

Forum:[Hack Forums](#)

Thread: [Command Line Scripts Tutorial/Source](#)

MiguelSanchez - January 22, 2022, 6:01 am

Here are a bunch of demo scripts, some practical, others useful as in introduction, starting from the simplest:

```
Code import sys #import the sys library
print(sys.argv) #prints the list argv
```

The above script, when run in Idle will simply return a list containing your script's filename.

However if run from command line, sys.argv will be a list where the first element is the filename and the remaining elements are the arguments from the command line.

With that in mind, the echo function in windows command line would look something like this.

```
Code import sys #import the sys library
```

```
s="" #Create an empty string
```

```
for argument in sys.argv[1:]:#with every item in sys.argv except for the first
```

```
    s+=argument+" " #add the argument to the string, followed by a space
```

```
print(s) #prints the string
```

Or for those more familiar with handling strings, using the join function

```
Code import sys #import the sys library
```

```
print(sys.argv[1:].join(" ")) #prints all the arguments joined together separated by a space, excluding the filename
```

Getting the current working directory is possible wherever you call the script from.

```
Code import os
```

```
print(os.getcwd())
```

Being on windows atm, I have this script in my "system environment variables" path so it can be called from any directory.

Finally, I'll take an intermission after concluding with something useful.

Being on windows it pained me to not have "curl";

The simplest method is Code from urllib.request import urlopen

```
import sys
```

```
site=sys.argv[1]#assign the first argument to the variable name "site"
```

```
myURL = urlopen(site)#open the site
```

```
print(myURL.read())#read the data from the site and print it to the active display
```

I'm going to take a break here and I'll comment a few more of my command line scripts to attach as a reply in the next 30 minutes.

Playing an audio file from command line

I hope I commented this well enough to be self-explainable.

It will require you to install pygame.

```
(run -> cmd -> "pip3 -install pygame")
```

```
Code import pygame
```

```
import sys
```

```
from time import sleep
```

```
if len(sys.argv)>1:#if there is at least 1 argument (excluding filename)
```

```
    pygame.mixer.init() #initialise the pygame mixer
```

```
    my_sound = pygame.mixer.Sound(sys.argv[1])#Load the command line argument into the sound
```

mixer

```
seconds=my_sound.get_length()#assign the length of the sound to a variable called seconds
my_sound.play()#play the sound
sleep(seconds)#pause the program for the duration of the sound
#end program
```

```
else:#if there isn't at least 1 argument (excluding filename)
    print("NO ARGUMENT")
```

Tuning.py (Produces a tone for tuning instruments)

Also requires pygame.

Usage:

tuning C2 // Plays note C2

tuning 322 // Plays the absolute pitch 332hz

tuning E5 436 // Plays the note E5 where A4 is tuned to 436HZ

tuning C3 432 // Plays the note C3 where A4 is tuned to 432HZ

Code from `__future__` import division

```
import pygame
```

```
import sys
```

```
from math import sin,pi
```

```
from time import sleep
```

```
def fold(List,Func,Arg):
```

```
    if List==[]:
```

```
        return Arg
```

```
    else:
```

```
        return fold(List[1:],Func,Func(Arg,List[0]))
```

```
def foldstr(Str,Func,Arg):
```

```
    if Str=="":
```

```
        return Arg
```

```
    else:
```

```
        return foldstr(Str[1:],Func,Func(Arg,Str[0]))
```

```
def displaymanual():
```

```
    print("""
```

Tuning:

Usage:

tuning C2 // Plays note C2

tuning 322 // Plays the absolute pitch 332hz

tuning E5 436 // Plays the note E5 where A4 is tuned to 436HZ

<Unimplemented>tuning E5 C4=260 // Plays the note E5 where C4 is tuned to 436Hz

```
""")
```

```
def playfreq(freq):
```

```
    pygame.mixer.init()
```

```
    my_sound = freqtosound(freq)
```

```
    my_sound.play()
```

```
def notetofreq(note,octave,rules="A4=440"):
```

```
    rules=rules.upper().strip()
```

```
    if rules[0:3]=="A4=":
```

```

tuning=int(rules[3:])
note=note[0].upper()+note[1:]
###49=A4=440hz Octave Starts on C and ends on B
notes={"C":0,"C#":1,"D":2,"D#":3,"E":4,"F":5,"F#":6,"G":7,"G#":8,"A":9,"A#":10,"B":11}
number=notes[note]+octave*12-8
return tuning*2**((number-49)/12.0)
else:
tuning=int(rules[3:])
centrenote=49-difference("A4",rules[:2])
note=note[0].upper()+note[1:]
###49=A4=440hz Octave Starts on C and ends on B
notes={"C":0,"C#":1,"D":2,"D#":3,"E":4,"F":5,"F#":6,"G":7,"G#":8,"A":9,"A#":10,"B":11}
number=notes[note]+octave*12-8
return tuning*2**((number-centrenote)/12.0)

```

```

def freqtosound(freq,samplerate=44100):
import wave
from struct import pack
tempname="temp.wav"
l=[(lambda a:abs(a)**2*(-1 if a<0 else 1))(sin(i/samplerate*freq*pi*2)) for i in range(samplerate*2)]
f=wave.open(tempname, mode="wb")
f.setparams((1, 4, 44100, 0, 'NONE', 'not compressed'))
for i in l:
frames=pack("<I",int(i*(2**30.9)))
f.writeframesraw(frames)
f.close()
return pygame.mixer.Sound(tempname)

```

```

def isnote(a):
a=a.upper()
if isinstance(a,str):
a=a.strip()
if 'A' in a or 'B' in a or 'C' in a or 'D' in a or 'E' in a or 'F' in a or 'G' in a:
return foldstr(a,lambda a,b: a and b in "ABCDEFGH#123456789",True)
else:
return False

```

```

def isnumstr(a):
if isinstance(a,str):
a=a.strip()
return foldstr(a,lambda a,b: a and b in "0123456789",True)
else:
return False

```

```

args=sys.argv[1:]
if len(args)==0:
displaymanual()
if len(args)==1:
if isnumstr(args[0]):
playfreq(int(args[0]))
elif isnote(args[0]):
if args[0][-1:] in "0123456789":
note=args[0][:-1]
octave=int(args[0][-1:])

```

```

        playfreq(notetofreq(note,octave))
    else:
        note=args[0]
        octave=4
        playfreq(notetofreq(note,octave))
    else:
        displaymanual()
elif len(args)==2:
    if isnote(args[0]) and isnumstr(args[1]):#Plays the note E5 where A4 is tuned to 436HZ
        if args[0][-1:] in "0123456789":
            note=args[0][:-1]
            octave=int(args[0][-1:])
            playfreq(notetofreq(note,octave,"A4=%s"%args[1]))
        else:
            note=args[0]
            octave=4
            playfreq(notetofreq(note,octave,"A4=%s"%args[1]))

sleep(2)

```