

Python Automation

Andrew Calder & Corey Forbes 20/11/2018

Announcements

- Discord
- Feedback
- Talks/meetings/workshops/projects/etc to present
 - Currently have nothing for next week

From last week:

```
Come pick from the books

3rd place: - (20 points)

2nd place: - (30 points)

1st place: - (67.89 points)
```

Counting Characters & words

Counting Chars + Words

```
#!/usr/bin/env python3
import tkinter as tk
import os
window = tk.Tk();
window.withdraw()
# Prints number of chracters and words in current selection
characters = len(window.selection_get())
words = len(window.selection_get().split())
print("characters: %s" % characters)
print("words: %s" % words)
os.system("notify-send 'wccount' 'words: %s, Characters %s'" % (words, characters))
# https://github.com/AR-Calder/PythonAutomation
```

Write a Cover letter & save as pdf

Writing a cover letter and saving as pdf

```
[andrew@localhost coverpy]$ cat basictemplate.txt
Dear {{tname}},
I wish to apply for the post of {{position}} advertised recently on {{source}}.
I am a {{study year}} year {{course}} student at {{university}} University currently tracking towards an First Class Honours degree with distinction.
A copy of my CV is enclosed for your perusal.
I look forward to hearing from you.
Best wishes,
{{vname}}
[andrew@localhost coverpy]$ ./coverpy.py -file basictemplate.txt -yname "Andrew Calder" -tname "Recruiter McRecruiterFace"
Enter value for 'position': Automation Engineer
Enter value for 'source': LinkedIn
Enter value for 'study year': 4th
Enter value for 'course': Ethical Hacking
Enter value for 'university': Abertay
Finished writing cover letter. That tooks aaaages ;)
Dear Recruiter McRecruiterFace,
I wish to apply for the post of Automation Engineer advertised recently on LinkedIn.
I am a 4th year Ethical Hacking student at Abertay University currently tracking towards an First Class Honours degree with distinction.
A copy of my CV is enclosed for your perusal.
I look forward to hearing from you.
Best wishes,
Andrew Calder
```

Creating
N random
strings of
X length

Creating N random strings of x length

```
#!/usr/bin/env python3
import argparse
import string
import random
def str generator(size=4, chars=string.ascii letters + string.digits):
  return (".join(random.choice(chars) for in range(size))) + '\n'
# Arg Parsing
parser = argparse.ArgumentParser()
group = parser.add_argument_group("Required Arguments")
group.add_argument("-l", "--patternlength", nargs='?', default=5, required=True)
group.add argument("-c", "--count", nargs='?', default=4, required=True)
args = parser.parse known args()[0]
with open("len%s count%s.txt" % (args.patternlength, args.count), "w") as outfile:
  for line in range(int(args.count)):
    outfile.write(str_generator(size=int(args.patternlength)))
# https://github.com/AR-Calder/PythonAutomation
```

Automate the things

General automation tips

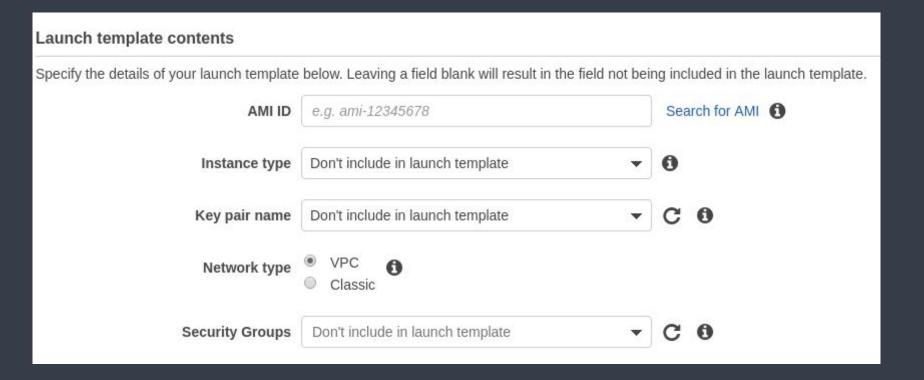
```
# Get output directly from a command
subprocess.run("command", stdout=subprocess.PIPE).stdout.decode('utf-8')

# open a file
with open("filename.extension") as f:
    For line in f: # Go through each line in a file

# Regex
# Makes many things easier
entries = re.findall("{{(.*?)}}", f) # find all words in file surrounded by {{word}}
```

