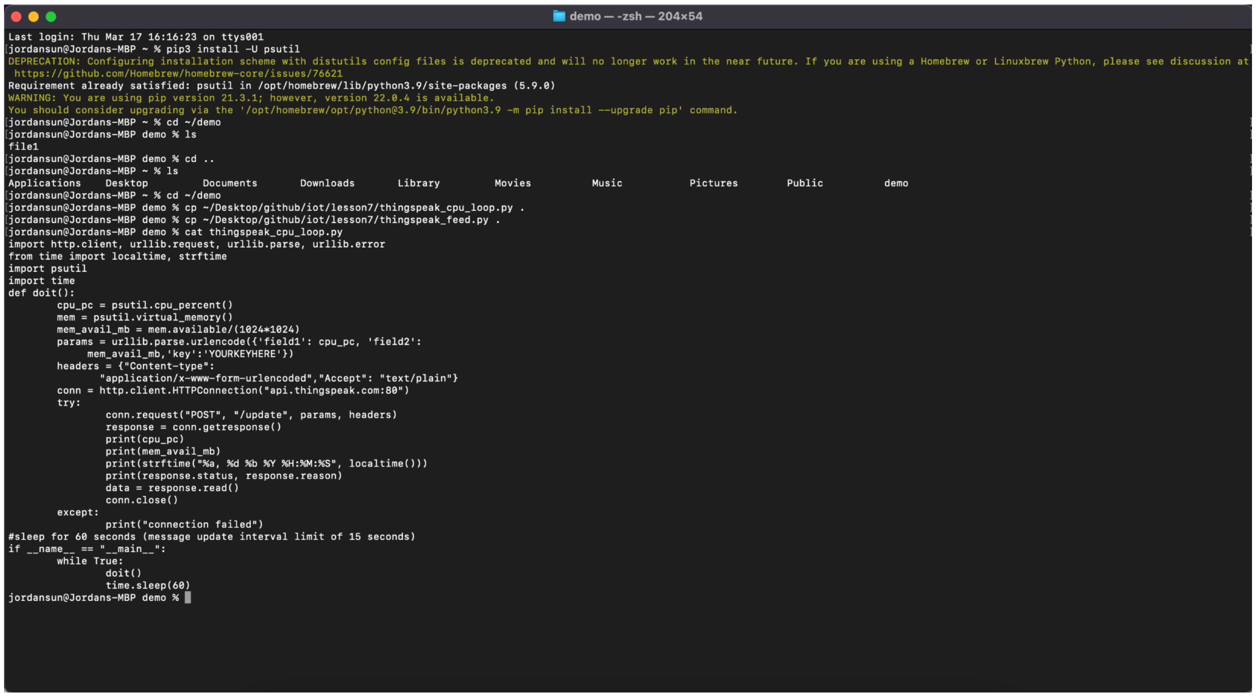
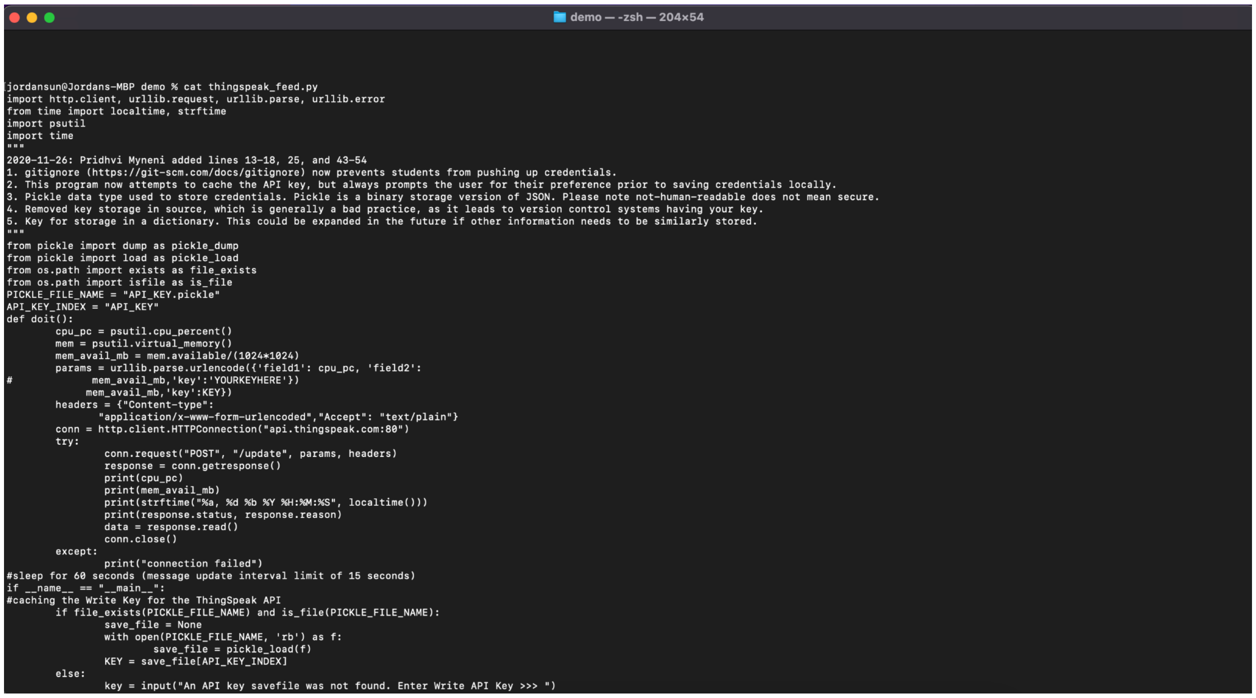
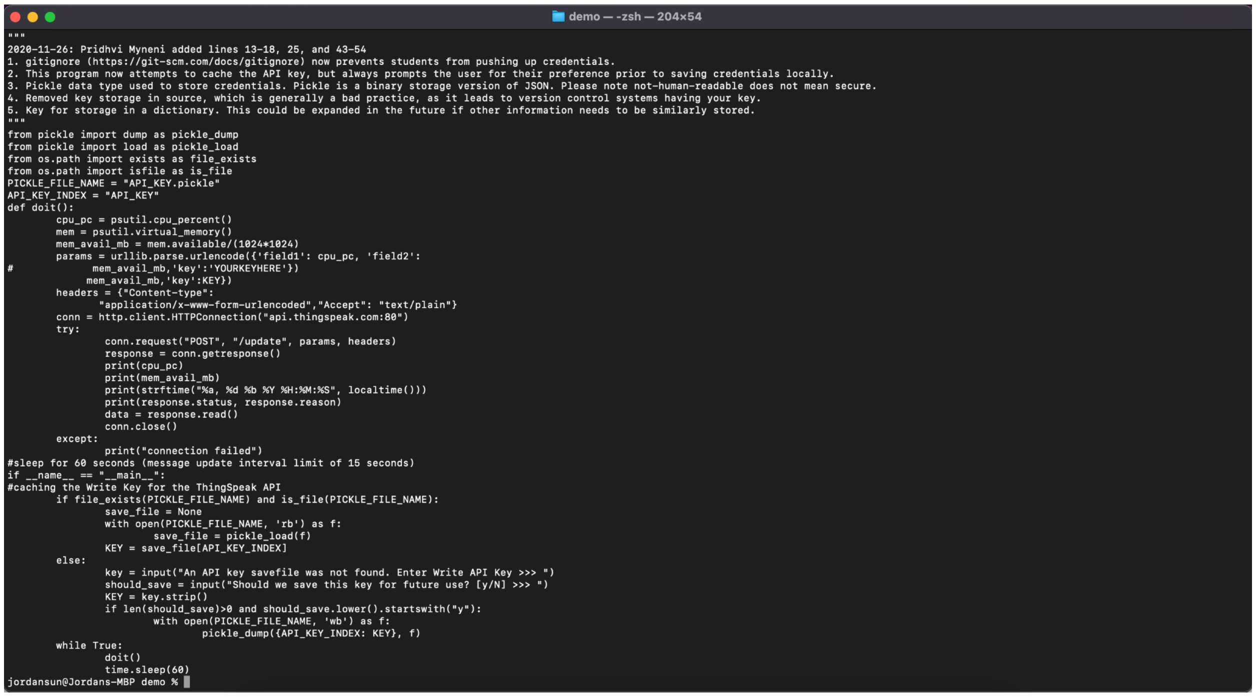
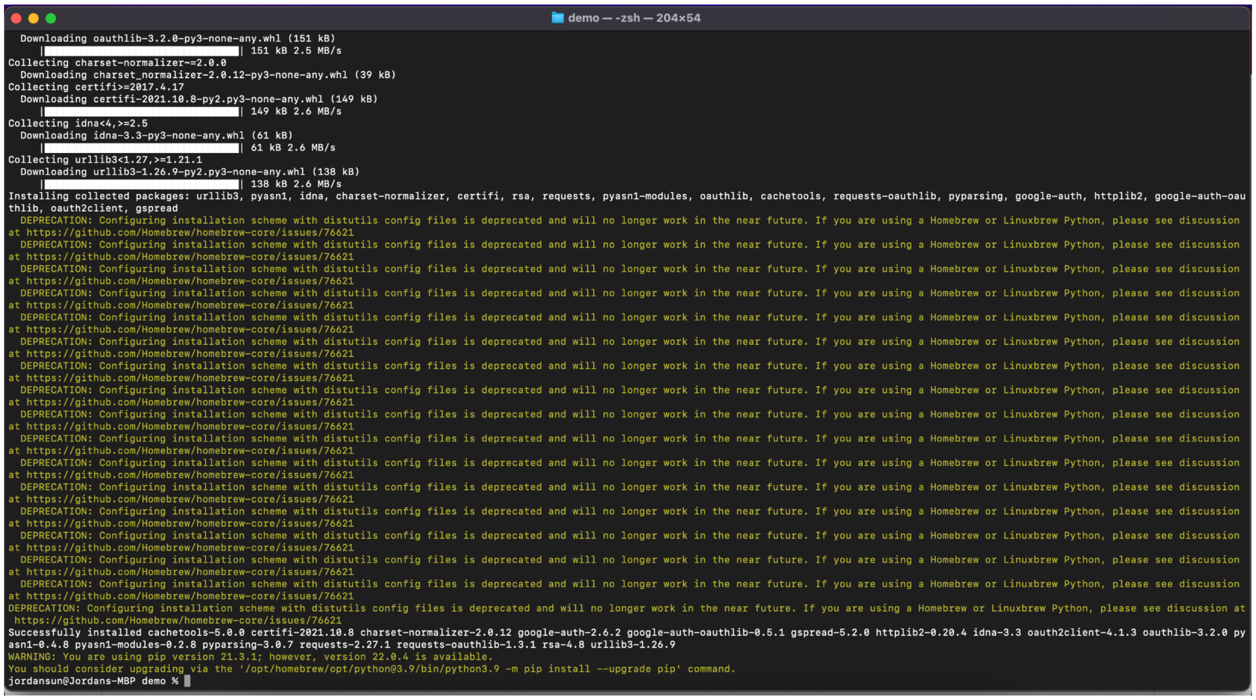


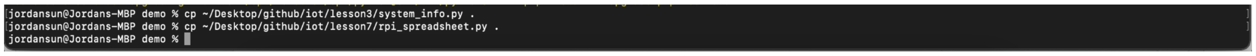
```
Last login: Thu Mar 17 16:16:23 on ttys001
jordansun@Jordans-MBP ~ % pip3 install -U psutil
DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion at
 https://github.com/Homebrew/homebrew-core/issues/76621
Requirement already satisfied: psutil in /opt/homebrew/lib/python3.9/site-packages (5.9.0)
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is available.
