PYTHON

1.2.6.1: Read an External File

In addition to user input, you can access a database, another computer program, or a file to provide input to your program. The **open()** function can be used to access a file using the following syntax:

```
open(name, [mode])
```

The *name* parameter is the name of the file to be opened. If the file is in a different directory than your script, you will also need to provide path information. For our purposes, we are only interested in three *mode* parameters:

- **r** - read the file (default mode if mode is omitted).
- **w** - write over the file, replacing the content of the file.
- **a** - append to the file.

Complete the following steps to read and print a file:

1. Open a blank script and save it as **07_file-access.py**.
2. Create a script to read and print the content of a file, as shown in Example 1.
3. The **devices.txt** file should be in the same directory as your script.
4. After printing the contents of the file, use the **close()** function to remove it from the computer's memory.

Note: The contents of the file are set to a variable named **file**. However, that variable can be called anything the programmer chooses.

Example 1: Read and Print a File

```
file=open("devices.txt","r")
for item in file:
    print(item)
file.close()
```

Run the script and troubleshoot, if necessary. Your output should look like Example 2.

Example 2: Output for the Script 07_file-access.py

```
========== RESTART: /home/user/Documents/GitHub/07_file-access.py ==========
Cisco 819 Router

Cisco 881 Router

Cisco 888 Router

Cisco 1100 Router

Cisco 4321 Router

Cisco 4331 Router

Cisco 4351 Router

Cisco 2960 Catalyst Switch

Cisco 3850 Catalyst Switch

Cisco 7700 Nexus Switch

Cisco Meraki MS220-8 Cloud Managed Switch

Cisco Meraki MX64W Security Appliance

Cisco Meraki MX84 Security Appliance

Csico Meraki MC74 VoIP Phone

Cisco 3860 Catalyst Switch
```

1.2.6.2: Remove Blank Lines from the Output

You may have noticed that Python added a blank line after each entry. We can remove this blank line using the **strip()** method. Edit your **07_file-access.py** script as shown in Example 1.

Example 1: Stripping the Blank Line

```
file=open("devices.txt","r")
for item in file:
   item=item.strip()
   print(item)
file.close()
```

Run and, if necessary, troubleshoot your script. Your output should look like Example 2.

Example 2: Output for the Script 07_file-access.py

```
========== RESTART: /home/user/Documents/GitHub/07_file-access.py ==========
Cisco 819 Router
Cisco 881 Router
Cisco 888 Router
Cisco 1100 Router
Cisco 4321 Router
Cisco 4331 Router
Cisco 4351 Router
Cisco 2960 Catalyst Switch
Cisco 3850 Catalyst Switch
Cisco 7700 Nexus Switch
Cisco Meraki MS220-8 Cloud Managed Switch
Cisco Meraki MX64W Security Appliance
Cisco Meraki MX84 Security Appliance
Csico Meraki MC74 VoIP Phone
Cisco 3860 Catalyst Switch
>>>
```

1.2.6.3: Copy File Content Into a List Variable

Most of the time when programmers access an external resource such as a database or file, they are wanting to copy that content into a local variable that can then be referenced and manipulated without impacting the original resource.

The **devices.txt** file is a list of Cisco devices that can easily be copied into a Python list using the following steps:

1. Create an empty list.
2. Use the **append** parameter to copy file content to the new list.

Modify your **07_file-access.py** as shown in Example 1.

Example 1: Copying a File into a List

```
devices=[]
file=open("devices.txt","r")
for item in file:
   item=item.strip()
   devices.append(item)
file.close()
print(devices)
```

Run and, if necessary, troubleshoot your script. Your output should look like Example 2.

Example 2: Output for the Script 07_file-access.py

```
========== RESTART: /home/user/Documents/GitHub/07_file-access.py ==========
['Cisco 819 Router', 'Cisco 881 Router', 'Cisco 888 Router', 'Cisco 1100 Router', 'Cisco 4321 Ro
uter', 'Cisco 4331 Router', 'Cisco 4351 Router', 'Cisco 2960 Catalyst Switch', 'Cisco 3850 Catal
yst Switch', 'Cisco 7700 Nexus Switch', 'Cisco Meraki MS220-8 Cloud Managed Switch', 'Cisco Mera
ki MX64W Security Appliance', 'Cisco Meraki MX84 Security Appliance', 'Csico Meraki MC74 VoIP Ph
one', 'Cisco 3860 Catalyst Switch']
```

1.2.6.4: Activity – Create a Script to Allow User to Add Devices

What if you want to add more devices to the **devices.txt** file? You can open the file in append mode and then ask the user to provide the name of the new devices. Complete the following steps to create a script:

1. Open a new file and save it as **07_file-access_actvity.py**.
2. For the **open()** function use the mode **a**, which will allow you to append a item to the **devices.txt** file.
3. Inside a **while True:** loop, embed an **input()** function command that asks the user for the new device.
4. Set the value of the user's input to a variable named **newItem**.
5. Use an if statement that breaks the loop if the user types **exit** and prints the statement **"All done!"**.
6. Use the command **file.write(newItem + "\n")** to add the new user provided device.

Run and troubleshoot your script until you get output similar to Example 1.

Example 1: Output for Script 07_file-access_activity.py

```
==== RESTART: /home/user/Documents/GitHub/07_file-access_sol_activity.py ====
Enter device name: Cisco 1941 Router
Enter device name: Cisco 2950 Catalyst Switch
Enter device name: exit
All done!
>>>
```