



Module 3 - System Administration Python

Session 5 - Practical Uses

Presented by Tim Medin

© SANS, Cyber Aces, Red Siege. All Rights Reserved. Redistribution Prohibited.

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

Welcome to the Module 3, System Administration. In this sub-section we'll be discussing Python. In this final session, we will discuss practical uses for Python.

SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems

- 01. Linux
- 02. Windows

2. Networking

3. System Administration

- 01. Bash
- 02. PowerShell
- 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of system administration and scripting. This session is part of Module 3, System Administration. This module is split into three sections, Bash, PowerShell, and Python. In this session, we will continue our examination of Python.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

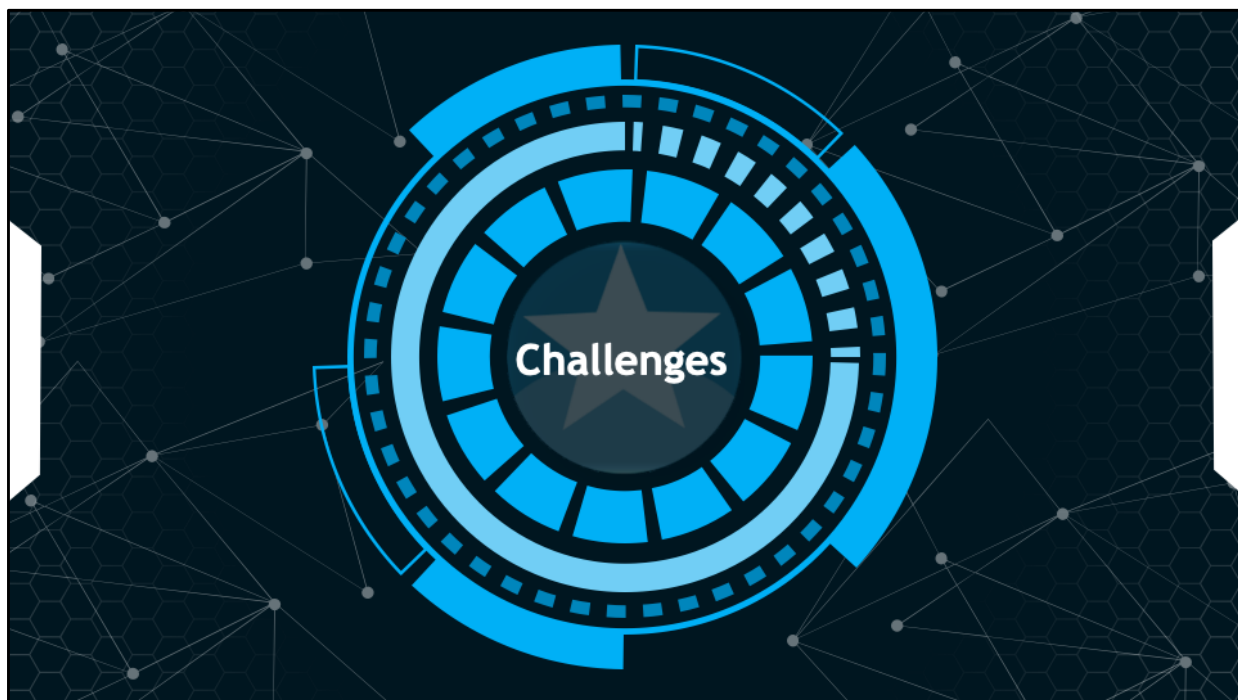
For more information about the Cyber Aces program, please visit the Cyber Aces website at <https://CyberAces.org/>.



Module 3 - System Administration Python

- Syntax & Data Types
- Flow Control
- Building a Script
- Modules
- Practical Uses

In this session, we will discuss some practical uses of Python.



This space intentionally left blank.



1. Converting Numbers



Write a script that will take an integer (base 10) and display the hex and binary equivalent

Your code should make sure it is an integer before converting

The number should be given as a command line argument

Your code should show an error if the user provides an incorrect number of command line arguments

Hint: You will need to use `bin()` and `hex()`

Hint: Use `sys.exit()` to exit your script when an error occurs

Example:

```
$ ./convert.py 997
0x3e5
0b1111100101
```

Write a script that will take an integer (base 10) and display the hex and binary equivalent. Your code should make sure it is an integer before converting.

