() returns the values (the second object in the key-value pair), and
  iii. .items() returns both the keys and the values as a tuple.
  iv. get() method to access a dictionary value if it exists. This method takes the key as the first argument and an optional default value as the second argument, and it returns the value for the specified key if key is in the dictionary. If the second argument is not specified and key is not found then None is returned.
  v. .pop() method. The method takes a key as an argument and removes it from the dictionary. At the same time, it also returns the value that it removes from the dictionary.

2. Using Python to interact with Operating System.

a) IT Support launchpad for future carrier - *Research by Harvard B-school and Accenture Burning Glass*

i. SysAdmin
ii. Cloud Solution
iii. Devops Architect
iv. Site Reliability Engineer

b) Automation – Consistency, Repeated Task

i. Disadvantage – Not when flexibility is needed (Centralized Solution)
ii. Use **shebang** on the start line of the script **"# !usr/bin/env Python3"**
iii. This is done to define path for the language installed

c) Automation examples:

i. To check free disk space on the machine (Linux)

```
#!usr/bin/env/python3
Import shutil
Du = shutil.disk_usage("/")
Print(Du)
```

ii. To check free CPU usage on the machine

```
#!usr/bin/env/python3
Import psutil
Cpu_percent(0.1)
```

d) Working with File

i.
```
Import OS
Os.remove(<filename>)
Os.rename(<filename>)
Os.path.exist(<filename>) #Returns Boolean
Os.path.getsize(<filename>) #Size of the file
Os.path.getmtimestamp(<filename>)
Os.getcwd((<filename>) #Check working directory
Os.mkdir(<filename>)
Os.path.join("<dir>","<filename>")
```

ii. Cheatsheet 1
iii. Cheatsheet 2

e) Reading from CSV

i. Import csv
ii. CSV cheat sheet

f) Data Streams

- **Taking input from users**
  - i. To read data interactively, use input() module in python
  - ii. It passes input as String hence if number then use int(input())
- **Standard Streams**
  - i. Act as a path between input(keyboard) & output(Screen)
  - ii. Three types.
    - a. STDIN (Standard Input), between program and source of input (text data)
    - b. STDOUT (Standard Output), between program and source of display
    - c. STDERR (Standard Error), same as STDOUT, but for Error messages.
- **Environment Variables**
  - i. Env command to check variables
  - ii. Environ method of OS library, stored as dictionary.
  - iii. Use get method for eg: - OS.environ.get[“SHELL”,””]
  - iv. This doesn’t return error if value is not present
  - v. Eviron Cheatsheet
- **Command Line Arugments**
  - i. Uses sys module to run command line arguments in a python script
  - ii. Exit Status is checked by $?
  - iii. Used so that script is generic and less user interactive
  - iv. Subprocess modules used for Command line arguments in the script
  - v. Returncode to check exit status
  - vi. Subprocess Docs
- **Processing Log File**
  - i. Filtering Using Regex
  - ii. First step to open the Log File

g) Testing in Python.

- To find defects in code
- Manual Testing and Automated Testing
- Automatic Testing tests code as we write
- Unit test, used to test small, isolated parts of code like functions or method
- Unit test, important characteristics are isolation
- *Our code, should never modify production environment*
- Write automatic tests in Python
  - i.) Create test file with convention <samefilename_test.py>
  - ii.) Import block from script file using *from* method
  - iii.) Import unittest module from python, this includes numerous testcases.
  - iv.) Unit Testing Cheatsheet

- https://www.coursera.org/learn/python-operating-system/supplement/P5e0m/more-about-tests
-

3. Bash Scripting

a. Redirecting Streams

Managing streams

These are the redirectors that we can use to take control of the streams of our programs

- command **>** file: redirects standard output, overwrites file
- command **>>** file: redirects standard output, appends to file
- command **<** file: redirects standard input from file
- command **2>** file: redirects standard error to file
- command1 **|** command2: connects the output of command1 to the input of command2

Operating with processes

- **ps:** lists the processes executing in the current terminal for the current user
- **ps** ax: lists all processes currently executing for all users
- **ps** e: shows the environment for the processes listed
- **kill** PID: sends the SIGTERM signal to the process identified by PID
- **fg**: causes a job that was stopped or in the background to return to the foreground
- **bg:** causes a job that was stopped to go to the background
- **jobs:** lists the jobs currently running or stopped
- **top:** shows the processes currently using the most CPU time (press "q" to quit)

b. Bash Scripts

- Ends with .sh
- Shebang = #/bin/bash
- Echo to print messages
- Pass commands in $() for echo

c. Using Variables & Globs

- User=Kaushik
- No spaces in between
- Echo $User
- Globs = used to match files, like * matches all files, sequences of files ? matches exactly one file

d. Conditional Execution

- If <Condition>; then
  Do this
  Else
  Do this
  Fi # Mandatory in bash scripting to end condtions
- Shell Scripting - Guide

e. While Loops & For loops

- While | for [Condition]; do

  Task

  Done

- "Basename" takes the filename and extension and return only filename
- For Eg – basename index.html .html > output: index