

IT 231- Project # 3A Python Scripts Fall 2018

We will create 11 scripts, take two screenshots for each scripts. The first screenshot should show the script, the second should show the output after you run the script.

1.gm.py

nano gm.py → file extension (.py) is not required in linux

#!/usr/bin/python → whereis python

print 'Good morning!'

Save the script and make it executable

chmod u+x gm.py or chmod 755 gm.py

./gm.py

Implied display:

python → Python interactive shell

```
>>> print 'Good morning'
```

```
>>> 'Good morning'
```

```
>>> print 2 + 2
```

```
>>> 2+ 2
```

```
>>> x = 'Hello'
```

```
>>> x
```

```
>>> int (8.999) →function: int(), exit(), float(), help(),len(), map(),  
max(),open(), range(), raw_input(), sorted(), type(), xrange()
```

```
>>> exit
```

You can also run a python program by specifying the name of the file or as a standard input to python.

```
# python gm.py
# python < gm.py
# cat gm.py | python
# python -c "print 'Good morning! '" → run program from the shell
```

2. raw_input.py

nano raw_input.py → create a script

```
#!/usr/bin/python
my_in = raw_input('Enter your name: ')
print 'Hello., ' + my_in + '!'
save the script and make it executable
```

```
./my_input.py
```

3. list.py

nano list.py

```
#!/usr/bin/python
a = ['bb', 'dd', 'zz', 'rr']
x = a[0]
print a
a.append('mm') →
# method: append(x), count(x), index(x), remove(x), reverse(), sort() ,
#a.reverse()
print a
b = sorted(a)
print len(b)
```

save the script and make it executable

```
./list.py
```

4. my_max.py

```
nano my_max.py
```

```
#!/usr/bin/python
```

```
my_rand_list = [5, 6, 4, 1, 7, 3, 2, 0, 9, 8]
```

```
largest = -1
```

```
for item in my_rand_list:
```

```
    if (item > largest):
```

```
        largest = item
```

```
        print largest,
```

```
print
```

```
print 'largest number is ', largest
```

save, make the script executable.

```
./my_max.py
```

5. if1.py

```
nano if1.py
```

```
#!/usr/bin/python
```

```
my_in = raw_input ('Enter your name: ')
```

```
if (my_in != ''):
```

```
    print 'Thank you, ' + my_in
```

```
print 'Program running, with or without your input.'
```

Save the script, make it executable

./if1.py

6. if2.py

nano if2.py

```
#!/usr/bin/python
```

```
my_in = raw_input ('Enter your name: ')
```

```
if (my_in != ''):
```

```
    print 'Thank you, ', my_in
```

```
else:
```

```
    print 'Program requires input to continue.'
```

```
    exit()
```

```
print 'Program running with your input.'
```

Save the script, make it executable.

./if2.py

7. bignum.py

nano bignum.py

```
#!/usr/bin/python
```

```
input = int (raw_input('Please enter a number: '))
```

```
if (input == 1):
```

```
    print 'You entered one.'
```

```
elif (input == 2):
```

```
    print 'You entered two.'
```

```
elif (input == 3):
```

```
    print 'You entered three.'
```

else:

```
    print 'You entered a big number...'
```

```
print 'End of program.'
```

Save the script, make it executable

8. count.py

```
nano count_num.py
```

```
#!/usr/bin/python
```

```
count = 0
```

```
while (count <= 10):
```

```
    print count, # stay on the same line
```

```
    count += 1
```

```
print
```

```
lis = ['turkey', 'pony', 'dog', 'fox']
```

```
for name in lis:
```

```
    print name
```

Save the script, make it executable

```
./count.py
```

9. my_function.py

```
nano my_function.py
```

```
#!/usr/bin/python
```

```
def add_ab(my_list):
```

```
    my_list.append('a')
```

```
    my_list.append('b')
```

```
a = [1,2,3]
```

```
add_ab(a) #call the function with the list a.
```

```
print a
```

Save the script and make it executable.

```
./my_function.py
```

10. new_function.py

```
nano new_function.py
```

```
#!/usr/bin/python
```

```
def add_one(x):
```

```
    return x + 1
```

```
a = add_one(2)
```

```
print a
```

Save the script, make it executable

```
./new_function.py
```

11. new_fun.py

```
nano new_fun.py
```

```
#!/usr/bin/python
```

```
def enter_number(val):
```

```
    num = int(raw_input('Please enter a number: '))
```

```
    if (num > val):
```

```
        print 'Too high'
```

```
        return 1
```

```
    elif (num < val):
```

```
        print 'Too low'
```

```
        return -1
```

```
    else:
```

```
        print 'Got it!'
```

```
    return 0
```

```
while (enter_number(2) != 0)
```

```
    continue
```

Save the script, then make it executable

```
./new_fun.py
```

3 → Too high

1 → too low

2 → Got it!