You should consider upgrading via the '/opt/homebrew/opt/python@3.9/bin/python3.9 -m pip install --upgrade pip' command.
jordansun@Jordans-MBP ~ % cd ~/demo
jordansun@Jordans-MBP demo % ls
file1
jordansun@Jordans-MBP demo % cd ..
jordansun@Jordans-MBP ~ % ls
Applications    Desktop         Documents       Downloads       Library         Movies          Music           Pictures        Public          demo
jordansun@Jordans-MBP ~ % cd ~/demo
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson7/thingspeak_cpu_loop.py .
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson7/thingspeak_feed.py .
jordansun@Jordans-MBP demo % cat thingspeak_cpu_loop.py
import http.client, urllib.request, urllib.parse, urllib.error
from time import localtime, strftime
import psutil
import time
def doit():
        cpu_pc = psutil.cpu_percent()
        mem = psutil.virtual_memory()
        mem_avail_mb = mem.available/(1024*1024)
        params = urllib.parse.urlencode({'field1': cpu_pc, 'field2':
             mem_avail_mb,'key':'YOURKEYHERE'})
        headers = {"Content-type":
               "application/x-www-form-urlencoded","Accept": "text/plain"}
        conn = http.client.HTTPConnection("api.thingspeak.com:80")
        try:
                conn.request("POST", "/update", params, headers)
                response = conn.getresponse()
                print(cpu_pc)
                print(mem_avail_mb)
                print(strftime("%a, %d %b %Y %H:%M:%S", localtime()))
                print(response.status, response.reason)
                data = response.read()
                conn.close()
        except:
                print("connection failed")
#sleep for 60 seconds (message update interval limit of 15 seconds)
if __name__ == "__main__":
        while True:
                doit()
                time.sleep(60)
jordansun@Jordans-MBP demo % █
```

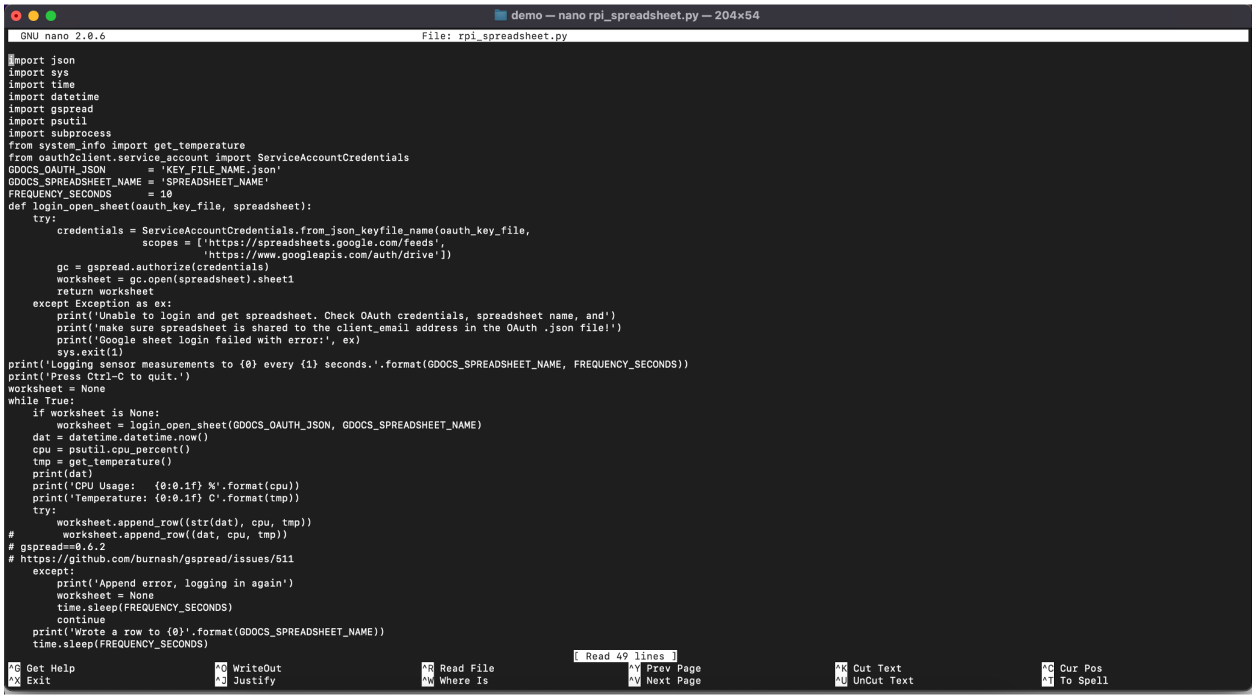


```
jordansun@Jordans-MBP demo % cat thingspeak_feed.py
import http.client, urllib.request, urllib.parse, urllib.error
from time import localtime, strftime
import psutil
import time
"""
2020-11-26: Pridhvi Myneni added lines 13-18, 25, and 43-54
1. gitignore (https://git-scm.com/docs/gitignore) now prevents students from pushing up credentials.