The number should be given as a command line argument. If the argument is missing, or there are too many, your code should show an error message.

Hint: You will need to use `bin()` and `hex()`

Hint: Use `sys.exit()` to exit your script when an error occurs

Example:

```
$ ./convert.py 1001
0x3e9
0b1111101001
```



2. Numbers in a File



We start with a text file containing four lines:

```
John Doe 90  
Jane Doe 89  
Amy Herning 99  
Freak Bean 97
```

There is a space between each field and no leading or trailing space.

The file name is provided as a command line argument, display an error if the argument is missing

Goal: Print the top score, bottom score, and the average

Output:

```
$ ./score.py scores.txt
```

```
Top:      99
```

```
Bottom:   89
```

```
Average: 93.75
```

Start with a text file containing four lines:

```
John Doe 90  
Jane Doe 89  
Amy Herning 99  
Freak Bean 97
```

There is a space between each field and no leading or trailing space.

Goal: Print the top score, bottom score, and the average.

Output:

```
$ ./score.py scores.txt
```

```
Top:      99
```

```
Bottom:   89
```

```
Average: 93.75
```



3. Dice Roller



In many games the dice used during a portion of the game are described as 3d6

- Three dice
- Six sided

Your goal is to take input in the format, then give the output

Hint: Use the "random" module and the "randint" method

In many games the dice used during a portion of the game are described as 3d6

Three dice

Six sided

Your goal is to take input in the format, then give the sum of the dice rolls.

Hint: Use the "random" module and the "randint" method

Example:

```
$ ./roller.py
```

```
What would you like to roll? 3d6
```

```
Rolling 3, 6 sided dice with a result of 12
```

```
What would you like to roll? 2d12
```

```
Rolling 2, 12 sided dice with a result of 13
```

```
What would you like to roll? 10d4
```

```
Rolling 10, 4 sided dice with a result of 23
```



This space intentionally left blank.



1. Converting Numbers Possible Solution

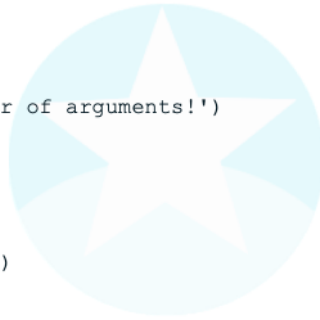


```
#!/usr/bin/env python3
import sys

if len(sys.argv) != 2:
    print('Invalid number of arguments!')
    sys.exit()

x = sys.argv[1]
if not x.isdigit():
    print('Not a number')
    sys.exit()

i = int(x)
print(hex(i))
print(bin(i))
```



Below is one possible solution

```
#!/usr/bin/env python3
import sys

if len(sys.argv) != 2:
    print('Invalid number of arguments!')
    sys.exit()

x = sys.argv[1]
if not x.isdigit():
    print('Not a number')
    sys.exit()

i = int(x)
print(hex(i))
print(bin(i))
```



2. Number in a File Possible Solution



```
#!/usr/bin/env python3
import sys

if len(sys.argv) != 2:
    print('Invalid number of arguments!')
    sys.exit()

max = -1
min = 99999
total = 0
count = 0

with open(sys.argv[1], 'r') as f:
    for line in f:
        score = int(line.split()[2])
        total += score
        count += 1
        if score > max:
            max = score
        if score < min:
            min = score

print('Top:      ' + str(max))
print('Bottom:   ' + str(min))
print('Average:  ' + str(total/count))
```

Possible solution

```
#!/usr/bin/env python3
import sys

if len(sys.argv) != 2:
    print('Invalid number of arguments!')
    sys.exit()

max = -1
min = 99999
total = 0
count = 0

with open(sys.argv[1], 'r') as f:
    for line in f:
        score = int(line.split()[2])
        total += score
        count += 1
        if score > max:
            max = score
        if score < min:
            min = score

print('Top:      ' + str(max))
print('Bottom:   ' + str(min))
```

```
print('Average: ' + str(total/count))
```



```
#!/usr/bin/env python3
import random
```

