

SBCC Computer Science Club
Python Workshop, 10/9/15
Supplemental

How to run a Python script in the command line:

```
$ python <script_name>
```

It's possible to run a Python script directly by adding the following line to the top of the script:

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

On Linux/Mac the file must be made executable. E.g., `chmod +x <script_name>`

now `./<script_name>` will function as `$ python <script_name>`

One way to simulate a `main()` function is to use the following condition:

```
import os # os.system(cmd)
if __name__ == '__main__': #the cond. body runs when invoked but not imported
    time = int(input('Input minutes until shutdown: '))
    os.system("shutdown /s /t " + str(60*time))
```

```
import sys # useful for sys.argv - enables access to command line arguments
```

Simple File IO Examples:

```
f = open(path, 'w')
f.write(string); f.close()
```

```
f = open('hostnames_linux.txt')
```

```
string = f.read(); f.close()
```

- There are many ways to read a file. E.g.,

```
lines = [line.strip() for line in f]
```

- it's possible to call `f.readline()` until the function returns "

Running system (Linux/Windows/Mac) commands:

```
import os # os.system(cmd)
os.system('chmod +x _sshCSLinux.sh')
```

Try this code out!

#Phi calculator example (1.6180339887498948...) - demos large integer handling

#Can be approximated by a ratio of successive fibonacci sequence elements

```
def phi(limit=10**6):
    a,b,c = 1,1,0
    for i in range(limit):
        c = a
        a = a + b
        b = c
    ratio = str((a*10**(len(str(a))-1) // b))
    return '1.' + ratio[1:]
```

```
def expo(base, power):
    if power < 1:
        return 1
    half = expo(base, power // 2)
    if power % 2 == 1:
        return base * half * half
    else:
        return half * half
```

```
def expo2(base, power):
    if power < 1:
        return 1
    result = 1
    for i in range(power):
        result = result * base
    return result
```