PYTHON SCRIPTING

Saturday, July 24, 2021 5:41 PM

1. **STRINGS-**

#!/bin/python3
#PRINT STRING
print("Hello World")
print("\n")
print('Hello, Worlld')
print("""This strings runs multiple lines!""")
print("this string is "+"awesome!")
	#Declaration of the program (PYTHON)
	# "#"-> means Comment in Python
	#Prints Hello World
	#Print to the next line
	
	#Prints the string in multiple lines
	
	#Concatenation of a string

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/STRINGS.py"
Hello World
Hello, Worlld
This strings runs
multiple lines!
this string is awesome!

[Done] exited with code=0 in 0.041 seconds
```

2. **MATH-**

#!/bin/python3
#MATHS WITH PYTHON
print(50+20)
print(200/5)
print(30-20)
print(2*100)
print(400//2)
print(40%3)
print(50+50-50*50/50)
	#Decleration of python
	
	#Addition
	#Division ->float
	#Sub
	#Multiplication
	#Double Divide ->int
	#Modulo
	#PEDMAS

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/MATHS.py"
70
40.0
10
200
200
1
50.0

[Done] exited with code=0 in 0.024 seconds
```

3. **VARIABLE AND METHODS-**

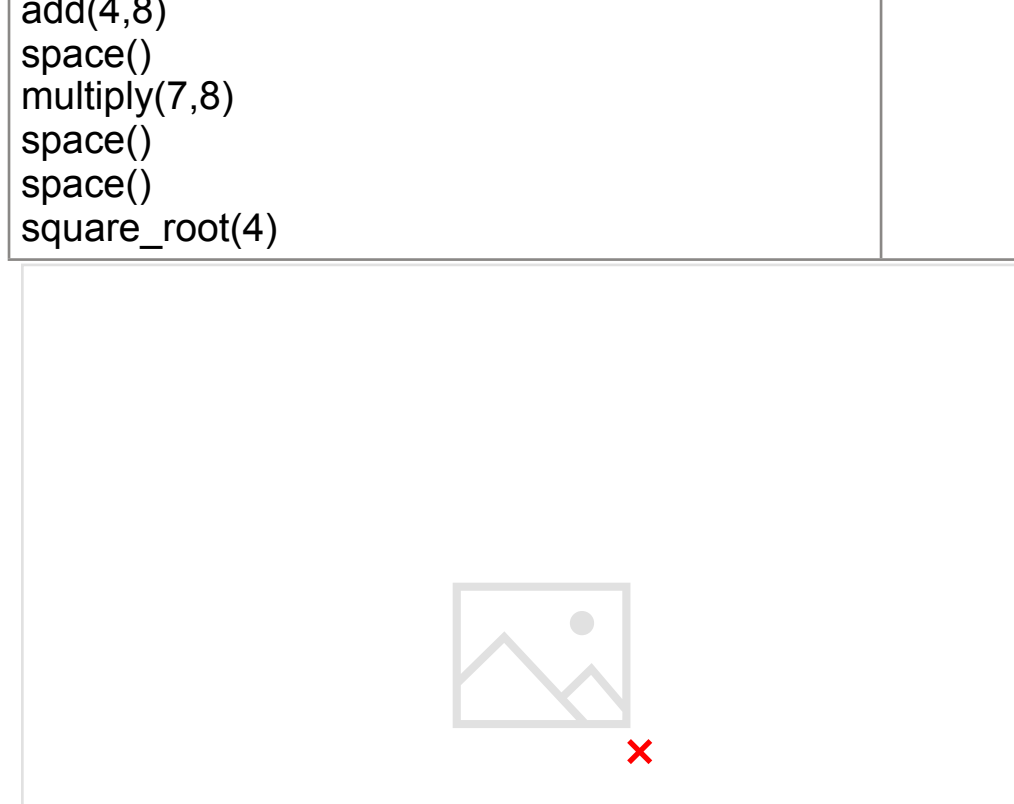
#!/bin/python3 a="Hello My Name Is ATOM" print(a) print(a.lower()) print(a.upper()) print(a.title()) print(len(a)) name="ATOM" age=17 gpa=4.0 print(int(age)) print(int(gpa)) print(float(age)) print(float(gpa)) print("My name is "+str(name)+" and I am "+str(age)+" year old.") age+=1 print(age) birthday=0 bithday=1 age+=bithday print(age)	#Assigning a variable #Makes it lower case #Make it Upper case #Make it Title Case #Prints the length #Integer #Float #Converts to int #Connverts to Float #Concatenation of string with converting to string #Adding one to age #Assign new value to birthday

