Network Automation

Introduction

https://w3schools.com

https://docs.python.org/3/

Topics of the week

- What is automation?
- Tools that integrate automation
- Why Python?
- Some Python applications
- Python downloads
- Network Automation framework
- Introduction to Python
- Building simple Python application
- Simple Python input/output and variables
- Input and output functions in Python

What is Network Automation?

- Repetitive administrative tasks
- Updating operating system versions
- Monitoring and Messaging automation
- Configuration of network devices
- Troubleshooting networks
- Backup and recovery automation
- Automating web logging processing

Tools that Integrate Automation

- Operating Systems' features
- Python programming environment
- Django (python) to develop web applications
- Ansible playbooks
- Teraform
- Selinium (web logging automation)
- Puppet for Python
- PyChef for Chef server automation

Why Python?

- Open source
- Easy to learn and use with simplicity in syntax
- Supports number of third party libraries
- Evolved with mature and supportive community of both professionals and hobbyists
- Supported APIs by renowned corporate sponsors (like Facebook, Amazon AWS, Google, Youtube etc.)
- Easy to write scripts
- Structured, Object Oriented language and script
- Supports graphical and text input and output
- Number of modules to connect operating systems directly and perform operating system functions
- SMTP API

Why Python (continued)?

- Huge number of Python Libraries and Framework that is increasing everyday (Paramiko, Netmiko, Matplotlib, SciPy, BeautifulSoup, NumPy, Django)
- Versatility, Efficiency, Reliability and Speed
- Big data, Machine Learning, Big data analytics and Data Science and Cloud Computing
- Automation
- First choice of language (academia and professional business related applications)

Some Python Applications

- Simple Network configuration
- Network device log file
- SMTP email application
- Operating System application

Interpretted vs. Compiled languages

- Scripting languages (e.g., sh, bash, PHP, Python, Javascript)
- Compiled languages (e.g. Java, C, C++)

- Python is interpreted scripting language

- Compiled languages are faster when executed

Tools to install

- Python interpreter – (python.org)
- Python documentation - https://docs.python.org/3/

- Integrated Development Environment (IDE) – Pycharm by jetbrains (jetbrains.com)
- Download pycharm community edition

https://www.jetbrains.com/pycharm/download/#section=windows

Python (partial) list of data types

- Character and String data type: str
- Numeric data types: int float complex
- Sequence type: list, tuple and range
- Mapping type: dict (dictionary)
- Set type: set frozenset
- Boolean type: bool
- Binary types: byte bytearray memoryview

Variables get their data types based on the value assigned to the variable

Examples of data types (w3schools.com)

Example	Data Type
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set
`x = frozenset({"apple", "banana", "cherry"})`	frozenset
`x = True`	bool
`x = b"Hello"`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview

Assigning (different datatype) values to variables

Python Data Types

w3schools.com/python/python_datatypes.asp

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP HOW TO W3.CSS JQUERY

Setting the Specific Data Type

If you want to specify the data type, you can use the following constructor functions:

Example

```
x = str("Hello World")
x = int(20)
x = float(20.5)
x = complex(1j)
x = list(("apple", "banana", "cherry"))
x = tuple(("apple", "banana", "cherry"))
x = range(6)
x = dict(name="John", age=36)
x = set(("apple", "banana", "cherry"))
x = frozenset(("apple", "banana", "cherry"))
x = bool(5)
x = bytes(5)
x = bytearray(5)
x = memoryview(bytes(5))
```

Python documentation, comments and conventions

- Multi-line comments
- Single line comments (or in-line comments)
- doc-string (__doc__)
- Documentation conventions

Strings and sub-strings

- Sequence of characters saved in contiguous memory locations ending with end-of-string character
- ‘This is string’ – 14 character including two spaces
- str1 = “This is string” – len(str1) will result in an integer value 14
- filename=“myfile.txt” – print (filename) will print myfile.txt
- filename=“myfile.txt” – print (filename[2:5]) prints fil (note index 5 is NOT printed, it prints from character at index 2 (‘f’) to character at index 4 (‘l’)
- Print (filename[-3:-2]) – here character at index -3 is ‘t’ (third character from the end) and character at index -2 is ‘x’

Sub-strings

```
fileList = ["myfile.txt", "myprogram.exe", "anotherfile.txt"]

for filename in fileList:
    if ".exe" in filename:
        print (filename)
```

```
name= "muhammad khan"
print(name.upper())
city='TORONTO'
print(city.lower())
print(name.capitalize())
print(len(name)) –prints number
                 of characters
```

```
MUHAMMAD KHAN
toronto
Muhammad khan

Process finished with exit code 0
```

Input and output statements (reading and displaying data)

- Reading data using ‘input()’ function
- Return type of ‘input()’ is string
- Use type-cast to convert string to other data types (e.g. int, float etc.)
- Displaying data using ‘print()’ function
- Formatting using format specifier for specific data type e.g. %s %d %f
- Displaying doc-string – print(__doc__)
- Displaying page/section divider – print (‘#’ * 100)