3. Dice Roller Possible Solution



```
while True:
    s = input('What would you like to roll? ')
    x = s.split('d')
    if len(x) != 2:
        print('Invalid input')
        continue

    num = x[0]
    if not num.isdigit():
        print('Invalid input')
        continue

    size = x[1]
    if not size.isdigit():
        print('Invalid input')
        continue

    total = 0
    for i in range(int(num)):
        total += random.randint(1, int(size))

    print('Rolling ' + num + ', ' + size + ' sided dice with a result of ' +
          str(total))
```

One possible solution is here:

```
#!/usr/bin/env python3
import random
while True:
    s = input('What would you like to roll? ')
    x = s.split('d')
    if len(x) != 2:
        print('Invalid input')
        continue

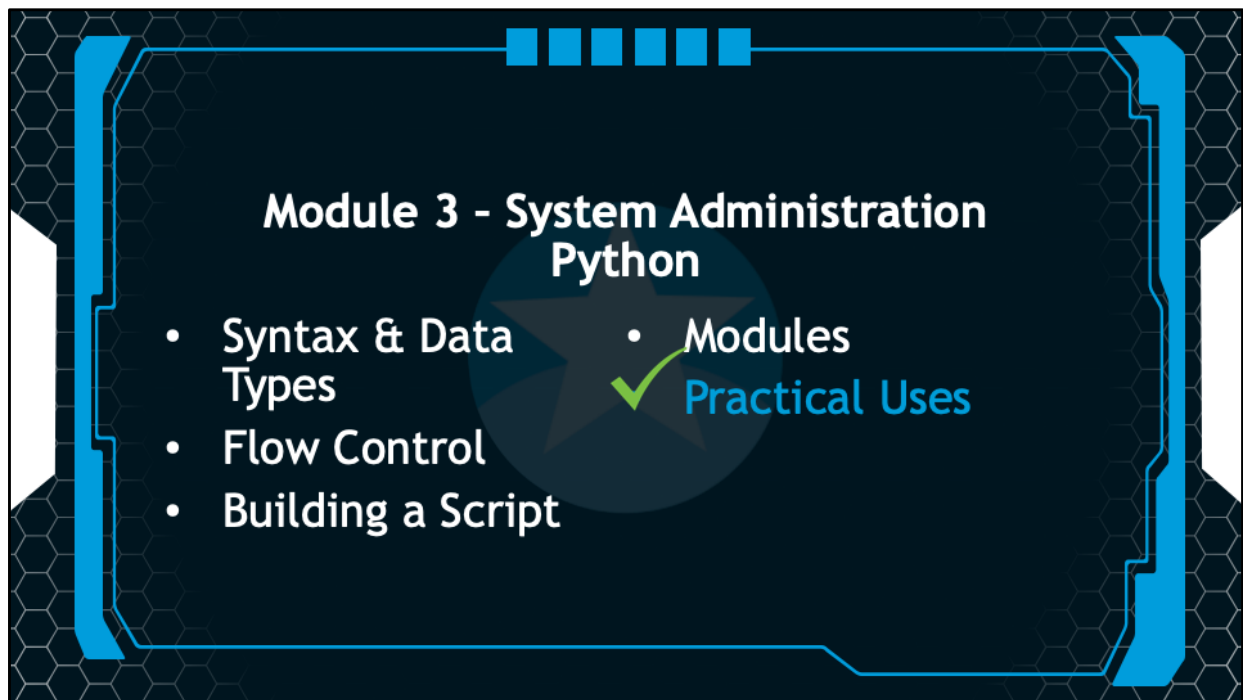
    num = x[0]
    if not num.isdigit():
        print('Invalid input')
        continue

    size = x[1]
    if not size.isdigit():
        print('Invalid input')
        continue

    total = 0
    for i in range(int(num)):
        total += random.randint(1, int(size))

    print('Rolling ' + num + ', ' + size + ' sided dice
```

with a result of ' + str(total))



We've completed our discussion of Python. If you've followed the sessions in order, you've completed not only Module 3, System Administration, but all of the Cyber Aces Modules!

SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems

- 01. Linux
- 02. Windows

2. Networking

3. System Administration

- 01. Bash
- 02. PowerShell
- 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of system administration and scripting. This session is part of Module 3, System Administration. This module is split into three sections, Bash, PowerShell, and Python. In this session, we will continue our examination of Python.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

For more information about the Cyber Aces program, please visit the Cyber Aces website at <https://CyberAces.org/>.