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/VARIABLES_AND_METHODS.py"
Hello My Name Is ATOM
hello my name is atom
HELLO MY NAME IS ATOM
Hello My Name Is Atom
21
17
4
17.0
4.0
My name is ATOM and I am 17 year old.
18
19

[Done] exited with code=0 in 0.043 seconds
```

4. **FUNCTIONS-**

#!/bin/python3
def who_am_i():
    name="Atom"
    age=18
    print("my name is",name,"and my age is",age)
 	
def add_one_hundred(num):
    print(num+100)
 	
def add(x,y):
    print(x+y)
 	
def multiply(x,y):
    return x*y
 	
def square_root(x):
    print(x**0.5)
 	
def space():
    print("\n")
who_am_i()
add_one_hundred(100)
add(10,8)
multiply(7,86)
print(multiply(7,86))
square_root(64)
space()
add(4,8)
space()
multiply(7,8)
space()
space()
square_root(4)
	#Defining a function
	#[]Indentation for the function
	
	#Defining a Function which requires a parameter
	
	#Calling The Function
	#Calling a function with a parameters
	
	#Does not print anything
	#Prints the returned value

a. 

5. **BOOLEAN-**

#!/bin/python3 bool1=True bool2=3*3==9 bool3=False bool4=3*3!=9 bool5="True" print(bool1,bool2,bool3,bool4) print(type(bool1),"\n",type(bool5))	#Assigning True & Type -> Boolean #True #Assigning False #False #String

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/BOOLEAN.PY"
True True False False
<class 'bool'>
 <class 'str'>

[Done] exited with code=0 in 0.04 seconds
```

6. **RELATION & BOOLEAN EXPRESSIONS-**

#!/bin/python3 greater_than=7>5 less_than=5<7 greater_than_and_equal_to=7>=7 less_than_and_equal_to=7<=7 test_and=(7>5) and (5<7) test_and2=(7>5) and (5>7) test_or=(7>5) or (5<7) test_or2=(7>5) or (5>7) test_or3=(7<5) or (1>2) test_not=not True test_not2=not False print(greater_than) print(less_than) print(greater_than_and_equal_to) print(less_than_and_equal_to) print(test_and) print(test_and2) print(test_or) print(test_or2) print(true_or3) print(test_not) print(test_not2)	 #True #False #True #True #True #False #True #True #False #False #True

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/RELATIONAL_AND_BOOLEAN.PY"
True
True
True
True
True
False
True
True
False
False
True

[Done] exited with code=0 in 0.024 seconds
```

7. **CONDITIONAL STATEMENTS-**

#!/bin/python3
def drink(money):
    if money>=2:
        return "you got a drink!"
    else:
        return "no drink for you"
def alcohol(money,age):
    if age >21 and money>=5:
        print("here is your drink")
    elif age >21 and money<5:
        print("you meet the age requitement but you dont have the money")
    elif age<21 and money>=5:
        print("no drinks for kids even if you have money")
    else:
        print("no drinks for kids and you dont even have the money")
alcohol(8,18)
alcohol(10,24)
alcohol(3,22)
alcohol(3,19)
print("heres your fanta\n",drink(5))
	
	#Condition with "If"
	
	#Condition with "If" and "and"

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/CONDITIONAL_STATEMENT.PY"
no drinks for kids even if you have money
here is your drink
you meet the age requitement but you dont have the money
no drinks for kids and you dont even have the money
heres your fanta
 you got a drink!
```