Automated Testing w/ AWS

Setup outside the Code



AWS connectivity via Boto3

```
#!/usr/bin/env python3
import sys
import boto3
from botocore.exceptions import ClientError
templateID = 'lt-0857d785b7ca2aefd'
client = boto3.client('ec2')
# Dry run to make sure I have permissions
try:
    client.run_instances(MaxCount=5, MinCount=5, DryRun=True,
                         LaunchTemplate={
                              'LaunchTemplateId': 'lt-0857d785b7ca2aefd',
                              'Version': '2'})
except ClientError as e:
    if 'DryRunOperation' not in str(e):
        print("You don't have permission to reboot instances.")
        raise
```

AWS connectivity via Boto3

After Parsing

```
Success {'Groups': [], 'Instances': [{'AmiLaunchIndex': 0, 'ImageId': 'ami-0274e11dced17bb5b', 'InstanceId':
'i-0c767e03d88d6ea2e', 'InstanceType': 't2.nano', 'LaunchTime': datetime.datetime(2018, 11, 20, 16, 59, 2,
tzinfo=tzutc()), 'Monitoring': {'State': 'disabled'}, 'Placement': {'AvailabilityZone': 'eu-west-2a',
'GroupName': '', 'Tenancy': 'default'}, 'PrivateDnsName': 'ip-172-31-31-250.eu-west-2.compute.internal',
'PrivateIpAddress': '172.31.31.250', 'ProductCodes': [], 'PublicDnsName': '', 'State': {'Code': 0, 'Name':
'pending'}, 'StateTransitionReason': '', 'SubnetId': 'subnet-12eb4d68', 'VpcId': 'vpc-afe878c7',
'Architecture': 'x86_64', 'BlockDeviceMappings': [], 'ClientToken': '', 'EbsOptimized': False, 'Hypervisor':
'xen', 'NetworkInterfaces': [{'Attachment': {'AttachTime': datetime.datetime(2018, 11, 20, 16, 59, 2,
tzinfo=tzutc()), 'AttachmentId': 'eni-attach-065e48ad1cd21c9e9', 'DeleteOnTermination': True, 'DeviceIndex':
0, 'Status': 'attaching'}, 'Description': '', 'Groups': [{'GroupName': 'default', 'GroupId': 'sg-3cb83456'}],
'Ipv6Addresses': [], 'MacAddress': '06:53:99:ea:1c:1a', 'NetworkInterfaceId': 'eni-069c7d43ac6049e4e',
'OwnerId': '809499123637', 'PrivateDnsName': 'ip-172-31-3<u>1-250.eu-west-2.compute.internal'.</u>
'PrivateIpAddress': '172.31.31.250', 'PrivateIpAddresses': [{'Primary': True, 'PrivateDnsName':
'ip-172-31-31-250.eu-west-2.compute.internal', 'PrivateIpAddress': '172.31.31.250'}], 'SourceDestCheck':
True, 'Status': 'in-use', 'SubnetId': 'subnet-12eb4d68', 'VpcId': 'vpc-afe878c7'}], 'RootDeviceName':
'/dev/xvda', 'RootDeviceType': 'ebs', 'SecurityGroups': [{'GroupName': 'default', 'GroupId': 'sg-3cb83456'}],
'SourceDestCheck': True, 'StateReason': {'Code': 'pending', 'Message': 'pending'}, 'Tags': [{'Key':
'aws:ec2launchtemplate:id', 'Value': 'lt-0857d785b7ca2aefd'}, {'Key': 'aws:ec2launchtemplate:version',
'Value': '2'}, {'Key': 'Testing', 'Value': 'Expendable'}], 'VirtualizationType': 'hvm', 'CpuOptions':
{'CoreCount': 1, 'ThreadsPerCore': 1}}], 'OwnerId': '809499123637', 'ReservationId': 'r-075234c8d2af3b298',
'ResponseMetadata': {'RequestId': '0d123a54-a003-44b0-ae44-0ac84a642be2', 'HTTPStatusCode': 200,
'HTTPHeaders': {'content-type': 'text/xml;charset=UTF-8', 'content-length': '4632', 'vary':
```

Any questions?