2. This program now attempts to cache the API key, but always prompts the user for their preference prior to saving credentials locally.
3. Pickle data type used to store credentials. Pickle is a binary storage version of JSON. Please note not-human-readable does not mean secure.
4. Removed key storage in source, which is generally a bad practice, as it leads to version control systems having your key.
5. Key for storage in a dictionary. This could be expanded in the future if other information needs to be similarly stored.
"""
from pickle import dump as pickle_dump
from pickle import load as pickle_load
from os.path import exists as file_exists
from os.path import isfile as is_file
PICKLE_FILE_NAME = "API_KEY.pickle"
API_KEY_INDEX = "API_KEY"
def doit():
        cpu_pc = psutil.cpu_percent()
        mem = psutil.virtual_memory()
        mem_avail_mb = mem.available/(1024*1024)
        params = urllib.parse.urlencode({'field1': cpu_pc, 'field2':
#             mem_avail_mb,'key':'YOURKEYHERE'})
             mem_avail_mb,'key':KEY})
        headers = {"Content-type":
               "application/x-www-form-urlencoded","Accept": "text/plain"}
        conn = http.client.HTTPConnection("api.thingspeak.com:80")
        try:
                conn.request("POST", "/update", params, headers)
                response = conn.getresponse()
                print(cpu_pc)
                print(mem_avail_mb)
                print(strftime("%a, %d %b %Y %H:%M:%S", localtime()))
                print(response.status, response.reason)
                data = response.read()
                conn.close()
        except:
                print("connection failed")
#sleep for 60 seconds (message update interval limit of 15 seconds)
if __name__ == "__main__":
#caching the Write Key for the ThingSpeak API
        if file_exists(PICKLE_FILE_NAME) and is_file(PICKLE_FILE_NAME):
                save_file = None
                with open(PICKLE_FILE_NAME, 'rb') as f:
                        save_file = pickle_load(f)
                KEY = save_file[API_KEY_INDEX]
        else:
                key = input("An API key savefile was not found. Enter Write API Key >>> ")
```


```
"""
2020-11-26: Pridhvi Myneni added lines 13-18, 25, and 43-54
1. gitignore (https://git-scm.com/docs/gitignore) now prevents students from pushing up credentials.
2. This program now attempts to cache the API key, but always prompts the user for their preference prior to saving credentials locally.
3. Pickle data type used to store credentials. Pickle is a binary storage version of JSON. Please note not-human-readable does not mean secure.
4. Removed key storage in source, which is generally a bad practice, as it leads to version control systems having your key.
5. Key for storage in a dictionary. This could be expanded in the future if other information needs to be similarly stored.