8. **LISTS-**

#!/bin/python3
games=['gta',"cod","battlefeild","wolfenestein2","wolfenestein youngblood"]
print(games[1])
print(games[0])
print(games[1:3])
print(games[:3])
print(games[1:])
print (games[::1])
print(games[::3])
print(games[-1])
print (len(games))
games.append("coc") #adds coc at the last
print(games)
games.pop() #pops last item
print(games)
games.pop(0) # pops first item
print(games)
	#List
	#Print the second element
	#Prints the 0 element
	#Prints from the 1st element till the 2nd element (3rd item not included)
	#Prints the 0 item till 2nd (3rd not included)
	#Prints from the 1st element till the last
	#Prints all the elements
	#Prints with a gap of 3
	#Prints the last element

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/LISTS.PY"
cod
gta
['cod', 'battlefeild']
['gta', 'cod', 'battlefeild']
['cod', 'battlefeild', 'wolfenestein2', 'wolfenestein youngblood']
['gta', 'cod', 'battlefeild', 'wolfenestein2', 'wolfenestein youngblood']
['gta', 'wolfenestein2']
wolfenestein youngblood
5
['gta', 'cod', 'battlefeild', 'wolfenestein2', 'wolfenestein youngblood', 'coc']
['gta', 'cod', 'battlefeild', 'wolfenestein2', 'wolfenestein youngblood']
['cod', 'battlefeild', 'wolfenestein2', 'wolfenestein youngblood']
```

9. **TUPLES-**

#!/bin/python3 grades=("a","b","c","d","f") grades1=10,47,5,7 print(grades[1:3]) print(type(grades),type(grades1))	#Immutble -> values cannot be changed after declaration #Tupe #Tuple #Prints 1st element till the 2nd element (3rd not included)

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/TUPLES.PY"
('b', 'c')
<class 'tuple'> <class 'tuple'>

[Done] exited with code=0 in 0.023 seconds
```

10. **LOOPING-**

#!/bin/python3
#FOR LOOP (START AND FINISH KNOWN)
veg=["cucumber","apple","mango","strawberry"]
for x in veg:
    print(x)
#WHILE LOOP (START KNOW BUT END NOT KNOWN {generally})
i=1
while i <10:
    print(i)
    i+=1
	
	#List
	#For loop
	#Prints all the elements of list "veg"
	
	#Initial value to enter the loop
	#While Loop works till "i" is less than 10
	
	#Increasing the value of "i" so that the loop breaks

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/LOOPING.PY"
cucumber
apple
mango
strawberry
1
2
3
4
5
6
7
8
9
```

11. **IMPORTING MODULES-**

#!/bin/python3 import sys from datetime import datetime as dt print(sys.version) print(dt.now()) sys.exit()	 #Importing Module (system functions and parameters) #[Import with aliases "as"]Importing a specific Function from a Module #Prints Python Version #Print Date & Time #For Clean Exit

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/IMPORTING_MODULES.PY"
3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110]
2021-08-21 23:18:08.972260
```

12. **ADVANCED STRINGS-**

#!/bin/python3 my_name="ATOM" print(my_name[0]) print(my_name[-1]) print(my_name[::-1]) sen="My name is ATOM" print(sen[:7]) print(sen.split()) sentence_split=sen.split() sen_join=" ".join(sentence_split) print(sen_join) quote="he said, \"give me all your money\"" #'\' for character escaping print(quote) too_much_space=" hello world " print(too_much_space.strip()) print("A" in "Apple") print("a" in "Apple") letter="A" word="Apple" print(letter.lower() in word.lower()) games="cod" print("My favourite game is {}.".format(games))	 #Print 1st character of the str #Print Last character of the str #Prints the whole str in reverse #Prints all the characters till the 6th (7th not included) #Splits the String with the delimiter "_"(space) #Joining all the elements of the List to a str #("\"string\"" colored string will be ignored and treated as a string) #Removes excess "_" (space) #True #False #Improved string checking for all cases #("{}" placeholder for string) -> ".format("string")