Python docstring

```
"""
Application Name:   Lecture1Example.py
Author/Developer:   Muhammad Khan (N0123456)
Date:               January 8th, 2023
Objective:      After displaying user name and ID, application gets user input (for various
data types). Application then processes information and displays outcome)
Description:    This application asks user to enter name and user enters name.
The application then asks user to enter student ID. User enters student ID.
Application asks user to enter item to purchase. User enters item name (which is string type).
Application then asks user to enter quantity of item to purchase. User enters the quantity of
the item to purchase. This is integer data type
Application then prompts user to enter price per item. User enters the price. This price is float
data type.
Display the information in the following format:
                                                        Name:
                                                        Student ID:
    Item Name               Item Quantity           Price per Item
    <name goes here>          <quantity goes here>      $<price goes here>

                                                        Tax Amount:     $<tax amount>
                                                        Total Amount:   $<total amount>
When displaying the price, use $ with it and two-digit display must be used to display price
in two digits. Similarly, tax amount (13% of the price) and total amount are also displayed
```

Example – source code

```
                                                    Tax Amount:     $<tax amount>
                                                    Total Amount:   $<total amount>
When displaying the price, use $ with it and two-digit display must be used to display price
in two digits. Similarly, tax amount (13% of the price) and total amount are also displayed
with $ symbol and real numbers are two-digits
"""



# Prompt user to enter name
name=input("Enter your name:\t\t")

student_ID=input("Enter your student ID:\t\t")  # prompt user to enter student ID
item_name=input("Enter item name:\t\t") # prompt user to enter item name
item_quantity=int(input("Enter item quantity:\t\t"))  # prompt user to enter quantity to purchase for the item
item_price=float(input("Enter price per item:\t\t$")) ### prompt user to enter price per item

tax_amount = 0.13 * item_price * item_quantity
total_amount = (item_price * item_quantity) + tax_amount

tax_amount_str= "Tax amount:    $%.2f"%(tax_amount)
print(type(tax_amount_str))

total_amount_str="Total amount paid:   $%.2f"%(total_amount)
```

```
py ×  Lecture1Example.py ×  test.py ×  Lecture1Example2.py ×
tax_amount = 0.13 * item_price * item_quantity
total_amount = (item_price * item_quantity) + tax_amount

tax_amount_str= "Tax amount:    $%.2f"%(tax_amount)
print(type(tax_amount_str))
total_amount_str="Total amount paid:   $%.2f"%(total_amount)
#display name and student ID in heading
display_name_ID_head="%120s\n"%(name)
display_name_ID_head +="%120s\n\n"%(student_ID)
#display heading
display_header="%40s%40s%40s\n"%("Item Name", "Item Quantity", "Price per item")

display_info="%40s%40d%40s\n"%(item_name, item_quantity, "$%.2f"%(item_price))

display_info+="%120s\n%120s\n"%(tax_amount_str, total_amount_str)

# Now displaying the information
print('-' * 125)
print("%70s"%("Displaying doc-string to display program documentation......"))
print('#' * 125)
print(__doc__)
print('#' * 125)
print(display_name_ID_head)
print(display_header)
print(display_info)
```

Python Example

Lecture1Example

```
C:\Python\python.exe C:/Users/mk_hu/OneDrive/Desktop/Winter2023/CCGC5003W23/Labs/Lab1/Lecture1Example.py
Enter your name:        Muhammad Khan
Enter your student ID:      N0123456
Enter item name:        Acer Computer
Enter item quantity:        6
Enter price per item:       $756.8378
```

Output of the Application

```
##########################################################################################################
                                                                                          Muhammad Khan
                                                                                               N0123456


                              Item Name                          Item Quantity                        Price per item

                           Acer Computer                                 6                                  $756.84
                                                                                            Tax amount:   $590.33
                                                                                     Total amount paid:  $5131.36


Process finished with exit code 0
```

Source code …

Lecture1Example.py

```python
# Prompt user to enter name
name=input("Enter your name:\t\t")

student_ID=input("Enter your student ID:\t\t")  # prompt user to enter student ID
item_name=input("Enter item name:\t\t") # prompt user to enter item name
item_quantity=int(input("Enter item quantity:\t\t"))  # prompt user to enter quantity to purchase for the item
item_price=float(input("Enter price per item:\t\t$")) ### prompt user to enter price per item

tax_amount = 0.13 * item_price * item_quantity
total_amount = (item_price * item_quantity) + tax_amount

tax_amount_str= "Tax amount:    $%.2f"%(tax_amount)
print(type(tax_amount_str))

total_amount_str="Total amount paid:   $%.2f"%(total_amount)

#display name and student ID in heading
display_name_ID_head="%120s\n"%(name)
display_name_ID_head +="%120s\n\n"%(student_ID)
#display heading
display_header="%40s%40s%40s\n"%("Item Name", "Item Quantity", "Price per item")

display_info="%40s%40d%40s\n"%(item_name, item_quantity, "$%.2f"%(item_price))
```

```python
# Now displaying the information
print('-' * 125)
print("%70s"%("Displaying doc-string to display program documentation......"))
print('#' * 125)
print(__doc__)
print('#' * 125)
print(display_name_ID_head)
print(display_header)
print(display_info)
```

Summary

- What is automation?
- Tools that integrate automation
- Why Python?
- Some Python applications
- Python downloads
- Network Automation framework
- Introduction to Python
- Building simple Python application
- Simple Python input/output and variables
- Input and output functions in Python