"""
from pickle import dump as pickle_dump
from pickle import load as pickle_load
from os.path import exists as file_exists
from os.path import isfile as is_file
PICKLE_FILE_NAME = "API_KEY.pickle"
API_KEY_INDEX = "API_KEY"
def doit():
        cpu_pc = psutil.cpu_percent()
        mem = psutil.virtual_memory()
        mem_avail_mb = mem.available/(1024*1024)
        params = urllib.parse.urlencode({'field1': cpu_pc, 'field2':
#             mem_avail_mb,'key':'YOURKEYHERE'})
             mem_avail_mb,'key':KEY})
        headers = {"Content-type":
               "application/x-www-form-urlencoded","Accept": "text/plain"}
        conn = http.client.HTTPConnection("api.thingspeak.com:80")
        try:
                conn.request("POST", "/update", params, headers)
                response = conn.getresponse()
                print(cpu_pc)
                print(mem_avail_mb)
                print(strftime("%a, %d %b %Y %H:%M:%S", localtime()))
                print(response.status, response.reason)
                data = response.read()
                conn.close()
        except:
                print("connection failed")
#sleep for 60 seconds (message update interval limit of 15 seconds)
if __name__ == "__main__":
#caching the Write Key for the ThingSpeak API
        if file_exists(PICKLE_FILE_NAME) and is_file(PICKLE_FILE_NAME):
                save_file = None
                with open(PICKLE_FILE_NAME, 'rb') as f:
                        save_file = pickle_load(f)
                KEY = save_file[API_KEY_INDEX]
        else:
                key = input("An API key savefile was not found. Enter Write API Key >>> ")
                should_save = input("Should we save this key for future use? [y/N] >>> ")
                KEY = key.strip()
                if len(should_save)>0 and should_save.lower().startswith("y"):
                        with open(PICKLE_FILE_NAME, 'wb') as f:
                                pickle_dump({API_KEY_INDEX: KEY}, f)
        while True:
                doit()
                time.sleep(60)
jordansun@Jordans-MBP demo % 
```

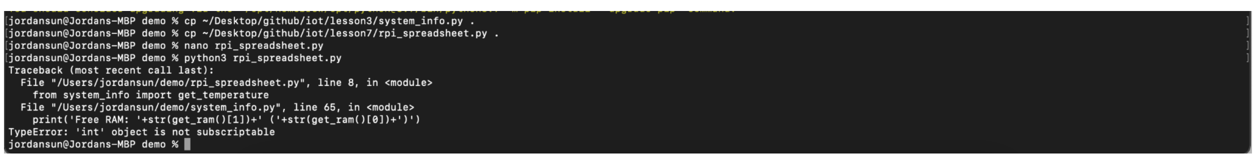

```
  Downloading oauthlib-3.2.0-py3-none-any.whl (151 kB)
     |████████████████████████████████| 151 kB 2.5 MB/s
Collecting charset-normalizer~=2.0.0
  Downloading charset_normalizer-2.0.12-py3-none-any.whl (39 kB)
Collecting certifi>=2017.4.17
  Downloading certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
     |████████████████████████████████| 149 kB 2.6 MB/s
Collecting idna<4,>=2.5
  Downloading idna-3.3-py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 2.6 MB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.9-py2.py3-none-any.whl (138 kB)
     |████████████████████████████████| 138 kB 2.6 MB/s
Installing collected packages: urllib3, pyasn1, idna, charset-normalizer, certifi, rsa, requests, pyasn1-modules, oauthlib, cachetools, requests-oauthlib, pyparsing, google-auth, httplib2, google-auth-oau
thlib, oauth2client, gspread
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
  DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion
at https://github.com/Homebrew/homebrew-core/issues/76621
DEPRECATION: Configuring installation scheme with distutils config files is deprecated and will no longer work in the near future. If you are using a Homebrew or Linuxbrew Python, please see discussion at
 https://github.com/Homebrew/homebrew-core/issues/76621
Successfully installed cachetools-5.0.0 certifi-2021.10.8 charset-normalizer-2.0.12 google-auth-2.6.2 google-auth-oauthlib-0.5.1 gspread-5.2.0 httplib2-0.20.4 idna-3.3 oauth2client-4.1.3 oauthlib-3.2.0 py
asn1-0.4.8 pyasn1-modules-0.2.8 pyparsing-3.0.7 requests-2.27.1 requests-oauthlib-1.3.1 rsa-4.8 urllib3-1.26.9
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is available.