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/ADVANCED_STRINGS.PY"
A
M
MOTA
My name
['My', 'name', 'is', 'ATOM']
My name is ATOM
he said, "give me all your money"
hello world
True
False
True
My favourite game is cod.
```

13. **DICTIONARIES-**

#!/bin/python3 drink={"frooty":60,"pepsi":50,"coka_cola":70} print(drink) employees={"finance":["bob","linda","tina"],"IT":["Gene","Louise","Teddy"],"HR":["jimmy","mort"]} print(employees) employees["legal"]=["mr. frond"] print(employees) employees.update({"sales":["Anie","ollie"]}) print(employees) drink["frooty"]=80 print(drink) print(drink.get("frooty")) print(drink["pepsi"])	#Dictionary (key:value pair) #Dictionary with values as list #Add new key value pair #Updating value for a specific key #Print value of passed key #Print value of passed key

a.
```
[Running] /bin/python3 "/home/atom/ATOM/CODING/PYTHON/DICTIONARIES.PY"
{'frooty': 60, 'pepsi': 50, 'coka_cola': 70}
{'finance': ['bob', 'linda', 'tina'], 'IT': ['Gene', 'Louise', 'Teddy'], 'HR': ['jimmy', 'mort']}
{'finance': ['bob', 'linda', 'tina'], 'IT': ['Gene', 'Louise', 'Teddy'], 'HR': ['jimmy', 'mort'], 'legal': ['mr. frond']}
{'finance': ['bob', 'linda', 'tina'], 'IT': ['Gene', 'Louise', 'Teddy'], 'HR': ['jimmy', 'mort'], 'legal': ['mr. frond'], 'sales': ['Anie', 'ollie']}
{'frooty': 80, 'pepsi': 50, 'coka_cola': 70}
80
50
```

14. **LIBRARY_SOCKET-**

#!/bin/python3 import socket host="127.0.0.1" port=7724 s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) s.connect((host,port))	 #Importing Socket Module #[AF_INET-> IPv4] [SOCK_STREAM-> Port] #Connecting to the host at the specific port

a.
```
┌──(atom㉿atom)-[~/ATOM]
└─$ sudo su
[sudo] password for atom:
┌──(root💀atom)-[/home/atom/ATOM]
└─# nc -nvlp 7751
listening on [any] 7751 ...
```

b.
```
┌──(atom㉿atom)-[~/ATOM]
└─$ sudo su
[sudo] password for atom:
┌──(root💀atom)-[/home/atom/ATOM]
└─# nc -nvlp 7751
listening on [any] 7751 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 35996
```

- **PORT SCANNER-**

#!/bin/python
import sys
import socket
from datetime import datetime
 	
#define our target
if len(sys.argv)==2:
    target=socket.gethostbyname(sys.argv[1])
else:
    print("Invalid amount of arguments.")
    print("Syntax: python3 scanner.py <ip>")
#ADD a preety banner
print("-"*50)
print("Scanning target "+target)
print("Time started:"+str(datetime.now()))
print("-"*50)
try:
    for port in range (1,61):
        s= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        socket.setdefaulttimeout(1)
        result=s.connect_ex((target,port))
        if result ==0:
            print("Port {} is open".format(port))
        s.close()
except KeyboardInterrupt:
    print("\nExiting Program.")
    sys.exit()
except socket.gaierror:
    print("Hostname could not be resolved.")
    sys.exit()
except socket.error:
    print("Couldn't connect to server.")
    sys.exit()
	#Importing sys
	#Importing Socket
	#Importing Date time
	
	#Translate Hostname to IPv4
	
	#Target
	
	#from port 1 to 60
	
	#Timeout of 1 sec
	#Return error indicator with we use (connect_ex)
	#When there is no error, then it returns 0 (result) ->OPEN PORT
	
	#Hitting [CTRL + C]
	
	#Clean Exit
	#DNS Failing us, No Hostname Resolution
	
	#Coudn't connect to the IP Address

```
┌──(root💀atom)-[/home/atom/ATOM/CODING/PYTHON]
└─# python3 ./PORT_SCANNER.PY 10.28.1.10
--------------------------------------------------
Scanning target 10.28.1.10
Time started:2021-08-22 00:28:52.197063
--------------------------------------------------
Port 22 is open
Port 53 is open
```