You should consider upgrading via the '/opt/homebrew/opt/python@3.9/bin/python3.9 -m pip install --upgrade pip' command.
jordansun@Jordans-MBP demo % 
```


```
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson3/system_info.py .
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson7/rpi_spreadsheet.py .
jordansun@Jordans-MBP demo % 
```



```
GNU nano 2.0.6                                    File: rpi_spreadsheet.py

import json
import sys
import time
import datetime
import gspread
import psutil
import subprocess
from system_info import get_temperature
from oauth2client.service_account import ServiceAccountCredentials
GDOCS_OAUTH_JSON       = 'KEY_FILE_NAME.json'
GDOCS_SPREADSHEET_NAME = 'SPREADSHEET_NAME'
FREQUENCY_SECONDS      = 10
def login_open_sheet(oauth_key_file, spreadsheet):
    try:
        credentials = ServiceAccountCredentials.from_json_keyfile_name(oauth_key_file,
                      scopes = ['https://spreadsheets.google.com/feeds',
                                'https://www.googleapis.com/auth/drive'])
        gc = gspread.authorize(credentials)
        worksheet = gc.open(spreadsheet).sheet1
        return worksheet
    except Exception as ex:
        print('Unable to login and get spreadsheet. Check OAuth credentials, spreadsheet name, and')
        print('make sure spreadsheet is shared to the client_email address in the OAuth .json file!')
        print('Google sheet login failed with error:', ex)
        sys.exit(1)
print('Logging sensor measurements to {0} every {1} seconds.'.format(GDOCS_SPREADSHEET_NAME, FREQUENCY_SECONDS))
print('Press Ctrl-C to quit.')
worksheet = None
while True:
    if worksheet is None:
        worksheet = login_open_sheet(GDOCS_OAUTH_JSON, GDOCS_SPREADSHEET_NAME)
    dat = datetime.datetime.now()
    cpu = psutil.cpu_percent()
    tmp = get_temperature()
    print(dat)
    print('CPU Usage:   {0:0.1f} %'.format(cpu))
    print('Temperature: {0:0.1f} C'.format(tmp))
    try:
        worksheet.append_row((str(dat), cpu, tmp))
#        worksheet.append_row((dat, cpu, tmp))
# gspread==0.6.2
# https://github.com/burnash/gspread/issues/511
    except:
        print('Append error, logging in again')
        worksheet = None
        time.sleep(FREQUENCY_SECONDS)
        continue
    print('Wrote a row to {0}'.format(GDOCS_SPREADSHEET_NAME))
    time.sleep(FREQUENCY_SECONDS)
                                                                                               [ Read 49 lines ]
^G Get Help          ^O WriteOut          ^R Read File          ^Y Prev Page          ^K Cut Text          ^C Cur Pos
^X Exit              ^J Justify           ^W Where Is           ^V Next Page          ^U UnCut Text        ^T To Spell
```



```
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson3/system_info.py .
jordansun@Jordans-MBP demo % cp ~/Desktop/github/iot/lesson7/rpi_spreadsheet.py .
jordansun@Jordans-MBP demo % nano rpi_spreadsheet.py
jordansun@Jordans-MBP demo % python3 rpi_spreadsheet.py
Traceback (most recent call last):
  File "/Users/jordansun/demo/rpi_spreadsheet.py", line 8, in <module>
    from system_info import get_temperature
  File "/Users/jordansun/demo/system_info.py", line 65, in <module>
    print('Free RAM: '+str(get_ram()[1])+' ('+str(get_ram()[0])+')')
TypeError: 'int' object is not subscriptable
jordansun@Jordans-MBP demo % 
```